

ДВНЗ «Донецький національний технічний університет»

Факультет комп'ютерно-інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету)

Кафедра прикладної математики та інформатики

(повна назва кафедри)

«До захисту допущено»

в.о. завідувача кафедри

(підпис) Наталія МАСЛОВА
(Ім'я та ПРІЗВИЩЕ)

“ _____ ” _____ 20__ р.

Випускна кваліфікаційна робота

магістр

(освітній ступінь)

на тему «Система дистанційного навчання мові каскадних таблиць стилів (CSS)»

Виконав (-ла): студент (-ка) 2 курсу, групи ІПЗзм21
(шифр групи)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми Інженерія програмного забезпечення

Широков Р.І.

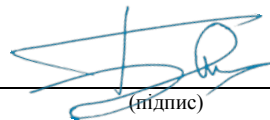
(прізвище та ініціали)



(підпис)

Керівник професор кафедри ПМІ, д.т.н., Євген БАШКОВ

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)



(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, Ім'я та ПРІЗВИЩЕ)

(підпис)

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

Студент



(підпис)

Покровськ – 2023 р.

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри

ЗАВДАННЯ
на дипломну (роботу) студента

1. Тема роботи: «Система дистанційного навчання мові каскадних таблиць стилів (CSS)»

Затверджені наказом від «17» лютого 2023 року № 70

2. Строк здачі студентом закінченого проекту (роботи, звіту) 11.05.2023 року

3. Вихідні дані до роботи результати переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1) Огляд предметної області.

2) Аналіз технологій та методів (розв'язання, створення, й т.п.) _____

3) Пояснення до обрання алгоритмів і методів для подальшої розробки (або методики дослідження)

4) Загальна схема та опис розробки.

5) Матеріали (тези) для публікації.

5. Перелік графічного матеріалу: робота містить 1 таблицю, 61 рисунок, 16 слайдів презентації та інший графічний матеріал, достатній для демонстрації сутності роботи

6. Консультанти розділів роботи

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

7. Дата видачі завдання 17.02.2023 року

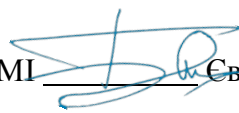
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури з предметної області	17.02.23 – 20.02.23	
2	Збір інформації і методичних відомостей щодо програмної реалізації системи дистанційного навчання	21.02.23 – 26.02.23	
3	Розробка структури програмного забезпечення, вибір даних для обробки та експериментів.	27.02.23 – 18.03.23	
4	Тестування системи	18.03.23 – 02.04.23	
5	Оформлення пояснювальної записки	03.04.2023 – 25.04.2023	
6	Нормоконтроль пояснювальної записки, її перевірка антиплагіатною системою	26.04.2023 – 11.05.2023	
7	Оформлення супутніх матеріалів (розробка графічного матеріалу, написання доповіді, тощо) та рецензування роботи	12.05.2023 – 24.05.2023	
8	Захист роботи	25.05.2023	

Студент:

гр. ПЗЗм21  Роман ШИРОКОВ

Керівник:

проф. кафедри ПМІ  Євген БАШКОВ

АНОТАЦІЯ

Широков Роман Ігорович. Система дистанційного навчання мові каскадних таблиць стилів (CSS) / Випускна кваліфікаційна робота здобувача освітнього ступеня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення». – ДВНЗ ДонНТУ, Покровськ, 2023.

Випускна кваліфікаційна робота магістра містить: 115 сторінок, 61 рисунок, 1 таблицю, 5 додатків, 27 посилань.

Роботу присвячено розробці поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

Мета роботи – розробка та створення поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

Об'єкт дослідження даної роботи – процеси дистанційного навчання в умовах сучасного, орієнтованого на використання комп'ютерно-інформаційних технології суспільства.

Предмет дослідження даної роботи – системи дистанційного навчання мові каскадних таблиць стилів (CSS) з використовуючи сучасні засоби анімації.

Задачами даної роботи є проведення порівняльного аналізу існуючих систем дистанційного навчання мові каскадних таблиць стилів (CSS), вивчення переваг та недоліків цих систем, пошук найкращих підходів побудови поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

Результатом роботи є поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS).

Наукова новизна полягає в комплексному використанні сучасних засобів візуалізації для супутньої анімації навчальних матеріалів.

Практичне значення полягає в поліпшені систем дистанційного навчання мові каскадних таблиць стилів (CSS).

Ключові слова: система навчання мові каскадних таблиць стилів (CSS), дослідження, аналіз, анімація, супутня анімація, переваги та недоліки.

Список публікацій здобувача:

1. Широков Р.І. Перспективи використання сучасних засобів анімації для поліпшення систем дистанційного навчання мові каскадних таблиць стилів (CSS) / / Моделювання та комп'ютерна графіка: збірник матеріалів Восьмої міжнародної науково-технічної конференції, 11-14 квітня 2023р. – м. Луцьк: ДВНЗ «ДонНТУ», 2023. – С. 79-84.

ABSTRACT

Roman Shirokov. System of distance learning of the language of cascading style sheets (CSS) / Graduation thesis for the degree of "master" in the speciality 121 "Software engineering". – SHEI DonNTU, Pokrovsk, 2023.

Master's final qualification thesis contains: 115 pages, 61 picture, 1 table, 5 applications, 27 references.

The work is devoted to the development of an improved distance learning system for the language of cascading style sheets (CSS).

The purpose of the work is to develop and create an improved system for remote learning of the Cascading Style Sheets (CSS) language.

The object of research of this work is the processes of distance learning in the conditions of a modern society oriented to the use of computer and information technologies.

The subject of research of this work is the system of remote learning of the language of cascading style sheets (CSS) using modern means of animation.

The tasks of this work are to carry out a comparative analysis of existing distance learning systems in the language of cascading style sheets (CSS), to study the advantages and disadvantages of these systems, to find the best approaches for building an improved system of distance learning in the language of cascading style sheets (CSS).

The result of the work is an improved distance learning system for the language of cascading style sheets (CSS).

The scientific novelty consists in the complex use of modern visualization tools for accompanying animation of educational materials.

The practical value lies in improved distance learning systems for the language of cascading style sheets (CSS).

Keywords: Cascading Style Sheets (CSS) language learning system, research, analysis, animation, accompanying animation, advantages and disadvantages.

