

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики

«До захисту допущено»
Завідувач кафедри ПМІ



(підпис)

Ольга ДМИТРИЄВА

(ім'я та прізвище)

«15» червня 2022 р.

**Випускна кваліфікаційна робота
бакалавра**

на тему Розробка web-додатку для автоматизації дистанційного навчання
учнів загальноосвітньої школи

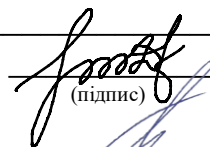
зі спецчастиною Розробка програмних модулів, інтерфейсу користувача та
бази даних засобами PHP, MySQL, JavaScript, HTML, CSS

Виконав (-ла): студент (-ка) 4 курсу, групи КН-18
(шифр групи)

спеціальності (напряму підготовки) 122 Комп'ютерні науки
(шифр і назва напряму підготовки, спеціальності)

Коверсун Д. В.

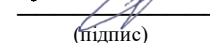
(прізвище та ініціали)



(підпис)

Керівник ст. викл. каф. ПМІ Євген ПАВЛОВСЬКИЙ

(посада, науковий ступінь, вчене звання, ім'я та прізвище)



(підпис)

Рецензент зав. каф. КІ, к.т.н., доц. Сергій КОВАЛЬОВ

(посада, науковий ступінь, вчене звання, ім'я та прізвище)



(підпис)

Нормоконтроль:

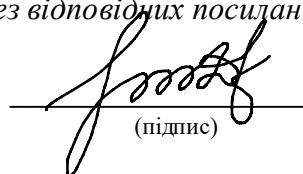


(підпис)

к.т.н., доц. Ірина НАЗАРОВА

*Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших
авторів без відповідних посилань.*

Студент



(підпис)

Луцьк – 2022 р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики
Освітньо-кваліфікаційний рівень (освітній ступінь) бакалавр
Напрямок підготовки (спеціальність) 122 Комп'ютерні науки
(шифр і назва)

ЗАТВЕРДЖУЮ:
Завідувачка кафедри

Дмитрієва О. А.

“ ”

2021 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТУ

Коверсун Діана Володимирівна

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка web-додатку для автоматизації
дистанційного навчання учнів загальноосвітньої школи

Спецчастина Розробка програмних модулів, інтерфейсу користувача та бази
даних засобами PHP, MySQL, JavaScript, HTML, CSS

Керівник проєкту (роботи) Євген ПАВЛОВСЬКИЙ, ст. викл. каф. ПМІ

(ім'я та прізвище, науковий ступінь, вчене звання)

затверджені наказом від “06” травня 2022 року № 180

2. Строк подання студентом проєкту (роботи) 8 червня 2022 року

3. Вихідні дані до проєкту (роботи) результати науково-дослідної роботи,
матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно
розробити) 1) Аналіз технічного завдання; 2) Проєктування системи; 3)
Розробка системи; 4) Опис роботи системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових
креслень)

1) діаграма варіантів використання; 2) діаграма розгортання; 3) діаграма
послідовності; 4) модель бази даних; 5) макети користувацького
інтерфейсу; 6) слайди презентації.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

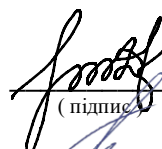
7. Дата видачі завдання 18 квітня 2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Аналіз технічного завдання. Постановка задач кваліфікаційної роботи	18.04.22-25.04.22	
2.	Проектування системи для автоматизації дистанційного навчання	26.04.22-02.05.22	
3.	Розробка системи	03.05.22-23.05.22	
4.	Тестування системи	24.05.22-30.05.22	
5.	Оформлення пояснювальної записки	01.06.22-09.06.22	
6.	Підготовка слайдів презентації	10.06.22-17.06.22	
7.	Нормоконтроль пояснювальної записки та додаткових матеріалів. Рецензування роботи	18.06.22-21.06.22	
8.	Захист кваліфікаційної роботи	22.06.22	

Студент

Керівник роботи


(підпис)

Діана КОВЕРСУН
(прізвище та ініціали)

Євген ПАВЛОВСЬКИЙ
(прізвище та ініціали)

АНОТАЦІЯ

Коверсун Д.В. Розробка web-додатку для автоматизації дистанційного навчання учнів загальноосвітньої школи / Випускна кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 122 Комп'ютерні науки. – ДВНЗ ДонНТУ, Луцьк, 2022.

Об'єктом дослідження є процеси розробки кросбраузерного web-додатку засобами PHP, MySQL, JavaScript, HTML, CSS.

Предметом дослідження є автоматизація дистанційного навчання учнів загальноосвітньої школи.

Метою роботи є проєктування та розробка web-додатку, що дозволяє автоматизувати дистанційне навчання для учнів загальноосвітньої школи.

Наукова новизна полягає у сучасних підходах до автоматизації дистанційного навчання.

Практичне значення полягає у розробці системи, яка дозволяє проводити навчання у дистанційному режимі у простому та зручному для користувача вигляді.

Ключові слова: база даних, браузер, web-додаток, інтернет, web-сервер, JavaScript, PHP, YII, HTML, web-сервіс.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	7
ВСТУП.....	8
1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ	10
1.1 Опис предметної області.....	10
1.2 Аналіз існуючих аналогів	11
1.2.1 Google Classroom	11
1.2.2 Moodle.....	13
1.2.3 Microsoft Teams	15
1.3 Аналіз засобів розробки.....	16
1.3.1 Клієнт-серверна архітектура.....	17
1.3.2 Системи керування базами даних	20
1.4 Порівняння і вибір інструментів для розробки	22
1.5 Постановка задачі	23
2 ПРОЄКТУВАННЯ СИСТЕМИ.....	25
2.1 Розробка специфікації вимог.....	25
2.1.1 Класи та характеристики користувачів	25
2.1.2 Операційне середовище	26
2.1.3 Дизайн та реалізація	26
2.1.4 Інтерфейси користувачів.....	26
2.1.5 Продуктивність	27
2.1.6 Безпека	27
2.1.7 Атрибути якості ПЗ	27
2.1.8 Допущення і залежності.....	27
2.1.9 Системні характеристики	28
2.2 Моделювання системи	28
2.2.1 Структура системи.....	28
2.2.2 Структура бази даних.....	31
2.2.3 Проєктування зовнішнього відображення сайту	32
2.2.4 Діаграма використання	36
2.2.5 Діаграма послідовності	38
2.2.6 Діаграма розгортання	39
3 РОЗРОБКА СИСТЕМИ.....	41

3.1 Засоби розробки	41
3.1.1 Мова програмування	41
3.1.2 Програмне середовище	41
3.1.3 Допоміжні компоненти	42
3.2 Формування таблиць бази даних	43
3.3 Опис класів	46
3.3.1 User	46
3.3.2 Students Group	47
3.3.3 Students Group Assignment	48
3.3.4 Course	48
3.3.5 Course Student Assignment	49
3.3.6 Course Task	50
3.3.7 Course Task Submission	51
3.3.8 Attachment	51
3.3.9 Course Time Slot	52
3.3.10 Student Mark	53
3.3.11 Course Chat Message	54
3.3.12 Notification	54
3.4 Тестування системи	55
3.4.1 Встановлення пакетів	56
3.4.2 Підготовка проєкту	56
4 ОПИС РОБОТИ СИСТЕМИ	59
ВИСНОВОК	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	68
Додаток А Зауваження нормоконтролера	70
Додаток Б Лістинг програми	71
Додаток В Роздатковий матеріал	128

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CSS – cascading style sheets (каскадні таблиці стилів);

HTML – hypertext markup language (мова розмітки гіпертекстових документів);

IDE – integrated development environment (інтегроване середовище розробки);

JS – JavaScript (динамічна, об'єктно-орієнтована прототипна мова програмування);

MVC – model-view-controller (модель–представлення–контролер);

PHP – hypertext preprocessor (гіпертекстовий препроцесор);

SQL – structured query language (мова структурованих запитів);

БД – база даних;

СКБД – система керування базами даних;

MVCC – Multi-Version Concurrency Control;

API – Application Programming Interface;

UI – UserInterface.

ВСТУП

В даний час дистанційне навчання стає все більш популярним способом отримання знань. Воно являє собою нову організацію освіти, що базується на використанні як традиційних методів отримання знань, так і нових інформаційних та телекомунікаційних технологій. Системи дистанційного навчання надають можливість навчати як школярів, студентів, так і людей, що працюють, без відриву від виробництва.

Дистанційне навчання має низку переваг перед іншими формами навчання. Так, практично не виходячи з дому чи не покидаючи свого робочого місця, можна регулярно підтримувати контакт з викладачем та одержувати структурований навчальний матеріал, представлений в електронному вигляді.

Також навчання через інтернет має такі переваги як:

- гнучкість;
- модульність;
- економічність;
- технологічність;
- інтерактивність.

Таким чином, розробка програмної системи для автоматизації дистанційного навчання є актуальною задачею даної кваліфікаційної роботи.

Виконання розробки даної системи передбачає такі кроки:

Для досягнення поставленої мети необхідно виконати наступні задачі:

- 1) Аналіз популярних існуючих додатків для дистанційного навчання.
- 2) Аналіз і вибір засобів розробки.
- 3) Проєктування та реалізація структури системи обраними засобами розробки.
- 4) Реалізація функціонала системи та зовнішнього відображення сайту.
- 5) Тестування розробленого додатку.

Структура кваліфікаційної роботи складається з вступу, чотирьох основних розділів, висновку та списку використаних літературних джерел.

У першому розділі наведено огляд існуючих систем для дистанційного навчання, а також аналізуються існуючі рішення та технології для розробки подібних систем. На основі цих досліджень визначаються мета і завдання роботи.

У другому розділі виконано розробку специфікацій до системи, а також проєктування її структури.

У третьому розділі наведено детальний опис реалізації окремих модулів і підсистем, та реалізації зовнішнього відображення сайту.

У четвертому розділі наведено опис роботи розробленого додатку.

1 АНАЛІЗ ТЕХНІЧНОГО ЗАВДАННЯ

1.1 Опис предметної області

Програмна система розробляється виключно для учнів та вчителів загальноосвітньої школи. Метою системи є надання можливості учням та вчителям проводити навчання у дистанційній формі.

Так як система розробляється для навчального закладу, можливість реєстрації користувачів повинна надаватися тільки адміністратору. Саме йому надають всю необхідну інформацію щодо учнів та вчителів.

Згідно з вищенаведеним, адміністратору необхідно створити обліковий запис (аккаунт) із відповідним ПІБ кожного користувача, його типом (учень чи вчитель), згенерувати логін і пароль та додати до відповідного класу (у випадку з учнем). Згодом цей аккаунт необхідно надати користувачеві для подальшої роботи в системі.

Також, адміністратори повинні мати можливість створювати шкільні предмети та призначати їх до відповідних вчителів та учнів.

Пройшовши процедуру авторизації, користувач повинен бачити наступну навігацію:

- предмети;
- розклад;
- журнал оцінок.

Перейшовши до сторінки «Предмети», користувач повинен мати можливість бачити список із призначеними предметами. Обравши конкретний предмет, у користувача має бути можливість переглянути його опис, надіслати повідомлення до чату даного предмета, та переглянути список із завданнями.

Учень повинен мати можливість переглянути детальний опис завдання та відправити результат його виконання на перевірку. Результат виконання може містити як текстовий коментар, так і відповідні прикріплені файли.

Вчитель, в свою чергу, повинен мати можливість створювати/редагувати/видаляти завдання та оцінювати результати його виконання.

Перейшовши до сторінки «Розклад», користувачу необхідно показати розклад на поточний тиждень (за замовчуванням) із можливістю перегляду іншого періоду. Вчитель повинен мати можливість редагувати розклад.

Перейшовши до «Журнал оцінок», учень повинен мати можливість переглянути свої оцінки. Вчителю необхідно надати можливість редагувати результат перевірки (оцінку).

Система повинна передбачити надання можливості переглядати як власні профілі, так і профілі інших користувачів. На сторінці профілю користувача повинна міститись наступна інформація:

- ПІБ;
- email адреса;
- тип (учень/вчитель);
- клас (у випадку з учнем).

Також, користувачі повинні мати можливість переглядати різноманітні повідомлення (нотифікації) в системі. Наприклад, при створенні вчителем нового завдання, всі учні повинні отримати відповідну нотифікацію.

1.2 Аналіз існуючих аналогів

На цей час кількість систем для дистанційного навчання стає дедалі більше, оскільки дана форма навчання стає зручнішою для багатьох людей. У кожній системі, які розглянемо далі, є як свої переваги, так і недоліки.

1.2.1 Google Classroom

Платформа Google Classroom – це сервіс, що пов’язує Google Docs, Google Drive та Gmail, дозволяє організувати онлайн-навчання, використовуючи відео-, текстову та графічну інформацію [1]. Викладач має змогу:

- проводити тестування;
- контролювати, систематизувати, оцінювати діяльність;
- переглядати результати виконання вправ;
- застосовувати різні форми оцінювання;
- коментувати й організовувати ефективне спілкування з учнями в режимі реального часу.

Основним елементом Google Classroom є групи. Функціонально групи нагадують структурою форуми, оскільки вони дозволяють користувачам легко відправляти повідомлення іншим учасникам процесу. Завдяки сервісу для спілкування Hangouts, учні та викладачі мають змогу вести онлайн-бесіди в режимі реального часу з комп'ютера або мобільного пристрою, учасники/учасниці команди можуть демонструвати свої екрани, дивитись і працювати разом. Така трансляція автоматично публікуватиметься на YouTube-каналі. Також платформа дозволяє за допомогою Google-форм збирати відповіді учнів і потім проводити автоматичне оцінювання результатів тестування (рис. 1.1).

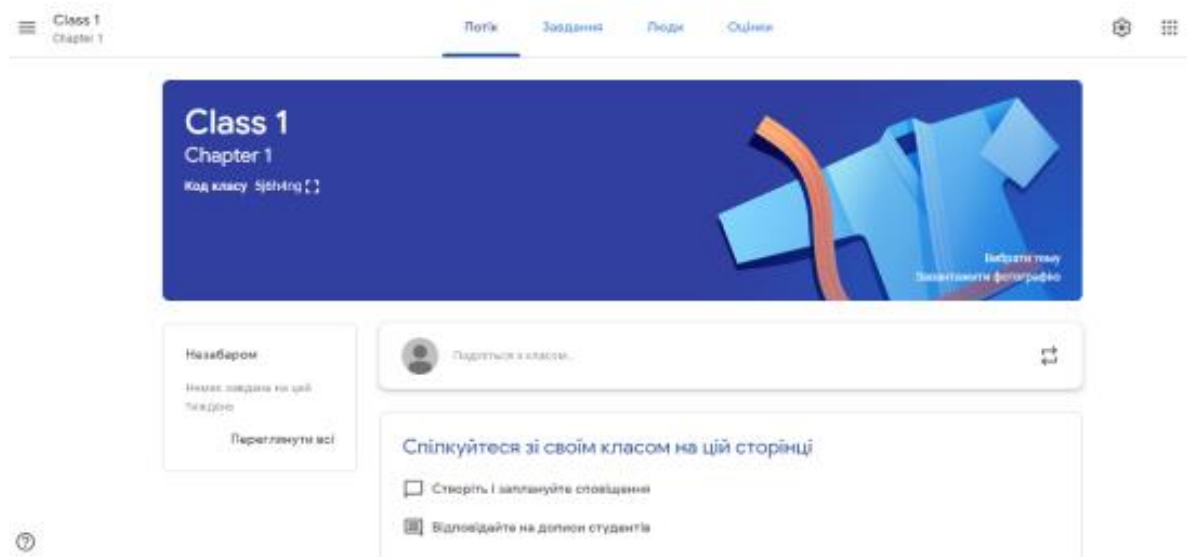


Рисунок 1.1 – Платформа Google Classroom

Серед переваг даного сервісу можна виділити наступні:

- безкоштовне використання сервісу;

- підтримка великої кількості мов інтерфейсу;
- зручний та інтуїтивно-зрозумілий інтерфейс;
- наявність журналу з оцінками учнів;
- додаток доступний для ПК, планшетів та смартфонів;
- інтеграція з багатьма сервісами Google.

До недоліків можна віднести наступне:

- ліміт на доступний обсяг для збереження файлів (15 ГБ);
- для реєстрації необхідно використовувати тільки обліковий запис Google;
- діти до 13 років мають мати дозвіл батьків (вхід з облікового запису батьків).

1.2.2 Moodle

Платформа Moodle – безкоштовна відкрита система керування дистанційним навчанням [2]. Дана платформа дозволяє використовувати широкий набір інструментів для освітньої взаємодії вчителів, учнів та адміністрації закладу освіти. Зокрема, надає можливість:

- подавати навчальний матеріал у різних форматах: (текст, презентація, відеоматеріал, web-сторінка);
- здійснювати тестування та опитування школярів з використанням питань закритого (множинний вибір правильної відповіді та зіставлення) і відкритого типів;
- учні можуть виконувати завдання з можливістю пересилати відповідні файли.

Крім того, система має широкий спектр інструментів моніторингу навчальної діяльності учнів, наприклад: щодо загального часу роботи учня/учениці з конкретним навчальним предметом, відповідними темами або складниками навчального матеріалу, загальної успішності учня/учениці або класу в процесі виконання тестових завдань тощо (рис. 1.2).

Moodle має у своєму інструментарії наступні можливості:

- форми здачі завдань;
- дискусійні форуми;
- завантаження файлів;
- журнал оцінювання;
- обмін повідомленнями;
- календар подій;
- новини та анонси;
- онлайн-тестування;
- Вікі-ресурси.

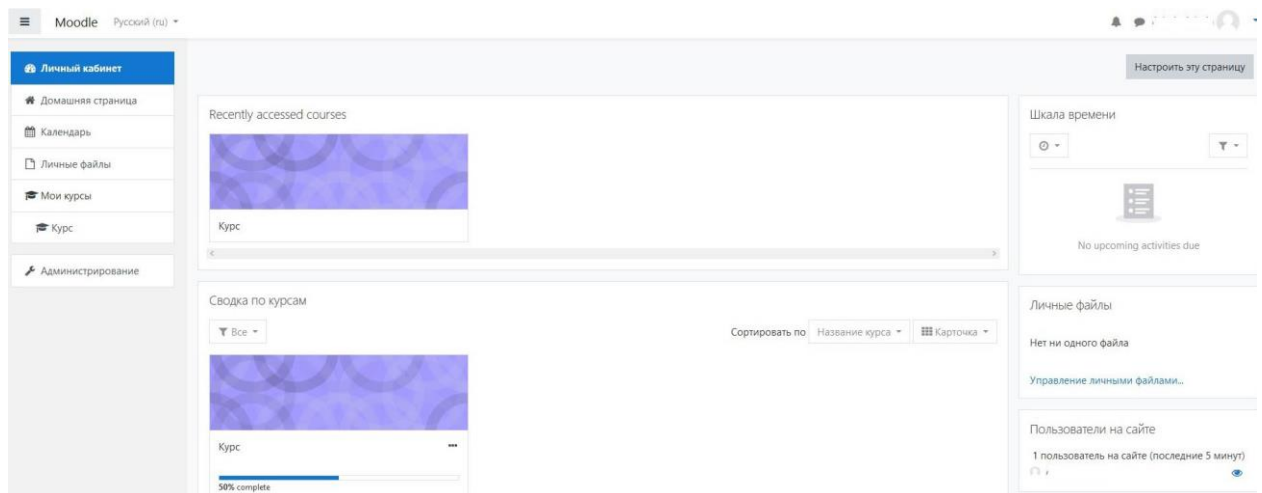


Рисунок 1.2 – Платформа Moodle

Серед переваг даної системи можна виділити наступні:

- безкоштовне використання сервісу;
- спрощення розповсюдження електронних матеріалів серед студентів;
- широкий вибір взаємодії зі студентами;
- контроль доступу до електронних матеріалів;
- наявність форуму.

До недоліків можна віднести наступне:

- складне адміністрування та налаштування, для якого необхідні технічні знання;

- дизайн можна вибрати тільки з готових рішень;
- в системі досить складний інтерфейс для роботи звичайних користувачів та викладачів, які завантажують свої матеріали для навчання.

1.2.3 Microsoft Teams

Платформа Microsoft Teams пропонує онлайн-простір для спільної роботи, в якому вчителі та учні можуть обмінюватися нотатками, чатами, зустрічами, завданнями та навіть програмами [3].

Спочатку запущена як платформа для підприємств, Microsoft Teams пройшла через безліч коригувань та оновлень і тепер є життєздатною платформою для використання у школах. Microsoft Teams for Education використовує ту ж базову безкоштовну платформу, яку може використовувати будь-який охочий, тільки адаптовану для шкіл (рис. 1.3).

Вчителі можуть створювати групи, які можуть бути цілим класом чи підгрупою всередині класу, та у будь-який час додавати учнів. У межах цих груп можна обмінюватися ресурсами, працювати над спільними проєктами, призначати завдання. SharePoint – це корисна система директорій, яка автоматично зберігає всі спільні файли, якими обмінюються в інших областях, наприклад, у чаті, для більш зручного пошуку.

Текстовий чат є ще однією важливою функцією, що дозволяє спілкуватися в режимі реального часу. Разом з тим також є і відеочат, в якому може бути присутній як весь клас, так і окремі співрозмовники.

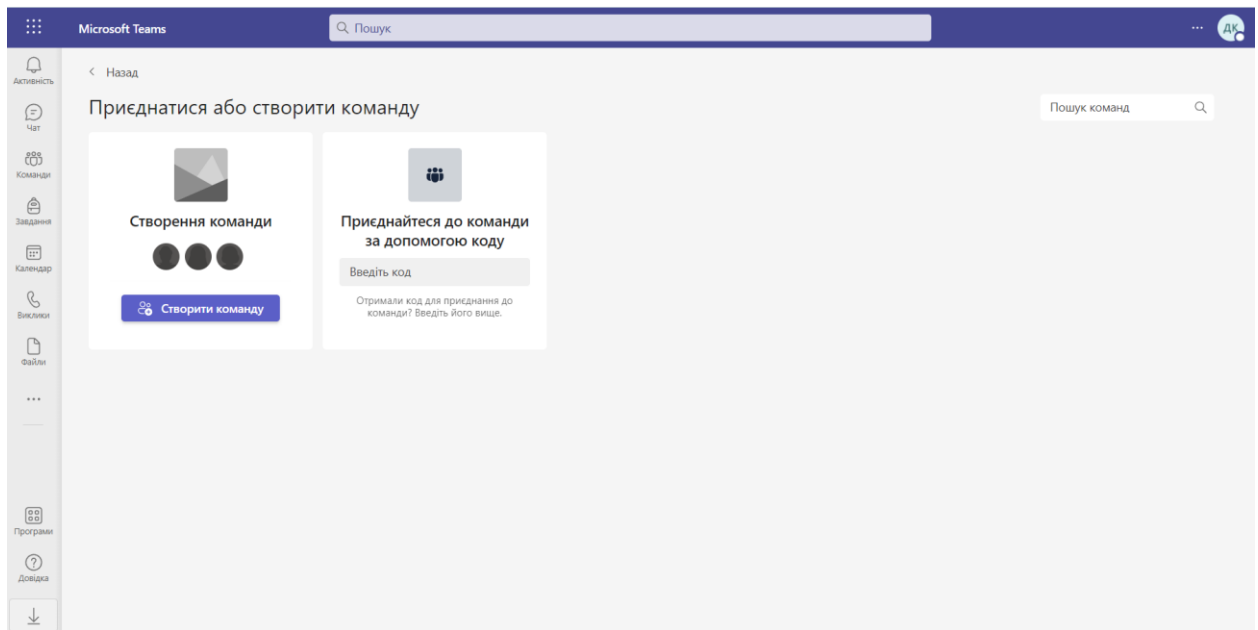


Рисунок 1.3 – Платформа Microsoft Teams

Серед переваг даної платформи можна виділити наступні:

- наявна безкоштовна версія;
- інтеграція всіх інструментів в одному місці;
- можливість планування дати проведення відео-конференції;
- можливість робити запис конференції та зберігати в окремій директорії;

- наявність додатку для ПК, планшетів та смартфонів.

До недоліків можна віднести наступне:

- заплутана структура файлів;
- ліміт на доступний обсяг зберігання файлів (до 2 ГБ);
- ліміт на кількість публічних та приватних груп;
- складності з налаштуванням доступу.

1.3 Аналіз засобів розробки

Для подальшої розробки web-системи слід проаналізувати відповідну клієнт-серверну архітектуру та системи керування базами даних. Далі наведено опис даних засобів розробки.

1.3.1 Клієнт-серверна архітектура

Сучасні сайти, додатки, сервіси стають все більш складними, динамічними, багатофункціональними. Вони повинні активно взаємодіяти з користувачем, бути здатними підлаштовуватися під мінливе середовище, що обумовлює постійний розвиток методів їх розробки, підтримки. Web-фреймворк [4] являє собою каркас, платформу для створення web-продуктів нового покоління, їх ефективної підтримки. Він надає базові модулі, на основі яких створюється гнучкий сайт з широкими можливостями модернізації, розширення функціоналу за рахунок приєднання додаткових додатків в майбутньому.

Стандартна архітектура фреймворку складається з трьох шарів:

- модель – відповідає за формування структури, правил бізнес-логіки;
- представлення – основною функцією є графічне відображення даних;
- контролер – реалізує зв'язок з користувачем, перетворюючи отриману від нього інформацію в команди для попередніх двох шарів.

Розробка сайту з використанням фреймворку отримала величезну популярність в останні роки, так як він зміг об'єднати переваги обох вищезгаданих методів:

- висока швидкість створення додатків, сервісів;
- можливість постійного розвитку продукту за рахунок інтеграції в структуру нових розширень;
- простий зручний супровід;
- максимальна безпека зберігання даних;
- застосування фреймворку для сайту забезпечує йому швидкість реагування, можливість витримувати значні навантаження.

Далі наведено опис найбільш популярних РНР фреймворків.

1.3.1.1 Symfony

Symfony – це РНР-фреймворк з відкритим кодом, розроблений для web-додатків та web-сайтів [5]. Даний фреймворк зарекомендував себе як

фундамент для створення різноманітних web-сервісів. Він містить багаторазові бібліотеки та компоненти PHP. Компоненти даного фреймворку використовуються у додатках таких проєктів, як Laravel та Drupal.

Переваги даного фреймворку:

- висока продуктивність;
- наявні багаторазові компоненти, які не пов'язані між собою;
- фреймворк на основі MVC;
- гнучка маршрутизація URI;
- підтримка кількох платформ баз даних;
- високорозвинений шаблонізатор Twig;
- якісна документація;
- стабільність.

Недоліки:

- фреймворк є досить складним в освоєнні незважаючи на наявність якісної документації.

1.3.1.2 Laravel

Laravel – це PHP-фреймворк, який дозволяє спростити аутентифікацію, кешування, маршрутизацію, архітектуру додатку та роботу з БД [6]. Даний фреймворк призначений для швидкої розробки додатків за допомогою MVC-архітектури. На сьогодні він є найбільш використовуваним з усіх PHP фреймворків.

Переваги:

- вбудовані механізми захисту;
- гнучка маршрутизація;
- легка міграція;
- висока продуктивність;
- надійна система шифрування даних;
- велика кількість вбудованих функціональних можливостей.

До недоліків цього фреймворку можна віднести досить складну для початківців документацію.

1.3.1.3 Yii2

Yii2 – це об’єктно-орієнтований компонентний фреймворк для PHP [7]. Фреймворк реалізує для використання MVC архітектурний шаблон та сприяє організації коду на основі цього шаблону. Даний фреймворк застосовує технологію «lazy loading», що сприяє високій швидкості роботи. Yii2 можна використовувати для розробки будь-якого виду web-додатків. Завдяки своїй основі компонентів, архітектури та складної підтримки кешування, фреймворк використовують для розробки великомасштабних проєктів, таких як портали, форуми, системи керування контентом (CMS), систем електронної комерції, RESTful web-сервісів тощо.

До переваг фреймворку можна віднести:

- просте встановлення та налаштування;
- фреймворк інтегрований з jQuery та поставляється з готовим набором AJAX функцій;
- наявний якісний генератор вихідного коду;
- фреймворк інтегрований з Codeception;
- потужні вбудовані механізми захисту, що сприяє в створенні безпечного додатку;
- якісна документація з багатьма прикладами використання компонентів;
- велика спільнота серед користувачів фреймворку.

До недоліків можна віднести:

- сильна зв’язність модулів у додатку;
- наявність різних антипатернів в проєкті;
- не дуже гнучка система маршрутизації.

1.3.2 Системи керування базами даних

Система керування базами даних (СКБД) – це спеціальний пакет програм, що забезпечує створення, супроводження і використання баз даних багатьма користувачами [8].

Основні характеристики СКБД:

- контроль за надлишковістю даних;
- несуперечливість даних;
- підтримка цілісності бази даних (коректність та несуперечливість);
- незалежність прикладних програм від даних;
- спільне використання даних;
- підвищений рівень безпеки.

Існує трирівнева система організації СКБД ANSI-SPARC, при якій існує незалежний рівень для ізоляції програми від особливостей представлення даних на нижчому рівні.

Рівні:

- зовнішній – представлення БД з точки зору користувача;
- концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем;
- внутрішній – фізичне представлення БД в комп'ютері.

Далі наведено найбільш популярні СКБД.

1.3.2.1 SQLite

SQLite – це вбудована кросплатформова БД, яка підтримує досить повний набір команд SQL і доступна у вихідних кодах [9]. Дана СКБД базується на файлах, що надає обширний набір інструментів для роботи з нею, у порівняння з іншими системами.

До переваг SQLite можна віднести:

- наявність файлової структури;

- повна база даних SQLite зберігається на одному крос-платформному диску;
- SQLite є автономним, що означає відсутність зовнішніх залежностей;
- система поставляється з нульовою конфігурацією, що означає відсутність у необхідності адміністрування.

Недоліком SQLite є відсутність системи користувачів – більші СКБД включають до свого складу системи керування правами доступу користувачів. Даний недолік обумовлюється використанням СКБД у невеликих додатках.

1.3.2.2 MySQL

MySQL – є системою керування базами даних, що дозволяє керувати реляційними базами даних [10]. Це програмне забезпечення з відкритим вихідним кодом за підтримки Oracle. MySQL є надійною, масштабованою та швидкою СКБД, яка може працювати на різних платформах, таких як Linux, Unix, Windows тощо.

Перевагами MySQL є:

- підтримка великої кількості типів таблиць, що робить дану СКБД досить гнучкою;
- вбудована велика кількість функцій, що забезпечують безпеку;
- можливість роботи з великими обсягами даних;
- масштабованість;
- є досить простою у використанні;
- висока швидкість роботи.

Недоліками MySQL є:

- деякі операції реалізовані менш надійно, ніж в інших СКБД;
- повільна розробка.

1.3.2.3 PostgreSQL

PostgreSQL – це об’єктно-реляційна СКБД, що базується на мові SQL та підтримує багаточисельні можливості [11]. Дана СКБД чудово підходить для використання зі складними операціями з великими обсягами даних. PostgreSQL є розширюваною системою, робота якої базується на catalog-driven підході. Вона зберігає інформацію не тільки про таблиці та стовпці, але і про типи даних, типи індексів, функціональні мови тощо. Також СКБД підтримує високопаралельні обчислення та NoSQL.

Переваги PostgreSQL:

- наявність функцій керування паралельним доступом за допомогою багатоверсійності (MVCC);
- можливість розробити власні плагіни та налаштувати СКБД під свої потреби;
- відкритий вихідний код;
- велике співтовариство;
- часті оновлення;
- дозволяє використовувати як об’єктно-орієнтовані, так й реляційні бази даних.

Недоліки PostgreSQL:

- продуктивність – виконання простих операцій читання може бути значно довшим ніж у інших СКБД;
- популярність – дана СКБД не є дуже популярною через складність її використання.