List of publications of the acquirer:

1. Roman Shirokov Prospects for the use of modern animation tools to improve distance learning systems in the language of cascading style sheets (CSS) // Modelling and computer graphics, a collection of materials of the Eighth International Scientific and Technical Conference, April 11-14, 2023. - Lutsk: DonNTU Higher Secondary School, 2023. - pp. 79-84

ЗМІСТ

ВСТУП	11
РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО МОВУ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS)	12
1.1 Мова каскадних таблиць стилів CSS	12
1.2 Основні перевагами та недоліки використання CSS	14
1.2.1 Основні переваги використання CSS	14
1.2.2 Основні недоліки використання CSS	15
1.3 Доцільність вивчення CSS	16
1.4 Висновки за розділом	17
РОЗДІЛ 2 СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ МОВІ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS).....	19
2.1 Сучасне дистанційне навчання	19
2.2 Основні переваги та недоліки дистанційного навчання.....	20
2.2.1 Основні переваги дистанційного навчання	20
2.2.2 Основні недоліки дистанційного навчання	21
2.2.3 Особливості засвоєння інформації при дистанційному навчанні	22
2.3 Аналіз систем дистанційного навчання мові каскадних таблиць стилів CSS	23
2.3.1 Онлайн курси в YouTube	24
2.3.2 Курси freeCodeCamp	26
2.3.3 Курси Code Basics	28
2.3.4 Курси Skillbox.....	30
2.3.5 Курси GeekBrains	32
2.3.6 Курси MateAcademy.....	35
2.4 Рекомендації щодо розробки поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)	37
2.5 Висновки за розділом	39

РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПОЛІПШЕНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ МОВИ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS).....	40
3.1 Інструменти створення поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	40
3.1.1 Текстовий редактор вихідного коду Visual Studio Code	40
3.1.2 Мова гіпертекстової розмітки HTML	42
3.1.3 Мова каскадних таблиць стилів CSS.....	43
3.1.4 Мова програмування JavaScript	43
3.2 Структура та використання поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	45
3.2.1 Навігаційна панель поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	54
3.2.2 Підключення відео-лекцій до поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	56
3.2.3 Надання методичних матеріалів поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	58
3.2.4 Принцип роботи та функціонування «анімаційних вікон» поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	61
3.2.5 Навігація по відео-лекції за допомогою «анімаційних вікон» поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	69
3.2.6 Організація роботи та функціонування «вікон з поясненнями» поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	71
3.2.7. Принцип роботи та функціонування «вікон імітації web-сторінок» поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS).....	75

3.2.8. Взаємодія «вікон з поясненнями» та «вікон імітації web-сторінок» з відео-програвачем поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)	80
3.3. Переваги поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)	82
3.4 Висновки за розділом	83
ВИСНОВКИ.....	85
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
Додаток А Зміст html-документу однієї зі сторінок з відео-лекціями поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).....	90
Додаток Б Зміст CSS-документу однієї зі сторінок з відео-лекціями поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).....	95
Додаток В Зміст JavaScript-документу однієї зі сторінок з відео-лекціями поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).....	101
Додаток Г Презентація поліпшеної система дистанційного навчання мові каскадних таблиць стилів (CSS).....	104
Додаток Д Зауваження нормоконтролера	115

ВСТУП

В сучасному світі все частіше за знаннями та навиками звертаються до світової мережі Інтернет.

В час коли для успішного існування людям необхідно бути гнучкими, опановувати нові спеціальності та отримувати навички без суттєвої зміни налаштованого ритму життя, саме дистанційне навчання дає можливість отримати необхідні навички не відриваючись від основної роботи та не змушує при цьому переїздити до місця отримання освіти.

В теперішній час одним з найактуальніших напрямків здобуття освіти (особливо в якості альтернативної спеціальності, де дистанційне навчання більш ніж доцільне) є ІТ напрямки, зокрема створення web-сайтів.

Процес дистанційного навчання потребує сучасних систем візуалізації (анімації, наглядних матеріалів та ін.), але більшість сучасних систем дистанційного навчання пропонують навчання з відео-лекціями та онлайн-системи тестування отриманих знань (питання – відповідь).

В цій роботі розглянуто та проаналізовано системи дистанційного навчання мові каскадних таблиць стилів (CSS) – для накопичення рекомендацій та розробки поліпшеної системи дистанційного навчання, з опорою при навчанні саме на візуалізацію процесу виконання вводу стилів, тегів за допомогою сучасних систем анімації.

РОЗДІЛ 1

ЗАГАЛЬНІ ВІДОМОСТІ ПРО МОВУ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS)

1.1 Мова каскадних таблиць стилів CSS

CSS (Cascading Style Sheets) [1], каскадні таблиці стилів – це спеціальна мова (мова стилів), за допомогою якої описують вигляд документів (як і де відображати елементи веб-сторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі створені за допомогою HTML, XHTML та XML.

CSS використовується при створенні («верстці») веб-сторінок для задання кольорів, шрифтів, розміщення і інших аспектів представлення документа.

Після появи всесвітньої мережі Internet, мова html, деякий час була єдиним інструментом створення web-документів, та використовувалась тільки для створення структури вмісту web-сторінок, тобто розробники могли використовувати теги для розмітки web-сторінок такі, як `<h1>` та `<p>`, для того щоб вказати браузеру, що помічений відповідним тегом текст – це заголовок, або параграф. Далі почали з'являтися спеціальні теги, які створювали дизайн web-документів, а не його структуру. Такі теги могли додавати кольори (наприклад тег `<bgcolor>`), або змінити шрифт (наприклад тег `<font>`), задати тип шрифту (наприклад тег `<b>`), також у більшості інших тегів з'явилися такі властивості, як `color` чи `border`, таким чином оформлення web-сторінок відбувалося безпосередньо в середині html-документу. Подальше прагнення додати до web-сторінок все більшої кількості дизайнерських рішень, призвело до того, що деякі теги, такі як, `<table>`, стали все більше використовуватися для створення дизайну web-сторінок, а не для структурування тексту, а сам код html став перевантаженим та нечитабельним, крім того багато нових тегів

підтримувалися лише сучасними браузером, а іноді навіть тільки одним браузером. Враховуючи вищевикладені незручності використання html, як для створення структури, так і для візуального оформлення web-документів, стало ясно, що такий підхід до створення web-сторінок є малоефективним. Постало питання розділення структури web-документа та його стилістичного оформлення [2].

З появою CSS стало можливим розділення змісту (написаного за допомогою HTML або іншого мови розмітки) та візуальне оформлення (написаного на CSS) web-документа (на рисунку 1.1-1.2 показані типові документи HTML та CSS зі стилями). Це розділення збільшує доступність документа, надає більшу гнучкість та можливість керування його оформлення, зменшити складність та повторюваність в структурному змісті.