1.4 Порівняння і вибір інструментів для розробки

Проаналізувавши найбільш популярні фреймворки, порівнявши їх переваги та недоліки, для подальшої реалізації web-додатку було обрано фреймворк Yii2 через його високу продуктивність, простоту використання та великого вбудованого функціоналу.

Для клієнтської частини системи обрана бібліотека JQuery, оскільки володіє низьким порогом входження, має кросбраузерну сумісність, і надає можливість нарощувати функціонал, підключаючи модулі [12].

У якості СУБД для web-додатку було обрано MySQL з декількох причин:

- можливість використання з Yii2 фреймворком;
- високий рівень захисту;
- простота використання;
- швидкість роботи.

1.5 Постановка задачі

В процесі виконання даної кваліфікаційної роботи, було розроблено технічне завдання, згідно з яким була визначена орієнтовна структура, дизайн та функціонал додатку.

Інтерфейс додатку повинен відповідати наступним сучасним вимогам:

- просте та зрозуміле управління;
- сучасний дизайн;
- логічна архітектура сайту;
- адаптивність під екрани планшетів та смартфонів.

Візуальне оформлення web-системи повинно бути витримане та не обмежувати інформативності: додаток має бути зручним для використання користувачами.

Всі дані, що генерують користувачі сайту, повинні зберігатися у базі даних.

Додаток повинен мати наступний функціонал:

- реєстрація користувачів за допомогою адміністратора;
- підтримка декількох типів користувачів (учень та вчитель);
- створення предметів за допомогою адміністратора;
- перегляд призначених предметів;

- додавання/редагування/видалення/оцінювання завдань до предмета (для вчителя);
- перегляд та виконання завдань до предмета (для учня);
- окремий чат для кожного предмета;
- редагування розкладу (для вчителя);
- перегляд розкладу (для учня);
- перегляд журналу оцінок (для учня);
- підтримка нотифікацій;
- перегляд профілю користувача.

Для підтримки додатку від персоналу не повинно вимагатися спеціальних технічних навичок, знання технологій або програмних продуктів, за винятком загальних навичок роботи з персональним комп'ютером і стандартним web-браузером (Chrome, Firefox, Opera, Safari).

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Розробка специфікації вимог

Специфікація вимог [13] створюється для однозначного розуміння вимог до програмного продукту, отримання закінченого опису поведінки web-сервісу, який потрібно розробити, для можливості точної оцінки етапів, планів і термінів проєкту. Вона описує функціональні та нефункціональні вимоги до web-системи.

У специфікації вимог наводиться структура ПЗ, вимоги до функцій, якості та документації. Специфікуються також системні вимоги, нефункціональні (додаткові) вимоги та вимоги до взаємодії з іншими компонентами й платформами (БД, СКБД, мережа й інше).

2.1.1 Класи та характеристики користувачів

У таблиці 2.1 наведені класи та опис користувачів.

Таблиця 2.1 – Класи та опис користувачів

Клас	Опис
Адміністратор	Клас користувачів, який має доступ до адміністративної частини. Має можливість створювати/редагувати користувачів та предмети.
Учень	Клас користувачів, яких реєструє адміністратор. Має можливість переглядати призначені предмети та виконувати завдання, переглядати розклад і журнал з оцінками.

Продовження таблиці 2.1

Вчитель	Клас користувачів, яких реєструє адміністратор. Має можливість переглядати призначені предмети, редагувати розклад/журнал оцінок, створювати та редагувати завдання до предметів і оцінювати результат виконання цих завдань.
---------	---

2.1.2 Операційне середовище

Вимоги до операційного середовища є наступні:

- web-система працює з усіма Інтернет-браузерами, окрім браузерів Internet Explorer 6 і нижче;
- web-система встановлена на сервері, що працює під управлінням Linux та Apache HTTP Server;
- web-система повинна надавати доступ користувачам як з внутрішньої мережі, так і через Інтернет-підключення з будинків користувачів.

2.1.3 Дизайн та реалізація

Вимоги щодо обмеження дизайну та реалізації є наступні:

- система повинна використовувати БД MySQL;
- весь код HTML повинен відповідати стандарту 5.0;
- усі сценарії повинні бути написані на PHP, JS;
- верстка сайту повинна бути адаптивною та кросбраузерною.

2.1.4 Інтерфейси користувачів

Вимоги до інтерфейсів користувачів є наступні:

- інтернет-сторінки повинні надавати повну можливість навігації по системі;
- екрани виводу користувачів повинні бути стандартних розмірів

різноманітних пристроїв (монітори ПК, смартфони, планшети).

2.1.5 Продуктивність

Вимоги до продуктивності системи є наступні:

- система повинна підтримувати тисячу підключень одночасно;
- всі інтернет-сторінки повинні завантажуватися не більше 10 секунд по повільному модемному з'єднанню 500 кб/сек;
- завантажування відповідей на запити на екрані повинна займати не більше 5 секунд після того, як користувач виконав запит.

2.1.6 Безпека

Можна виділити наступні вимоги до безпеки у системі:

- всі мережеві з'єднання, що включають приватну інформацію клієнтів повинні шифруватися;
- всі поля, що відправляються на сервер, та мають приватні дані, повинні шифруватися спеціальними методами, передбачені мовою програмування.

2.1.7 Атрибути якості ПЗ

Атрибутами якості даного ПЗ є наступні:

- web-сайт повинен бути доступним користувачам 99,9% часу;
- якщо зв'язок перервався до того, як користувач підтвердив операцію, система повинна надати можливість при повторному підключенні продовжити заповнення інформації.

2.1.8 Допущення і залежності

До залежностей та допущень системи можна віднести наступне:

- система функціонує в штатному режимі тільки за умови, що все необхідне для неї обладнання знаходиться у робочому стані;

– система функціонує в штатному режимі тільки за умови, що між її вузлами та призначеної для користувача стороною існує стабільний (тобто не переривається протягом сесії використання системи) канал зв'язку.

2.1.9 Системні характеристики

Можна виділити наступні вимоги до системних характеристик:

- web-система працює з усіма Інтернет-браузерами, окрім браузерів Internet Explorer 6 і нижче;
- web-система встановлений на сервері, що працює під управлінням Linux та Apache HTTP Server;
- web-система повинен допускати доступ користувачів як з внутрішньої мережі компанії, так і через Інтернет-підключення з будинків користувачів;
- система керування базами даних: у якості системи керування базами даних повинні виступати MySQL версії 5.7 і вище;
- інтерфейси обміну даних: система повинна надавати інтерфейси обміну інформацією для клієнтських додатків – web-інтерфейсу сайту і мобільних клієнтів.

2.2 Моделювання системи

Першим кроком процесу моделювання ПЗ є визначення структурних компонентів та зв'язків між ними. Результатом уточнення структури може бути наведений на структурних та функціональних схемах.

2.2.1 Структура системи

Додатки, що реалізовані засобами Yii фреймворку, організовані згідно шаблону проєктування MVC, де:

- моделі є компонентами, що відповідають за дані та бізнес логіку;
- представлення є компонентами, що відповідають за взаємодію з користувачем;

– контролери приймають вхідні дані від користувача і перетворюють їх в зрозумілий для моделей формат і команди.

До структурної схеми Yii-додатків можна віднести елементи, що наведено у таблиці 2.2.

Таблиця 2.2 – Елементи структурної схеми Yii-додатків

Елемент	Опис
Вхідні скрипти	PHP скрипти, які доступні кінцевому користувачеві програми. Вони відповідальні за запуск та обробку вхідного запиту.
Додатки	Об'єкти, що є глобально доступними. Вони здійснюють коректну роботу різних компонентів програми та їх координацію для обробки запиту.
Компоненти програми	Об'єкти, що зареєстровані в додатку і надають різні можливості для обробки поточного запиту.
Модулі	Самодостатні пакети, які містять повністю всі засоби для MVC. Додаток може бути організовано за допомогою декількох модулів.
Фільтри	Код, який повинен бути виконаний до і після обробки запиту контролерами.
Віджети	Об'єкти, які можуть включати до себе представлення. Вони можуть містити різну логіку і бути використані в різних представленнях.

На рисунку 2.1 наведено візуальний вигляд структурної схеми

програми.

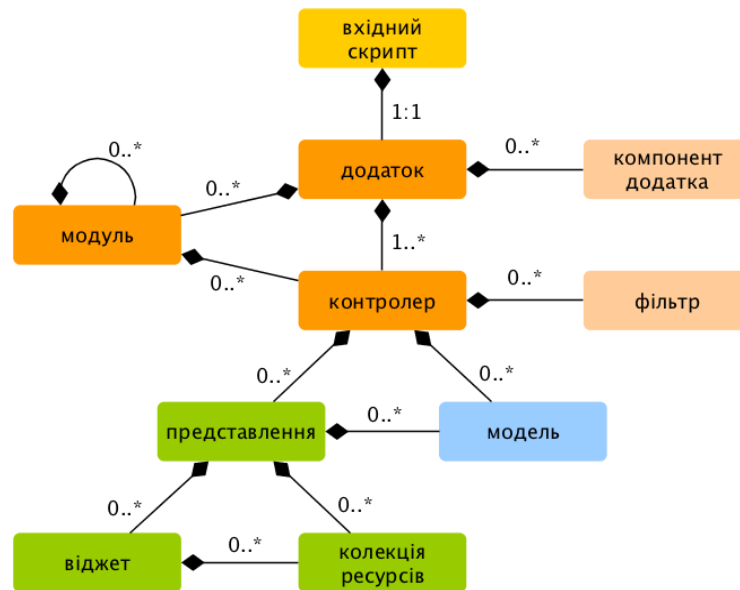


Рисунок 2.1 – Структурна схема програми

Також, Yii використовує front-контролер, або додаток (application), який інкапсулює контекст обробки запиту. Додаток збирає інформацію про запит і передає її для подальшої обробки до відповідного контролера.

На рисунку 2.2 зображено діаграму, що відображає структуру додатка Yii.

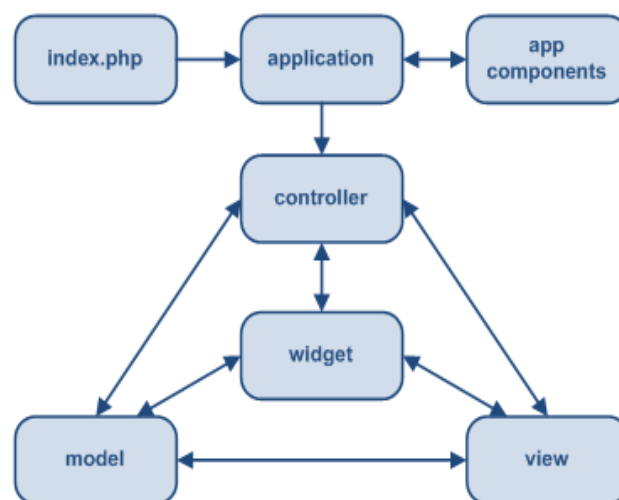


Рисунок 2.2 – Структура додатка Yii

2.2.2 Структура бази даних

База даних містить таблиці, кожна з яких є сутністю предметної області. Кожна таблиця має свої атрибути, а ті, в свою чергу, домени. Нижче зазначені основні сутності бази даних системи:

- «user» – містить дані користувача;
- «course» – містить дані стосовно предмета;
- «course_chat_message» – зберігає повідомлення чату;
- «course_task» – містить інформацію стосовно завдання до предмета;
- «course_task_submission» – містить інформацію щодо результату виконання задачі;
- «course_time_slot» – містить інформацію щодо проведення предмета у запланований час;
- «course_student_assignment» – містить інформацію стосовно призначеного до предмета учня;
- «student_mark» – містить інформацію щодо оцінки учня;
- «attachment» – містить інформацію щодо прикладеного файлу до певного запису
- «students_group» – містить інформацію щодо класу учнів;
- «students_group_assignment» – містить інформацію щодо призначеного до класу учня;
- «notification» – містить інформацію щодо нотифікації користувача.

Загальна схема бази даних зображена на рисунку 2.3.

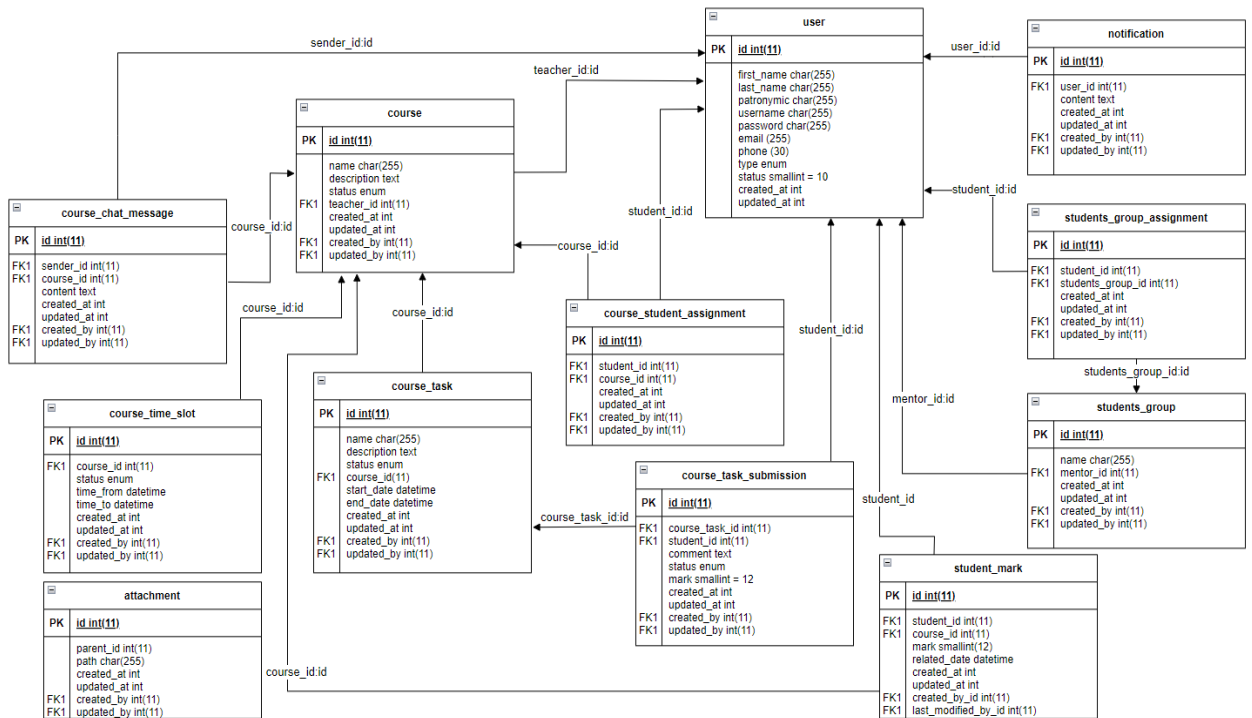


Рисунок 2.3 – База даних системи

Слід відзначити, що фреймворк автоматично згенерує більшу частину сутностей, які будуть призначатись для реєстрації та прав доступу.

2.2.3 Проектування зовнішнього відображення сайту

Зовнішній вигляд сайту є досить важливою складовою. Сайт має бути адаптивним, тобто нормально відображати структуру сторінок на різних пристроях.

В цілому, сайт має наступні складові:

- верхня частина – область, яка містить навігаційну панель;
- основна частина – область, що містить відповідний контент обраної сторінки.

2.2.3.1 Макет сторінки авторизації

Для подальшого користування системи, користувач повинен авторизуватися до неї. Авторизація відбувається на відповідній сторінці входу. Макет даної сторінки наведений на рисунку 2.4.

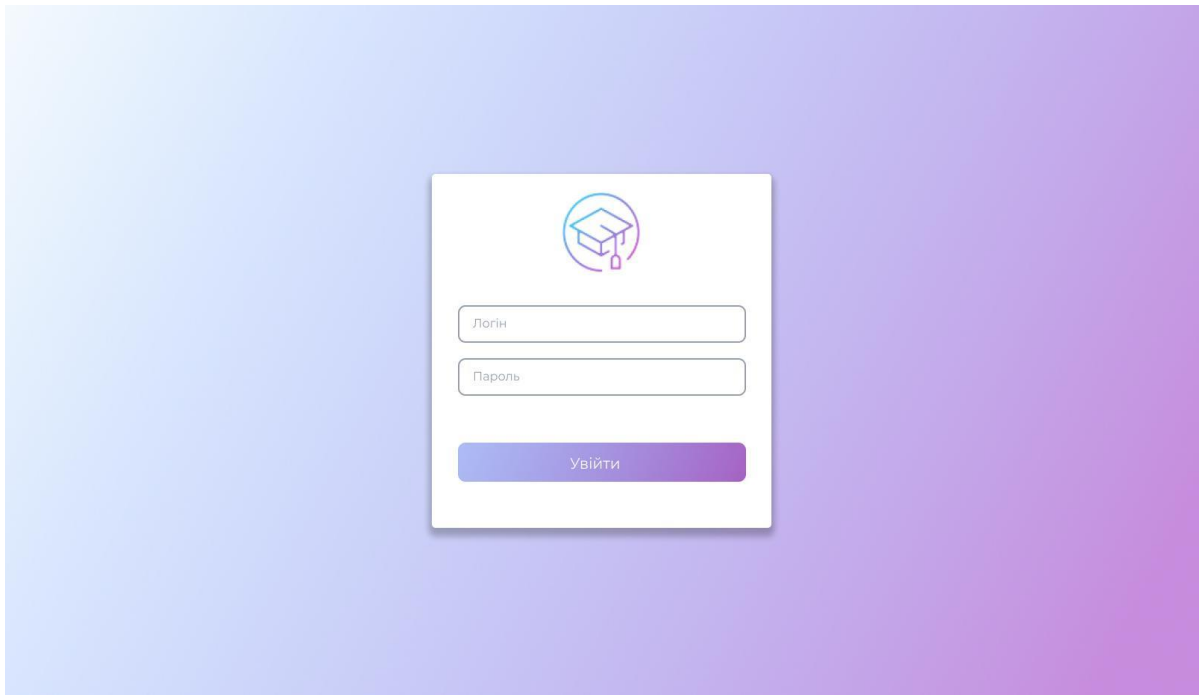


Рисунок 2.4 – Макет сторінки авторизації

2.2.3.2 Макет сторінки предметів

Сторінку «Предмети» можна побачити обравши однойменну кнопку в навігаційному меню. Дана сторінка відкривається за замовчуванням, коли користувач заходить до системи. На рисунку 2.5 представлений макет цієї сторінки.

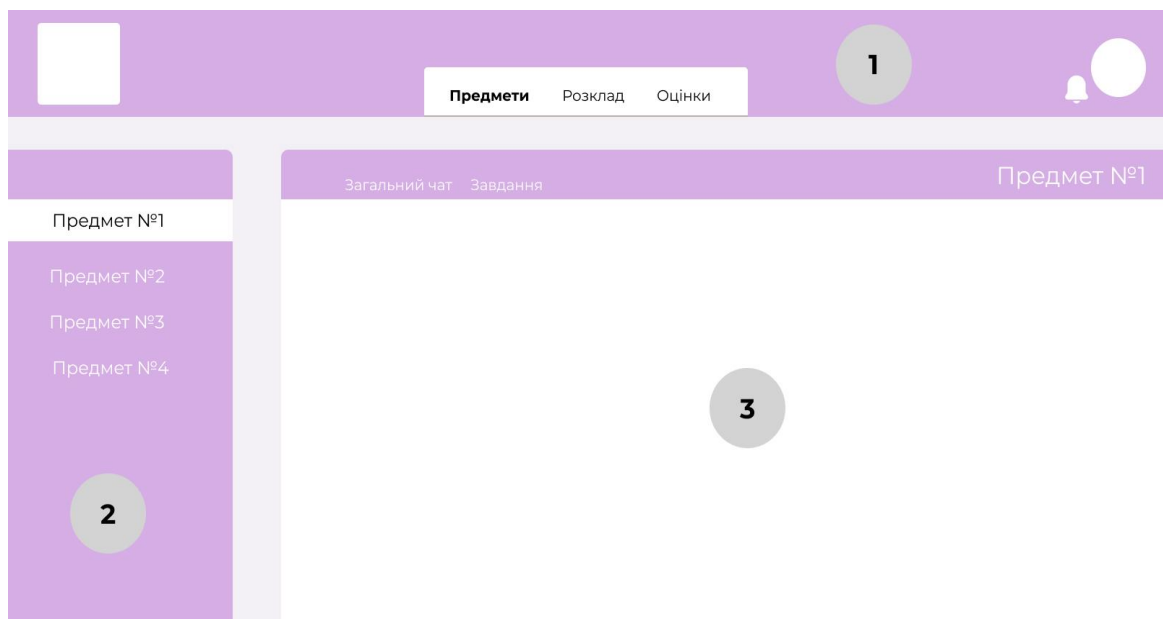


Рисунок 2.5 – Макет сторінки предметів

Даний макет містить наступні елементи:

- навігаційна панель;
- список всіх призначених предметів;
- деталі обраного предмета.

2.2.3.3 Макет сторінки розкладу

Сторінку «Розклад» можна побачити обравши однойменну кнопку у навігаційному меню. Макет даної сторінки наведений на рисунку 2.6.

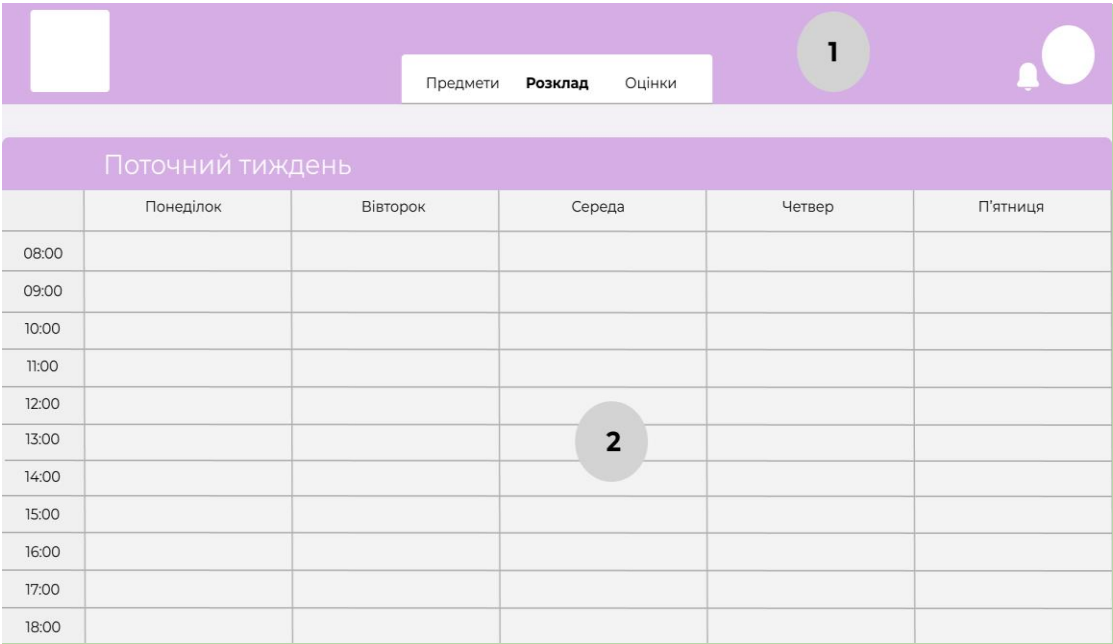


Рисунок 2.6 – Макет сторінки розкладу

Даний макет містить наступні елементи:

- 1) навігаційна панель;
- 2) календар типу «Тиждень».

2.2.3.4 Макет сторінки журналу з оцінками

Сторінку «Оцінки» можна побачити обравши однойменну кнопку в навігаційному меню. Макет даної сторінки наведений на рисунку 2.7.

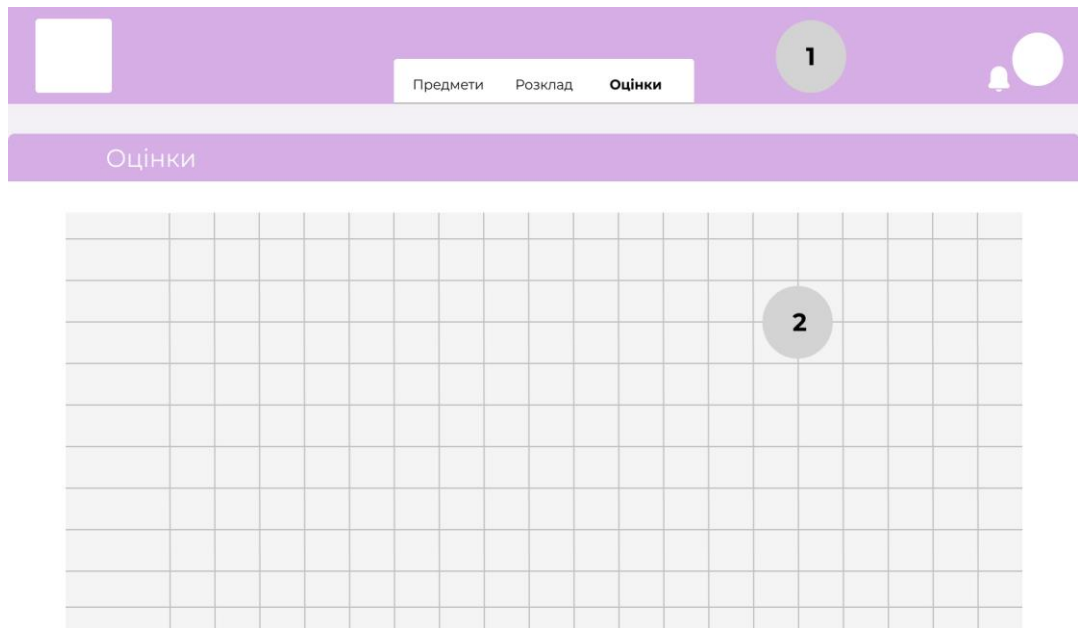


Рисунок 2.7 – Макет сторінки з оцінками

Даний макет містить наступні елементи:

- 1) навігаційна панель;
- 2) таблиця з оцінками.

2.2.3.5 Макет сторінки профілю користувача

Дану сторінку можна побачити відкривши деталі користувача в системі. Макет цієї сторінки наведений на рисунку 2.8.

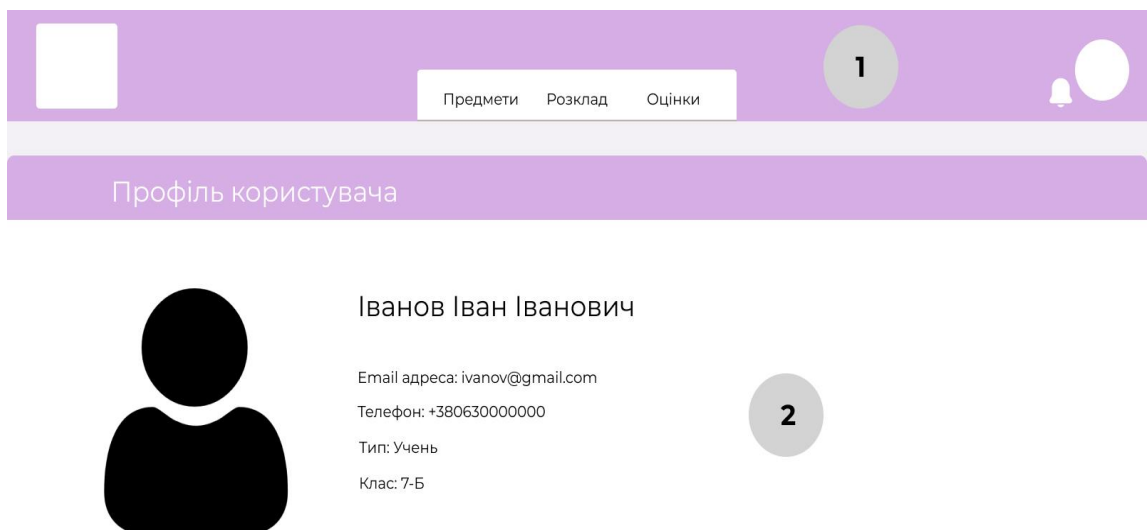


Рисунок 2.8 – Макет профілю користувача

2.2.3.6 Макет сторінки адміністрування

Дана сторінка доступна лише для адміністратора системи. Макет цієї сторінки можна побачити на рисунку 2.9.

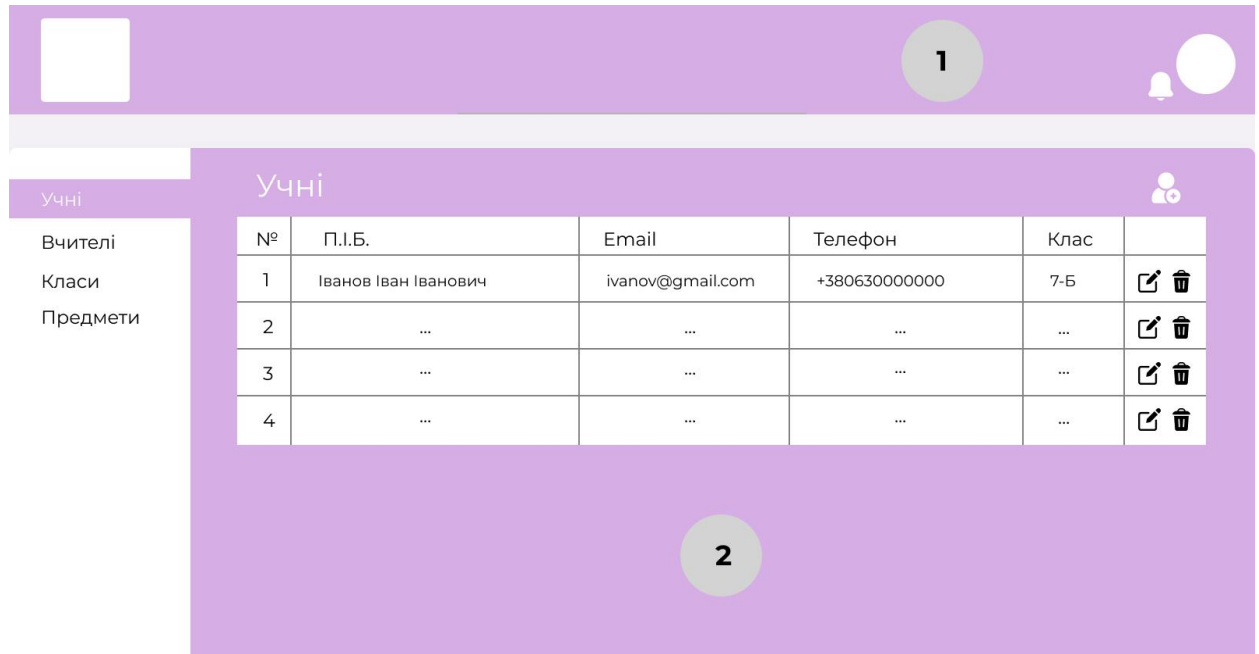


Рисунок 2.9 – Макет сторінки адміністрування

Даний макет містить наступні елементи:

- 1) навігаційна панель;
- 2) таблиці користувачів та предметів.

2.2.4 Діаграма використання

Діаграма використання [14] показує дієвих осіб (людей або інших користувачів системи), варіанти використання (сценарії використання системи) та їх взаємодію. Варіанти використання є описом типових елементів взаємодії користувачів системи з самою системою. Вони відповідають зовнішньому інтерфейсу системи та визначають форму вимог до того, що має робити система.

На рисунку 2.10 зображена діаграма використання системи учнем.