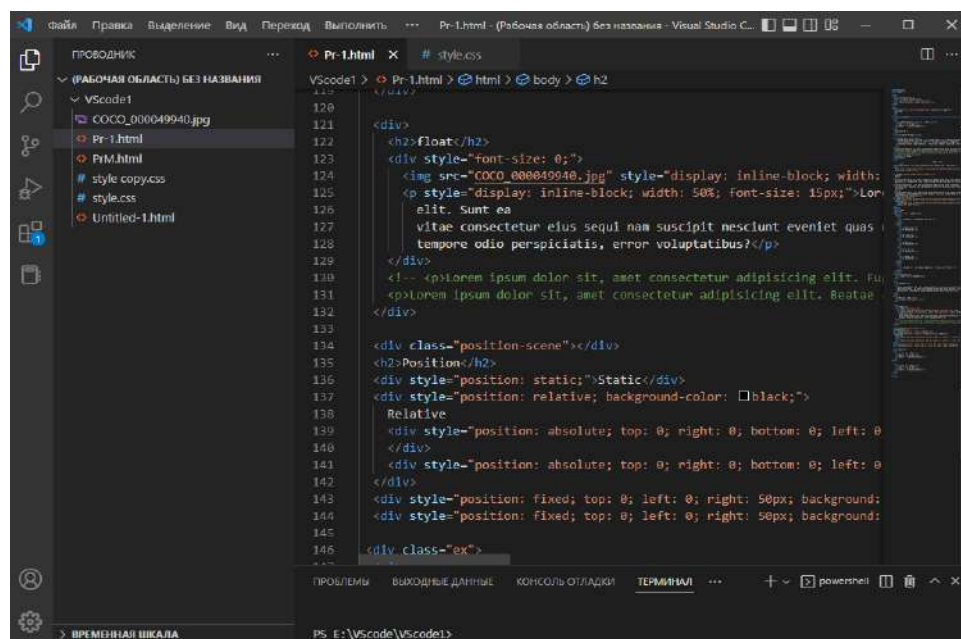


Рисунок 1.1 – Сторінка HTML зі стилями

Джерело: [Матеріал автору]

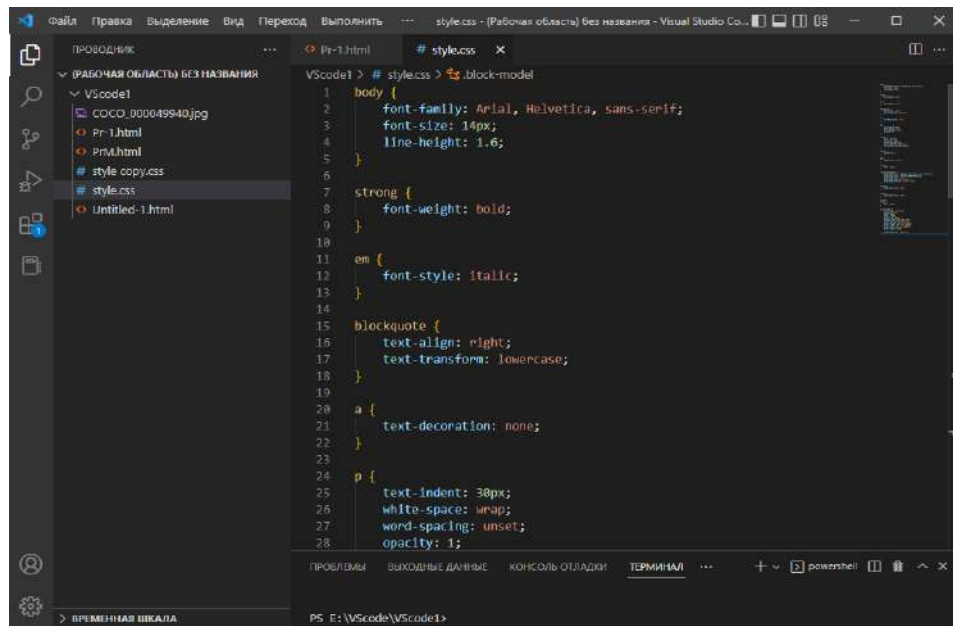


Рисунок 1.2 – Сторінка файлу CSS зі стилями

Джерело: [Матеріал автору]

1.2 Основні перевагами та недоліки використання CSS

1.2.1 Основні переваги використання CSS

До основних переваг CSS відносять:

- розділення оформлення та змісту сайту, тобто код HTML, буде вільним від елементів оформлення документів (колір, розмір шрифту та інші параметрів), таким чином веб-сторінка містить тільки теги логічного форматування, а вигляд елементів задається через стилі, такий підхід зокрема дає можливість паралельного проектування сайту та формування дизайну;

- єдине оформлення документів, тобто використання одноманітного оформлення заголовків, основного тексту та інших елементів створює спадкоємність між сторінками та полегшує роботу користувачів з сайтом та його сприйняття в цілому, з іншого боку використання стилів CSS істотно полегшує проектування дизайну;

- централізоване зберігання, тобто всі стилі проекту зберігаються в одному, або в декількох спеціальних файлах, посилання на які вказується в усіх документах сайту, такий підхід дозволяє коректувати стиль сайту в

одному місці, при цьому оформлення стилів документів змінюється автоматично на всіх сторінках, котрі пов'язані з вказаним файлом;

- розширені можливості, тобто на відміну від HTML стилі CSS, мають набагато більше можливостей, що до оформлення веб-сторінок (простими засобами можливо змінити колір фону елемента, його прозорість, орієнтованість, встановити пакети шрифтів та ін.);

- прискорення роботи, тобто при збереженні стилів в окремому файлі, він кешується та при повторному звертанні до нього вилучається з кеша браузера, за рахунок кешування та того, що стилі зберігаються в окремому файлі, зменшується код веб-сторінок та знижується час завантаження сторінки [2].

1.2.2 Основні недоліки використання CSS

До основних недоліків CSS відносять:

- часто виникає необхідність на практиці виправляти не тільки один CSS-файл, але і теги HTML та серверний код, що складним та ненаглядним способом пов'язані з селекторами CSS, це значно збільшує час редагування та тестування;

- складність опанування CSS полягає в його постійній взаємодії з HTML, методи підключення стилів, чітка структура проекту, потребує використання додаткових автоматизованих інструментів для автоматичного додавання префіксів замовника;

- структура файлів: CSS вимагає, щоб стилі були відокремлені від HTML-коду, що може зробити структуру файлів складнішою, особливо для великих проектів;

- строковість, CSS-код може стати досить довгим та складним для розуміння, особливо якщо міститься багато стилів для різних елементів веб-сторінки, це може зробити код більш вразливим до помилок та складним для зміни;

- сумісність з браузерами, різні браузери можуть трактувати CSS-код по-різному, що може призвести до відображення сторінок по-різному, це може вимагати додаткових зусиль, щоб забезпечити правильне відображення сторінок в різних браузерах;

- об'єм коду, CSS може займати багато місця, особливо якщо сторінки містять багато складних елементів, це може призвести до повільного завантаження сторінок та погіршення їх продуктивності;

- підтримка старих браузерів, деякі старі версії браузерів можуть не підтримувати деякі нові функції CSS, що може призвести до неправильного відображення сторінок або навіть їх непрацездатності [1, 2].

1.3 Доцільність вивчення CSS

Використання html в сукупності з CSS дозволяти робити з web-документом неймовірні речі, опанування CSS дозволить створювати деякий функціонал web-сторінок, набагато простіше ніж навіть з використанням JavaScript.

Опанування CSS дозволяє:

- відображати один і той же документ в різних стилях;
- створювати дизайн web-сторінок для різних гаджетів (комп'ютерів, смартфонів, планшетів, ноутбуків то що), сторінка буде відображатися по-різному в залежності від розміру екрану;
- зменшити час завантаження сайту (оскільки використання стилів в окремому CSS-файлі, дозволить браузеру завантажувати тільки структуру та дані які знаходяться на сторінці, а стилі цих даних завантажуються браузером тільки один раз, а потім кешуються);
- отримати додаткові можливості оформлення (наприклад прибрати підкреслення посилань, додати зміну стилю посилання при наведенні на нього курсору, та інші), які недоступні при використанні стилів в середині html-документу.

Крім того усі правила в CSS працюють по каскаду. Це дозволяє контролювати результат, за умови коли одному структурному елементу, надається декілька стилів.

Усі вищеперераховані можливості стають доступними тільки при використанні зв'язки html та CSS при створенні web-документів.

Вивчення CSS не тільки можливо, але й важливо для будь-якого веб-розробника чи дизайнера. CSS є фундаментальною частиною веб-розробки і дизайну, та відіграє вирішальну роль у створенні візуально привабливих, і зручних веб-сайтів.

CSS підтримується всіма сучасними веб-браузерами. CSS постійно розвивається, і регулярно доповнюється новими функціями та можливостями. Вивчення CSS — це не одноразове завдання, а постійний процес, щоб бути в курсі останніх тенденцій і технік. Однак, маючи хорошу основу CSS, ви зможете швидко адаптуватися до нових розробок у цій галузі.

Таким чином, вивчення CSS є цілком здійсненним і необхідним для всіх, хто цікавиться веб-розробкою чи дизайном. Завдяки великій кількості ресурсів, доступних в Інтернеті, вивчити CSS порівняно легко, і це може відкрити багато цікавих можливостей у цій галузі.

Виходячи з цього, незважаючи на деякі недоліки та складність опанування CSS, вивчення цього інструменту створення web-сторінок, є необхідним [1, 2].

1.4 Висновки за розділом

В цьому розділі були зібрана та проаналізована основна інформація про мову каскадних таблиць стилів CSS, проаналізований функціонал та передумови її появи.

В результаті аналізу були виявлені основні переваги та недоліки використання CSS, а також обґрунтована доцільність її опанування в сучасних умовах.

Наступний розділ присвячено, як дистанційному навчанню мові каскадних таблиць стилів, так і дистанційному навчанню в цілому, їх видам, перевагам та недолікам.

РОЗДІЛ 2

СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ МОВИ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS)

2.1 Сучасне дистанційне навчання

Дистанційне навчання [3] – освітній процес з використанням сукупності телекомунікаційних технологій, що мають ціллю надання можливості абітурієнтам засвоїти основний об'єм необхідної інформації з виключенням безпосереднього контакту абітурієнта та викладача під час процесу навчання (який може відбуватися як у синхронній так і у асинхронній формі). Процес може бути як самостійною формою навчання, так і доповненням до іншої більш традиційної форми навчання (очної, очно-заочної, заочної та екстернату), за необхідності надає можливість людині вивчити курс підготовки, перепідготовки або підвищення кваліфікації, без суттєвої зміни звичного способу життя.

Дистанційне навчання поділяється на синхронне та асинхронне.

Синхронне дистанційне навчання – це форма дистанційного навчання, при якій студенти взаємодіють з викладачем та іншими студентами в режимі реального часу, використовуючи онлайн-інструменти, такі як відео конференції, чати, форуми та інші.

У синхронному дистанційному навчанні викладач проводить вебінари та відео конференції, під час яких студенти можуть задавати питання та отримувати відповіді від викладача в режимі реального часу. Викладач також може використовувати інші інтерактивні інструменти, такі як групові проекти та дискусії, щоб забезпечити активну участь студентів у процесі навчання.

Однією з головних переваг синхронного дистанційного навчання є можливість навчання в реальному часі, що дозволяє студентам отримувати миттєву зворотну зв'язок від викладача та інших студентів. Це допомагає

студентам більш ефективно засвоювати матеріал та вирішувати проблеми під час навчання.

Однак, синхронне дистанційне навчання має свої недоліки, такі як технічні проблеми та обмежена гнучкість в плані графіку та місця проведення занять. Крім того, він може бути важким для студентів, які мають різний рівень знань та вмінь.

Асинхронне навчання – це форма дистанційного навчання, при якій студенти навчаються у своєму власному темпі та часі, використовуючи онлайн-інструменти, такі як відео-лекції, електронні підручники, відеозаписи лекцій та інші.

У асинхронному навчанні студенти отримують доступ до навчального матеріалу, який можуть вивчати у своєму власному темпі та часі. Вони можуть звертатися до викладачів через електронну пошту, чат або форуми, щоб отримати відповіді на свої запитання. Часто викладачі також відстежують прогрес студентів за допомогою електронних засобів та надають фідбек на основі результати тестів та інших завдань.

Однією з головних переваг асинхронного навчання є гнучкість, яка дозволяє студентам навчатися у зручний для них час та з будь-якого місця з доступом до Інтернету. Це дозволяє студентам зберігати баланс між роботою, сімейними обов'язками та навчанням. Крім того, студенти можуть зосередитися на конкретних темах, які вони вважають найбільш важливими та використовувати різні джерела для додаткової інформації.

Однак, асинхронне навчання також має свої недоліки, такі як відсутність миттєвої зворотної зв'язку та взаємодії з викладачем та іншими студентами. Це може призвести до відчуття ізоляції та складнощів у вирішенні проблем.

2.2 Основні переваги та недоліки дистанційного навчання

2.2.1 Основні переваги дистанційного навчання

До основних переваг дистанційного навчання відносять:

- собівартість дистанційного навчання нижче ніж традиційні форми навчання, так як може бути відсутня орендна плата за приміщення де буде відбуватися навчання;

- знижуються витрати на організацію власне занять, є можливість кожному викладачу одночасно займатися з більшою кількістю абітурієнтів та виключаються деякі інші фактори, що прямо чи опосередковано впливають на підсумкову вартість навчання;

- скорочується час на навчання (час на дорогу до навчального закладу то що);

- дистанційне навчання може носити індивідуальний характер, отже надає можливість більш ефективно налаштувати процес навчання, даючи можливість абітурієнту підібрати зручний час та темп навчання;

- підвищення якості навчання за рахунок сучасних засобів, об'ємних електронних бібліотек то що;

- така форма навчання зручна для людей з обмеженими фізичними можливостями;

- абітурієнт може в стислі терміни одночасно навчатися в декількох навчальних організаціях та за декількома напрямками;

- дає можливість постійно збільшувати власний рівень кваліфікації;

- активне використання зображень, тексту, звуку та відео посібників в учбовому матеріалі підвищує якість засвоєння нової інформації [3, 4].