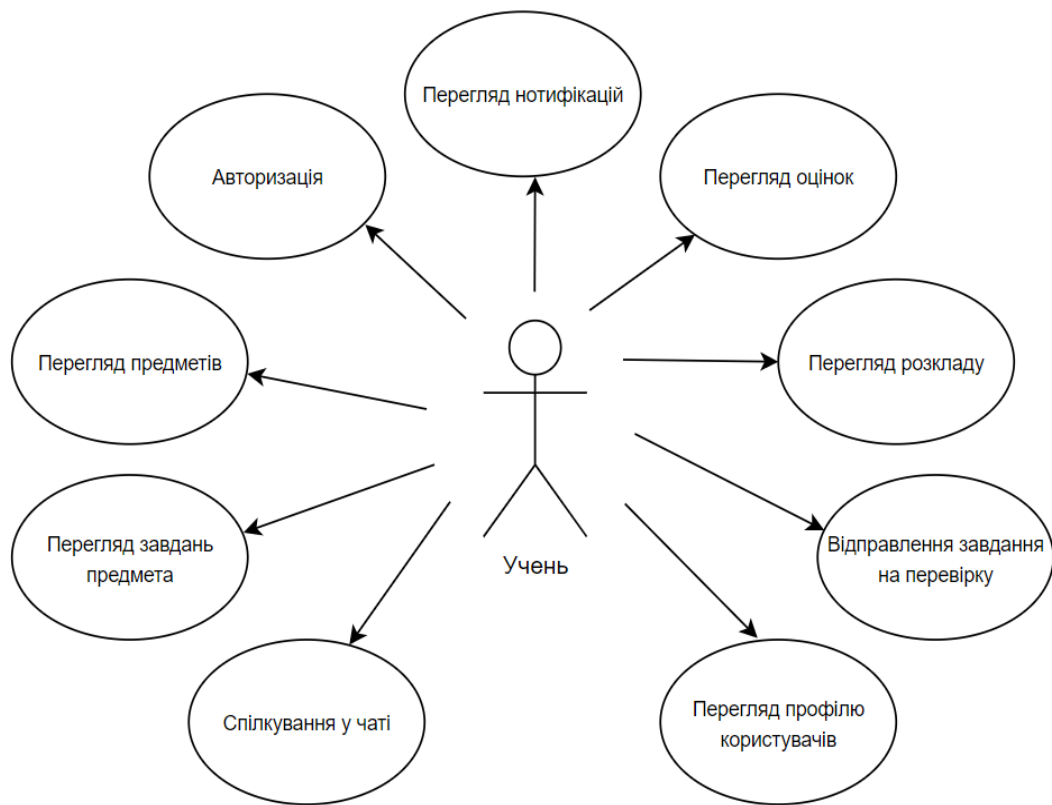


Рисунок 2.10 – Діаграма використання системи учнем

На рисунку 2.11 зображена діаграма використання системи вчителем.

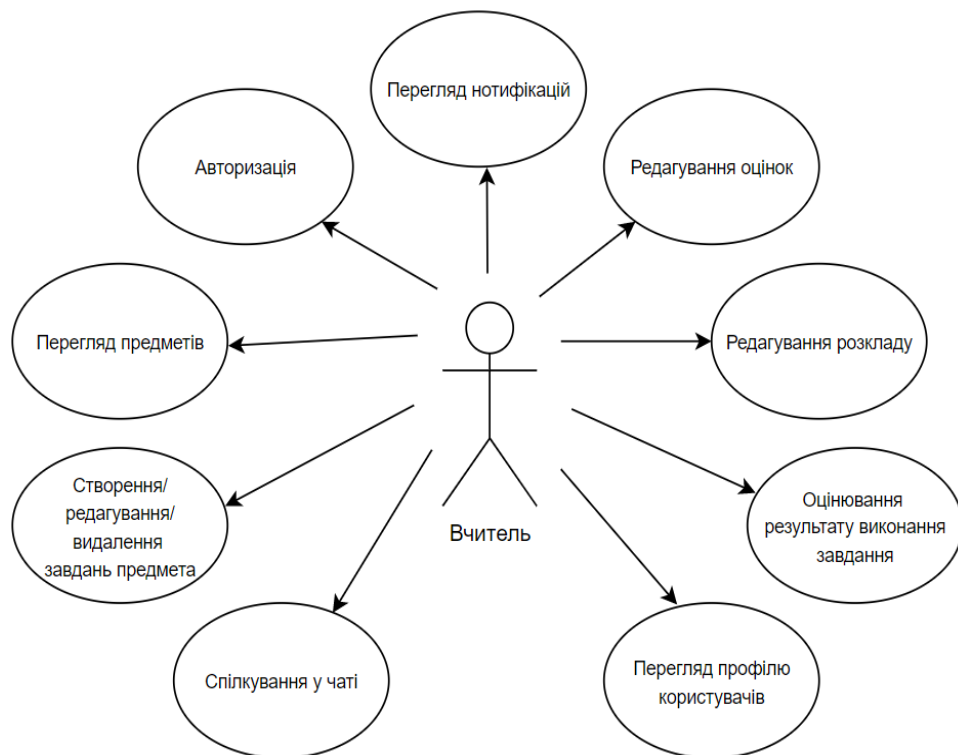


Рисунок 2.11 – Діаграма використання системи вчителем

2.2.5 Діаграма послідовності

Діаграма послідовності [15] графічно демонструє порядок взаємодії певних об'єктів програми у часі. Як правило, в цій діаграмі демонструється, як саме користувачі (актори з діаграми варіантів використання) взаємодіють з іншими компонентами програми під час реалізації тих чи інших варіантів використання програми, та як при цьому взаємодіють інші компоненти програмної системи.

Діаграми послідовності є одним із способів формалізації сценаріїв використання. Її перевага заключається в тому, що на ранніх стадіях опису сценаріїв можливо з'ясувати склад взаємодіючих компонентів та описати потік повідомлень від одних компонентів до інших. Ці компоненти та потоки повідомлень в подальшому будуть трансформовані в конкретні класи (об'єкти), методи цих об'єктів.

Наприклад, послідовність дій для прецеденту «Авторизація», показує послідовність дій для авторизації у web-системі:

1. Користувач заходить на сайт;
2. Заповнення форми;
3. Якщо дані введені не вірно, то виникає альтернативний потік:
 - 3.1. Скористатися функцією відновлення паролю;
 - 3.2. Відновити пароль;
4. Повідомлення про успішну авторизацію.

На рисунку 2.12 продемонстрована діаграма послідовності дій для прецеденту «Авторизація».

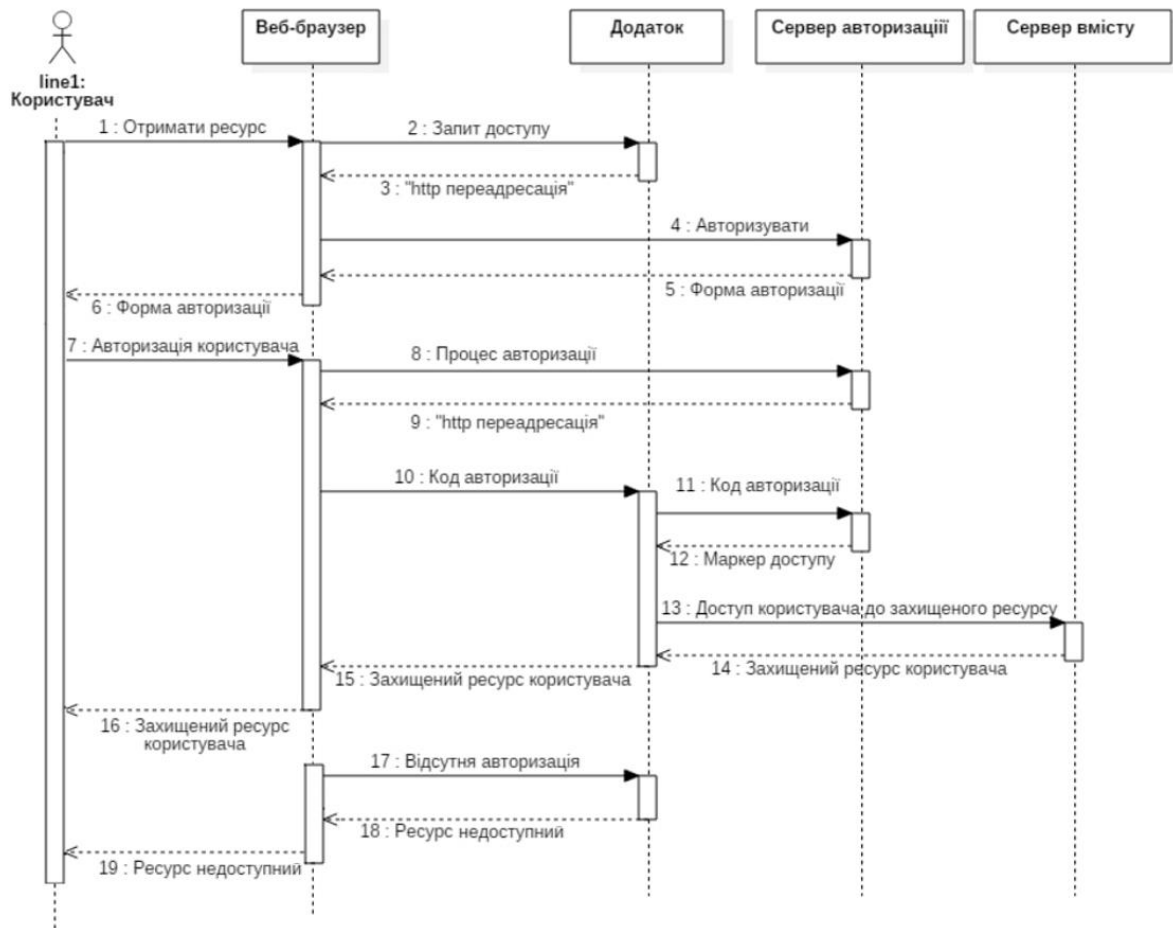


Рисунок 2.12 – Діаграма послідовності дій для прецеденту «Авторизація»

2.2.6 Діаграма розгортання

Діаграма розгортання [16] відображає обчислювальні вузли під час роботи програми, компоненти та об'єкти, що виконуються на цих вузлах.

Кожен вузол на діаграмі являє собою певний тип обчислювального пристрою, в більшості випадків частина апаратури. Ця апаратура може бути простим пристроєм або датчиком, а може бути й основним комп'ютером.

Діаграма розгортання моделює фізичне розгортання артефактів на вузлах.

На рисунку 2.13 наведена діаграма розгортання розроблюваної web-системи.

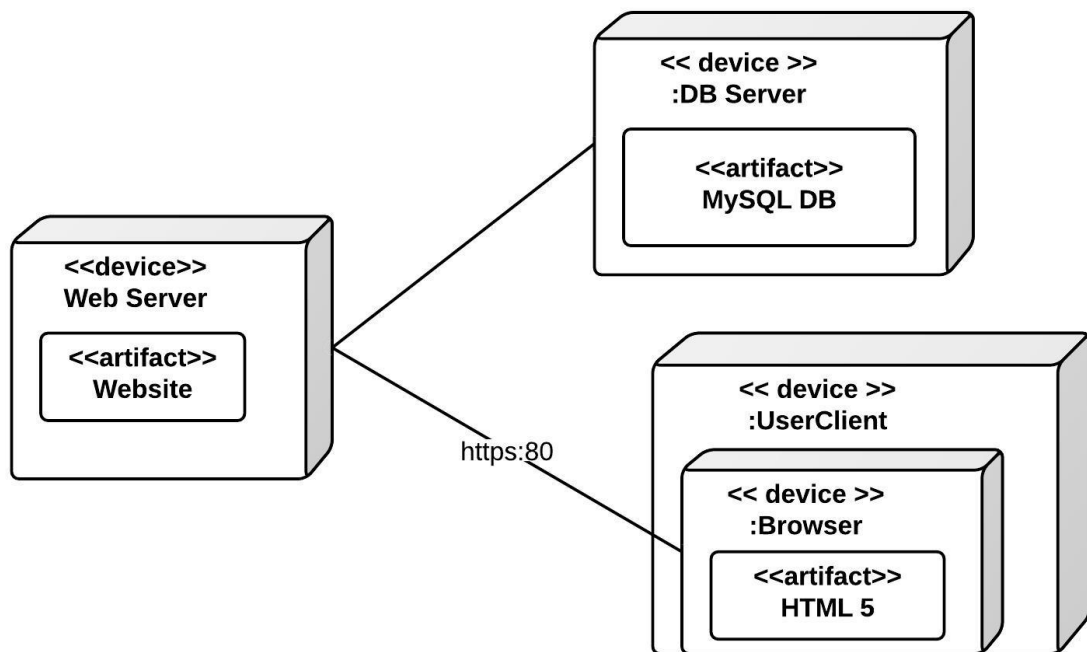


Рисунок 2.13 – Діаграма розгортання web-системи

Всі дані нашої системи зберігаються в базі даних, що уявляє собою компонент MySQL DB, який знаходиться в блоці DB Server. Цей блок не співпрацює напряму з комп'ютером користувачів, адже це може бути небезпечно з точки зору захисту інформації та її цілісності. Блок DB Server співпрацює з блоком Web Server за допомогою TCP/IP протоколу. Зв'язок між програмами системи та базою даних відбувається за допомогою протоколу HTTP.

Компонент може мати більше одного інтерфейсу, у цьому випадку видно, які компоненти взаємодіють з кожним інтерфейсом.

3 РОЗРОБКА СИСТЕМИ

3.1 Засоби розробки

Досить важливим етапом у розробці системи є правильний вибір засобів її розробки. Правильно обрані засоби сприяють більш гнучкому, швидкому та комфортному процесу розробки системи. Далі наведено опис таких засобів.

3.1.1 Мова програмування

Для генерації HTML-сторінок додатку на web-сервері та для роботи з базою даних було вирішено використовувати мову PHP, так як вона вже досить довго займає провідні позиції у сфері web-програмування. PHP-скрипти виконуються на сервері та генерують HTML-код, що повертається на клієнтську частину. Фрагменти PHP-коду можливо також додавати в HTML-код, що надає певної гнучкості.

PHP доречно використовувати для вирішення широкого спектра завдань завдяки своїй багатій функціональності. Також дана мова програмування має реалізацію ефективних засобів безпеки, в тому числі ряд надійних механізмів шифрування. Завдяки тому, що сценарії компілюються на стороні сервера, їх вихідний текст не можна переглянути в браузері. Крім усіх вищезазначених переваг, PHP підтримує велику кількість баз даних (більше 20 видів).

3.1.2 Програмне середовище

При розробці web-додатку варто приділити час обранню середовища програмування для втілення проєкту. Основна причина вибору інтегрованого середовища розробки (IDE) залежить від мови програмування й цільової платформи.

Оскільки у якості основної мови було обрано PHP, то було вирішено використовувати IDE для PHP – JetBrains PhpStorm [17]. Дане середовище розробки має наступні переваги:

- забезпечує найкраще автодоповнення коду;
- аналізує код в реальному часі;
- надає швидку навігацію по коду;
- є кросплатформним середовищем розробки;
- наявна можливість налаштовувати різні кольори для виділення різних мов програмування;
- містить набір інструментів для рефакторингу коду;
- робота з віддаленими файлами через протоколи FTP, SFTP, FTPS і т.д.;
- вбудована підтримка різних систем контролю версій (Git, SVN, Mercurial);
- підтримка фреймворків (MVC view для Symfony2, Yii) і спеціалізовані плагіни для провідних PHP фреймворків (Symfony, Magento, Drupal, Yii, CakePHP і багато інших);
- містить інструменти роботи з базами даних, SQL-редактор.

3.1.3 Допоміжні компоненти

Процес розробки web-додатку передбачає використання локального сервера. Цей інструмент надає можливість переглянути роботу майбутнього сайту та здійснити тестування проєкту, не завантажуючи його на хостинг. Для виконання вищезазначених дій було вирішено використовувати Open Server Panel [18]. Даний інструмент є портативним програмним комплексом, який створений для того, щоб сприяти розробці, налагодженні та тестуванні сайтів безпосередньо на комп'ютері (навіть якщо на ньому немає інтернету) під керуванням ОС Windows.

Ключовими перевагами локального сервера Open Server є:

- можливість одночасної роботи з будь-якою кількістю сайтів;

- не потребує встановлення (портативність);
- робота на локальному/мережевому/зовнішньому IP;
- захист сервера від зовнішнього доступу;
- підтримка SSL без додаткового налаштування;
- набір популярних сторонніх розширень PHP;
- планувальник завдань.

3.2 Формування таблиць бази даних

Для початку необхідно створити БД, використовуючи клієнтську програму phpMyAdmin, та підключити її до обраного програмного середовища (рис. 3.1).

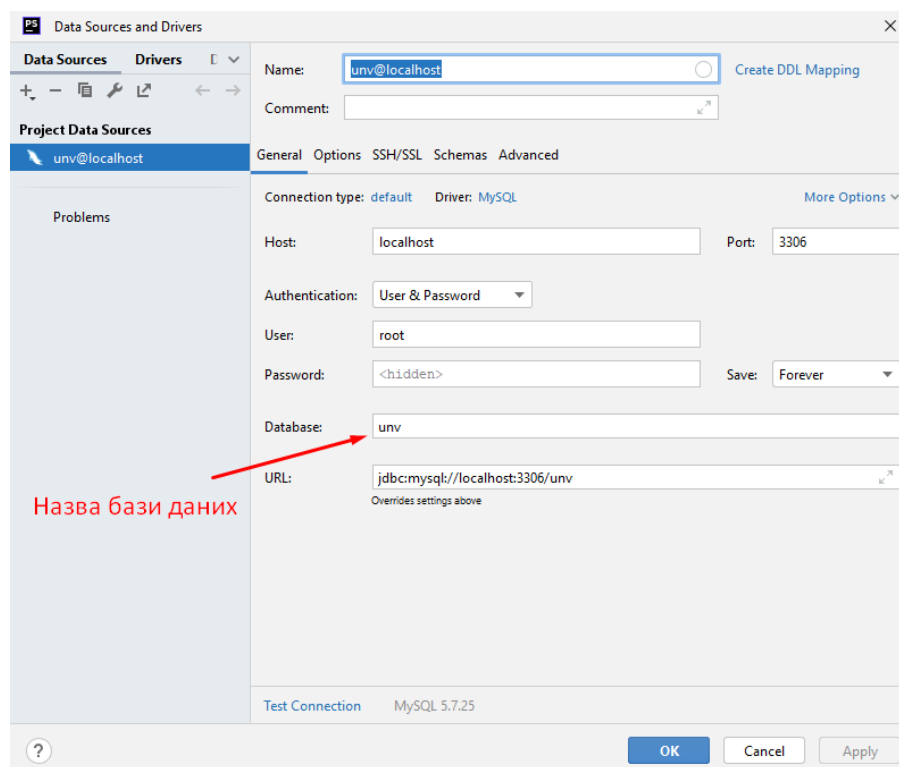


Рисунок 3.1 – Підключення БД до IDE

Після підключення БД до IDE, необхідно зв'язати web-додаток зі створеною БД (рис. 3.2).

```
'db' => [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=unv',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
]
```

Рисунок 3.2 – Налаштування підключення до БД

Фреймворк Yii2 надає засоби для здійснення змін структури БД використовуючи термінал PHPStorm. Дані засоби дозволяють робити наступне з міграціями БД:

- створювати нові міграції;
- застосовувати міграції;
- скасовувати міграції;
- застосовувати міграції повторно;
- показувати історію та статус міграцій.

Всі дані інструменти доступні за допомогою команди `yii migrate`. Таким чином, ввівши команду `yii migrate/create` і вказавши назву таблиці, сформується файл міграції, де необхідно додати код для реалізації таблиці та її атрибутів.

Створення сутностей відбувається подібно SQL-запиту в СКБД, який повністю описує сутності та їх атрибути, а також ключі та зв'язки. На рисунку 3.3 наведено приклад створення таблиці «course_task» в БД. Дана таблиця наведена на рисунку 3.4.

```
$this->createTable('{{course_task}}', [
    'id' => $this->primaryKey(),
    'name' => $this->string()->notNull(),
    'course_id' => $this->integer()->notNull(),
    'description' => $this->text(),
    'start_date' => $this->dateTime(),
    'end_date' => $this->dateTime(),
    'status' => "ENUM('Draft','Active','Archived') NOT NULL DEFAULT 'Draft'",
    'created_at' => $this->integer()->notNull(),
    'updated_at' => $this->integer()->notNull(),
    'created_by_id' => $this->integer()->notNull(),
    'last_modified_by_id' => $this->integer()->notNull()
]);
```

Рисунок 3.3 – Механізм створення таблиці «course_task» в БД

	id	name	course_id	description	start_date	end_date	created_at	updated_at	created_by	updated_by	status
1	5	Test	8	testsetsat	2022-05-08 23...	2022-05-08 ...	1652032683	1652054788	3		3 Active
2	7	Test	8	testasfdsfa@sdf...	2022-05-07 07...	2022-05-07 ...	1652034985	1652054782	3		3 Active
3	8	test	8	test	2022-05-20 11...	2022-05-19 ...	1652038036	1652038036	3		3 Draft
4	11	Драфт за...	7	Тестовий опис з...	2022-05-13 11...	2022-05-13 ...	1652050132	1653170501	3		3 Active
5	15	Завдання...	9	Опис завдання	2022-05-10 07...	2022-05-10 ...	1652054869	1652054876	3		3 Active
6	16	1235465	7	13546	2022-05-22 10...	2022-05-22 ...	1653170640	1653170647	3		3 Active
7	17	Завдання...	7	test	2022-05-22 09...	2022-05-23 ...	1653170963	1653170972	3		3 Active
8	18	ПК №3	7	Test	2022-05-22 09...	2022-05-22 ...	1653172179	1653172179	3		3 Active
9	20	ASG-9151...	8	TEST	2022-05-22 22...	2022-05-23 ...	1653248116	1653248116	3		3 Active

Рисунок 3.4 – Таблиця «course_task»

Для того, щоб зв'язати між собою таблиці «course_task» та «course», необхідно додати зовнішній ключ до «course_task» сутності та додати його відповідний індекс (рис. 3.5).

```

$this->createIndex(
    'idx-course_task-course_id',
    'course_task',
    'course_id'
);

$this->addForeignKey(
    'fk-course_task-course_id',
    'course_task',
    'course_id',
    'course',
    'id',
    'CASCADE',
    'CASCADE'
);

```

Рисунок 3.5 – Зв'язування таблиць «course_task» та «course»

Результат зв'язування таблиці «course_task» за допомогою зовнішнього ключа «course_id» із таблицею «course» наведений на рис. 3.6.

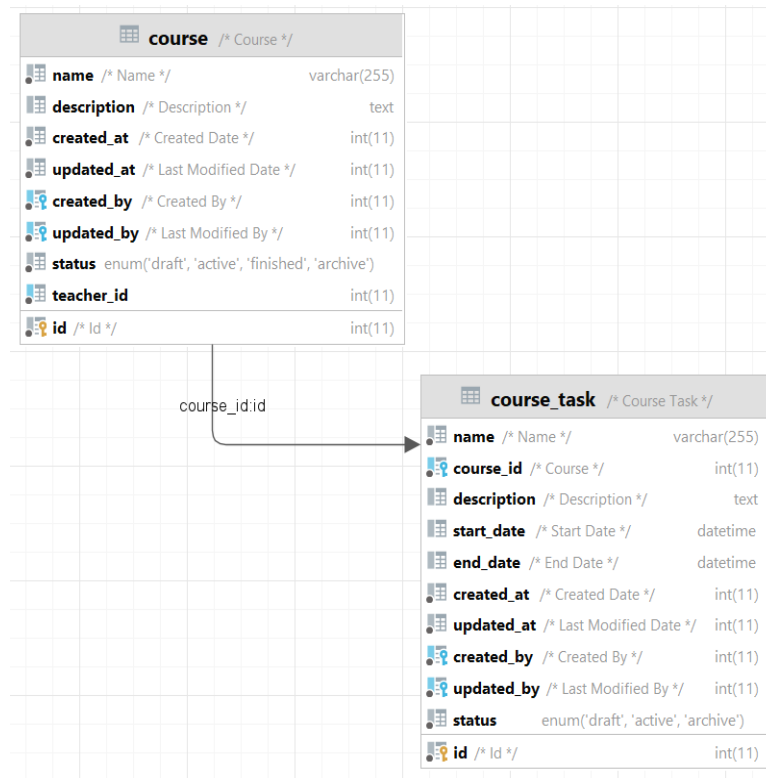


Рисунок 3.6 – Результат зв’язування таблиць «course_task» та «course»

Аналогічним чином були реалізовані усі інші сутності системи.

3.3 Опис класів

Далі наведено опис класів-сутностей, які є моделями таблиць з БД.

3.3.1 User

Даний клас використовується для зберігання інформації стосовно користувача. У таблиці 3.1 наведено опис атрибутів даного класу.

Таблиця 3.1 – Опис атрибутів класу «User»

Атрибут	Опис
id	Ідентифікатор запису
username	Логін

Продовження таблиці 3.1

email	Поштова адреса
phone	Номер телефону
type	Тип («Учень», «Вчитель», «Адміністратор»)
first_name	Ім'я
last_name	Прізвище
patronymic	Ім'я по батькові
status	Статус користувача
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі

3.3.2 Students Group

Даний клас використовується для зберігання інформації стосовно групи (класу) учнів. У таблиці 3.2 наведено опис атрибутів даного класу.

Таблиця 3.2 – Опис атрибутів класу «Students Group»

Атрибут	Опис
id	Ідентифікатор запису
mentor_id	Ідентифікатор керівника класу
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис

Продовження таблиці 3.2

updated_by	Ідентифікатор користувача, що оновив запис
------------	--

3.3.3 Students Group Assignment

Даний клас використовується для зберігання інформації стосовно призначення певного учня до певної групи (класу). У таблиці 3.3 наведено опис атрибутів даного класу.

Таблиця 3.3 – Опис атрибутів класу «Students Group Assignment»

Атрибут	Опис
id	Ідентифікатор запису
student_id	Ідентифікатор учня
students_group_id	Ідентифікатор групи (класу)
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.4 Course

Даний клас використовується для зберігання інформації стосовно предмета. У таблиці 3.4 наведено опис атрибутів даного класу.

Таблиця 3.4 – Опис атрибутів класу «Course»

Атрибут	Опис
id	Ідентифікатор запису

Продовження таблиці 3.4

name	Назва предмета
description	Опис предмета
status	Статус («Не активний», «Активний», «Закінчений», «Архівний»)
teacher_id	Ідентифікатор вчителя предмета
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.5 Course Student Assignment

Даний клас використовується для зберігання інформації стосовно призначення певного учня до певного предмета. У таблиці 3.5 наведено опис атрибутів даного класу.

Таблиця 3.5 – Опис атрибутів класу «Course Student Assignment»

Атрибут	Опис
id	Ідентифікатор запису
student_id	Ідентифікатор учня
course_id	Ідентифікатор предмета
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі

Продовження таблиці 3.5

created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.6 Course Task

Даний клас використовується для зберігання інформації стосовно завдання предмета. У таблиці 3.6 наведено опис атрибутів даного класу.

Таблиця 3.6 – Опис атрибутів класу «Course Task»

Атрибут	Опис
id	Ідентифікатор запису
name	Назва завдання
course_id	Ідентифікатор предмета
status	Статус («Не активне», «Активне», «Архівне»)
description	Опис завдання
start_date	Дата початку терміну виконання завдання
end_date	Дата закінчення терміну виконання завдання
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.7 Course Task Submission

Даний клас використовується для зберігання інформації стосовно виконання певного завдання. У таблиці 3.7 наведено опис атрибутів даного класу.

Таблиця 3.7 – Опис атрибутів класу «Course Task Submission»

Атрибут	Опис
id	Ідентифікатор запису
course_task_id	Ідентифікатор завдання
student_id	Ідентифікатор учня
comment	Коментар учня
status	Статус виконання завдання («Очікує перевірки», «Виконано», «Повернено»)
mark	Результат виконання завдання
completion_date	Дата виконання завдання
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.8 Attachment

Даний клас використовується для зберігання інформації стосовно прикладеного файлу до певного запису. У таблиці 3.8 наведено опис атрибутів даного класу.

Таблиця 3.8 – Опис атрибутів класу «Attachment»

Атрибут	Опис
id	Ідентифікатор запису
parent_id	Ідентифікатор батьківського запису
path	Шлях до файлу
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.9 Course Time Slot

Даний клас використовується для зберігання інформації стосовно планування проведення певного предмета. У таблиці 3.9 наведено опис атрибутів даного класу.

Таблиця 3.9 – Опис атрибутів класу «Course Time Slot»

Атрибут	Опис
id	Ідентифікатор запису
course_id	Ідентифікатор предмета
status	Статус планування («Не активне», «Активне»)
time_from	Дата і час початку
time_to	Дата і час закінчення

Продовження таблиці 3.9

description	Опис щодо планування
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.10 Student Mark

Даний клас використовується для зберігання інформації стосовно оцінки певного учня. У таблиці 3.10 наведено опис атрибутів даного класу.

Таблиця 3.10 – Опис атрибутів класу «Student Mark»

Атрибут	Опис
id	Ідентифікатор запису
student_id	Ідентифікатор учня
course_id	Ідентифікатор предмета
mark	Оцінка
related_date	Дата, до якої належить оцінка
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.11 Course Chat Message

Даний клас використовується для зберігання інформації стосовно повідомлення користувача до певного предмета. У таблиці 3.11 наведено опис атрибутів даного класу.

Таблиця 3.11 – Опис атрибутів класу «Course Chat Message»

Атрибут	Опис
id	Ідентифікатор запису
sender_id	Ідентифікатор користувача
course_id	Ідентифікатор предмета
content	Вміст повідомлення
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.3.12 Notification

Даний клас використовується для зберігання інформації стосовно сповіщення для користувача. У таблиці 3.12 наведено опис атрибутів даного класу.

Таблиця 3.12 – Опис атрибутів класу «Notification»

Атрибут	Опис
id	Ідентифікатор запису

Продовження таблиці 3.12

user_id	Ідентифікатор користувача
content	Вміст сповіщення
created_at	Дата створення запису в unix-форматі
updated_at	Дата оновлення запису в unix-форматі
created_by	Ідентифікатор користувача, що створив запис
updated_by	Ідентифікатор користувача, що оновив запис

3.4 Тестування системи

Тестування є важливим етапом розробки ПЗ, бо воно дозволяє перевірити, чи відповідає ПЗ вимогам проєкту, чи є воно надійним, витривалим та легким у використанні, і чи готове до представлення широкому загалу. Можна перевіряти функціонал всієї системи вручну, але найбільш ефективним методом буде забезпечення додатку автоматичними тестами. Без таких тестів не обійтись при розробці великих проєктів.

Yii2 фреймворк підтримує інтеграцію з фреймворком для тестування Codeception [20]. Даний фреймворк дозволяє проводити наступні типи тестування:

- модульне – перевіряє, що окремий модуль системи працює коректно;
- функціональне – перевіряє користувацькі сценарії шляхом емуляції браузера;
- приймальне – перевіряє користувацькі сценарії в браузері.

У Codeception включений модуль PhpBrowser, що являє собою "віртуальний" браузер, і, якого може бути цілком достатньо для виконання тестів, які не потребують перевірки JavaScript.

3.4.1 Встановлення пакетів

Для використання самого пакету Codeception, його спочатку потрібно встановити. Це можна зробити за допомогою Composer. Для того, щоб не встановлювати даний пакет та не налаштовувати його для кожного проєкту окремо, рекомендується встановлювати це розширення глобально (рис. 3.7).

```
composer global require "codeception/codeception=2.0.*"
composer global require "codeception/specify=*"
composer global require "codeception/verify=*"
```

Рисунок 3.7 – Встановлення розширення codeception

3.4.2 Підготовка проєкту

Для перевірки, що конфігурація є правильною, та не заглиблюватися в процес написання тестів, можна використати готові тести, які входять до шаблону advanced. Під час налаштування URL, необхідно винести правила їх зміни до файлу .htaccess (рис. 3.8).

```
RewriteEngine on
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . index.php
```

Рисунок 3.8 – Вміст файлу .htaccess

Це надає певної зручності, якщо сайт розміщується на хостингу, що не надає доступ до налаштувань Apache. Так як на локальній машині є доступ до цих налаштувань, то зручніше відмовитися від використання .htaccess і перенести всі настройки з нього до конфігураційного файлу Apache (рис. 3.9).


```

<VirtualHost *:80>
    DocumentRoot "c:/wamp/www/testing/frontend/web"
    ServerName testing.local
    <Directory "c:/wamp/www/testing/frontend/web">
        Order Allow,Deny
        Allow from all
        AddDefaultCharset utf-8
        RewriteEngine on
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule . index.php
    </Directory>
</VirtualHost>

```

Рисунок 3.9 – Вміст конфігураційного файлу Apache

З такою конфігурацією, при запуску тестів буде використовуватися вхідний скрипт `index.php` замість `index-text.php`. Тому, для тестування можна створити окремий віртуальний хост, який буде працювати с тією ж директорією, але використовувати скрипт `index-test.php`. Дана конфігурація наведена на рисунку 3.10.

```

<VirtualHost *:80>
    DocumentRoot "c:/wamp/www/testing/frontend/web"
    ServerName test.testing.local
    <Directory "c:/wamp/www/testing/frontend/web">
        Order Allow,Deny
        Allow from all
        AddDefaultCharset utf-8
        DirectoryIndex index-test.php
        RewriteEngine on
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule . index-test.php
    </Directory>
</VirtualHost>

```

Рисунок 3.10 – Відредагований вміст конфігураційного файлу Apache

Коли тестова база сконфігурована та підключена, її структуру необхідно привести до того ж самого вигляду, що й основна. Для цього в директорії `tests/codeception/bin` потрібно виконати команду `yii migrate/up`. Після цього, потрібно виправити адресу вхідного скрипту у файлі

codeception.yml, що знаходиться у директорії tests/codeception/frontend, як наведено на рисунку 3.11.

```
test_entry_url: http: //test.testing.local/index-test.php
```

Рисунок 3.11 – Виправлена адреса вхідного скрипту

Тепер, коли є все необхідне для тестування web-додатку, можна почати запуск тестів. Для цього необхідно виконати команду codecept run. Результат виконання даної команди наведено на рис 3.12.

```
$ codecept run
Codeception PHP Testing Framework v2.1.11
Powered by PHPUnit 5.3.5 by Sebastian Bergmann and contributors.
Acceptance Tests (1) -----
Ensure that frontpage works (WelcomeCept)                                OK
-----
Functional Tests (0) -----
-----
Unit Tests (0) -----
-----
Time: 554 ms, Memory: 10.25MB
OK (1 test, 2 assertions)
```

Рисунок 3.12 – Результати виконання тесту

4 ОПИС РОБОТИ СИСТЕМИ

Увійшовши до web-системи, користувач опиняється на сторінці авторизації. Авторизація виконується через відповідну форму, що наведена на рисунку 4.1.

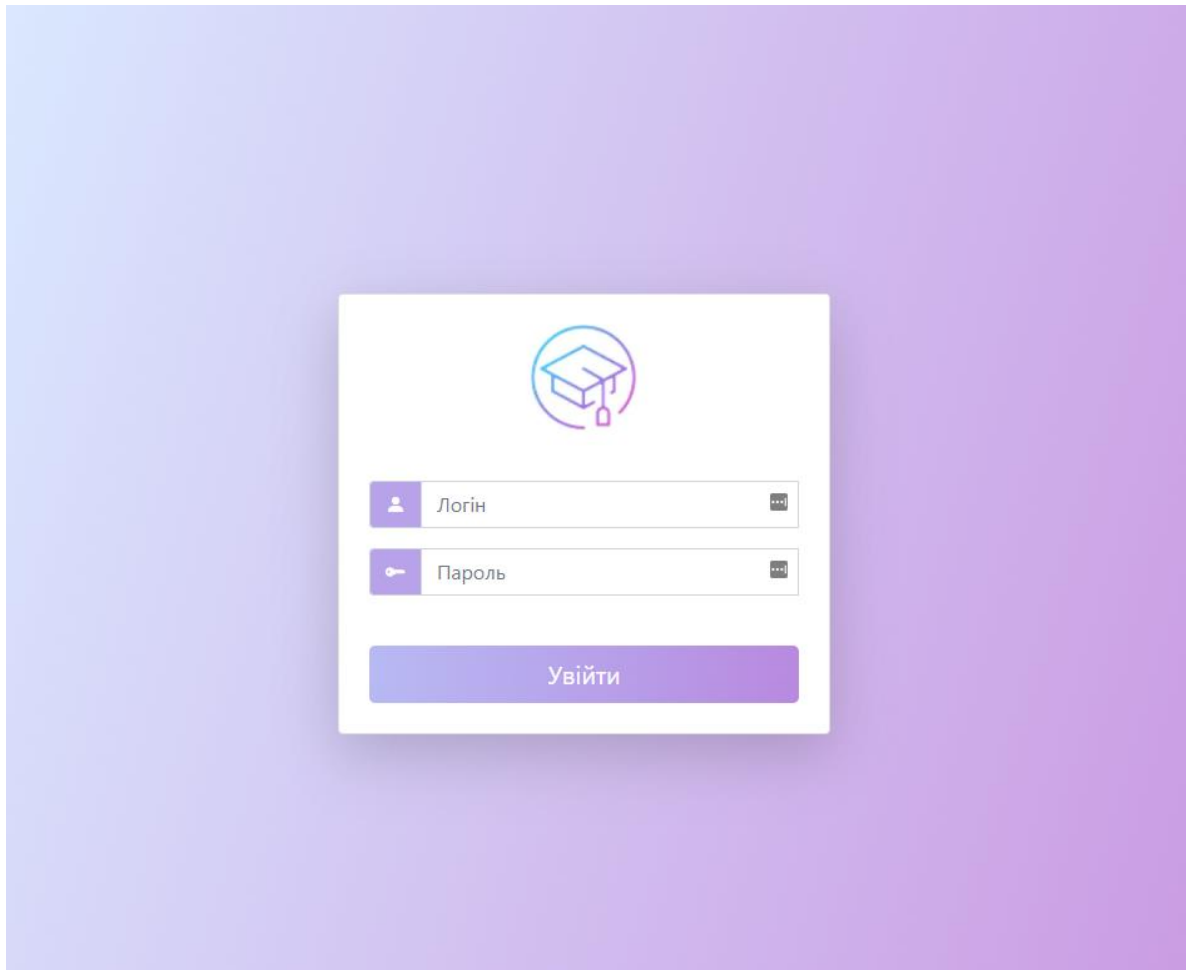


Рисунок 4.1 – Форма для авторизації користувача

Для простої роботи з формами розроблений спеціальний віджет ActiveForm. Його застосування можна побачити в додатку Б на прикладі форми для авторизації користувача, що міститься у файлі frontend/views/site/login.php.

Авторизувавшись до системи, користувач автоматично потрапляє на сторінку із призначеними предметами (рис. 4.2). Відповідна навігаційна

панель для переходу до інших сторінок знаходиться зверху.

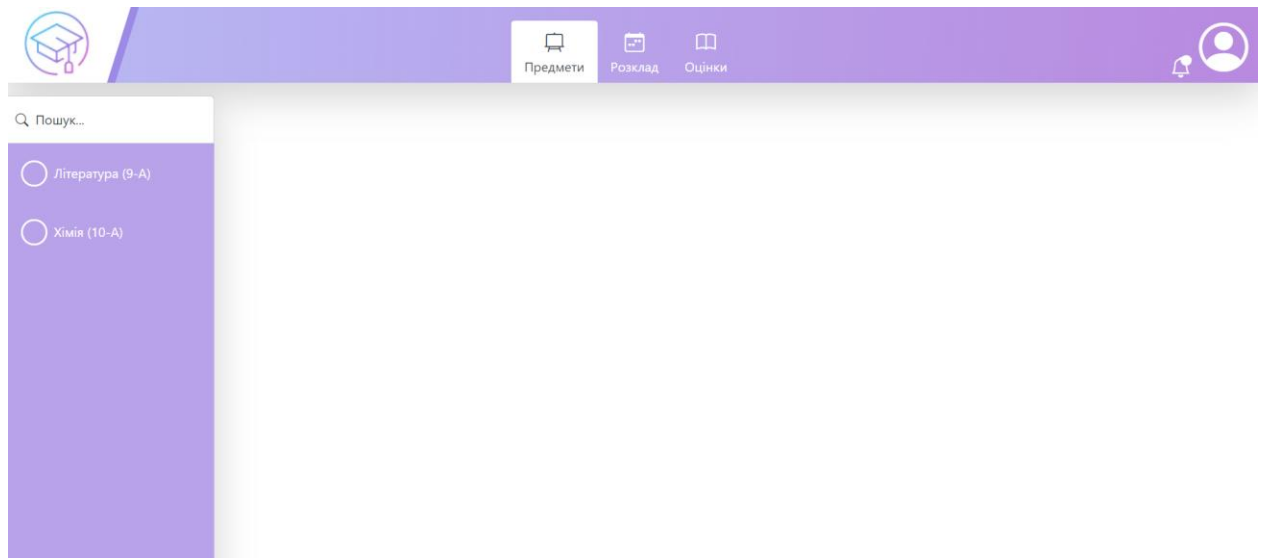


Рисунок 4.2 – Сторінка із предметами користувача

За формування навігації на сайті відповідає файл `frontend/views/layout/_header.php`. Саме у цьому файлі побудована структура меню.

За створення сторінки із предметами відповідає файл `frontend/views/course/index.php`. Здійснення контенту на сторінках web-додатку виконується за одним сценарієм. Основний `php` файл сторінки містить індивідуальний код та підключає додаткові елементи, що формують шаблон.

Формування даних (призначених предметів) здійснюється в контролері `frontend/controllers/CourseController.php` у функції `actionIndex()`.

Обравши один з предметів, можна побачити його відповідні деталі (рис. 4.3).

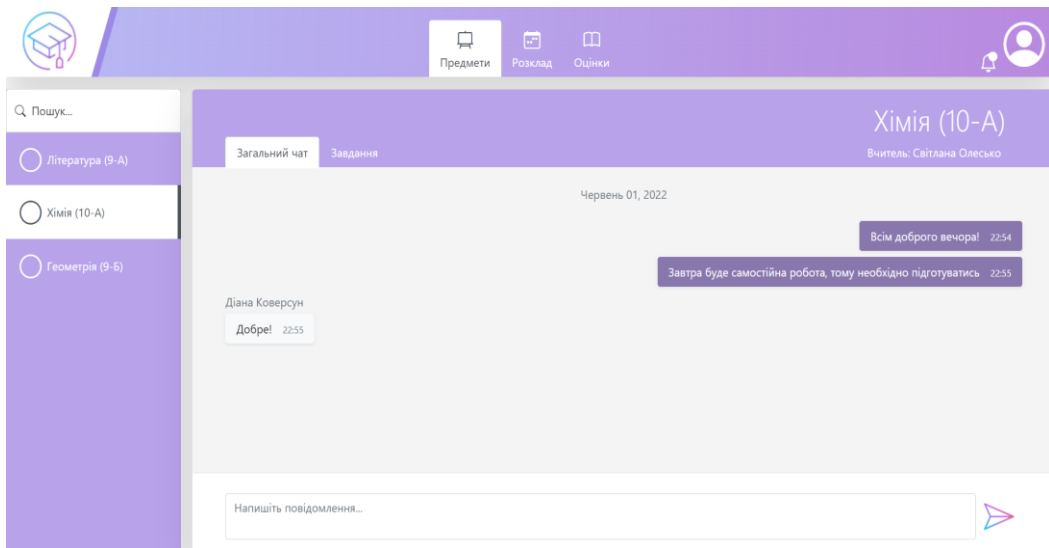


Рисунок 4.3 – Деталі предмета

Відкривши деталі предмета, користувач спочатку бачить загальний чат, де знаходяться повідомлення від всіх користувачів призначених до предмета. Реалізація даного чату наведена у додатку Б, а саме у файлі `frontend/views/course/_general_chat_tab.php`.

Для перегляду завдань предмета необхідно перейти до вкладки «Завдання». Якщо поточний користувач є учнем, то він побачить список завдань із відповідними статусами та статистикою їх виконання (рис. 4.4). Реалізація даного списку завдань наведена у додатку Б, а саме у файлі `frontend/views/course-task/course_tasks_list.php`. Відповідним контролером є `frontend/controllers/CourseTaskController.php`.

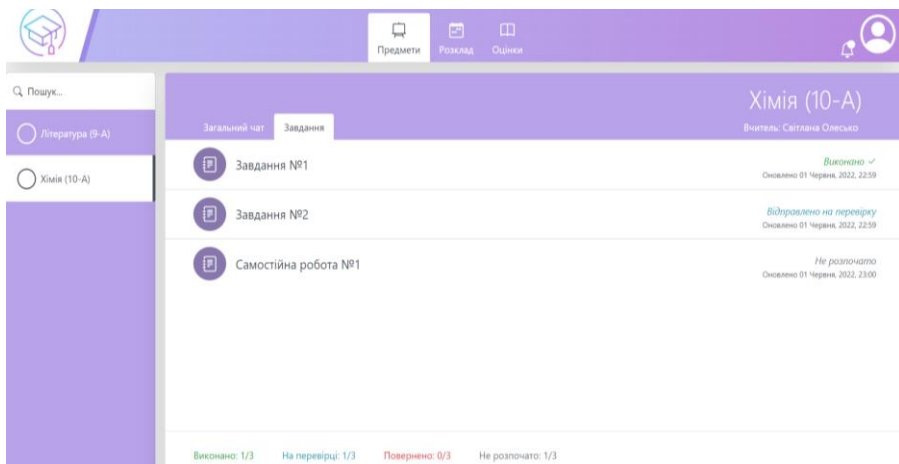


Рисунок 4.4 – Список завдань для учня

Вчитель, в свою чергу, побачить список існуючих завдань із можливістю додавання нових завдань. Приклад наведений на рисунку 4.5.

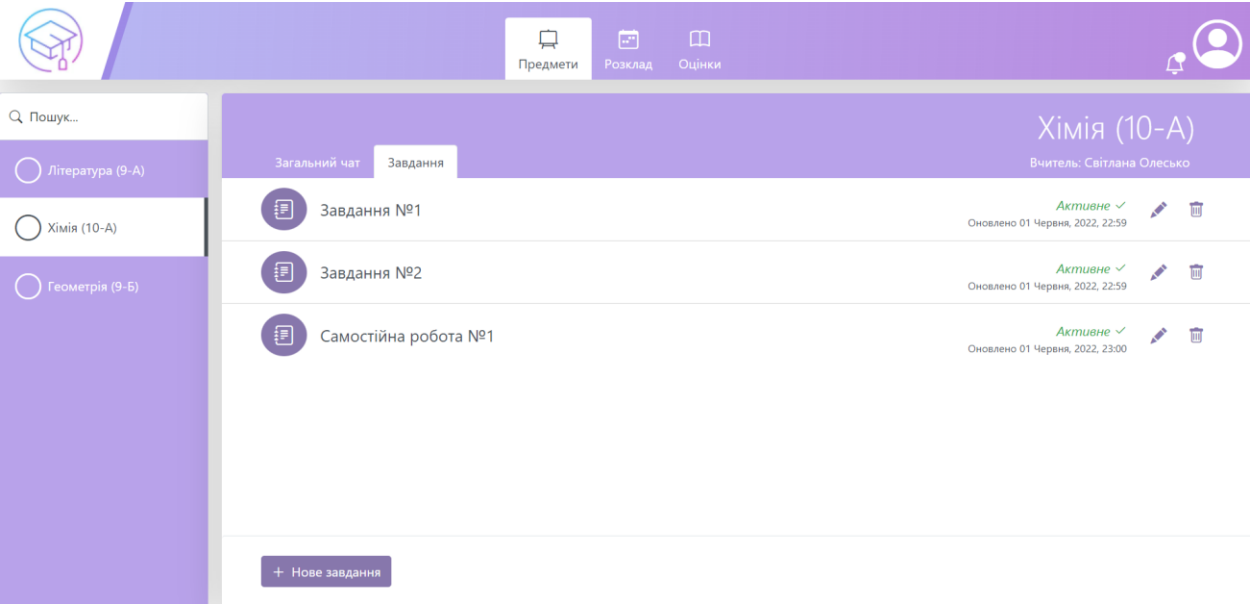


Рисунок 4.5 – Список завдань для вчителя

Користувачу надається можливість переглянути деталі завдання. Таким чином, учень може відправляти завдання на перевірку та переглянути статус його виконання, як наведено на рисунках 4.6 – 4.7.

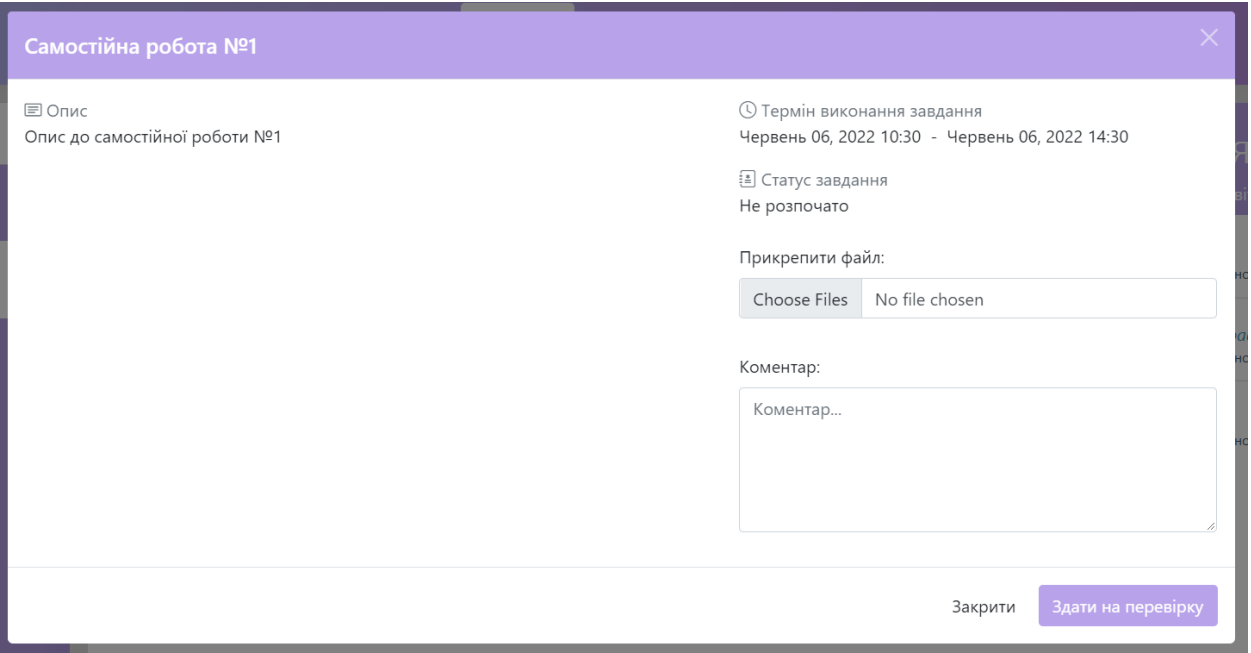


Рисунок 4.6 – Здача завдання на перевірку

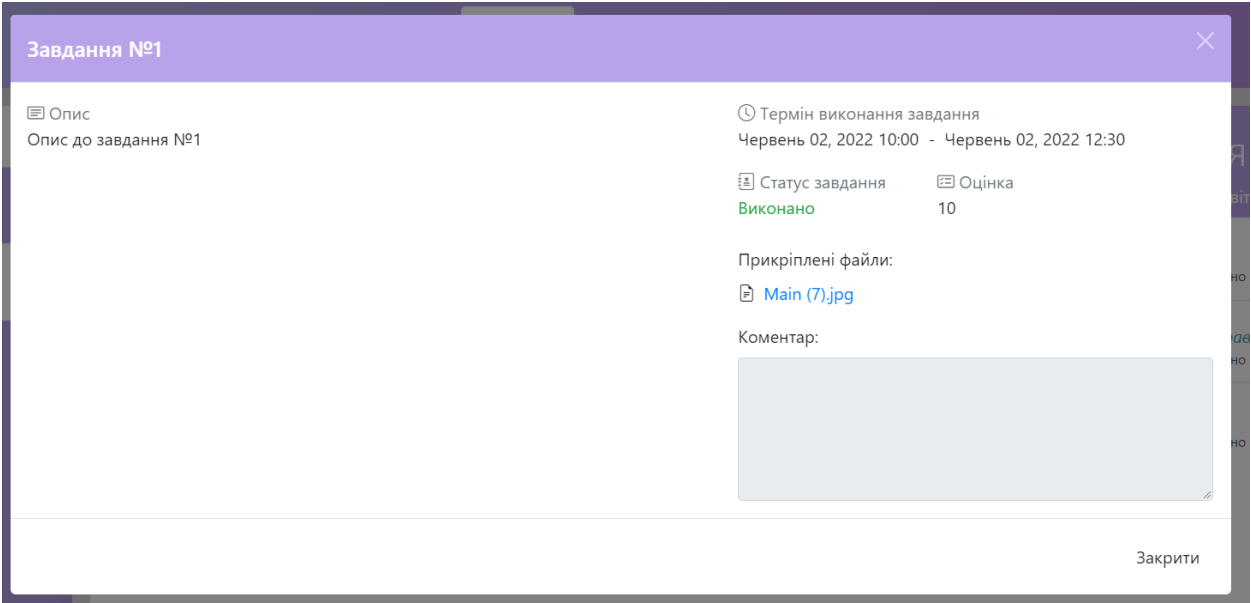


Рисунок 4.7 – Перегляд учнем результату виконання завдання

Реалізацію даного функціоналу можна побачити у лістингу Б, а саме у файлі frontend/views/course-task/view.php. Відповідним контролером є frontend/controllers/CourseTaskController.php.

Відкривши деталі завдання, вчитель може побачити його відповідний опис та статус, а також таблицю з результатами виконання даного завдання (рис. 4.8 – 4.9).

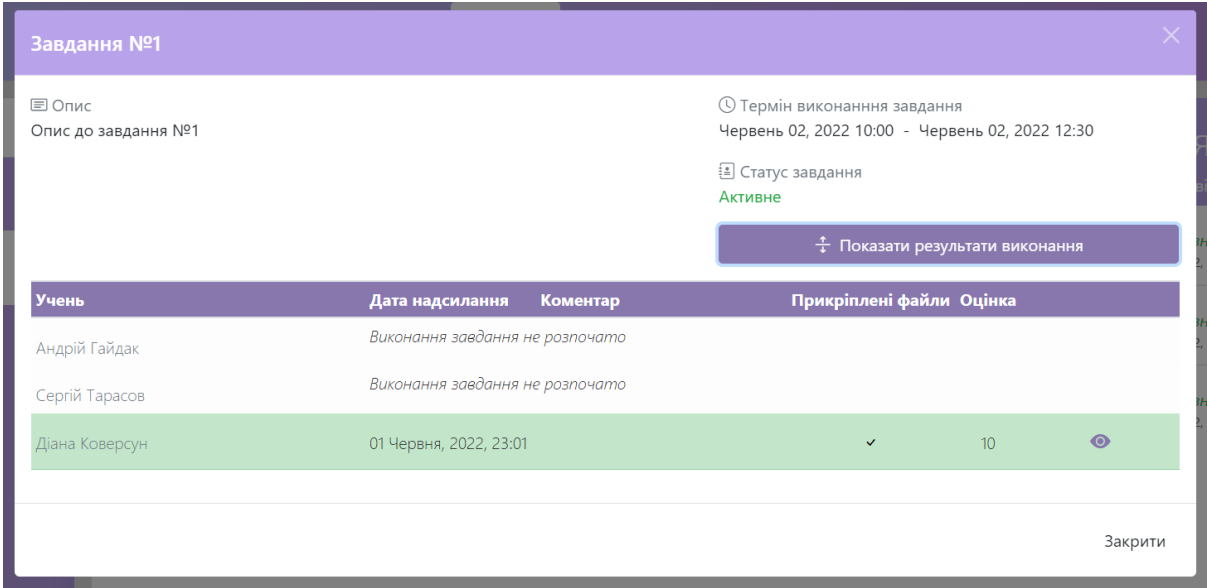


Рисунок 4.8 – Перегляд вчителем результатів виконання завдання

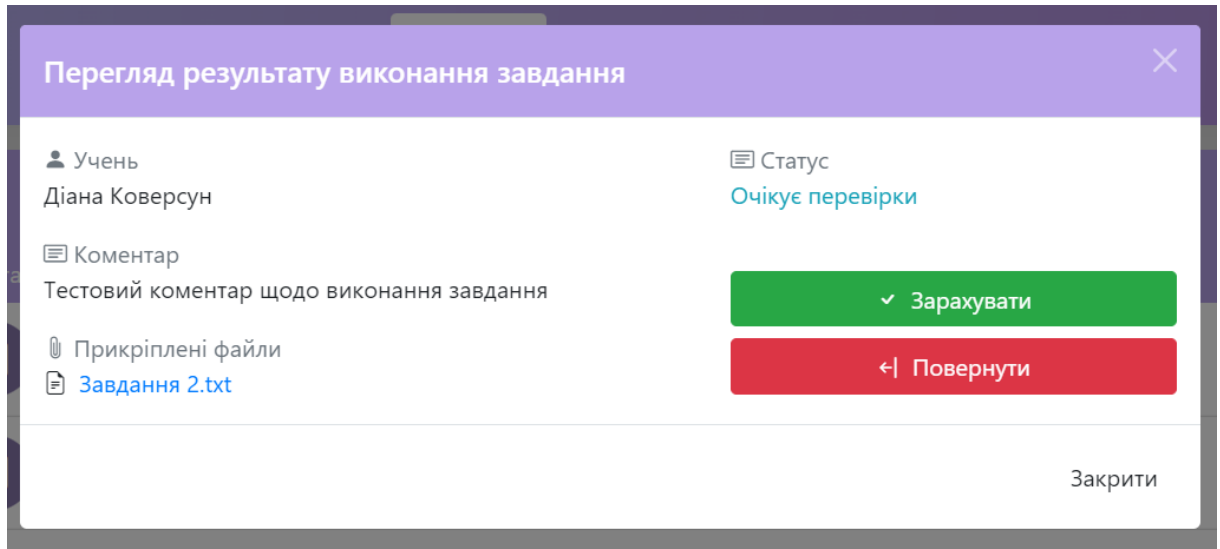


Рисунок 4.9 – Оцінювання виконання завдання

Реалізація функціоналу щодо оцінювання результатів виконання завдання наведена у лістингу Б, а саме у файлі `frontend/views/course-task-submission/view.php`. Відповідним контролером є `frontend/controllers/CourseTaskSubmissionController.php`.

Перейшовши до «Розклад» у навігаційному меню, користувачу надається інформація щодо запланованих проведення предметів (рис. 4.10).

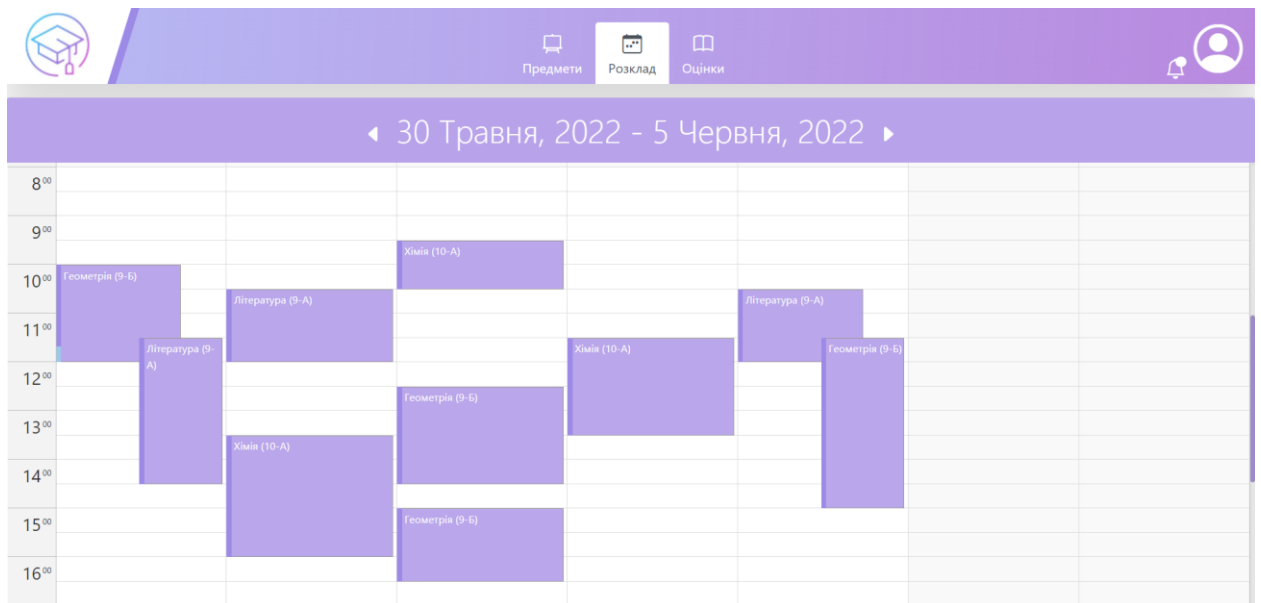


Рисунок 4.10 – Планування проведення предметів

Вчителю також надається можливість створювати, редагувати та

видаляти планування проведення предметів. Функціонал даного модулю наведений у додатку Б, а саме у файлі `frontend/views/calendar/index.php`. Відповідним контролером даного модуля є `frontend/controllers/CalendarController.php`. Слід зазначити, що у якості готового вирішення роботи із календарем було використано DayPilot компонент [19] з адаптацією під даний web-додаток.

Перейшовши до «Оцінки» у навігаційному меню, користувачу надається інформація щодо оцінок (рис. 4.11).

◀ Травень, 2022 ▶																															
Назва предмета	01 нд	02 пн	03 вт	04 ср	05 чт	06 пт	07 сб	08 нд	09 пн	10 вт	11 ср	12 чт	13 пт	14 сб	15 нд	16 пн	17 вт	18 ср	19 чт	20 пт	21 сб	22 нд	23 пн	24 вт	25 ср	26 чт	27 пт	28 сб	29 нд	30 пн	31 вт
Література (9-А)					10				11					8		12				11										10	
Хімія (10-А)													11			10			9	10			10				11				

Рисунок 4.11 – Інформація щодо отриманих оцінок учня

Вчителю надається можливість виставляти та редагувати оцінки учням з відповідних предметів (рис. 4.12).

Предмети

Розклад

Оцінки

◀ Травень, 2022 ▶

✕

✓ Зберегти

Література (9-А) ▾	01 нд	02 пн	03 вт	04 ср	05 чт	06 пт	07 сб	08 нд	09 пн	10 вт	11 ср	12 чт	13 пт	14 сб	15 нд	16 пн	17 вт	18 ср	19 чт	20 пт	21 сб	22 нд	23 пн	24 вт	25 ср	26 чт	27 пт	28 сб	29 нд
Андрій Гайдак												7							9										
Діана Коверсун					10				11							12			10	11									
Сергій Тарасов														8				8											

Рисунок 4.12 – Редагування оцінок учнів

Функціонал модуля з оцінками можна побачити у лістингу Б, а саме у файлі `frontend/views/marks/index.php`. Відповідним контролером даного модуля є `frontend/controllers/MarksController.php`.