2.2.2 Основні недоліки дистанційного навчання

До основних недоліків дистанційного навчання відносять:

- необхідність розбиратися в цифрових технологіях, тобто необхідність вчитися навчатися, для деяких абітурієнтів;

- нестача особистого спілкування, віддалене спілкування через месенджери та канали, не дає достатнього рівня контакту між викладачем та студентом;

- відсутність розмежування між робочим (навчальним) та вільним часом, необхідність чіткого виділення часу на навчання, працю, відпочинок [3, 4].

2.2.3 Особливості засвоєння інформації при дистанційному навчанні

Процес засвоєння інформації – це складний психологічний процес, який включає в себе отримання, розуміння та запам'ятовування інформації. Цей процес може бути різним для кожної людини і залежить від таких факторів, як індивідуальні особливості, рівень уваги та інтерес до теми, яку необхідно засвоїти.

Основні етапи процесу засвоєння інформації:

- отримання інформації, цей етап включає в себе отримання інформації з різних джерел, таких як книги, веб-сайти, відео та інші, важливо мати на увазі, що не всі джерела інформації є достовірними та корисними;
- розуміння інформації, цей етап передбачає зрозуміння інформації, яку необхідно засвоїти, для цього можна використовувати різні методи, такі як узагальнення, класифікацію, порівняння та інші;
- запам'ятовування інформації, цей етап передбачає збереження інформації в пам'яті для подальшого використання, для цього можна використовувати різні методи, такі як повторення, асоціацію, структурування та інші;
- використання інформації, цей етап передбачає використання засвоєної інформації у практичній діяльності, це може бути використання інформації у навчанні, роботі або інших сферах життя.

Якщо у людини процес засвоєння інформації на високому рівні, засвоєна інформація в його свідомості буде повною та коректною. В такому разі йому не доведеться часто повертатися до першоджерела для уточнення деяких даних, або поповнення інформації, якої бракує. При такому рівні засвоєння інформації людина запам'ятовує, сприймає, та розуміє все так як почув чи побачив.

Зовсім інша ситуація, коли навик засвоєння інформації низький. В цьому разі інформація буде сприйматися викривленою, тобто отримані данні будуть неповними. Залишаться проблеми для заповнення яких мозок людини може навіть вигадати щось, чого від самого початку не було. Людина буде неправильно розуміти те, що до нього хотіли донести та відповідно не зможе доцільно та ефективно використати отримані знання.

Люди, які обирають для навчання, дистанційне навчання, частіше за все крім здобуття нових навичок мають ще декілька повсякденних задач (основна робота, побутові питання, можливо освоєння декількох спеціальностей чи вмінь одночасно), тому за таких обставин рівень засвоєння матеріалу у таких здобувачів освіти, може перебувати не на високому рівні.

Для ефективного засвоєння інформації слід визивати емоції та створювати образи у студента, для реалізації цього процесу, як найкраще сприятимуть використання сучасних систем візуалізації навчального матеріалу (анімація тощо) [5].

2.3 Аналіз систем дистанційного навчання мові каскадних таблиць стилів CSS

В якості систем дистанційного навчання мові каскадних таблиць стилів CSS (далі СДН CSS), для подальшого аналізу обираємо декілька шкіл програмування (як платних так і безкоштовних), а також онлайн курси в YouTube в якості одного з найдоступніших інтерактивних засобів навчання за будь якими напрямками. Серед обраних СДН CSS проведемо аналіз по функціональним характеристикам.

Об'єкти дослідження були визначені за критерієм відповідності поставленій задачі: «Вибір систем дистанційного навчання мові каскадних таблиць стилів CSS для подальшого аналізу». Остаточний вибір об'єктів дослідження засновується на особистому досвіді автора роботи та рейтингу навчальних систем в мережі internet.

Серед безкоштовних курсів для подальшого аналізу обираємо freeCodeCamp та Code Basics.

Серед платних курсів обираємо Skillbox, MateAcademy та GeekBrains.

Також в якості об'єкту дослідження будуть виступати онлайн курси в YouTube.

2.3.1 Онлайн курси в YouTube

YouTube [6] – портал YouTube був заснований в 2005 році в межах розробки проектів епохи «Web 2.0». В цей період інтернет-користувачі отримують можливість створювати контент самостійно. Однак основною причиною росту популярності порталу, стала можливість дивитися відеоролики на будь якій сторінці Інтернет. Саме YouTube першим створив вікно онлайн-плеєра, який можна розташувати на будь якому сайті.

Згідно статистичним дослідженням, що були проведені в 2019 році в США 94% людей в віці від 18 до 35 років, є активними користувачами YouTube, а 36% з них щоденно використовують освітній контент на порталі (фільмами, освітніми каналами то що). Для прикладу на рисунку 2.1. показана сторінка пошуку YouTube за запитом: «курси CSS».

Виходячи з вищесказаного, YouTube сьогодні виступає в якості звичного та комфортного самоосвітнього середовища [6].

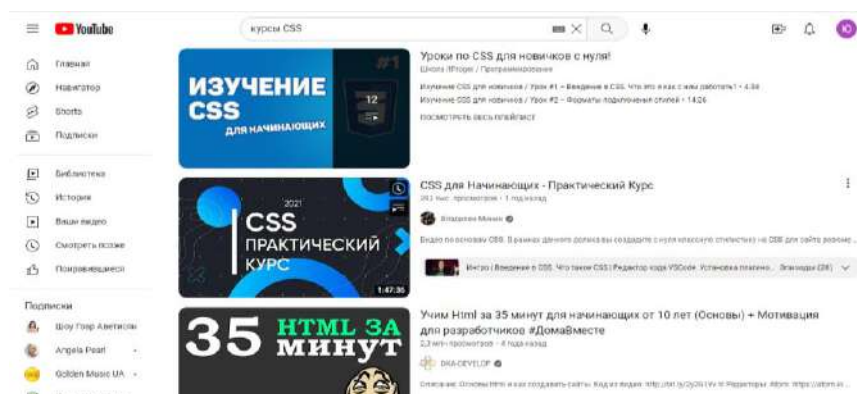


Рисунок 2.1 – Сторінка пошуку YouTube за запитом: «курси CSS»

Джерело: [6]

Основні переваги навчання в середовищі YouTube:

- широкий спектр відеоматеріалів від відео-лекцій, які створені спеціально для навчання мові каскадних таблиць стилів CSS до блогів, які створюють блогери, що мають досвід роботи та необхідні знання з CSS;
- доступність використання, для навчання достатньо гаджету, який може співпрацювати з мережею Інтернет (ПК, ноутбук, смартфон то що), та власне доступ до мережі Інтернет;
- обмеженість відеороликів за часом, більшість роликів розбито на частини прийнятної довжини [6].

Основні недоліки навчання в середовищі YouTube:

- неясність протоколів авторських прав, а також он-лайн матеріалів;
- відсутність можливості об'єктивно оцінити цінність та ефективність навчання конкретного каналу чи блогеру, до початку навчання;
- відсутність прямого контакту з викладачам, при виникненні питань;
- недостатня візуалізація дії наставника через відео-плеєр, що ускладнює сприйняття команд та тегів, які потрібно додати до файлів, щоб отримати той чи інший результат, тобто постійно потрібно зупиняти відео рядок на конкретному місці (такий підхід не завжди є ефективним, через можливості плеєра та особливості створення відеоматеріалів, дивись рисунок 2.2) [6, 7].

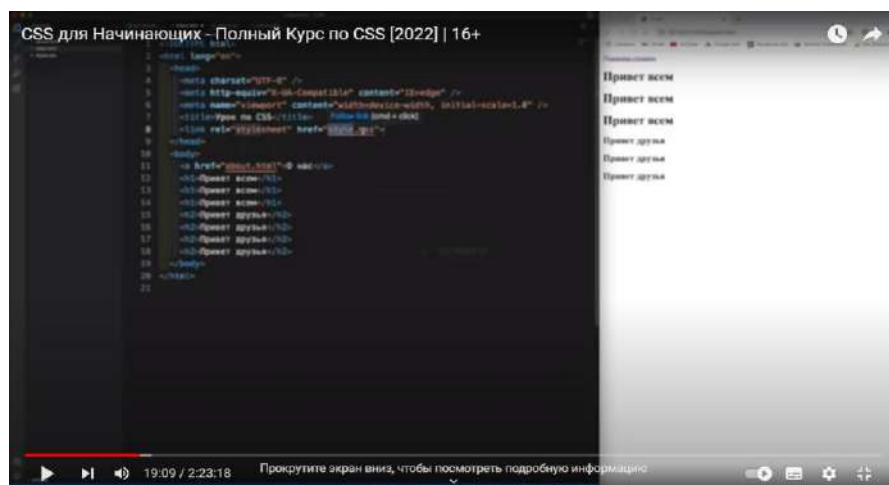


Рисунок 2.2 – Сторінка відео уроку на YouTube курсів CSS

Джерело: [6]

2.3.2 Курси freeCodeCamp

freeCodeCamp [8, 9, 10] (дивись рисунок 2.3) – це некомерційна організація, яка складається з інтерактивної навчальної веб-платформи, онлайн-форуму, чатів, онлайн-публікацій. Мета компанії, зробити навчання веб-розробці доступним для будь кого.

Починаючи з навчальних посібників, які знайомлять з HTML, CSS і JavaScript, студенти переходять до проектних завдань, які вони виконують самостійно або в парах. Після виконання всіх завдань проекту студенти співпрацюють з іншими некомерційними організаціями для створення веб-додатків, надаючи студентам практичний досвід розробки.



Рисунок 2.3 – Головна сторінка freeCodeCamp

Джерело: [8]

Після реєстрації студент навчається у зручному йому темпі. Після завершення блоку навчання можна зробити п'ять перевірочних завдань та отримати сертифікат.

Кожен урок містить пояснення теорії та практичне завдання, яке потрібно вирішити (приклад уроку на freeCodeCamp показано на рисунку 2.4). Якщо студент не може знайти рішення, йому доступні підказки.

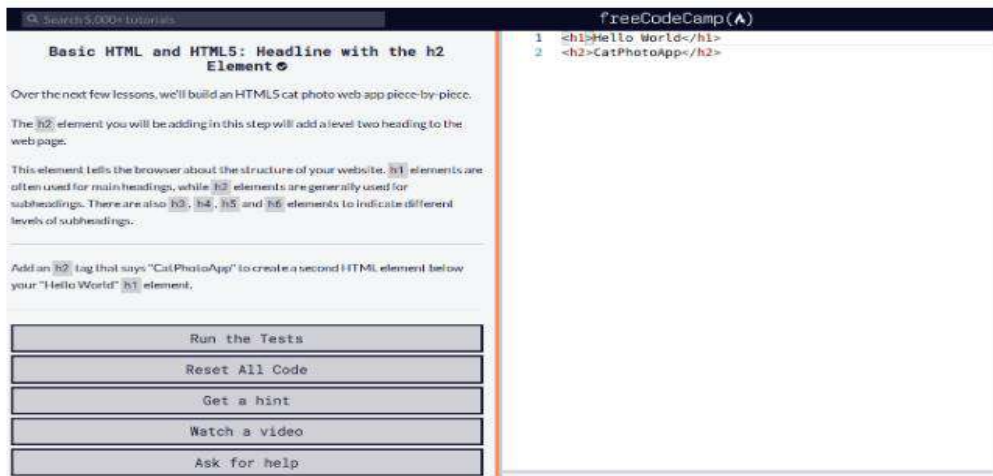


Рисунок 2.4 – Урок на freeCodeCamp

Джерело: [8]

Основні переваги навчання на freeCodeCamp:

- навчання повністю безкоштовне;
- сайт англomовний, це корисно для розробників;
- ви не зможете рухатися далі, поки не розберетеся з поточним завданням;
- freeCodeCamp привчає студентів до вирішення практичних вправ [8, 9, 10].

Основні недоліки навчання на freeCodeCamp:

- ресурс безкоштовний, тому на ньому немає підтримки викладачів та менторів, на запитання студентів відповідають користувачі зі статусом модератора та прості студенти;
- сайт виключно англomовний, що затрудняє процес засвоєння матеріалу;
- на freeCodeCamp дається мінімум теорії, вивчати програмування лише на цьому сайті без роботи з додатковими джерелами не вдасться;
- недостатня візуалізація навчального процесу, відсутнє використання засобів анімації та можливостей відео [8, 9, 10].

2.3.3 Курси Code Basics

Code Basics [11] (дивись рисунок 2.5) – це онлайн-платформа для навчання програмуванню. Вона пропонує різні курси з програмування на різних мовах, таких як Python, Java, JavaScript, PHP та інші (дивись рисунок 2.6). Курси розроблені для початківців, але можуть бути корисними для всіх, хто бажає поглибити свої знання з програмування.