Користувач також може переглядати власний профіль та профілі інших користувачів (рис. 4.13). Даний функціонал наведений у лістингу Б, а саме у

файлі frontend/views/user/view.php. Відповідним контролером даного модуля є frontend/controllers/UserController.php.

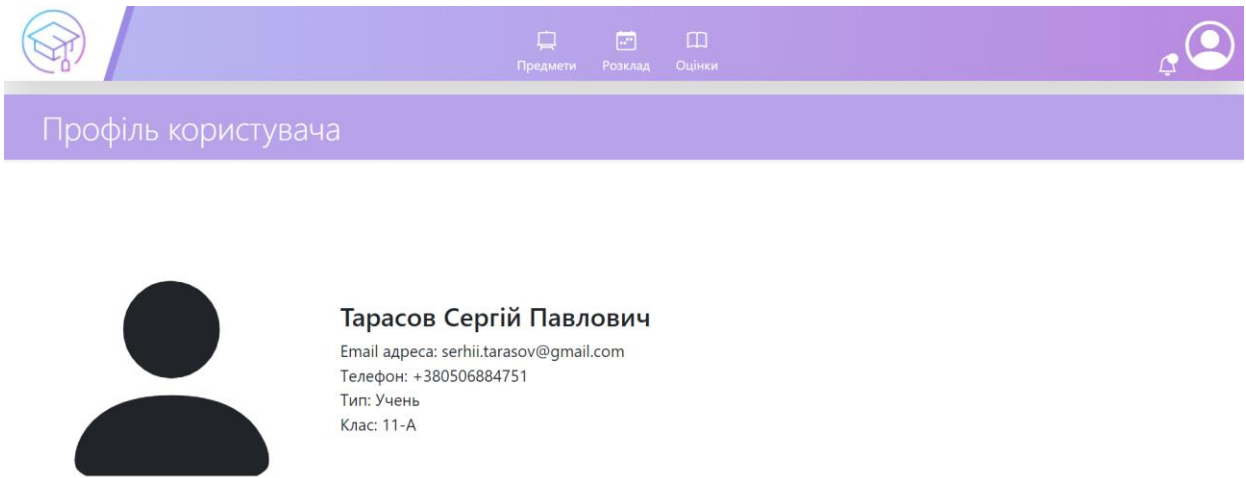


Рисунок 4.13 – Перегляд профілю користувача

Для користувачів, що є адміністраторами, було розроблено окремий модуль для роботи із записами в системі. Даний модуль знаходиться в директорії backend, де містяться відповідні контролери (backend/controllers/) та представлення (backend/views/). Наприклад, роботу із записами учнів наведено на рисунку 4.14.

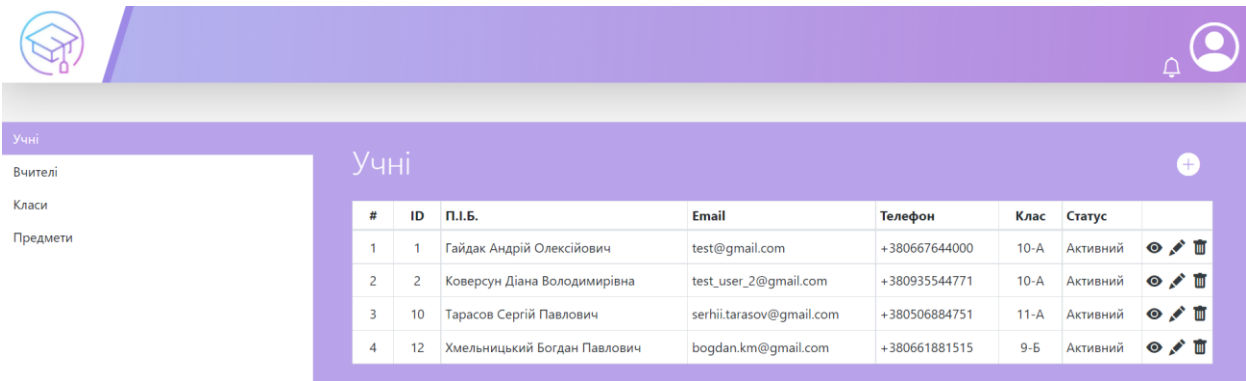


Рисунок 4.14 – Робота із записами учнів

ВИСНОВОК

В даній кваліфікаційній роботі було досліджено загальну інформацію стосовно розробки web-систем та розроблено web-додаток для автоматизації дистанційного навчання учнів загальноосвітньої школи.

Були вирішені наступні основні задачі:

1) проведено аналіз існуючих додатків для дистанційного навчання та досліджено основні засоби розробки для реалізації додатку;

2) розроблено специфікації системи та спроектовано й реалізовано структуру системи обраними засобами розробки;

3) реалізовано функціонал системи із відповідним графічним інтерфейсом;

4) проведене тестування розробленої системи.

Для реалізації web-додатку були обрані наступні програмні засоби та технології:

- система керування базами даних MySQL;
- мови PHP, JavaScript, CSS, HTML;
- фреймворк Yii2;
- програмне середовище JetBrains PhpStorm;
- локальний сервер OpenServer.

Проаналізувавши отримані результати функціонування системи, було виявлено ряд недоліків, що обов'язково будуть усунені в майбутньому. До таких недоліків можна віднести наступні:

- відсутня можливість відмічати пропуски учнів;
- відсутня можливість прикріплення файлів до опису завдання;
- не зовсім гнучкий функціонал оцінювання учнів.

В результаті виконання дипломної роботи було розроблено систему, яка вирішує завдання у сфері освітнього процесу, а саме – автоматизація дистанційного навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Платформа Google Classroom [Електронний ресурс]. – Режим доступу: <https://uk.tech-sensor.com/10020677-google-classroom-review>. – Огляд Google Classroom.
2. Платформа Moodle [Електронний ресурс]. – Режим доступу: <https://hyperhost.ua/info/uk/cms-moodle-sistema-dlya-navchannya-i-testuvannya-onlayn-oglyad-funktsionalu>. – CMS Moodle – система для навчання і тестування онлайн. Огляд функціоналу.
3. Платформа Microsoft Teams [Електронний ресурс]. – Режим доступу: <https://www.techlearning.com/features/microsoft-teams-what-is-it-and-how-does-it-work-for-students-and-teachers/>. – Microsoft Teams: What is it and How Does it Work for Students and Teachers?
4. Web-фреймворки для розробки [Електронний ресурс]. – Режим доступу: <https://voll.com.ua/uk/blog/frejmworki-dlya-veb-rozrobki>. – Фреймворки для веб-розробки.
5. Matthias N. A Year with Symfony / N. Matthias. – Matthias Noback, 2013. – 230 с.
6. Кіріченко О. В. Laravel для web-розробників. Практичний посібник із створення сайтів / О. В. Кіріченко, Є. В. Дубовик. – Наука і техніка, 2021. – 432 с.
7. Структура Yii2 фреймворку [Електронний ресурс]. – Режим доступу: <https://yiiframework.com.ua/uk/doc/guide/2/structure-overview/>. – Yii framework. Огляд.
8. Системи керування базами даних [Електронний ресурс]. – Режим доступу: https://pidru4niki.com/81326/tehnika/sistemi_keruvannya_bazami_danih. – Системи керування базами даних.
9. Sibsankar H. SQLite Database System Design and Implementation (Second Edition, Version 1) / H. Sibsankar. – Sibsankar Haldar, 2015. – 256 p.

- 10.Маклафлін Б. PHP та MySQL. Виключний посібник / Б. Маклафлін. – Пітер, 2014. – 544 с.
- 11.PostgreSQL СКБД [Електронний ресурс]. – Режим доступу: <https://intellect.icu/postgresql-obzor-bazy-dannykh-osnovy-6837>. – PostgreSQL – обзор базы данных, основы кратко
- 12.Дакетт Д. Javascript та jQuery. Інтерактивна web-розробка / Д. Дакетт. – Ексмо, 2018. – 640 с.
- 13.Специфікація програмних вимог SRS [Електронний ресурс]. – Режим доступу: <https://visuresolutions.com/uk/software-requirement-specification-srs-tips-template/>. – Специфікація вимог до програмного забезпечення (SRS): Поради та шаблон.
- 14.Розенберг Д. Застосування об'єктного моделювання з використанням UML та аналіз прецедентів / Д. Розенберг. – М.: ДМК Пресс, 2019. – 160 с.
- 15.Діаграми послідовності [Електронний ресурс]. – Режим доступу: <https://evergreens.com.ua/ua/articles/uml-diagrams.html>. – UML для бізнес-моделювання: для чого потрібні діаграми процесів.
- 16.Діаграми розгортання [Електронний ресурс]. – Режим доступу: <https://jak.koshachek.com/articles/13-diagrami-rozgortannja.html>. – 13 Діаграми розгортання.
- 17.JetBrains PhpStorm IDE [Електронний ресурс]. – Режим доступу: <https://www.sitepoint.com/phpstorm-review-and-give-away/>. – PhpStorm – Review and Give Away
- 18.Програмний комплекс Open Server Panel [Електронний ресурс]. – Режим доступу: <https://ospanel.io/>. – Open server.
- 19.DayPilot компонент [Електронний ресурс]. – Режим доступу: <https://doc.daypilot.org/calendar/>. – Event Calendar.
- 20.Codeception фреймворк [Електронний ресурс]. – Режим доступу: <https://codeception.com/blog/> – Blog.

Додаток А

Зауваження нормоконтролера

[illegible]

Додаток Б

Лістинг програми

Контролери:

frontend/controllers/AlertNotificationController.php

```
<?php

namespace frontend\controllers;

use Yii;
use yii\web\Controller;

class AlertNotificationController extends Controller {
    const SUCCESS_TYPE = 'success';
    const ERROR_TYPE = 'error';
    const WARNING_TYPE = 'warning';

    public function actionView() {
        return $this->renderAjax('view');
    }

    public static function setNotification($notificationType, $message) {
        Yii::$app->session->setFlash($notificationType, $message);
    }
}
```

frontend/controllers/AttachmentController.php

```
<?php

namespace frontend\controllers;

use Yii;
use yii\web\Controller;

class AttachmentController extends Controller {
    public function actionDownload() {
        $path = Yii::$app->request->get('path');

        if (file_exists($path)) {
            return Yii::$app->response->sendFile($path);
        } else {
            throw new \yii\web\NotFoundHttpException("{ $path } is not found!");
        }
    }
}
```

frontend/controllers/CalendarController.php

```
<?php

namespace frontend\controllers;

use common\models\CourseTimeSlot;
use yii\web\Controller;

class CalendarController extends Controller {
    public function actionIndex() {
        return $this->render('index', [
            'timeSlots' => self::getTimeSlots()
        ]);
    }
}
```

```

    public static function getTimeSlots() {
        $assignedCourseIds = [];

        foreach (CourseUtils::getAssignedCourses() as $assignedCourse) {
            $assignedCourseIds[] = $assignedCourse->id;
        }

        return CourseTimeSlot::find()->where(['status' =>
CourseTimeSlot::ACTIVE_STATUS])
                                ->andWhere(['in', 'course_id',
$assignedCourseIds])
                                ->all();
    }
}

```

frontend/controllers/CourseChatMessageController.php

```

<?php

namespace frontend\controllers;

use common\models\CourseChatMessage;
use Yii;
use yii\web\Controller;
use yii\filters\VerbFilter;

class CourseChatMessageController extends Controller {
    public function behaviors() {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ],
        );
    }

    public function actionCreate() {
        $courseId = Yii::$app->request->post('courseId');
        $message = Yii::$app->request->post('message');

        $chatMessage = new CourseChatMessage();
        $chatMessage->course_id = $courseId;
        $chatMessage->content = $message;
        $chatMessage->sender_id = Yii::$app->user->getId();
        $chatMessage->save();

        return $this->renderAjax('_course_chat_messages_list', [
            'courseChatMessages' => CourseChatMessage::findAll(['course_id' =>
$courseId])
        ]);
    }
}

```

frontend/controllers/CourseController.php

```

<?php

namespace frontend\controllers;

use common\models\Course;
use Yii;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

class CourseController extends Controller {

```



```

public function behaviors() {
    return array_merge(
        parent::behaviors(),
        [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ]
    );
}

public function actionIndex() {
    $assignedCourses = CourseUtils::getAssignedCourses();

    return $this->render('index', [
        'assignedCourses' => $assignedCourses
    ]);
}

public function actionView() {
    $courseId = Yii::$app->request->post('id');

    return $this->renderAjax('_details', [
        'course' => $this->findModel($courseId),
    ]);
}

protected function findModel($id) {
    if (($model = Course::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
```

frontend/controllers/CourseTaskController.php

```

<?php

namespace frontend\controllers;

use common\models\Attachment;
use common\models\CourseTask;
use common\models\CourseTaskSubmission;
use Yii;
use yii\helpers\Json;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;
use yii\web\UploadedFile;

class CourseTaskController extends Controller {
    public function behaviors() {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ]
        );
    }
}
```

```

public function actionPreviewCreationForm() {
    $courseId = Yii::$app->request->post('courseId');

    return $this->renderAjax('create', [
        'courseId' => $courseId
    ]);
}

public function actionPreviewUpdateForm() {
    $courseTaskId = Yii::$app->request->post('id');

    return $this->renderAjax('update', [
        'courseTask' => $this->findModel($courseTaskId)
    ]);
}

public function actionPreviewDeleteForm() {
    $courseTaskId = Yii::$app->request->post('id');

    return $this->renderAjax('delete', [
        'courseTask' => $this->findModel($courseTaskId)
    ]);
}

public function actionView() {
    $courseTask = $this->findModel(Yii::$app->request->post('id'));

    return $this->renderAjax('view', [
        'courseTask' => $courseTask
    ]);
}

public function actionCreate() {
    $courseTask = new CourseTask();

    if ($this->request->isPost && $courseTask->load($this->request->post()) &&
    $courseTask->save()) {
        $courseTask = $this->findModel($courseTask->id);

        $courseTask->status = CourseTask::ONGOING_STATUS;
        $courseTask->save();

        AlertNotificationController::setNotification(
            AlertNotificationController::SUCCESS_TYPE,
            'Завдання було успішно створено'
        );

        $this->sendNotifications(
            $courseTask->course->courseStudentAssignments,
            'Призначено нове завдання "' . $courseTask->name . '" для курсу "' .
            $courseTask->course->name . '"'
        );

        return $this->renderAjax('_course_tasks_list', [
            'courseTasks' => CourseTask::findAll(['course_id' => $courseTask->course_id]),
            'courseId' => $courseTask->course_id
        ]);
    } else {
        AlertNotificationController::setNotification(
            AlertNotificationController::ERROR_TYPE,
            Json::encode($courseTask->errors)
        );
    }
}

protected function sendNotifications($studentAssignments, $message) {
    foreach ($studentAssignments as $studentAssignment) {
        NotificationUtils::sendNotification(
            $studentAssignment->student_id,

```

```

        $message
    );
}

}

public function actionSubmit() {
    $sourceSubmission = new CourseTaskSubmission();

    if ($this->request->isPost && $sourceSubmission->load($this->request->post()))
    {
        $courseTask = $this->findModel($sourceSubmission->course_task_id);
        $student = Yii::$app->user->identity;

        $submissionToModify = $this->fetchAppliedCourseTaskSubmission(
            $sourceSubmission->course_task_id,
            $sourceSubmission->student_id
        );

        if ($submissionToModify !== null) {
            $submissionToModify->comment = $sourceSubmission->comment;
            $this->deleteAttachments($submissionToModify);
        } else {
            $submissionToModify = $sourceSubmission;
        }

        $submissionToModify->status = CourseTaskSubmission::PENDING_STATUS;
        $submissionToModify->mark = 0;

        if ($submissionToModify->save()) {
            $attachments = UploadedFile::getInstances($sourceSubmission,
'attachments');

            if ($attachments !== null && !empty($attachments)) {
                $this->uploadSubmissionAttachment(
                    $submissionToModify,
                    $attachments
                );
            }

            AlertNotificationController::setNotification(
                AlertNotificationController::SUCCESS_TYPE,
                'Завдання було успішно відправлено на перевірку'
            );

            NotificationUtils::sendNotification(
                $courseTask->course->teacher_id,
                $student->getName() . ' надіслав завдання "' . $courseTask->name .
'" на перевірку'
            );

            return $this->renderAjax('_course_tasks_list', [
                'courseTasks' => CourseTask::findAll(['course_id' => $courseTask-
>course_id]),
                'courseId' => $courseTask->course_id
            ]);
        } else {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($courseTask->errors)
            );
        }
    }
}

protected function fetchAppliedCourseTaskSubmission($courseTaskId, $studentId) {
    return CourseTaskSubmission::findOne([
        'student_id' => $studentId,
        'course_task_id' => $courseTaskId
    ]);
}

```

```

protected function uploadSubmissionAttachment($submission, $attachmentContents) {
    $basePath = 'uploads/course-task-submissions';

    if (!file_exists($basePath)) {
        mkdir($basePath, 0777, true);
    }

    foreach ($attachmentContents as $attachmentContent) {
        $attachmentName = $submission->id . '-' . $attachmentContent->name;

        $file = pathinfo($basePath . '/' . $attachmentName);

        if ($file !== null) {
            $filename = $file['filename'];
            $counter = 1;

            while (file_exists($basePath . '/' . $filename . '.' .
                $file['extension'])) {
                $filename = $file['filename'] . "($counter)";
                $counter++;
            }

            $attachmentName = $filename . '.' . $attachmentContent->extension;
        }

        $attachmentPath = $basePath . '/' . $attachmentName;

        $attachmentContent->saveAs($attachmentPath);

        $attachment = new Attachment();
        $attachment->parent_id = $submission->id;
        $attachment->path = $attachmentPath;
        $attachment->save();
    }
}

protected function deleteAttachments($submission) {
    foreach ($submission->getAttachments() as $attachment) {
        $attachment->delete();
    }
}

public function actionUpdate()
{
    $courseTask = $this->findModel($this->request->post()['CourseTask']['id']);

    if ($this->request->isPost && $courseTask->load($this->request->post()) &&
        $courseTask->save()) {
        AlertNotificationController::setNotification(
            AlertNotificationController::SUCCESS_TYPE,
            'Завдання було успішно оновлено'
        );

        return $this->renderAjax('_course_tasks_list', [
            'courseTasks' => CourseTask::findAll(['course_id' => $courseTask->course_id]),
            'courseId' => $courseTask->course_id
        ]);
    } else {
        AlertNotificationController::setNotification(
            AlertNotificationController::ERROR_TYPE,
            Json::encode($courseTask->errors)
        );
    }
}

public function actionDelete() {
    $courseTaskId = Yii::$app->request->post('id');

    $courseTask = $this->findModel($courseTaskId);
    $courseId = $courseTask->course_id;
}

```

```

        if ($courseTask->delete()) {
            AlertNotificationController::setNotification(
                AlertNotificationController::SUCCESS_TYPE,
                'Завдання було успішно видалено'
            );

            return $this->renderAjax('_course_tasks_list', [
                'courseTasks' => CourseTask::findAll(['course_id' => $courseId]),
                'courseId' => $courseId
            ]);
        } else {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($courseTask->errors)
            );
        }
    }

    protected function findModel($id)
    {
        if (($model = CourseTask::findOne(['id' => $id])) !== null) {
            return $model;
        }

        throw new NotFoundHttpException('The requested page does not exist.');
    }
}

```

frontend/controllers/CourseTaskSubmissionController.php

```

<?php

namespace frontend\controllers;

use common\models\CourseTaskSubmission;
use common\models\StudentMark;
use DateTime;
use Yii;
use yii\helpers\Json;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

class CourseTaskSubmissionController extends Controller {
    public function behaviors() {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ],
        );
    }

    public function actionView() {
        return $this->renderAjax('view', [
            'courseTaskSubmission' => $this->findModel(Yii::$app->request-
>post('id')),
        ]);
    }

    public function actionUpdate() {
        $courseTaskSubmission = $this->findModel(Yii::$app->request->post('id'));
        $type = Yii::$app->request->post('type');
        $notificationMessage = '';
    }
}

```

```

        if ($type === 'APPROVAL') {
            $courseTaskSubmission->status = CourseTaskSubmission::COMPLETED_STATUS;
            $courseTaskSubmission->mark = Yii::$app->request->post('mark');
            $courseTaskSubmission->completion_date = (new DateTime())->format('Y-m-d
H:i:s');
            $this->createStudentMark($courseTaskSubmission);

            $notificationMessage = 'Заповнено завдання "' . $courseTaskSubmission-
>courseTask->name . '"';
        } else if ($type === 'REJECT') {
            $courseTaskSubmission->status = CourseTaskSubmission::REJECTED_STATUS;

            $notificationMessage = 'Повернуто завдання "' . $courseTaskSubmission-
>courseTask->name . '"';
        }

        if ($courseTaskSubmission->save()) {
            NotificationUtils::sendNotification($courseTaskSubmission->student_id,
            $notificationMessage);

            $courseTaskAssignees = [];

            foreach ($courseTaskSubmission->courseTask->course-
>courseStudentAssignments as $courseStudentAssignment) {
                $courseTaskAssignees[] = $courseStudentAssignment->student;
            }

            return $this->renderAjax('_course_task_submissions_list', [
                'courseTaskSubmissions' =>
                CourseTaskSubmission::findAll(['course_task_id' => $courseTaskSubmission-
>course_task_id]),
                'courseTaskAssignees' => $courseTaskAssignees
            ]);
        } else {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($courseTaskSubmission->errors)
            );
        }
    }

    protected function createStudentMark($courseTaskSubmission) {
        $affectedStudentMark = $this->fetchExistingStudentMark($courseTaskSubmission);

        if ($affectedStudentMark !== null) {
            return;
        }

        $affectedStudentMark = new StudentMark();
        $affectedStudentMark->course_id = $courseTaskSubmission->courseTask-
>course_id;
        $affectedStudentMark->student_id = $courseTaskSubmission->student_id;
        $affectedStudentMark->mark = $courseTaskSubmission->mark;
        $affectedStudentMark->related_date = $courseTaskSubmission->completion_date;

        if (!$affectedStudentMark->save()) {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($affectedStudentMark->errors)
            );
        }
    }

    protected function fetchExistingStudentMark($courseTaskSubmission) {
        $affectedStudentMarks = StudentMark::find()
            ->where(['student_id' =>
            $courseTaskSubmission->student_id])
            ->andWhere(['course_id' =>
            $courseTaskSubmission->courseTask->course_id])
            ->all();
    }

```

```

        $affectedStudentMark = null;

        if ($affectedStudentMarks !== null) {
            foreach ($affectedStudentMarks as $studentMark) {
                $relatedDate = (new DateTime($studentMark->related_date))->format('Y-
m-d');
                $completionDate = (new DateTime($courseTaskSubmission-
>completion_date))->format('Y-m-d');

                if ($relatedDate === $completionDate) {
                    $affectedStudentMark = $studentMark;
                    break;
                }
            }

            return $affectedStudentMark;
        }

        protected function findModel($id) {
            if (($model = CourseTaskSubmission::findOne(['id' => $id])) !== null) {
                return $model;
            }

            throw new NotFoundHttpException('The requested page does not exist.');
```

frontend/controllers/CourseTimeSlotController.php

```

<?php

namespace frontend\controllers;

use common\models\Course;
use common\models\CourseTimeSlot;
use Yii;
use yii\helpers\Json;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

class CourseTimeSlotController extends Controller {
    public function behaviors() {
        return array_merge(
            parent::behaviors(),
            [
                'verbs' => [
                    'class' => VerbFilter::className(),
                    'actions' => [
                        'delete' => ['POST'],
                    ],
                ],
            ],
        );
    }

    public function actionPreviewCreationForm() {
        $startDate = Yii::$app->request->post('startDate');
        $endDate = Yii::$app->request->post('endDate');
        $availableCourses = $this->getAvailableCourses();

        return $this->renderAjax('create', [
            'availableCourses' => $availableCourses,
            'startDate' => $startDate,
            'endDate' => $endDate
        ]);
    }

    protected function getAvailableCourses() {
```

```

$courses = [];

foreach (Yii::$app->user->identity->courses as $course) {
    if ($course->status === Course::ONGOING_STATUS) {
        $courses[] = $course;
    }
}

return $courses;
}

public function actionPreviewUpdateForm() {
    $courseTimeSlot = $this->findModel(Yii::$app->request->post('id'));
    $startDate = Yii::$app->request->post('startDate');
    $endDate = Yii::$app->request->post('endDate');

    return $this->renderAjax('update', [
        'courseTimeSlot' => $courseTimeSlot,
        'startDate' => $startDate,
        'endDate' => $endDate
    ]);
}

public function actionPreviewDeletionForm() {
    $courseTimeSlot = $this->findModel(Yii::$app->request->post('id'));

    return $this->renderAjax('delete', [
        'courseTimeSlot' => $courseTimeSlot
    ]);
}

public function actionView() {
    $timeSlot = $this->findModel(Yii::$app->request->post('id'));

    return $this->renderAjax('view', [
        'courseTimeSlot' => $timeSlot
    ]);
}

public function actionCreate() {
    $timeSlot = new CourseTimeSlot();

    if ($this->request->isPost && $timeSlot->load($this->request->post()) &&
    $timeSlot->save()) {
        $timeSlot = $this->findModel($timeSlot->id);

        $this->sendNotifications(
            $timeSlot->course->courseStudentAssignments,
            'Призначено нову зустріч для курсу "' . $timeSlot->course->name . '"'
        );

        AlertNotificationController::setNotification(
            AlertNotificationController::SUCCESS_TYPE,
            'Планування було успішно створено'
        );

        return $this->renderAjax('../calendar/_course_time_slots_list', [
            'timeSlots' => CalendarController::getTimeSlots()
        ]);
    } else {
        AlertNotificationController::setNotification(
            AlertNotificationController::ERROR_TYPE,
            Json::encode($timeSlot->errors)
        );
    }
}
}

```



```

public function actionUpdate() {
    $timeSlot = $this->findModel($this->request->post()['CourseTimeSlot']['id']);

    if ($this->request->isPost && $timeSlot->load($this->request->post()) &&
    $timeSlot->save()) {
        $this->sendNotifications(
            $timeSlot->course->courseStudentAssignments,
            'Оновлено зустріч для курсу "' . $timeSlot->course->name . '"'
        );

        AlertNotificationController::setNotification(
            AlertNotificationController::SUCCESS_TYPE,
            'Планування було успішно відредаговано'
        );

        return $this->renderAjax('../calendar/_course_time_slots_list', [
            'timeSlots' => CalendarController::getTimeSlots()
        ]);
    } else {
        AlertNotificationController::setNotification(
            AlertNotificationController::ERROR_TYPE,
            Json::encode($timeSlot->errors)
        );
    }
}

public function actionDelete() {
    $timeSlot = $this->findModel(Yii::$app->request->post('id'));

    if ($timeSlot->delete()) {
        $this->sendNotifications(
            $timeSlot->course->courseStudentAssignments,
            'Видалено зустріч для курсу "' . $timeSlot->course->name . '"'
        );

        AlertNotificationController::setNotification(
            AlertNotificationController::SUCCESS_TYPE,
            'Планування було успішно видалено'
        );

        return $this->renderAjax('../calendar/_course_time_slots_list', [
            'timeSlots' => CalendarController::getTimeSlots()
        ]);
    } else {
        AlertNotificationController::setNotification(
            AlertNotificationController::ERROR_TYPE,
            Json::encode($timeSlot->errors)
        );
    }
}

protected function sendNotifications($studentAssignments, $message) {
    foreach ($studentAssignments as $studentAssignment) {
        NotificationUtils::sendNotification(
            $studentAssignment->student_id,
            $message
        );
    }
}

protected function findModel($id)
{
    if (($model = CourseTimeSlot::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('The requested page does not exist.');
```

frontend/controllers/CourseUtils.php

```

<?php

namespace frontend\controllers;

use common\controllers\UserUtils;
use common\models\Course;
use Yii;

class CourseUtils {
    public static function getAssignedCourses() {
        $assignedCourses = [];

        if (UserUtils::currentUserIsStudent()) {
            $assignedCourses = self::getAssignedToStudentCourses(Yii::$app->user->identity);
        } else if (UserUtils::currentUserIsTeacher()) {
            $assignedCourses = self::getAssignedToTeacherCourses(Yii::$app->user->identity);
        }

        return $assignedCourses;
    }

    protected static function getAssignedToStudentCourses($student) {
        $courses = [];

        if (!empty($student->courseStudentAssignments)) {
            $courseIds = [];

            foreach ($student->courseStudentAssignments as $assignment) {
                $courseIds[] = $assignment->course_id;
            }

            $courses = Course::find()->where(['status' => Course::ONGOING_STATUS])
                ->andWhere(['in', 'id', $courseIds])
                ->all();
        }

        return $courses;
    }

    protected static function getAssignedToTeacherCourses($teacher) {
        $courses = [];

        foreach ($teacher->courses as $course) {
            if ($course->status === Course::ONGOING_STATUS) {
                $courses[] = $course;
            }
        }

        return $courses;
    }
}

```

frontend/controllers/MarksController.php

```

<?php

namespace frontend\controllers;

use common\controllers\UserUtils;
use common\models\Course;
use common\models\StudentMark;
use yii\helpers\Json;
use yii\web\Controller;
use Yii;

class MarksController extends Controller {
    public function actionIndex() {
        return $this->render('index');
    }
}

```

```

    }

    public function actionFetchMarksForSpecificPeriod() {
        $startDate = new \DateTime(Yii::$app->request->get('startDate'));
        $endDate = new \DateTime(Yii::$app->request->get('endDate'));

        $assignedCourses = CourseUtils::getAssignedCourses();
        $studentMarks = $this->getRelatedStudentMarks($startDate, $endDate, $this->getCourseIds($assignedCourses));

        $params = [
            'assignedCourses' => $assignedCourses,
            'studentMarks' => $studentMarks,
            'startDate' => $startDate,
            'endDate' => $endDate
        ];

        if (UserUtils::currentUserIsTeacher()) {
            $params['appliedCourse'] = $this->getAppliedCourse($assignedCourses);
        }

        return $this->renderAjax('_marks_table', $params);
    }

    protected function getCourseIds($courses) {
        $courseIds = [];

        foreach ($courses as $course) {
            $courseIds[] = $course->id;
        }

        return $courseIds;
    }

    protected function getRelatedStudentMarks($startDate, $endDate, $courseIds) {
        $query = StudentMark::find()
            ->where(['in', 'course_id', $courseIds])
            ->andWhere(['>=', 'related_date', $startDate->format('Y-m-d')]);

        $query ->andWhere(['<=', 'related_date', $endDate->format('Y-m-d')]);

        if (UserUtils::currentUserIsStudent()) {
            $query->andWhere(['student_id' => Yii::$app->user->identity]);
        }

        return $query->all();
    }

    protected function getAppliedCourse($assignedCourses) {
        $appliedCourse = new Course();
        $appliedCourseId = Yii::$app->request->get('appliedCourseId');

        if ($appliedCourseId === null && !empty($assignedCourses)) {
            $appliedCourse = $assignedCourses[0];
        } else if ($appliedCourseId !== null) {
            $appliedCourseId = intval($appliedCourseId);

            foreach ($assignedCourses as $course) {
                if ($course->id !== $appliedCourseId) {
                    continue;
                }

                $appliedCourse = $course;
                break;
            }
        }

        return $appliedCourse;
    }
}

```

```

public function actionSave() {
    $marksToUpsert = Json::decode(Yii::$app->request->post('marksToUpsert'));
    $marksToDelete = Json::decode(Yii::$app->request->post('marksToDelete'));

    foreach ($marksToUpsert as $mark) {
        $studentMark = new StudentMark();

        if ($mark['id'] !== null) {
            $studentMark = StudentMark::findOne(['id' => $mark['id']]);
        }

        $studentMark->course_id = intval($mark['course_id']);
        $studentMark->student_id = intval($mark['student_id']);
        $studentMark->mark = intval($mark['mark']);
        $studentMark->related_date = $mark['related_date'];

        if (!$studentMark->save()) {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($studentMark->errors)
            );
        }
    }

    foreach ($marksToDelete as $mark) {
        $studentMark = StudentMark::findOne(['id' => $mark['id']]);

        $studentMark->delete();
    }
}
}

```

frontend/controllers/NotificationUtils.php

```

<?php

namespace frontend\controllers;

use common\models\Notification;
use yii\helpers\Json;

class NotificationUtils {
    public static function sendNotification($userId, $message) {
        $notification = new Notification();

        $notification->user_id = $userId;
        $notification->content = $message;

        if (!$notification->save()) {
            AlertNotificationController::setNotification(
                AlertNotificationController::ERROR_TYPE,
                Json::encode($notification->errors)
            );
        }
    }
}

```

frontend/controllers/UserController.php

```

<?php

namespace frontend\controllers;

use common\models\User;
use yii\web\Controller;
use Yii;
use yii\web\NotFoundHttpException;

class UserController extends Controller {
    public function actionView()

```

```

{
    $id = Yii::$app->request->get('id');

    return $this->render('view', [
        'user' => $this->findModel($id),
    ]);
}

protected function findModel($id)
{
    if (($model = User::findOne(['id' => $id])) !== null) {
        return $model;
    }

    throw new NotFoundHttpException('Даний користувач не знайдений у системі.');
}
}

```

frontend/controllers/SiteController.php

<?php

```

namespace frontend\controllers;

use Yii;
use yii\web\Controller;
use yii\filters\VerbFilter;
use yii\filters\AccessControl;
use common\models\LoginForm;

class SiteController extends Controller {
    public function behaviors() {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'rules' => [
                    [
                        'actions' => ['login', 'error', 'signup'],
                        'allow' => true,
                    ],
                    [
                        'actions' => ['logout', 'index'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'logout' => ['post'],
                ],
            ],
        ];
    }

    public function actions() {
        return [
            'error' => [
                'class' => 'yii\web>ErrorAction',
            ],
            'captcha' => [
                'class' => 'yii\captcha\CaptchaAction',
                'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
            ],
        ];
    }

    public function actionIndex() {
        $this->redirect(['/course/index']);
    }
}

```

```

public function actionLogin() {
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new LoginForm();
    if ($model->load(Yii::$app->request->post()) && $model->login()) {
        return $this->goBack();
    }

    $model->password = '';
    $this->layout = 'login';

    return $this->render('login', [
        'model' => $model,
    ]);
}

public function actionLogout() {
    Yii::$app->user->logout();

    return $this->goHome();
}
}

```

Представления:

frontend/views/alert-notification/view.php

```

<?php ?>

<div class="alert-notification fixed-top p-4">
    <?php if (Yii::$app->session->hasFlash('success')): ?>
        <div class="alert alert-success alert-dismissible">
            <button aria-hidden="true" data-dismiss="alert" class="close"
type="button">×</button>

                <div class="d-flex align-items-center gap-3">
                    <i class="icon fa fa-check"></i>

                    <span class="alert-message">
                        <?= Yii::$app->session->getFlash('success') ?>
                    </span>
                </div>
            <?php endif; ?>

            <?php if (Yii::$app->session->hasFlash('error')): ?>
                <div class="alert alert-danger alert-dismissible">
                    <button aria-hidden="true" data-dismiss="alert" class="close"
type="button">×</button>

                        <div class="d-flex align-items-center gap-3">
                            <i class="bi bi-x-circle"></i>

                            <span class="alert-message">
                                <?= Yii::$app->session->getFlash('error') ?>
                            </span>
                        </div>
                    </div>
                <?php endif; ?>
            </div>

```

frontend/views/calendar/_course_time_slots_list.php

```

<?php

/** @var $this View */
/** @var $timeSlots \common\models\CourseTimeSlot[] */

use yii\helpers\Json;
use yii\web\View;

```

```

$events = Array();

if (!empty($timeSlots)) {
    foreach ($timeSlots as $timeSlot) {
        $events[] = [
            'text' => $timeSlot->course->name,
            'start' => $timeSlot->time_from,
            'end' => $timeSlot->time_to,
            'id' => $timeSlot->id
        ];
    }
}

$this->registerJs(
    'var timeSlots = ' . Json::encode($events) . ';',
    View::POS_HEAD
);

?>

<div class="modal fade course-time-slot-modal-content"
    id="course-time-slot-modal"
    data-bs-backdrop="static"
    data-bs-keyboard="false"
    tabindex="-1"
    aria-labelledby="course-task-submission-modal-label"
    aria-hidden="true"
></div>

```

frontend/views/calendar/index.php

```

<?php

/** @var $this \yii\web\View */
/** @var $timeSlots \common\models\CourseTimeSlot[] */

use yii\bootstrap5\BootstrapAsset;
use yii\web\View;

$this->title = 'Розклад';

$styles = [
    'calendar_g',
    'calendar_green',
    'calendar_traditional',
    'calendar_transparent',
    'calendar_white'
];

foreach ($styles as $style) {
    $path = '@web/css/calendar-theme/' . $style . '.css';
    $this->registerCssFile($path, ['depends' => [BootstrapAsset::class]]);
}

$this->registerJs(
    'var userType = "' . Yii::$app->user->identity->getUserType() . '";',
    View::POS_HEAD
);

$this->registerJsFile(Yii::getAlias('@web') . '/js/calendar-page.js', [
    'depends' => [yii\web\JqueryAsset::class]
]);

?>

<div class="h-100 pt-3">
    <div class="card d-flex flex-column shadow-lg h-100">
        <div class="container-fluid rounded-top bg-purple shadow-sm px-0 px-sm-5 px-
md-5 px-lg-5 pt-3 pb-2">

```

```

3">
    <div class="d-flex align-items-center justify-content-center gap-1 gap-md-
        <i class="bi bi-caret-left-fill period-navigation-button fs-4"
            onclick="handlePeriodButtonClick('previous') "
        ></i>

        <h1 class="display-6 text-center"
            id="current-period-text"
        ></h1>

        <i class="bi bi-caret-right-fill period-navigation-button fs-4"
            onclick="handlePeriodButtonClick('next') "
        ></i>
    </div>
</div>

<div class="flex-grow-1 overflow-auto bg-purple-scroll p-0 h-100">
    <div id="time-slots-container">
        <?= $this->render('_course_time_slots_list', [
            'timeSlots' => $timeSlots
        ]) ?>
    </div>

    <div id="dp"></div>
</div>
</div>
</div>

```

frontend/views/course/_courses_list.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courses \common\models\Course[] */

?>

<div class="card d-flex flex-column bg-purple shadow-lg h-100 border-0">
    <div class="form-floating text-dark">
        <input class="form-control rounded-top"
            id="search-input"
            type="text"
            placeholder="Пошук"
            aria-label="Пошук"
            oninput="handleCourseSearchInputChange(event) "
        >

        <label for="search-input">
            <i class="bi bi-search"></i>
            <span class="ms-1">Пошук...</span>
        </label>
    </div>

    <div class="list-group list-group-flush flex-grow-1 overflow-auto courses-list bg-
purple-scroll"
        id="courses"
        role="tablist"
    >
        <?php
            foreach ($courses as $course):
                echo $this->render('_courses_list_item', [
                    'course' => $course
                ]);
            endforeach;
        ?>
    </div>
</div>

```


frontend/views/course/_courses_list_item.php

```

<?php

/** @var $this \yii\web\View */
/** @var $course \common\models\Course */

?>

<button class="list-group-item list-group-item-action lh-tight course-option"
        data-bs-toggle="list"
        type="button"
        role="tab"
        data-id="<?= $course->id ?>"
        data-name="<?= $course->name ?>"
        onclick="handleCoursesListItemClick(event, <?= $course->id ?>)"
    >
    <div class="d-flex align-items-center justify-content-start">
        <div class="subject-icon me-2">
            <i class="bi bi-circle fs-2"></i>
        </div>

        <div class="subject-name text-truncate">
            <?= $course->name ?>
        </div>
    </div>
</button>

```

frontend/views/course/_details.php

```

<?php

/** @var \yii\web\View $this */
/** @var $course \common\models\Course */

?>

<div class="card d-flex flex-column shadow-lg h-100"
        id="course-details"
        data-course-id="<?= $course->id ?>"
    >
    <div class="container-fluid rounded-top bg-purple shadow-sm px-0 px-sm-5 px-md-5
px-lg-5 pt-3">
        <nav id="course-details-tabs">
            <div class="nav nav-tabs align-items-end justify-content-start border-0"
                id="nav-tab"
                role="tablist"
            >
                <button class="nav-link text-white text-truncate col-12 col-sm-auto
col-md-auto active"
                    id="nav-course-general-chat-tab"
                    data-bs-toggle="tab"
                    data-bs-target="#nav-course-general-chat"
                    type="button"
                    role="tab"
                    aria-controls="nav-course-general-chat"
                    aria-selected="true"
                >
                    <span>Загальний чат</span>
                </button>

                <button class="nav-link text-white text-truncate col-12 col-sm-auto
col-md-auto"
                    id="nav-course-tasks-tab"
                    data-bs-toggle="tab"
                    data-bs-target="#nav-course-tasks"
                    type="button"
                    role="tab"
                    aria-controls="nav-course-tasks"
                    aria-selected="false"
                >

```

```

        <span>Завдання</span>
    </button>

    <div class="col-12 col-sm-auto ms-md-auto align-self-stretch">
        <div class="d-flex flex-column align-items-center h-100 text-
white">

            <h1 class="display-6">
                <?= $course->name ?>
            </h1>

            <span class="pe-3 pb-2">
                Вчитель: <?= $course->teacher->getLinkedName('text-white')
?>
            </span>
        </div>
    </div>
</div>
</nav>
</div>

<div class="tab-content flex-grow-1 overflow-auto h-100" id="nav-tab-content">
    <div class="tab-pane h-100 show active"
        id="nav-course-general-chat"
        role="tabpanel"
        aria-labelledby="nav-course-general-chat-tab"
    >
        <?= $this->render('_general_chat_tab', [
            'course' => $course
        ]) ?>
    </div>

    <div class="tab-pane h-100 course-tasks-container"
        id="nav-course-tasks"
        role="tabpanel"
        aria-labelledby="nav-course-tasks-tab"
    >
        <?= $this->render('_tasks_tab', [
            'course' => $course
        ]) ?>
    </div>
</div>
</div>

```

frontend/views/course/_general_chat_tab.php

```

<?php

/** @var \yii\web\View $this */
/** @var $course \common\models\Course */

?>

<div class="d-flex flex-column justify-content-between h-100">
    <div class="flex-grow-1 overflow-auto course-chat-messages-list-container bg-
purple-scroll px-5">
        <?= $this->render('../course-chat-message/_course_chat_messages_list', [
            'courseChatMessages' => $course->courseChatMessages
        ]); ?>
    </div>

    <div class="input-area px-5 py-4 rounded border-top">
        <div class="d-flex align-items-center">
            <div class="flex-grow-1">
                <textarea class="form-control"
                    id="course-chat-message-input"
                    placeholder="Напишіть повідомлення..."
                    onkeyup="handleCoursesChatMessageTyping(event, <?= $course-
>id ?>)"
                ></textarea>
            </div>

```

```

        <div class="p-2">
            <button class="btn text-white shadow-none p-0"
                type="button"
                onclick="handleCoursesChatMessageSending(event, <?= $course-
>id ?>)"
            >
                
            </button>
        </div>
    </div>
</div>
</div>
</div>

```

frontend/views/course/_tasks_tab.php

```

<?php

/** @var \yii\web\View $this */
/** @var $course \common\models\Course */

?>

<div class="d-flex flex-column justify-content-between h-100">
    <div class="course-tasks-list-container h-100">
        <?= $this->render('../course-task/_course_tasks_list', [
            'courseTasks' => $course->courseTasks,
            'courseId' => $course->id
        ]) ?>
    </div>
</div>

```

frontend/views/course/index.php

```

<?php

/** @var $this \yii\web\View */
/** @var $assignedCourses \common\models\Course[] */

use yii\bootstrap5\BootstrapAsset;

$this->registerJsFile(Yii::getAlias('@web') . '/js/courses-page.js', [
    'depends' => [yii\web\jQueryAsset::class]
]);

$this->registerCssFile('@web/css/courses.css', [
    'depends' => [BootstrapAsset::class]
]);

$this->title = 'Курс';

?>

<div class="h-100 pt-3">
    <div class="d-flex flex-wrap h-100">
        <div class="col-12 col-sm-3 col-md-3 col-lg-2 h-100 p-0">
            <?= $this->render('_courses_list', [
                'courses' => $assignedCourses
            ]) ?>
        </div>

        <div class="col-12 col-sm-9 col-md-9 col-lg-10 h-100 p-0">
            <div class="ms-0 ms-sm-3 ms-md-3 ms-lg-3 mt-3 mt-sm-0 h-100 course-
details-container">
                </div>
            </div>
        </div>
    </div>

```

frontend/views/course-chat-message/_course_chat_messages_list.php

```
<?php

/** @var \yii\web\View $this */
/** @var $courseChatMessages \common\models\CourseChatMessage[] */

use common\controllers\DateTimeUtils;

$dateToCourseChatMessages = array();

foreach ($courseChatMessages as $chatMessage) {
    $sendDateKey = (new DateTime())->setTimestamp($chatMessage->created_at)->format('Y-m-d');

    if ($dateToCourseChatMessages[$sendDateKey] === null) {
        $dateToCourseChatMessages[$sendDateKey] = array();
    }

    $dateToCourseChatMessages[$sendDateKey][] = $chatMessage;
}

?>

<?php if (empty($courseChatMessages)) : ?>
    <div class="col-12 text-center pt-4">
        <span class="text-muted fs-5">
            Поки що ні одного повідомлення...
        </span>
    </div>
<?php else : ?>
    <div class="vstack gap-2 my-2">
        <?php
            foreach (array_keys($dateToCourseChatMessages) as $key) {
                $currentDate = new DateTime($key);
                $formattedDate = DateTimeUtils::formatFullDate($currentDate);

                echo '<div class="text-muted text-center py-3">' . $formattedDate .

                foreach ($dateToCourseChatMessages[$key] as $courseChatMessage) {
                    echo $this->render('_course_chat_messages_list_item', [
                        'courseChatMessage' => $courseChatMessage
                    ]);
                }
            }
        ?>
    </div>
<?php endif ?>
```

frontend/views/course-chat-message/_course_chat_messages_list_item.php

```
<?php

/** @var \yii\web\View $this */
/** @var $courseChatMessage \common\models\CourseChatMessage */

$isCurrentUserMessage = ($courseChatMessage->sender_id == Yii::$app->user->getId());

?>

<?php if ($isCurrentUserMessage): ?>
    <div class="col-12 col-sm-10 col-md-8 d-flex justify-content-end p-0 ms-auto">
        <div class="d-inline-flex message">
            <div class="d-flex flex-column">
                <div class="d-flex align-items-end gap-3 rounded px-3 py-2 shadow-sm bg-purple-dark">
                    <div class="message-text">
                        <?= nl2br($courseChatMessage->content) ?>
                    </div>
                    <span class="message-time text-white">
```

```

        <small>
            <?= date("H:i", $courseChatMessage->updated_at) ?>
        </small>
    </span>
</div>
</div>
</div>
<?php else: ?>
    <div class="col-12 col-sm-10 col-md-8 d-flex justify-content-end p-0 me-auto">
        <div class="d-inline-flex message me-auto">
            <div class="d-flex flex-column">
                <?= $courseChatMessage->sender->getLinkedName('text-muted mb-1') ?>

                <div class="d-flex align-items-end gap-3 rounded px-3 py-2 shadow-sm
bg-light">
                    <div class="message-text">
                        <?= nl2br($courseChatMessage->content) ?>
                    </div>

                    <span class="message-time text-muted">
                        <small>
                            <?= date("H:i", $courseChatMessage->updated_at) ?>
                        </small>
                    </span>
                </div>
            </div>
        </div>
    </div>
<?php endif; ?>

```

frontend/views/course-task/_course_tasks_list.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTasks \common\models\CourseTask[] */
/** @var $courseId string */

use common\models\CourseTask;
use common\controllers\UserUtils;

$affectedCourseTasks = [];

if (UserUtils::currentUserIsStudent()) {
    foreach ($courseTasks as $courseTask) {
        if ($courseTask->status !== CourseTask::ONGOING_STATUS) {
            continue;
        }

        $affectedCourseTasks[] = $courseTask;
    }
} else if (UserUtils::currentUserIsTeacher()) {
    $affectedCourseTasks = $courseTasks;
}

?>

<div class="d-flex flex-column justify-content-between h-100">
    <div class="flex-grow-1 overflow-auto bg-purple-scroll">
        <div class="list-group list-group-flush course-tasks-list">
            <?php
                if (UserUtils::currentUserIsStudent()) {
                    foreach ($affectedCourseTasks as $courseTask) {
                        echo $this->render('_student_course_tasks_list_item', [
                            'courseTask' => $courseTask
                        ]);
                    }
                } else if (UserUtils::currentUserIsTeacher()) {
                    foreach ($affectedCourseTasks as $courseTask) {
                        echo $this->render('_teacher_course_tasks_list_item', [

```

```

                'courseTask' => $courseTask
            ]);
        }
    }
    ?>
</div>
</div>

<div class="course-tasks-statistics">
    <?= $this->render('_statistics', [
        'courseTasks' => $affectedCourseTasks,
        'courseId' => $courseId
    ]) ?>
</div>

<div class="modal fade course-task-modal-content"
    id="course-task-modal"
    data-bs-backdrop="static"
    data-bs-keyboard="false"
    tabindex="-1"
    aria-labelledby="course-task-modal-label"
    aria-hidden="true"
></div>

<div class="modal fade course-task-submission-modal-content"
    id="course-task-submission-modal"
    data-bs-backdrop="static"
    data-bs-keyboard="false"
    tabindex="-1"
    aria-labelledby="course-task-submission-modal-label"
    aria-hidden="true"
></div>
</div>

```

frontend/views/course-task/_statistics.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTasks \common\models\CourseTask[] */
/** @var $courseId string */

use common\controllers\UserUtils;

?>

<div class="px-5 py-4 rounded border-top">
    <?php if (UserUtils::currentUserIsStudent()): ?>
        <?= $this->render('_student_statistics', [
            'courseTasks' => $courseTasks
        ]) ?>
    <?php elseif (UserUtils::currentUserIsTeacher()): ?>
        <?= $this->render('_teacher_statistics', [
            'courseTasks' => $courseTasks,
            'courseId' => $courseId
        ]) ?>
    <?php endif; ?>
</div>

```

frontend/views/course-task/_student_course_tasks_list_item.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

use common\models\CourseTaskSubmission;
use common\controllers\DateTimeUtils;

$currentUserId = Yii::$app->user->id;

```

```

$appliedTaskSubmission = null;

if (!empty($courseTask->courseTaskSubmissions)) {
    foreach ($courseTask->courseTaskSubmissions as $taskSubmission) {
        if ($taskSubmission->student_id === $currentUserId) {
            $appliedTaskSubmission = $taskSubmission;
            break;
        }
    }
}

$taskSubmissionStatus = $appliedTaskSubmission === null ? 'NULL' :
$appliedTaskSubmission->status;

$statusLabel = [
    'NULL' => 'Не розпочато',
    CourseTaskSubmission::PENDING_STATUS => 'Відправлено на перевірку',
    CourseTaskSubmission::COMPLETED_STATUS => 'Виконано',
    CourseTaskSubmission::REJECTED_STATUS => 'Повернено'
];

$statusClass = [
    'NULL' => 'text-muted',
    CourseTaskSubmission::PENDING_STATUS => 'text-info',
    CourseTaskSubmission::COMPLETED_STATUS => 'text-success',
    CourseTaskSubmission::REJECTED_STATUS => 'text-danger'
];

$statusIcon = [
    CourseTaskSubmission::COMPLETED_STATUS => 'bi bi-check2'
];

?>

<div class="d-flex gap-3 align-items-center list-group-item list-group-item-action px-
5 course-task-item"
    data-id="<?= $courseTask->id ?>"
    onclick="handleCourseTasksListItemClick(event, <?= $courseTask->id ?>)"
>
    <div class="fs-4 py-2 px-3 rounded-circle bg-purple-dark">
        <i class="bi bi-journal-text"></i>
    </div>

    <div class="fs-5">
        <?= $courseTask->name ?>
    </div>

    <div class="d-flex flex-column ms-auto align-self-end">
        <div class="text-end">
            <div class="<?= $statusClass[$taskSubmissionStatus] ?>">
                <i><?= $statusLabel[$taskSubmissionStatus] ?></i>
                <i class="<?= $statusIcon[$taskSubmissionStatus] ?>"></i>
            </div>
        </div>

        <small>
            Оновлено <?= DateTimeUtils::formatFullDateTimeRelated((new DateTime()) -
>setTimestamp($courseTask->updated_at) ?>
        </small>
    </div>
</div>

```

frontend/views/course-task/_student_details_view.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

use common\models\CourseTaskSubmission;
use common\controllers\DateTimeUtils;

```



```

        <?= DateTimeUtils::formatFullDateTime(new
DateTime($courseTask->end_date)) ?>
    </div>
</div>

<div class="mt-3 d-flex flex-wrap gap-3 gap-lg-5">
    <div class="task-status">
        <div class="text-muted fs-6">
            <i class="bi bi-journal-medical"></i>
            <span>Статус завдання</span>
        </div>

        <div>
            <span class="<?= $statusClass[$taskSubmissionStatus]
?>">

                <?= $statusLabel[$taskSubmissionStatus] ?>
            </span>
        </div>
    </div>

    <?php if ($taskSubmissionStatus !== NULL &&
$appliedTaskSubmission->status === CourseTaskSubmission::COMPLETED_STATUS): ?>
        <div class="submission-mark">
            <div class="text-muted fs-6">
                <i class="bi bi-card-checklist"></i>
                <span>Оцінка</span>
            </div>

            <div>
                <?= $appliedTaskSubmission->mark ?>
            </div>
        </div>
    <?php endif; ?>
</div>

<?php if ($taskNeedsToBeSubmitted):
    $courseTaskSubmission = new CourseTaskSubmission();

    $form = ActiveForm::begin([
        'id' => 'course-task-submission-form',
        'options' => [
            'enctype' => 'multipart/form-data',
            'novalidate' => true
        ]
    ]);
?>
    <?= $form->field($courseTaskSubmission, 'course_task_id',
['options' => ['class' => 'mb-0']])
        ->hiddenInput(['value' => $courseTask->id])
        ->label(false)
    ?>

    <?= $form->field($courseTaskSubmission, 'student_id',
['options' => ['class' => 'mb-0']])
        ->hiddenInput(['value' => Yii::$app->user->id])
        ->label(false)
    ?>

    <div class="submission-content-uploader d-grid mt-4">
        <label for="submission-content-input" class="form-
label">Прикріпити файл:</label>

        <?= $form->field($courseTaskSubmission,
'attachments[]')
            ->fileInput(['multiple' => true])
            ->label(false)
        ?>
    </div>

    <div class="submission-comments mt-3">

```

```

        <label for="task-submission-comments-input"
class="form-label">Коментар:</label>

        <?= $form->field($courseTaskSubmission, 'comment')
            ->textarea([
                'rows' => '5',
                'placeholder' => 'Коментар...'
            ])
            ->label(false)
        ?>
    </div>
    <?php ActiveForm::end(); ?>
    <?php else: ?>
        <div class="submission-content-uploader d-grid mt-4">
            <label for="submission-content-input" class="form-
label">Прикріплені файли:</label>

            <?php foreach ($appliedTaskSubmission-
>getAttachments() as $attachment): ?>
                <div class="d-flex gap-2">
                    <i class="bi bi-file-earmark-text"></i>
                    <a href="<?= Yii::$app->urlManager-
>createUrl(['attachment/download', 'path' => $attachment->path]) ?>"
                        <?= $attachment->getName() . '.' .
$attachment->getExtension() ?>
                        </a>
                    </div>
                <?php endforeach; ?>
            </div>

            <div class="submission-comments mt-3">
                <label for="task-submission-comments-input"
class="form-label">Коментар:</label>
                <textarea class="form-control"
                    id="task-submission-comments-input"
                    placeholder="<?= $appliedTaskSubmission-
>comment ?>"
                    rows="5"
                    readonly
                ></textarea>
            </div>
        <?php endif; ?>
    </div>
</div>
</div>

<div class="modal-footer justify-content-start">
    <button class="btn ms-auto" type="button" data-bs-
dismiss="modal">Закрити</button>

    <?php if ($taskNeedsToBeSubmitted): ?>
        <button class="btn bg-purple text-white"
            type="button"
            onclick="handleCourseTaskSubmission(event, <?= $courseTask->id
?>) ">
            >
            Здати на перевірку
        </button>
    <?php endif; ?>
</div>
</div>
</div>

```

frontend/views/course-task/_student_statistics.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTasks \common\models\CourseTask[] */

```

```

use common\models\CourseTaskSubmission;

$currentUserId = Yii::$app->user->id;
$appliedCourseTaskSubmissions = [];

foreach ($courseTasks as $courseTask) {
    if (empty($courseTask->courseTaskSubmissions)) {
        continue;
    }

    foreach ($courseTask->courseTaskSubmissions as $submission) {
        if ($submission->student_id === $currentUserId) {
            $appliedCourseTaskSubmissions[] = $submission;
        }
    }
}

$courseTasksCount = count($courseTasks);

$notStartedCourseTasks = $courseTasksCount - count($appliedCourseTaskSubmissions);
$pendingCourseTasks = 0;
$completedCourseTasks = 0;
$rejectedCourseTasks = 0;

foreach ($appliedCourseTaskSubmissions as $submission) {
    if ($submission->status === CourseTaskSubmission::PENDING_STATUS) {
        $pendingCourseTasks++;
    }

    if ($submission->status === CourseTaskSubmission::COMPLETED_STATUS) {
        $completedCourseTasks++;
    }

    if ($submission->status === CourseTaskSubmission::REJECTED_STATUS) {
        $rejectedCourseTasks++;
    }
}

?>

```

```

<div class="d-flex align-items-center gap-5">
    <div class="text-success">
        Виконано: <?= $completedCourseTasks .'/' . $courseTasksCount ?>
    </div>

    <div class="text-info">
        На перевірці: <?= $pendingCourseTasks .'/' . $courseTasksCount ?>
    </div>

    <div class="text-danger">
        Повернено: <?= $rejectedCourseTasks .'/' . $courseTasksCount ?>
    </div>

    <div class="text-muted">
        Не розпочато: <?= $notStartedCourseTasks .'/' . $courseTasksCount ?>
    </div>
</div>

```

frontend/views/course-task/_teacher_course_tasks_list_item.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

use common\models\CourseTask;
use common\controllers\DateTimeUtils;

$statusLabel = [

```

```

CourseTask::DRAFT_STATUS => 'Не активне',
CourseTask::ONGOING_STATUS => 'Активне',
CourseTask::ARCHIVED_STATUS => 'Архівне'
];

$statusClass = [
    CourseTask::DRAFT_STATUS => 'text-muted',
    CourseTask::ONGOING_STATUS => 'text-success'
];

$statusIcon = [
    CourseTask::ONGOING_STATUS => 'bi bi-check2'
];

?>

<div class="d-flex gap-3 align-items-center list-group-item list-group-item-action px-5 course-task-item"
    data-id="<?=$courseTask->id ?>"
    onclick="handleCourseTasksListItemClick(event, <?=$courseTask->id ?>)"
>
    <div class="fs-4 py-2 px-3 rounded-circle bg-purple-dark">
        <i class="bi bi-journal-text"></i>
    </div>

    <div class="fs-5">
        <?=$courseTask->name ?>
    </div>

    <div class="d-flex flex-column ms-auto align-self-end">
        <div class="text-end">
            <div class="<?=$statusClass[$courseTask->status] ?>">
                <i><?=$statusLabel[$courseTask->status] ?></i>
                <i class="<?=$statusIcon[$courseTask->status] ?>"></i>
            </div>
        </div>

        <small>
            Оновлено <?=$DateTimeUtils::formatFullDateTimeRelated((new DateTime()) -
>setTimestamp($courseTask->updated_at) ?>
        </small>
    </div>

    <div class="d-flex align-items-center actions">
        <div class="edit-action">
            <button class="btn"
                type="button"
                title="Редагувати"
                onclick="handleCourseTaskUpdateTriggering(event, <?=$courseTask->id ?>)"
            >
                <i class="bi bi-pencil-fill fs-5 text-purple-dark"></i>
            </button>
        </div>

        <div class="delete-action">
            <button class="btn"
                type="button"
                title="Видалити"
                onclick="handleCourseTaskDeleteTriggering(event, <?=$courseTask->id ?>)"
            >
                <i class="bi bi-trash fs-5 text-purple-dark"></i>
            </button>
        </div>
    </div>
</div>

```



```

        <span>Статус завдання</span>
    </div>

    <div>
        <span class="<?= $statusClass[$courseTask->status]
?>">

            <?= $statusLabel[$courseTask->status] ?>
        </span>
    </div>
</div>

<?php if ($courseTask->status !== CourseTask::DRAFT_STATUS):
?>

    <button class="btn bg-purple-dark text-white"
        type="button"
        data-bs-toggle="collapse"
        data-bs-target="#course-task-submissions"
        aria-expanded="false"
        aria-controls="course-task-submissions"
    >
        <div class="d-flex gap-2 justify-content-center">
            <i class="bi bi-arrows-expand"></i>
            <span>Показати результати виконання</span>
        </div>
    </button>
    <?php endif; ?>
</div>
</div>

    <div class="col-12 collapse" id="course-task-submissions">
        <div class="mt-3 course-task-submissions-list-container">
            <?= $this->render('../course-task-
submission/_course_task_submissions_list', [
                'courseTaskSubmissions' => $courseTask-
>courseTaskSubmissions,
                'courseTaskAssignees' => $courseTaskAssignees
            ]) ?>
        </div>
    </div>
</div>
</div>

    <div class="modal-footer justify-content-start">
        <button class="btn ms-auto" type="button" data-bs-
dismiss="modal">Закрити</button>
    </div>
</div>
</div>

```

frontend/views/course-task/_teacher_statistics.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTasks \common\models\CourseTask[] */
/** @var $courseId string */

?>

<div class="d-flex align-items-center">
    <div>
        <button class="btn bg-purple-dark text-white"
            type="button"
            onclick="handleCourseTaskCreationTriggering(event, <?= $courseId ?>)"
        >
            <div class="d-flex gap-2">
                <i class="bi bi-plus-lg"></i>
                <span>Нове завдання</span>
            </div>
        </button>
    </div>
</div>

```

</div>

frontend/views/course-task/create.php

[illegible]

```

        'autoclose' => true,
        'startDate' => date('Y-m-d H:i:s')
    ]
    )->label(false);
    ?>
</div>

<div class="mb-3">
    <span class="display-6 text-muted">
        -
    </span>
</div>

<div>
    <?= $form->field($courseTask, 'end_date')->widget(
        DateTimePicker::class,
        [
            'language' => 'uk',
            'convertFormat' => false,
            'pluginOptions' => [
                'todayHighlight' => true,
                'autoclose' => true,
                'startDate' => date('Y-m-d H:i:s')
            ]
        ]
    )->label(false);
    ?>
</div>
</div>
</div>

<div class="description-input-group">
    <div class="text-muted fs-6">
        <i class="bi bi-card-text"></i>
        <span>Опис</span>
    </div>

    <div class="d-grid mt-1">
        <?= $form->field($courseTask, 'description')
        ->textarea(['rows' => '5'])
        ->label(false)
    </div>
    ?>
</div>
</div>
</div>

<div class="modal-footer">
    <button class="btn"
        type="button"
        data-bs-dismiss="modal"
    >
        Закрити
    </button>

    <?= Html::submitButton(
        'Створити',
        [
            'class' => 'btn bg-purple text-white',
            'name' => 'creation-button'
        ]
    ) ?>
</div>
<?php ActiveForm::end(); ?>
</div>
</div>