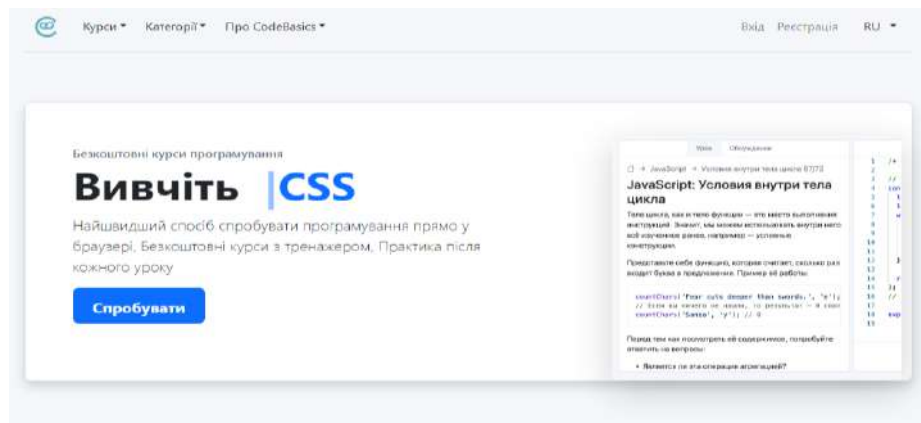


Рисунок 2.5 – Головна сторінка Code Basics

Джерело: [11]



Рисунок 2.6 – Сторінка Code Basics з курсами

Джерело: [11]

Курси на Code Basics включають в себе практичні завдання та тести, що допомагають студентам отримати практичні навички у програмуванні та

перевірити свій рівень знань (для прикладу на рисунку 2.7 показаний онлайн-урок на Code Basics). Крім того, платформа пропонує можливість комунікації з викладачами та іншими студентами, що допомагає зрозуміти складні концепції та отримати додаткову підтримку.

Code Basics також має безкоштовний доступ до деяких курсів, але для отримання повного доступу до всіх курсів потрібно придбати платний план. Курси на Code Basics добре підходять для тих, хто хоче вивчити програмування на власному темпі та за допомогою онлайн-ресурсів.

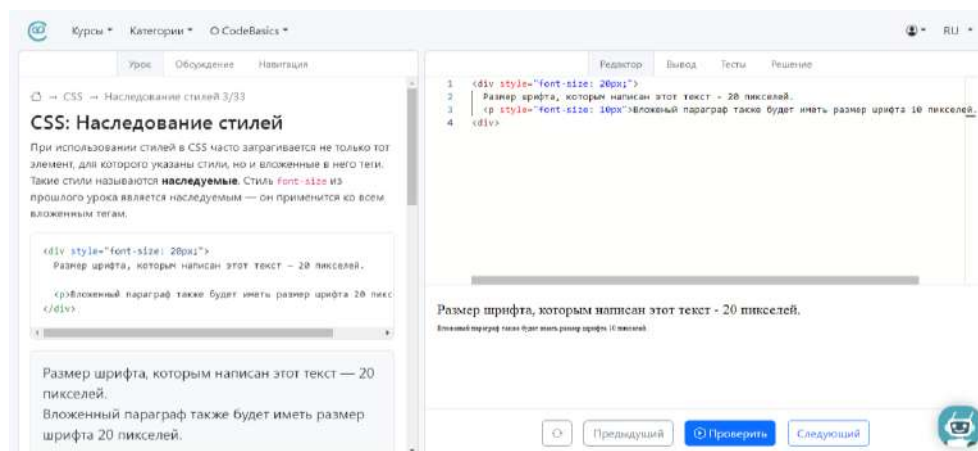


Рисунок 2.7 – Урок Code Basics

Джерело: [11]

Основні переваги навчання з Code Basics:

- можливість навчатися безкоштовно;
- структурованість теорії, перехід від легкого до важкого;
- направленість проекту на здобуття практичних навичок, обмежена кількість теоретичних матеріалів;
- непоганий рівень візуалізації, завдання надається у вигляді питання з прописаними тегамі подібними до завдання, що потрібно виконати;
- швидкий відклик на перевірку завдання, на відповідь витрачається 2-3 секунди;
- можливість перевірити себе подивившись рішення вчителя;

- рішення вчителя доступне не відразу, а протягом 20 хвилин, частіше за все, студент встигає вирішити завдання самостійно, до появи рішення вчителя [11, 12].

Основні недоліки навчання з Code Basics:

- обмежена теоретична база, необхідна допомога інших сайтів;
- складність виконання завдання полягає не в самому завданні, а в тому як воно поставлено, більше часу витрачаєш на те щоб зрозуміти, що від тебе потрібно, а не на виконання самого завдання;
- проект не надає сертифікацію;
- недостатня візуалізація навчання, повністю відсутня анімація процесу, матеріал надається у вигляді готового коду, слабка взаємодія між завданням та панеллю виконання задач [11, 12].

2.3.4 Курси Skillbox

Skillbox [13] (дивись рисунок 2.8) – інтернет-компанія, що працює в сфері онлайн-освіти, заснована в 2016 році, входить до освітнього холдингу Skillbox Holding Limited. Skillbox працює на базі власної платформи для онлайн-навчання. Сервіс пропонує асинхронний формат навчання, тобто заздалегідь записані відеоматеріали.

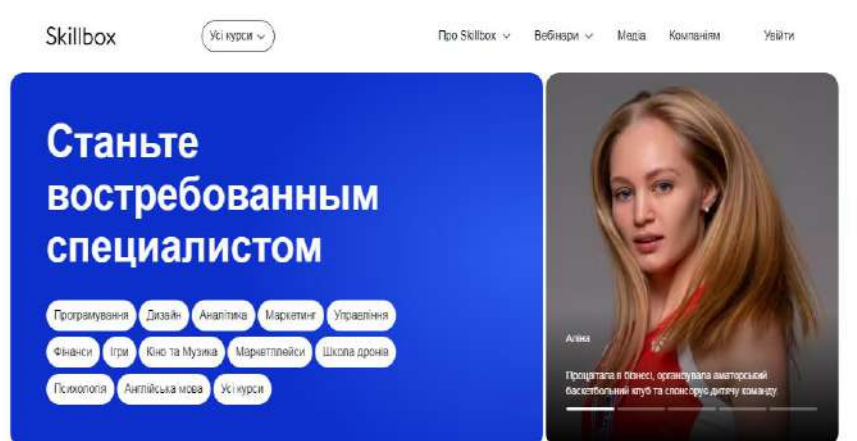


Рисунок 2.8 – Головна сторінка Skillbox

Джерело: [14]

Курси відбуваються у вигляді відео-лекцій (для прикладу на рисунку 2.9 показана відео-лекція на Skillbox) та вебінарів, після проходження кожного з етапів курсу студент виконує домашнє завдання, виконані завдання перевіряє куратор, або автор програми, після виправлення всіх недоліків та зауважень студент переходить до наступного завдання.