```



```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

?>

<div class="modal-dialog modal-dialog-centered modal-lg">
    <div class="modal-content border-0">
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Видалення завдання
            </h5>

            <button type="button" class="btn-close btn-close-white" data-bs-
dismiss="modal" aria-label="Close"></button>
        </div>

        <div class="modal-body">
            <div class="text-center fs-5">
                Ви впевнені, що хочете видалити завдання "<i><?= $courseTask->name
?></i>"?
            </div>
        </div>

        <div class="modal-footer justify-content-center">
            <button class="btn"
                type="button"
                data-bs-dismiss="modal"
            >
                Hi
            </button>

            <button class="btn bg-purple text-white"
                type="button"
                onclick="handleCourseTaskDelete(event, <?= $courseTask->id ?>)"
            >
                Так
            </button>
        </div>
    </div>
</div>

```

frontend/views/course-task/update.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

use common\models\CourseTask;
use kartik\datetime\DateTimePicker;
use yii\bootstrap5\ActiveForm;
use yii\bootstrap5\Html;

$statusOptions = [
    CourseTask::DRAFT_STATUS => 'Не активне',
    CourseTask::ONGOING_STATUS => 'Активне',
    CourseTask::ARCHIVED_STATUS => 'Архівне'
];

?>

<div class="modal-dialog modal-lg">
    <div class="modal-content border-0">
        <?php $form = ActiveForm::begin([
            'id' => 'update-course-task-form',
            'action' => '/course-task/update',
            'options' => [
                'novalidate' => true
            ]
        ])

```

```

    ]
  }); ?>
  <div class="modal-header bg-purple">
    <h5 class="modal-title">
      Оновлення завдання
    </h5>

    <button type="button" class="btn-close btn-close-white" data-bs-
dismiss="modal" aria-label="Close"></button>
  </div>

  <div class="modal-body">
    <div class="d-flex flex-column gap-0">
      <?= $form->field($courseTask, 'id', ['options' => ['class' => 'mb-
0']])->hiddenInput()->label(false) ?>

      <div class="name-input-group">
        <div class="text-muted fs-6">
          <i class="bi bi-journal-medical"></i>
          <span>Назва</span>
        </div>

        <div class="d-grid mt-1">
          <?= $form->field($courseTask, 'name')->label(false) ?>
        </div>
      </div>

      <div class="status-input-group">
        <div class="text-muted fs-6">
          <i class="bi bi-journal-medical"></i>
          <span>Статус</span>
        </div>

        <div class="d-grid mt-1">
          <?= $form->field($courseTask, 'status')
            ->dropDownList($statusOptions)
            ->label(false)
          ?>
        </div>
      </div>

      <div class="duration-input-group">
        <div class="text-muted fs-6">
          <i class="bi bi-clock"></i>
          <span>Термін виконання завдання</span>
        </div>

        <div class="d-flex align-items-center gap-2 mt-1">
          <div>
            <?= $form->field($courseTask, 'start_date')->widget(
              DateTimePicker::class,
              [
                'language' => 'uk',
                'convertFormat' => false,
                'pluginOptions' => [
                  'todayHighlight' => true,
                  'autoclose' => true
                ]
              ]
            )->label(false);
          ?>
        </div>

        <div class="mb-3">
          <span class="display-6 text-muted">
            -
          </span>
        </div>
      </div>
    </div>
  </div>

```

```

        <div>
            <?=$form->field($courseTask, 'end_date')->widget(
                DateTimePicker::class,
                [
                    'language' => 'uk',
                    'convertFormat' => false,
                    'pluginOptions' => [
                        'todayHighlight' => true,
                        'autoclose' => true
                    ]
                ]
            )->label(false);
        ?>
    </div>
</div>

<div class="description-input-group">
    <div class="text-muted fs-6">
        <i class="bi bi-card-text"></i>
        <span>Опис</span>
    </div>

    <div class="d-grid mt-1">
        <?=$form->field($courseTask, 'description')
            ->textarea(['rows' => '5'])
            ->label(false)
        ?>
    </div>
</div>
</div>

<div class="modal-footer">
    <button class="btn"
        type="button"
        data-bs-dismiss="modal"
    >
        Закрити
    </button>

    <?=$Html::submitButton(
        'Зберегти',
        [
            'class' => 'btn bg-purple text-white',
            'name' => 'update-button'
        ]
    ) ?>
</div>
<?php ActiveForm::end(); ?>
</div>
</div>

```

frontend/views/course-task/view.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTask \common\models\CourseTask */

use common\controllers\UserUtils;

if (UserUtils::currentUserIsStudent()) {
    echo $this->render('_student_details_view', [
        'courseTask' => $courseTask
    ]);
} else if (UserUtils::currentUserIsTeacher()) {
    echo $this->render('_teacher_details_view', [
        'courseTask' => $courseTask
    ]);
}

```

```

    });
}

?>

```

frontend/views/course-task-submission/_course_task_submissions_list.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTaskSubmissions CourseTaskSubmission[] */
/** @var $courseTaskAssignees User[] */

use common\models\CourseTaskSubmission;
use common\models\User;

$assigneeToSubmission = array();

foreach ($courseTaskSubmissions as $submission) {
    $assigneeToSubmission[$submission->student_id] = $submission;
}

?>

<table class="table table-hover table-sm table-responsive-sm">
    <thead class="bg-purple-dark">
        <tr>
            <th scope="col">Учень</th>
            <th scope="col" class="column-fit">Дата надсилання</th>
            <th scope="col" class="ps-2">Коментар</th>
            <th scope="col" class="column-fit">Прикріплені файли</th>
            <th scope="col" class="column-fit">Оцінка</th>
            <th scope="col"></th>
        </tr>
    </thead>
    <tbody>
        <?php foreach ($courseTaskAssignees as $assignee) {
            echo $this->render('_course_task_submissions_list_item', [
                'student' => $assignee,
                'submission' => $assigneeToSubmission[$assignee->id]
            ]);
        }
    <?php
    </tbody>
</table>

```

frontend/views/course-task-submission/_course_task_submissions_list_item.php

```

<?php

/** @var \yii\web\View $this */
/** @var $submission common\models\CourseTaskSubmission */
/** @var $student common\models\User */

use common\models\CourseTaskSubmission;
use common\controllers\DateTimeUtils;

$submissionIsFound = ($submission !== null);

$statusTheme = [
    'NULL' => 'table-light',
    CourseTaskSubmission::PENDING_STATUS => 'table-info',
    CourseTaskSubmission::COMPLETED_STATUS => 'table-success',
    CourseTaskSubmission::REJECTED_STATUS => 'table-danger',
];

$theme = $statusTheme[$submissionIsFound ? $submission->status : 'NULL'];

?>

<tr class="<?= $theme ?> fw-light">

```

```

<td class="text-muted pt-3">
    <?= $student->getName() ?>
</td>

<?php if ($submissionIsFound): ?>
    <td class="column-fit pt-3">
        <?= DateTimeUtils::formatFullDateTimeRelated((new DateTime())-
>setTimestamp($submission->updated_at)) ?>
    </td>

    <td class="ps-2 pt-3">
        <?= nl2br($submission->comment) ?>
    </td>

    <td class="pt-3">
        <div class="d-flex justify-content-center gap-2">
            <?php if (!empty($submission->getAttachments())): ?>
                <i class="bi bi-check"></i>
            <?php else: ?>
                <i class="bi bi-dash-lg"></i>
            <?php endif; ?>
        </div>
    </td>

    <td class="text-center pt-3">
        <?= $submission->mark ?>
    </td>

    <td>
        <div class="d-flex justify-content-center">
            <div class="view-action">
                <button class="btn"
                    type="button"
                    title="Переглянути"
                    onclick="handleCourseTaskSubmissionView(event, <?=
$submission->id ?>)"
                >
                    <i class="bi bi-eye-fill fs-5 text-purple-dark"></i>
                </button>
            </div>
        </div>
    </td>
<?php else: ?>
    <td colspan="5">
        <i>Виконання завдання не розпочато</i>
    </td>
<?php endif; ?>
</tr>

```

frontend/views/course-task-submission/view.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTaskSubmission CourseTaskSubmission */

use common\models\CourseTaskSubmission;
use yii\bootstrap5\ActiveForm;

$courseTask = $courseTaskSubmission->courseTask;
$student = $courseTaskSubmission->student;

$statusLabel = [
    CourseTaskSubmission::PENDING_STATUS => 'Очікує перевірки',
    CourseTaskSubmission::COMPLETED_STATUS => 'Виконано',
    CourseTaskSubmission::REJECTED_STATUS => 'Повернено'
];

$statusColor = [
    CourseTaskSubmission::PENDING_STATUS => 'text-info',
    CourseTaskSubmission::COMPLETED_STATUS => 'text-success',

```

```

        CourseTaskSubmission::REJECTED_STATUS => 'text-rejected'
    ];

    ?>

<div class="modal-dialog modal-lg">
    <div class="modal-content border-0">
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Перегляд результату виконання завдання
            </h5>

            <button class="btn-close btn-close-white"
                type="button"
                data-bs-dismiss="modal"
                aria-label="Close"
                onclick="handleCourseTaskSubmissionModalClose()" ">
            ></button>
        </div>

        <div class="modal-body">
            <div class="row">
                <div class="col-7">
                    <div class="d-flex flex-column gap-3">
                        <div class="student">
                            <div class="text-muted fs-6">
                                <i class="bi bi-person-fill"></i>
                                <span>Учень</span>
                            </div>

                            <div>
                                <?= $student->getName() ?>
                            </div>
                        </div>

                        <div class="submission-comment">
                            <div class="text-muted fs-6">
                                <i class="bi bi-card-text"></i>
                                <span>Коментар</span>
                            </div>

                            <div>
                                <?= nl2br($courseTaskSubmission->comment) ?>
                            </div>
                        </div>

                        <div class="submission-attachment">
                            <div class="text-muted fs-6">
                                <i class="bi bi-paperclip"></i>
                                <span>Прикріплені файли</span>
                            </div>

                            <div>
                                <?php foreach ($courseTaskSubmission->getAttachments()
as $attachment): ?>
                                    <div class="d-flex gap-2">
                                        <i class="bi bi-file-earmark-text"></i>
                                        <a href="<?= Yii::$app->urlManager-
>createUrl(['attachment/download', 'path' => $attachment->path]) ?>">
                                            <?= $attachment->getName() . '.' .
$attachment->getExtension() ?>
                                        </a>
                                    </div>
                                <?php endforeach; ?>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

<div class="col-5">
  <div class="d-flex flex-column gap-3">
    <div class="submission-status">
      <div class="text-muted fs-6">
        <i class="bi bi-card-text"></i>
        <span>Статус</span>
      </div>

      <div class="<?= $statusColor[$courseTaskSubmission-
>status] ?>">
        <?= $statusLabel[$courseTaskSubmission->status] ?>
      </div>
    </div>

    <?php if ($courseTaskSubmission->status ===
CourseTaskSubmission::COMPLETED_STATUS): ?>
      <div class="submission-mark">
        <div class="text-muted fs-6">
          <i class="bi bi-card-checklist"></i>
          <span>Оцінка</span>
        </div>

        <div>
          <?= $courseTaskSubmission->mark ?>
        </div>
      </div>

      <?php elseif ($courseTaskSubmission->status ===
CourseTaskSubmission::PENDING_STATUS): ?>
        <div class="d-flex flex-column gap-2 submission-actions
mt-4">
          <div class="complete-action d-grid">
            <button class="btn btn-success"
              type="button"
              data-bs-toggle="collapse"
              data-bs-target="#course-task-submission-
mark"
              aria-expanded="false"
              aria-controls="course-task-submission-
mark">
              <div class="d-flex gap-2 justify-content-
center">
                <i class="bi bi-check"></i>
                <span>Зархувати</span>
              </div>
            </button>

            <div class="collapse" id="course-task-submission-
mark">
              <?php $form = ActiveForm::begin([
                'id' => 'course-task-submission-form',
                'options' => [
                  'class' => 'mt-2',
                  'novalidate' => true
                ]
              ]); ?>

              <?= $form->field($courseTaskSubmission,
                'mark', [
                  'template' =>
                    <div class="input-group">
                      {input}
                      <span class="input-group-
text input-group-icon bg-purple save-mark-icon"
onclick="handleCourseTaskSubmissionApproval(' . $courseTaskSubmission->id . ')"
                      >
                        <i class="bi bi-check-
lg text-white"></i>
                      </span>

```

```

                                {error}{hint}
                                </div>'
                                ]->textInput([
                                    'type' => 'number',
                                    'autofocus' => true,
                                    'placeholder' => 'Оцінка',
                                    'min' => 1,
                                    'max' => 12,
                                    'id' => 'course-task-submission-
mark-input'
                                ]->label(false)
                                ?>

                                <?php ActiveForm::end(); ?>
                                </div>
                                </div>

                                <div class="reject-action d-grid">
                                    <button class="btn btn-danger"
                                        type="button"

onclick="handleCourseTaskSubmissionReject(<?= $courseTaskSubmission->id ?>)"
                                        >
                                        <div class="d-flex gap-2 justify-content-
center">
                                            <i class="bi bi-arrow-bar-left"></i>
                                            <span>Повернути</span>
                                        </div>
                                    </button>
                                </div>
                                </div>
                                <?php endif; ?>
                                </div>
                                </div>
                                <div class="modal-footer justify-content-start">
                                    <button class="btn ms-auto"
                                        type="button"
                                        data-bs-dismiss="modal"
                                        onclick="handleCourseTaskSubmissionModalClose()"
                                    >
                                        Закрити
                                    </button>
                                </div>
                                </div>
                                </div>
                                </div>

```

frontend/views/course-time-slot/create.php

```

<?php

/** @var \yii\web\View $this */
/** @var $availableCourses \common\models\Course */
/** @var $startDate DateTime */
/** @var $endDate DateTime */

use common\models\CourseTimeSlot;
use kartik\datetime\DateTimePicker;
use yii\bootstrap5\ActiveForm;
use yii\bootstrap5\Html;

$courseTimeSlot = new CourseTimeSlot();
$courseTimeSlot->time_from = $startDate;
$courseTimeSlot->time_to = $endDate;

$courseOptions = Array();

if (!empty($availableCourses)) {
    foreach ($availableCourses as $availableCourse) {

```



```

        $courseOptions[$availableCourse->id] = $availableCourse->name;
    }
}

?>

<div class="modal-dialog modal-lg">
    <div class="modal-content border-0">
        <?php $form = ActiveForm::begin([
            'id' => 'creation-course-time-slot-form',
            'action' => '/course-time-slot/create',
            'options' => [
                'novalidate' => true
            ]
        ]); ?>
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Створення планування
            </h5>

            <button class="btn-close btn-close-white" type="button" data-bs-
dismiss="modal" aria-label="Close"></button>
        </div>

        <div class="modal-body">
            <div class="d-flex flex-column gap-2">
                <?= $form->field($courseTimeSlot, 'status', ['options' => ['class'
=> 'mb-0']])
                    ->hiddenInput(['value' => CourseTimeSlot::ACTIVE_STATUS])
                    ->label(false)
                ?>

                <div class="course-input-group">
                    <div class="text-muted fs-6">
                        <i class="bi bi-journal-medical"></i>
                        <span>Курс</span>
                    </div>

                    <div class="d-grid mt-1">
                        <?= $form->field($courseTimeSlot, 'course_id')
                            ->dropDownList($courseOptions)
                            ->label(false)
                        ?>
                    </div>
                </div>

                <div class="duration-input-group">
                    <div class="text-muted fs-6">
                        <i class="bi bi-clock"></i>
                        <span>Час планування</span>
                    </div>

                    <div class="d-flex align-items-center gap-2 mt-1">
                        <div>
                            <?= $form->field($courseTimeSlot, 'time_from')-
>widget(
                                DateTimePicker::class,
                                [
                                    'language' => 'uk',
                                    'convertFormat' => false,
                                    'pluginOptions' => [
                                        'todayHighlight' => true,
                                        'autoclose' => true,
                                        'startDate' => date('Y-m-d H:i:s')
                                    ]
                                ]
                            )->label(false);
                        ?>
                    </div>

```

```

        <div class="mb-3">
            <span class="display-6 text-muted">
                -
            </span>
        </div>

        <div>
            <?=$form->field($courseTimeSlot, 'time_to')->widget(
                DateTimePicker::class,
                [
                    'language' => 'uk',
                    'convertFormat' => false,
                    'pluginOptions' => [
                        'todayHighlight' => true,
                        'autoclose' => true,
                        'startDate' => date('Y-m-d H:i:s')
                    ]
                ]
            )->label(false);
            ?>
        </div>
    </div>
</div>
</div>
</div>

    <div class="modal-footer justify-content-start">
        <button class="btn ms-auto" type="button" data-bs-
dismiss="modal">Закрити</button>

        <?=$Html::submitButton(
            'Створити',
            [
                'class' => 'btn bg-purple text-white',
                'name' => 'creation-button'
            ]
        ) ?>
    </div>
    <?php ActiveForm::end(); ?>
</div>
</div>

```

frontend/views/course-time-slot/delete.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTimeSlot \common\models\CourseTimeSlot */

?>

<div class="modal-dialog modal-dialog-centered modal-lg">
    <div class="modal-content border-0">
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Видалення планування
            </h5>

            <button type="button" class="btn-close btn-close-white" data-bs-
dismiss="modal" aria-label="Close"></button>
        </div>

        <div class="modal-body">
            <div class="text-center fs-5">
                Ви впевнені, що хочете видалити планування для "<i><?=$
courseTimeSlot->course->name ?></i>"?
            </div>
        </div>

        <div class="modal-footer justify-content-center">
            <button class="btn"

```

```

        type="button"
        data-bs-dismiss="modal"
    >
        Hi
    </button>

    <button class="btn bg-purple text-white"
        type="button"
        onclick="handleEventCompleteDeletion(event, <?= $courseTimeSlot->id ?>)"
    >
        Так
    </button>
</div>
</div>
</div>

```

frontend/views/course-time-slot/update.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTimeSlot CourseTimeSlot */
/** @var $startDate DateTime */
/** @var $endDate DateTime */

use common\models\CourseTimeSlot;
use kartik\datetime\DateTimePicker;
use yii\bootstrap5\ActiveForm;
use yii\bootstrap5\Html;

$courseTimeSlot->time_from = $startDate;
$courseTimeSlot->time_to = $endDate;

?>

<div class="modal-dialog modal-lg">
    <div class="modal-content border-0">
        <?php $form = ActiveForm::begin([
            'id' => 'update-course-time-slot-form',
            'action' => '/course-time-slot/update',
            'options' => [
                'novalidate' => true
            ]
        ]); ?>
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Редагування планування
            </h5>

            <button class="btn-close btn-close-white" type="button" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>

        <div class="modal-body">
            <div class="d-flex flex-column gap-2">
                <?= $form->field($courseTimeSlot, 'id', ['options' => ['class' => 'mb-0']])->hiddenInput()->label(false) ?>

                <div class="course-input-group">
                    <div class="text-muted fs-6">
                        <i class="bi bi-journal-medical"></i>
                        <span>Курс</span>
                    </div>

                    <div class="d-grid mt-1">
                        <?= $courseTimeSlot->course->name ?>
                    </div>
                </div>
            </div>

```

```

<div class="duration-input-group">
  <div class="text-muted fs-6">
    <i class="bi bi-clock"></i>
    <span>Час планування</span>
  </div>

  <div class="d-flex align-items-center gap-2 mt-1">
    <div>
      <?=$form->field($courseTimeSlot, 'time_from')->widget(
        DateTimePicker::class,
        [
          'language' => 'uk',
          'convertFormat' => false,
          'pluginOptions' => [
            'todayHighlight' => true,
            'autoclose' => true,
            'startDate' => date('Y-m-d H:i:s')
          ]
        ]
      )->label(false);
    <?>
  </div>

  <div class="mb-3">
    <span class="display-6 text-muted">
      -
    </span>
  </div>

  <div>
    <?=$form->field($courseTimeSlot, 'time_to')->widget(
      DateTimePicker::class,
      [
        'language' => 'uk',
        'convertFormat' => false,
        'pluginOptions' => [
          'todayHighlight' => true,
          'autoclose' => true,
          'startDate' => date('Y-m-d H:i:s')
        ]
      ]
    )->label(false);
    <?>
  </div>
</div>
</div>
</div>
</div>

<div class="modal-footer justify-content-start">
  <button class="btn ms-auto"
    type="button"
    data-bs-dismiss="modal"
    onclick="handleEventModificationRejection()"
  >
    Закрити
  </button>

  <?=$Html::submitButton(
    'Зберегти',
    [
      'class' => 'btn bg-purple text-white',
      'name' => 'update-button'
    ]
  ) ?>
</div>
<?php ActiveForm::end(); ?>
</div>
</div>

```

frontend/views/course-time-slot/view.php

```

<?php

/** @var \yii\web\View $this */
/** @var $courseTimeSlot \common\models\CourseTimeSlot */

use common\controllers\DateTimeUtils;

?>

<div class="modal-dialog">
    <div class="modal-content border-0">
        <div class="modal-header bg-purple">
            <h5 class="modal-title">
                Перегляд планування
            </h5>

            <button class="btn-close btn-close-white" type="button" data-bs-
dismiss="modal" aria-label="Close"></button>
        </div>

        <div class="modal-body">
            <div class="d-flex flex-column gap-2">
                <div>
                    <div class="text-muted fs-6">
                        <i class="bi bi-card-text"></i>
                        <span>Курс</span>
                    </div>

                    <div>
                        <?= $courseTimeSlot->course->name ?>
                    </div>
                </div>

                <div class="duration">
                    <div class="text-muted fs-6">
                        <i class="bi bi-clock"></i>
                        <span>Час проведення</span>
                    </div>

                    <div>
                        <?= DateTimeUtils::formatFullDateTime(new
DateTime($courseTimeSlot->time_from)) ?>
                        &nbsp;-&nbsp;
                        <?= DateTimeUtils::formatFullDateTime(new
DateTime($courseTimeSlot->time_to)) ?>
                    </div>
                </div>
            </div>

            <div class="modal-footer justify-content-start">
                <button class="btn ms-auto" type="button" data-bs-
dismiss="modal">Закрити</button>
            </div>
        </div>
    </div>
</div>

```

frontend/views/layouts/_header.php

```

<?php

use common\models\Notification;
use common\controllers\DateTimeUtils;
use common\controllers\UserUtils;
use yii\bootstrap5\BootstrapAsset;
use yii\helpers\Url;

/** @var \yii\web\View $this */

```

```

$this->registerCssFile('@web/css/header.css', [
    'depends' => [BootstrapAsset::class]
]);

$currentUser = Yii::$app->user->identity;

$notifications = Notification::find()
    ->where(['user_id' => $currentUser->id])
    ->orderBy('id DESC')
    ->limit(5)
    ->all();

?>

<nav class="navbar navbar-expand-md navbar-light bg-purple-gradient shadow-lg p-0">
    <div class="container-fluid p-0 pe-0 pe-md-3">

        <!--? Logo -->
        <div class="navbar-brand col-12 col-md-auto p-0" >
            <div class="position-relative">
                <div class="d-flex align-items-center justify-content-between logo-
container px-4 py-2">
                    

                    <div>
                        <button class="navbar-toggler"
                            type="button"
                            data-bs-toggle="collapse"
                            data-bs-target="#navbar-toogler"
                            aria-controls="navbar-toogler"
                            aria-expanded="false"
                            aria-label="Toggle navigation"
                        >
                            <span class="navbar-toggler-icon"></span>
                        </button>
                    </div>
                </div>

                <div class="logo-container-border"></div>
            </div>
        </div>

        <div class="collapse navbar-collapse align-self-end " id="navbar-toogler">
            <!--? Navigation Tabs -->
            <div class="col-12 col-md-auto align-self-end ms-md-auto me-md-auto p-0">
                <?php if (UserUtils::currentUserIsStudent()): ?>
                    <?= $this->render('student_navigation') ?>
                <?php elseif (UserUtils::currentUserIsTeacher()): ?>
                    <?= $this->render('teacher_navigation') ?>
                <?php endif; ?>
            </div>

            <!--? User Avatar -->
            <div class="col-12 col-md-auto">
                <div class="d-flex align-items-center justify-content-center pb-2 pb-
md-0">
                    <div class="align-self-end me-0 me-md-2">
                        <div class="dropdown">
                            <button class="btn text-white shadow-none p-0"
                                type="button"
                                id="notifications-dropdown-button"
                                data-bs-toggle="dropdown"
                                aria-expanded="false"
                            >
                                <i class="bi bi-bell notification-bell"></i>

                                <?php if (!empty($notifications)): ?>

```

```

        <span class="position-absolute new-alerts-circle
translate-middle p-1 bg-white border border-2 border-light rounded-circle">
            <span class="visually-hidden">New
alerts</span>

        </span>
        <?php endif; ?>
    </button>

    <ul class="dropdown-menu dropdown-menu-end"
        aria-labelledby="notifications-dropdown-button"
    >
        <li>
            <?php if (!empty($notifications)): ?>
                <?php foreach ($notifications as
$notification):
                    $notificationDate = new DateTime();
                    $notificationDate-
>setTimestamp($notification->created_at);
                    ?>
                    <div class="dropdown-item dropdown-
selectable-option">
                        <div class="d-flex flex-column">
                            <div class="notification-content">
                                <?= $notification->content ?>
                            </div>
                            <div class="notification-date">
                                <small>
                                    <?=
DateTimeUtils::formatFullDateTime($notificationDate) ?>
                                </small>
                            </div>
                        </div>
                    </div>
                <?php endforeach; ?>
            <?php else: ?>
                <div class="dropdown-header">
                    Сповіщень немає
                </div>
            <?php endif; ?>
        </li>
    </ul>
</div>
</div>
</div>

<div>
    <div class="dropdown">
        <button class="btn text-white shadow-none p-0"
            type="button"
            id="avatar-dropdown-button"
            data-bs-toggle="dropdown"
            aria-expanded="false"
        >
            <i class="bi bi-person-circle user-avatar"></i>
        </button>
        <ul class="dropdown-menu dropdown-menu-end" aria-
labelledby="avatar-dropdown-button">
            <li class="border-bottom text-center fs-6 pb-2">
                <?= $currentUser->getLinkedName('text-muted') ?>
            </li>
            <li>
                <a class="dropdown-item dropdown-selectable-
option"
                    href="<?= Url::to(['site/logout']) ?>"
                    data-method="post"
                >
                    <div class="d-flex align-items-center justify-
content-center">

```



```

?>

<table class="table table-bordered table-sm table-responsive-sm">
    <thead>
        <tr>
            <th scope="col" class="table-light text-center">
                Назва курсу
            </th>

            <?php
                $currentPeriod = new DateTime($startDate->format('Y-m-d'));

                while ($currentPeriod <= $endDate):
                    ?>
                        <td class="table-light day-header-cell">
                            <?= $currentPeriod->format('d') ?>
                            <br>
                            <?= DateTimeUtils::$WEEK_DAY_TRANSLATION[$currentPeriod->format('D')]
?>
                        </td>
                        <?php
                            $currentPeriod->add(new DateInterval('P1D'));
                        endwhile;
                    ?>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($assignedCourses as $course): ?>
                    <tr>
                        <td class="table-cell text-center">
                            <?= $course->name ?>
                        </td>

                        <?php
                            $currentPeriod = new DateTime($startDate->format('Y-m-d'));

                            while ($currentPeriod <= $endDate):
                                ?>
                                    <td class="table-cell mark-cell">
                                        <?php
                                            $courseKey = $course->id;
                                            $dateKey = $currentPeriod->format('Y-m-d');
                                            $relatedStudentMark =
$courseToDateStudentMarkRelation[$courseKey][$dateKey];

                                            if ($relatedStudentMark != null):
                                                ?>
                                                    <span>
                                                        <?= $relatedStudentMark->mark ?>
                                                    </span>
                                                <?php else: ?>
                                                    <span></span>
                                                <?php endif; ?>
                                        </td>
                                        <?php
                                            $currentPeriod->add(new DateInterval('P1D'));
                                        endwhile;
                                    ?>
                                </tr>
                            <?php endforeach; ?>
                        </tbody>
                    </table>

```

frontend/views/marks/_teacher_view.php

```

<?php

/** @var $this \yii\web\View */
/** @var $assignedCourses \common\models\Course[] */

```

```

/** @var $appliedCourse \common\models\Course[] */
/** @var $studentMarks \common\models\StudentMark[] */
/** @var $startDate DateTime */
/** @var $endDate DateTime */

use common\controllers\DateTimeUtils;

if (empty($assignedCourses)) {
    return;
}

$studentToDateStudentMarkRelation = array();

foreach ($studentMarks as $studentMark) {
    $courseId = $studentMark->course_id;
    $studentId = $studentMark->student_id;
    $relatedDate = new DateTime($studentMark->related_date);

    if ($studentToDateStudentMarkRelation[$courseId] === null) {
        $studentToDateStudentMarkRelation[$courseId] = array();
    }

    if ($studentToDateStudentMarkRelation[$courseId][$studentId] === null) {
        $studentToDateStudentMarkRelation[$courseId][$studentId] = array();
    }

    $studentToDateStudentMarkRelation[$courseId][$studentId][$relatedDate->format('Y-m-d')] = $studentMark;
}

$courseAssignees = [];

foreach ($appliedCourse->courseStudentAssignments as $courseStudentAssignment) {
    $courseAssignees[] = $courseStudentAssignment->student;
}

?>

<table class="table table-bordered table-sm table-responsive-sm">
    <thead>
        <tr>
            <th scope="col" class="table-light text-center">
                <nav class="navbar navbar-expand">
                    <div class="collapse navbar-collapse justify-content-center"
id="courses-dropdown">
                        <ul class="navbar-nav">
                            <li class="nav-item dropdown">
                                <span class="dropdown-toggle text-dark"
id="applied-course"
data-id="<?= $appliedCourse->id ?>"
role="button"
data-bs-toggle="dropdown"
aria-expanded="false"
                                >
                                    <?= $appliedCourse->name ?>
                                </span>
                                <ul class="dropdown-menu" aria-labelledby="course-
dropdown-menu">
                                    <?php foreach($assignedCourses as $course):
                                        if ($course->id === $appliedCourse->id) {
                                            continue;
                                        }
                                    ?>
                                    <li>
                                        <span class="dropdown-item dropdown-
selectable-option"
                                            onclick="handleCourseApplying(<?=
$course->id ?>) "
                                        >

```

```

                <?= $course->name ?>
            </span>
        </li>
    <?php endforeach; ?>
</ul>
</li>
</ul>
</div>
</nav>
</th>

<?php
    $currentPeriod = new DateTime($startDate->format('Y-m-d'));

    while ($currentPeriod <= $endDate):
    ?>
        <td class="table-light day-header-cell cell-fit">
            <?= $currentPeriod->format('d') ?>
            <br>
            <?= DateTimeUtils::$WEEK_DAY_TRANSLATION[$currentPeriod->format('D')]
?>
        </td>
    <?php
        $currentPeriod->add(new DateInterval('P1D'));
    endwhile;
    ?>
</tr>
</thead>
<tbody>
<?php foreach ($courseAssignees as $assignee): ?>
    <tr>
        <td class="table-cell text-center">
            <?= $assignee->getName() ?>
        </td>

        <?php
            $currentPeriod = new DateTime($startDate->format('Y-m-d'));

            while ($currentPeriod <= $endDate):
                $dateKey = $currentPeriod->format('Y-m-d');
                $courseKey = $appliedCourse->id;
                $studentKey = $assignee->id;
                $relatedStudentMark =
$studentToDateStudentMarkRelation[$courseKey][$studentKey][$dateKey];

                $elementId = $studentKey . '_' . $dateKey;

                if ($relatedStudentMark !== null) {
                    $elementId = $elementId . '_' . $relatedStudentMark->id;
                }
            ?>
            <td class="table-cell mark-cell"
                data-id="<?= $elementId ?>"
                onclick="handleMarkEditing(event, '<?= $elementId ?>', <?=
$relatedStudentMark->mark ?>)"
            >
                <span data-id="<?= $elementId ?>">
                    <?= $relatedStudentMark->mark ?>
                </span>
            </td>
        <?php
            $currentPeriod->add(new DateInterval('P1D'));
        endwhile;
    ?>
    </tr>
<?php endforeach; ?>
</tbody>
</table>

```

frontend/views/marks/index.php

```

<?php

/** @var $this \yii\web\View */

use common\controllers\UserUtils;
use yii\bootstrap5\BootstrapAsset;
use yii\web\View;

$this->title = 'Оцінки';

$this->registerJs(
    'var userType = "' . Yii::$app->user->identity->getUserType() . '";',
    View::POS_HEAD
);

$this->registerJsFile(Yii::getAlias('@web') . '/js/marks-page.js', [
    'depends' => [yii\web\jQueryAsset::class]
]);

$this->registerCssFile('@web/css/marks.css', [
    'depends' => [BootstrapAsset::class]
]);

?>

<div class="h-100 pt-3">
    <div class="card d-flex flex-column shadow-lg h-100">
        <div class="container-fluid rounded-top bg-purple shadow-sm px-0 px-sm-5 px-md-5 px-lg-5 pt-3 pb-2">
            <div class="d-flex align-items-center justify-content-center gap-1 gap-md-3 position-relative">
                <i class="bi bi-caret-left-fill period-navigation-button fs-4"
                    onclick="handlePeriodButtonClick('previous') ">
                </i>

                <h1 class="display-6 text-center"
                    id="current-period-text">
                </h1>

                <i class="bi bi-caret-right-fill period-navigation-button fs-4"
                    onclick="handlePeriodButtonClick('next') ">
                </i>

                <?php if (UserUtils::currentUserIsTeacher()): ?>
                    <div class="position-absolute top-25 end-0 d-none"
                        id="marks-editing-actions">
                        >
                            <div class="d-flex gap-2">
                                <button class="btn btn-light"
                                    type="button"
                                    onclick="handleMarksEditCanceling() ">
                                >
                                    <div class="d-flex gap-2 justify-content-center">
                                        <i class="bi bi-x-lg"></i>
                                    </div>
                                </button>

                                <button class="btn btn-light"
                                    type="button"
                                    onclick="handleMarksSaving() ">
                                >
                                    <div class="d-flex gap-2 justify-content-center">
                                        <i class="bi bi-check"></i>
                                        <span>Зберегти</span>
                                    </div>
                                </button>
                            </div>
                        </div>
                    <?php endif; ?>
                </div>
            </div>
        </div>
    </div>

```



```

        <?php if (UserUtils::currentUserIsTeacher() || $user->id ===
Yii::$app->user->id): ?>
            <div class="phone fs-5">
                Телефон: <?= $user->phone ?>
            </div>
        <?php endif; ?>

        <div class="type fs-5">
            Тип: <?= $typeLabel[$user->getUserType()] ?>
        </div>

        <?php if ($user->getUserType() === User::STUDENT_USER_TYPE): ?>
            <div class="class fs-5">
                Клас: <?= $user->getStudentsGroup()->name ?>
            </div>
        <?php endif; ?>
    </div>
</div>
</div>
</div>
</div>

```

Додаток В
Роздатковий матеріал

**ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики**

Випускна кваліфікаційна робота бакалавра

на тему: Розробка web-додатку для автоматизації дистанційного навчання учнів
загальноосвітньої школи

зі спецчастиною: Розробка програмних модулів, інтерфейсу користувача та бази даних
засобами PHP, MySQL, JavaScript, HTML, CSS

Автор: ст. гр. КН-18 Коверсун Д.В.

Керівник: ст. викл. каф. ПМІ Павловський Є.В.

Рисунок В.1 – Титульний лист

Актуальність роботи, об'єкт та предмет дослідження

Дистанційне навчання представляє собою нову організацію освіти, що базується на використанні як традиційних методів отримання знань, так і нових інформаційних та телекомунікаційних технологій. Системи дистанційного навчання надають можливість навчати як школярів, студентів, так і людей, що працюють, без відриву від виробництва.

Актуальність даної роботи полягає у потребі в автоматизації дистанційного навчання для учнів загальноосвітньої школи.

Об'єктом дослідження є процеси розробки кросбраузерного web-додатку засобами PHP, MySQL, JavaScript, HTML, CSS.

Предметом дослідження є автоматизація дистанційного навчання учнів загальноосвітньої школи.

Рисунок В.2. – Актуальність роботи, об'єкт та предмет дослідження

Мета та основні задачі роботи

Мета даної кваліфікаційної роботи - проєктування та розробка web-додатку, що дозволяє автоматизувати дистанційне навчання для учнів загальноосвітньої школи.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- 1) Аналіз існуючих популярних додатків для дистанційного навчання.
- 2) Аналіз засобів розробки.
- 3) Проєктування та реалізація структури системи обраними засобами розробки.
- 4) Реалізація функціонала системи та зовнішнього відображення сайту.
- 5) Тестування розробленого додатку.

3

Рисунок В.3 – Мета та основні задачі

Аналіз існуючих аналогів

Google Classroom

Дана платформа поєднує у собі сервіси Google, що дозволяють організовувати онлайн навчання.

Можливості платформи:

- створення класів/курсів;
- запис учнів на курси;
- проведення тестування;
- створення завдань для учнів;
- наявність різних форм оцінювання;
- спілкування з учнями в режимі реального часу.



Google Classroom

4

Рисунок В.4 – Опис платформи Google Classroom

Аналіз існуючих аналогів

Moodle

Дана платформа є безкоштовною відкритою системою керування дистанційним навчанням, яка використовується в багатьох країнах світу.

Moodle має у своєму інструментарії:

- форми здачі завдань;
- дискусійні форуми;
- завантаження файлів;
- журнал оцінювання;
- обмін повідомленнями;
- календар подій;
- онлайн тестування;



5

Рисунок В.5 – Опис платформи Moodle

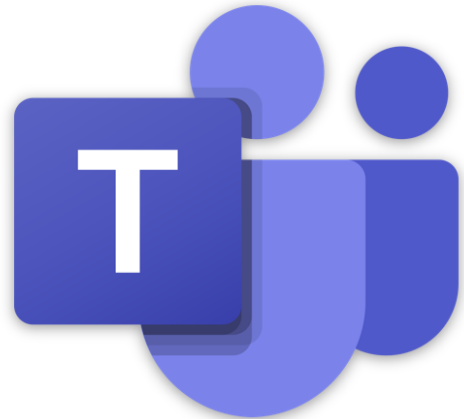
Аналіз існуючих аналогів

Microsoft Teams

Дана платформа надає безкоштовний онлайн-простір для спільної роботи, в якому вчителі та учні можуть обмінюватися нотатками, чатами, завданнями та навіть програмами.

Можливості платформи:

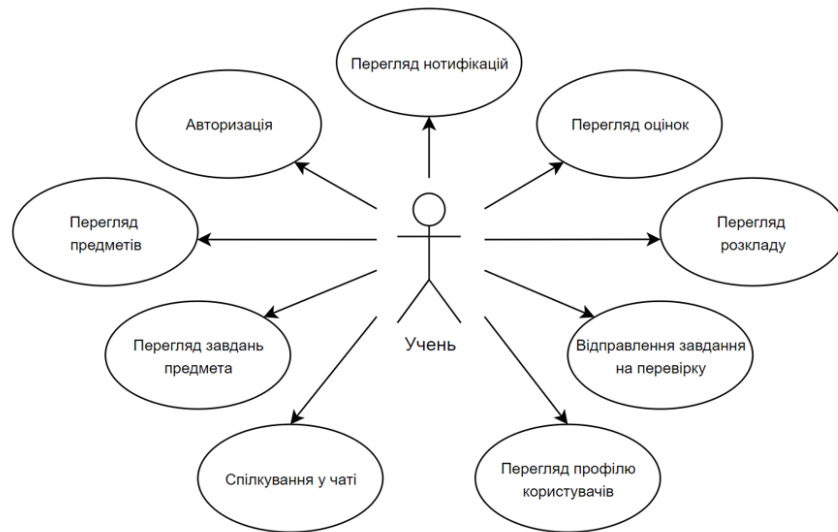
- створення груп/класів;
- наявність текстового та відеочату;
- планування дати проведення відео-конференції;
- наявність додатку для ПК, планшетів та смартфонів;
- інтеграція з Microsoft 365 додатками.



6

Рисунок В.6 – Опис платформи Microsoft Teams

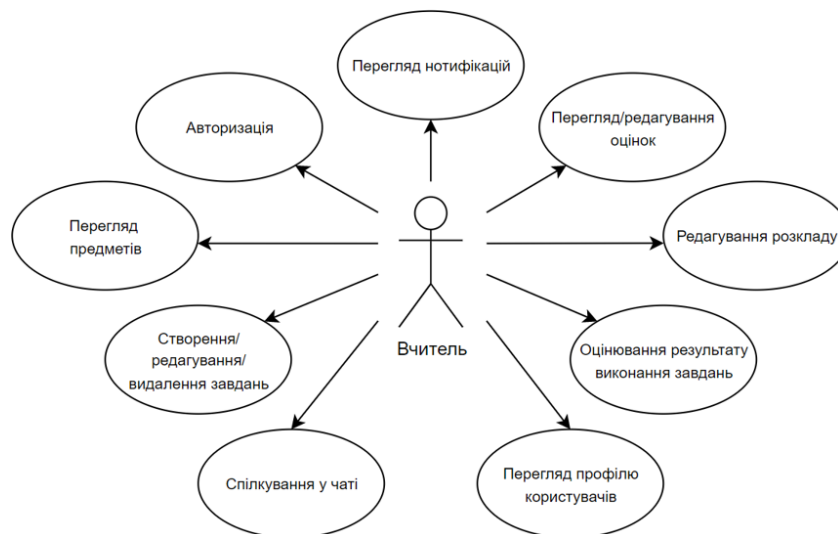
Діаграма варіантів використання системи



7

Рисунок В.7 – Діаграма використання для актора «Учень»

Діаграма варіантів використання системи



8

Рисунок В.8 – Діаграма використання для актора «Вчитель»

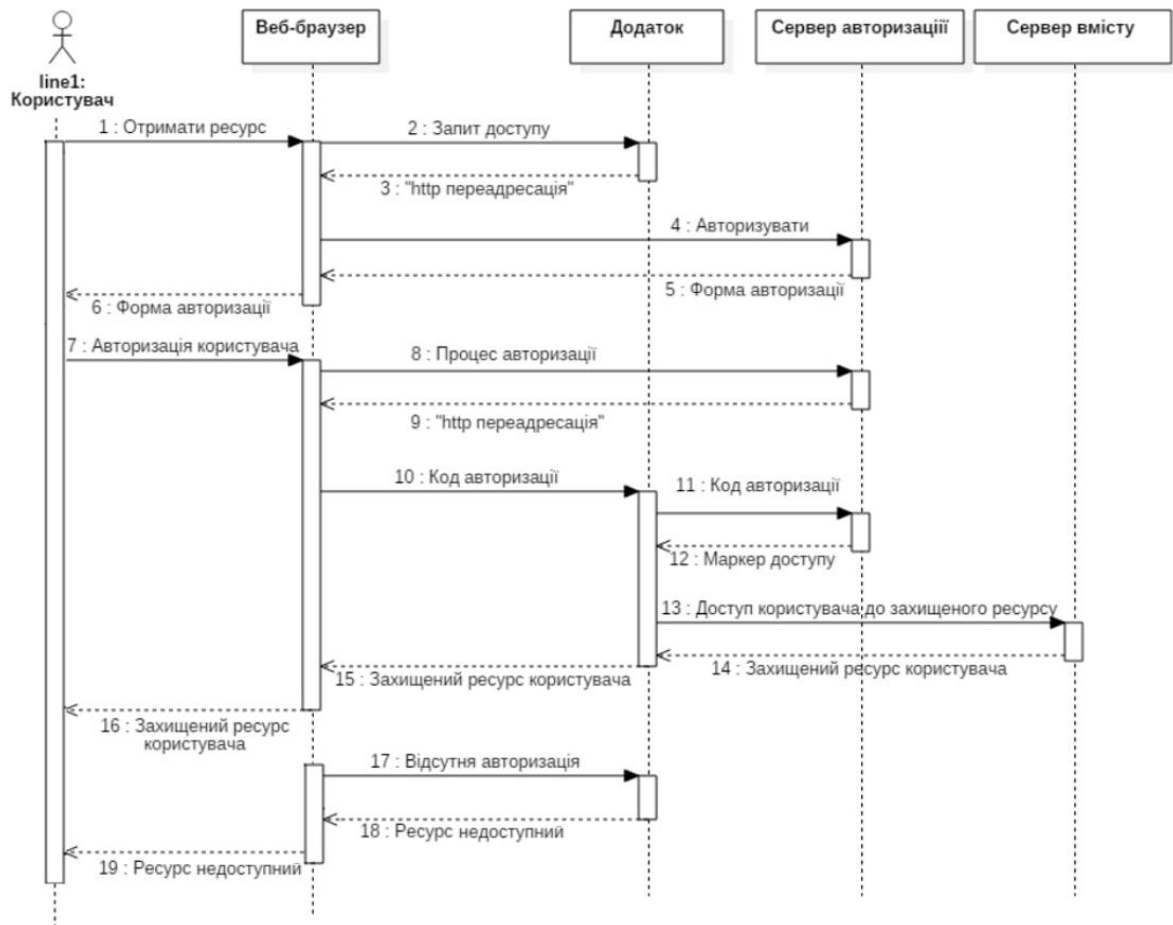


Рисунок В.9 – Діаграма послідовності

Діаграма розгортання системи

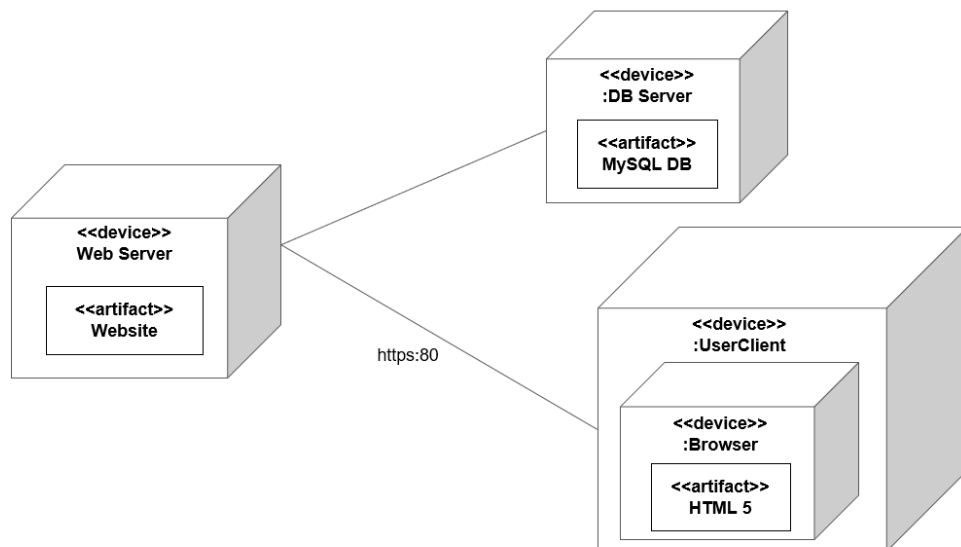
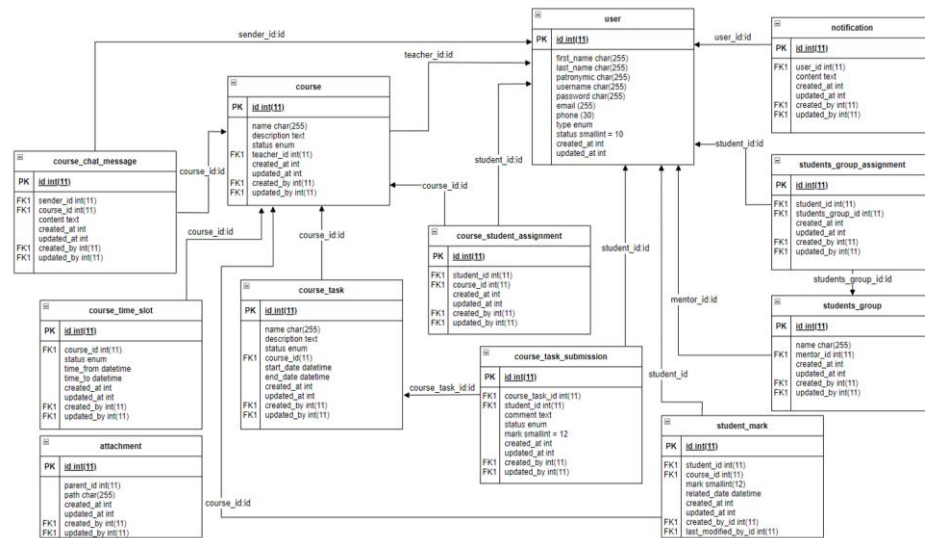


Рисунок В.10 – Діаграма розгортання системи

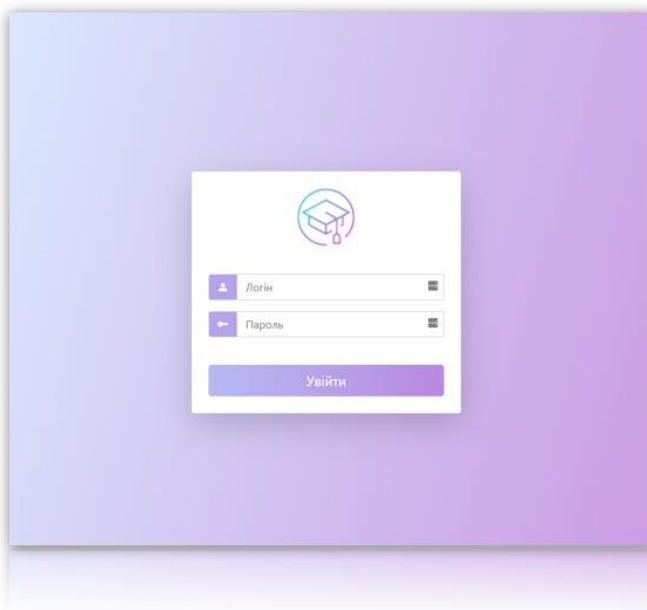
Схема бази даних



11

Рисунок В.11 – Схема бази даних

Огляд розробленої системи



Авторизація

Увійшовши до системи, користувач опиняється на сторінці авторизації.

12

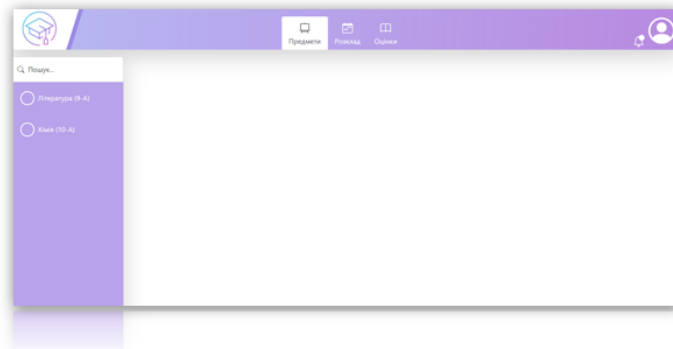
Рисунок В.12 – Огляд сторінки «Авторизація»

Огляд розробленої системи

Предмети користувача

Авторизувавшись до системи, користувач автоматично потрапляє на сторінку із призначеними предметами.

Відповідна навігаційна панель знаходиться зверху.



13

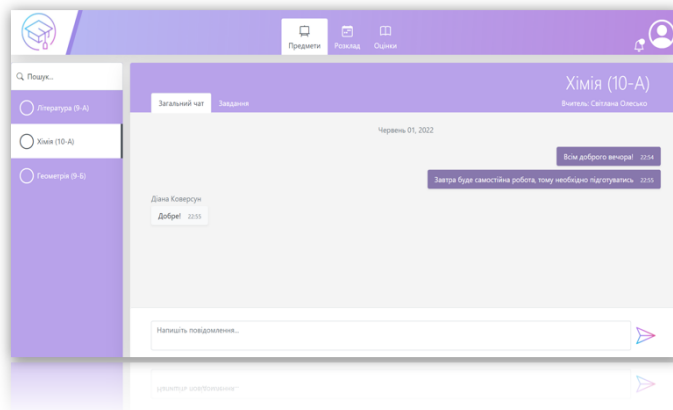
Рисунок В.13 – Огляд сторінки «Предмети користувача»

Огляд розробленої системи

Деталі предмета

Відкривши деталі предмета, користувач має можливість побачити:

- призначеного вчителя;
- загальний чат;
- список із всіма призначеними завданнями.



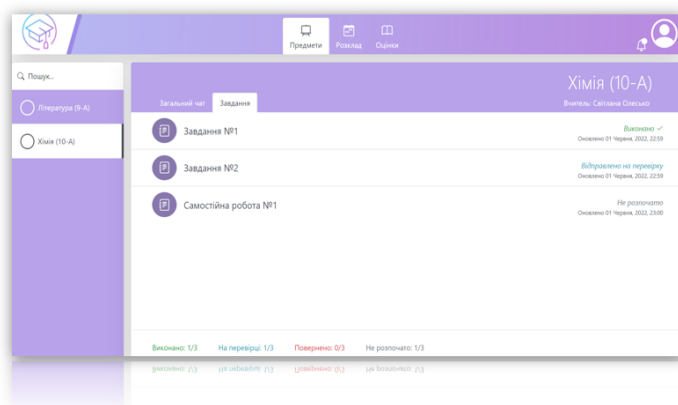
14

Рисунок В.14 – Огляд сторінки «Деталі предмета»

Огляд розробленої системи

Список завдань для учня

Якщо поточний користувач є учнем, то він побачить список завдань із відповідними статусами та статистикою їх виконання.



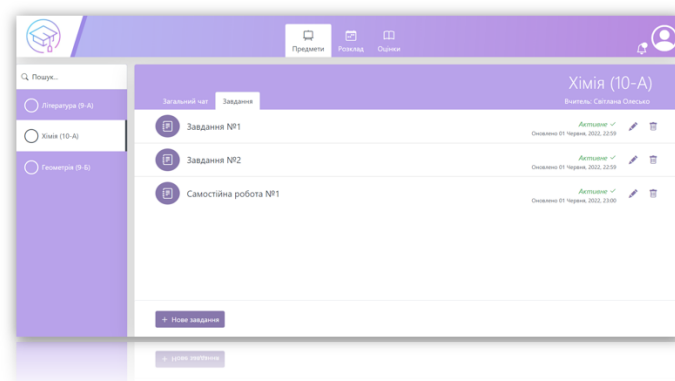
15

Рисунок В.15 – Огляд сторінки «Список завдань для учня»

Огляд розробленої системи

Список завдань для вчителя

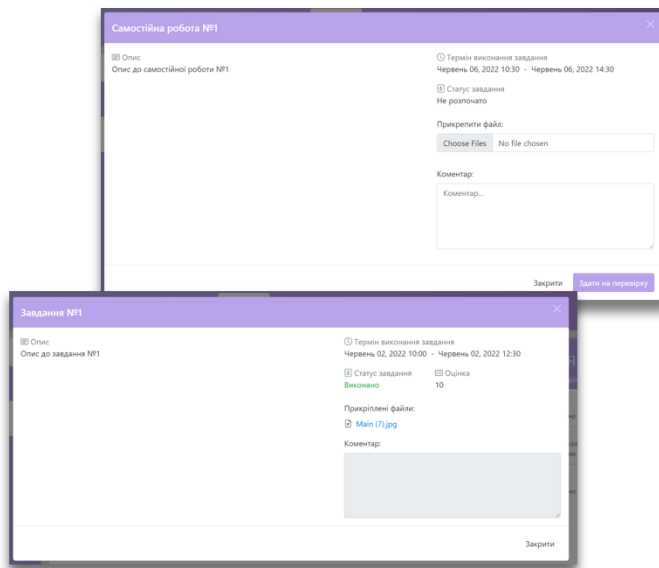
Якщо поточний користувач є вчителем, то він побачить список існуючих завдань із можливістю додавання нових завдань.



16

Рисунок В.16 – Огляд сторінки «Список завдань для вчителя»

Огляд розробленої системи



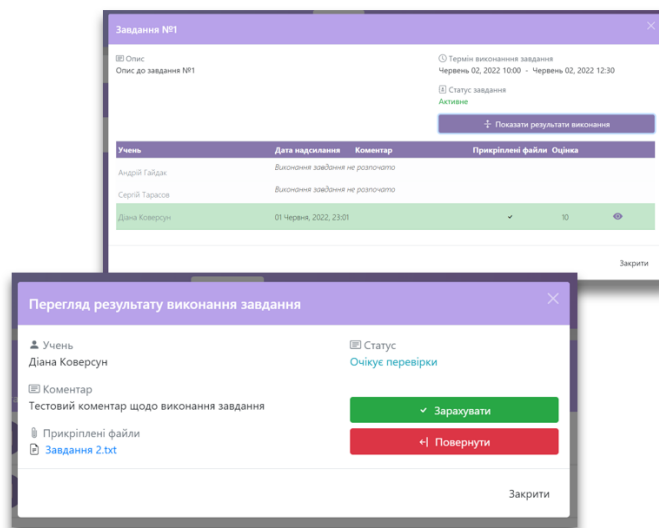
Деталі завдання

Користувачу надається можливість переглянути деталі завдання. Таким чином, учень може відправляти завдання на перевірку та переглядати статус його виконання.

17

Рисунок В.17 – Огляд сторінки «Деталі завдання»

Огляд розробленої системи



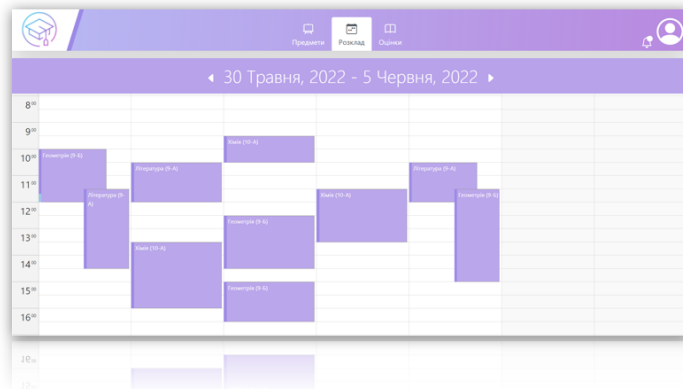
Оцінювання завдання

Відкривши деталі завдання, вчитель може побачити його відповідний опис та статус, а також таблицю з результатами виконання даного завдання.

18

Рисунок В.18 – Огляд сторінки «Оцінювання завдання»

Огляд розробленої системи



Планування проведення предметів

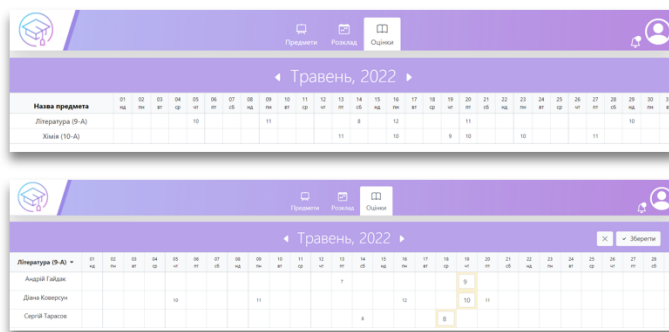
Перейшовши до «Розклад» у навігаційному меню, користувачу надається інформація щодо запланованих проведення предметів.

Вчителю також надається можливість редагувати та видаляти планування.

19

Рисунок В.19 – Огляд сторінки «Планування проведення предметів»

Огляд розробленої системи



Оцінки

Перейшовши до «Оцінки» у навігаційному меню, учню надається інформація щодо оцінок.

Вчителю, в свою чергу, надається можливість виставляти та редагувати оцінки учням з відповідних предметів.

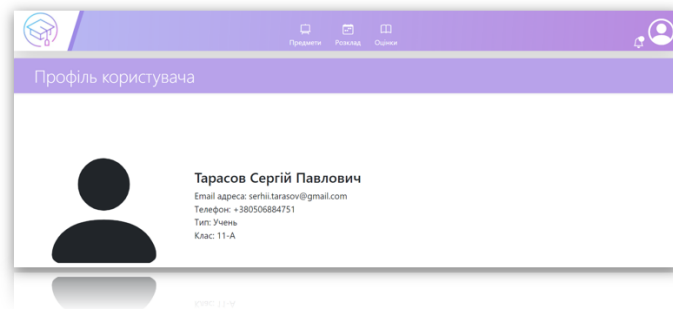
20

Рисунок В.20 – Огляд сторінки «Оцінки»

Огляд розробленої системи

Профіль користувача

Користувач також має можливість переглядати власний профіль та профілі інших користувачів.



21

Рисунок В.21 – Огляд профілю користувача

Огляд розробленої системи

Адміністрування

Для користувачів, що є адміністраторами в системі, також розроблений графічний інтерфейс для роботи із записами в системі.



Адміністратори можуть створювати, редагувати та видаляти записи учнів, вчителів, класів і предметів.

22

Рисунок В.22 – Вид системи від адміністратора

Отримані результати

- Проведено аналіз існуючих додатків для дистанційного навчання та досліджено основні засоби розробки для реалізації додатку.
- Розроблено специфікації системи та спроектовано й реалізовано структуру системи обраними засобами розробки.
- Реалізовано функціонал системи із відповідним графічним інтерфейсом.
- Проведене тестування розробленої системи.

23

Рисунок В.23 – Отримані результати

Висновки

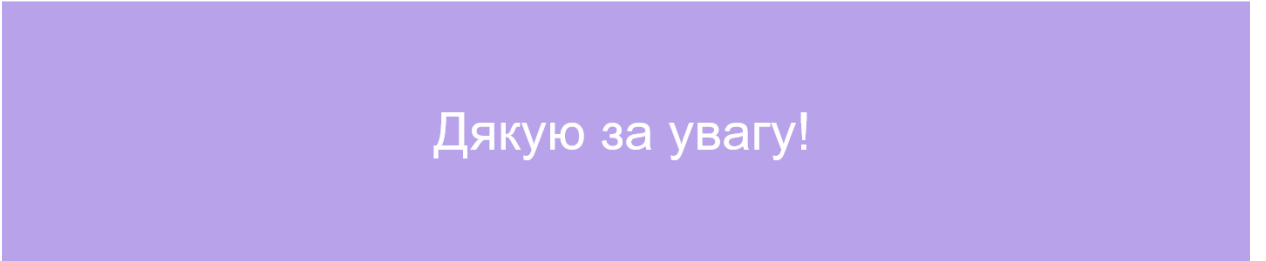
В даній кваліфікаційній роботі було досліджено загальну інформацію стосовно розробки web-систем та розроблено web-додаток для автоматизації дистанційного навчання учнів загальноосвітньої школи.

Для реалізації web-додатку були обрані наступні програмні засоби та технології:

- система керування базами даних MySQL;
- мови PHP, JavaScript, HTML, CSS;
- фреймворк Yii2;
- програмне середовище JetBrains PhpStorm;
- локальний сервер OpenServer.

24

Рисунок В.24 – Висновки



Дякую за увагу!

Рисунок В.25 – Завершення презентації