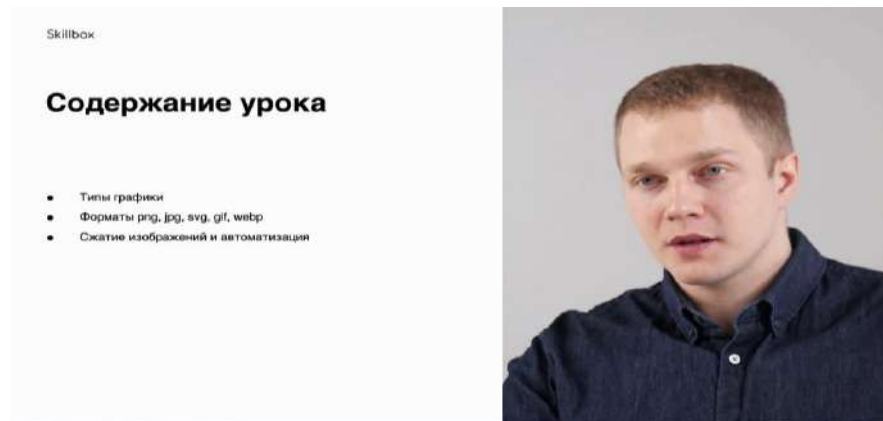


Рисунок 2.9 – Відео урок на Skillbox

Джерело: [14]

На деякому етапі навчання студент виконує практичне завдання від потенційного роботодавця.

Останній етап навчання виконання дипломної роботи по технічному завданню потенційного роботодавця [13, 14].

Основні переваги навчання зі Skillbox:

- строгий контроль викладачів за виконанням роботи, доки невиконано завдання Вас не допускають до наступних розділів занять;
- доволі зручний особистий кабінет;
- структурованість теорії, перехід від легкого до важкого[13, 14].

Основні недоліки навчання зі Skillbox:

- висока ціна навчання;
- система навчання заснована на викладенні відео лекційних матеріалів з демонстрацією стилів викладачем на його гаджеті, слухати кваліфікованих

викладачів цікаво, але за підсумками лекції засвоюється невелика частина матеріалу;

- постійно відбувається збільшення складності домашнього завдання по відношенню до викладеного матеріалу;

- недоліком також є те що наставники прямо не відповідають на поставлені питання, замість цього вони підштовхують до пошуку правильної відповіді, такий підхід дратує;

- недостатня візуалізація, демонстрацію коду відбувається через відео-програвач залишає бажати кращого, намагаючись виконувати верстку слідом за викладачем, постійно доводиться зупиняти відео ряд та намагатися спіймати код, що не завжди виходить;

- в процесі викладення матеріалу не використовується візуалізація навчального процесу за допомогою анімації [13, 14].

2.3.5 Курси GeekBrains

GeekBrains [15, 16] (дивись рисунок 2.10) – Компанія у сфері онлайн-освіти, заснована в 2010 році. Освітня платформа GeekBrains пропонує курси з інформаційних технологій, програмуванню, аналітики, тестуванню, маркетингу, управлінню та дизайну.

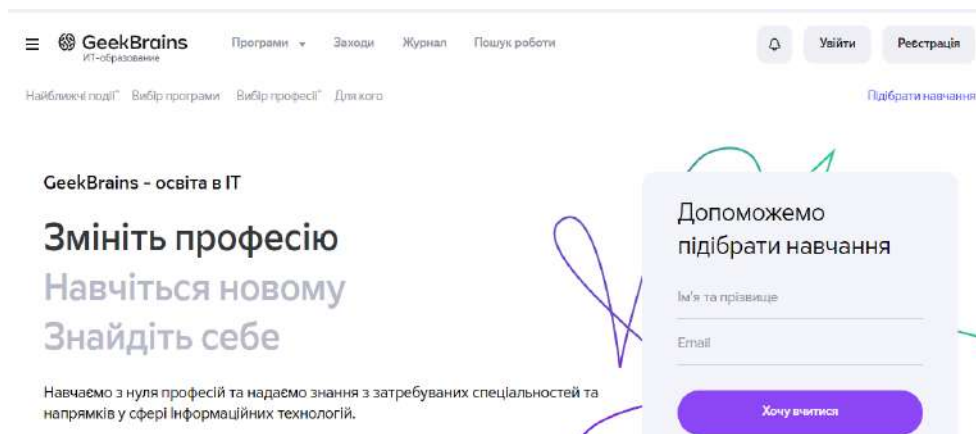


Рисунок 2.10 – Головна сторінка GeekBrains

Джерело: [16]

Курс включає в себе вісім онлайн-лекцій, за підсумками яких студент отримує методичні вказівки та домашнє завдання. Записи лекцій залишаються доступними в особистому кабінеті студента на протязі півтора роки після завершення курсу (для прикладу на рисунку 2.11 показаний особистий кабінет користувача курсів GeekBrains).

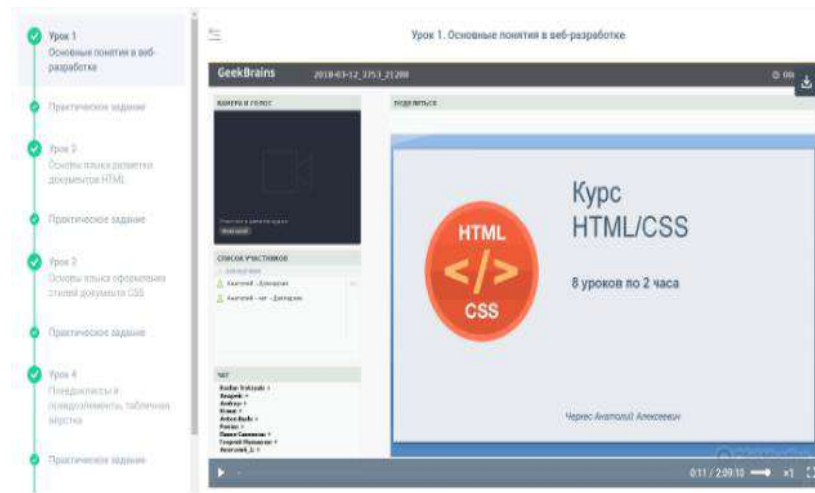


Рисунок 2.11 – Записи лекцій в особистому кабінеті GeekBrains

Джерело: [16]

Основні переваги навчання з GeekBrains:

- зручний особистий кабінет, в якому зберігаються записи лекцій, методичні вказівки, виконані домашні завдання;
- є можливість задавати питання викладачам під час онлайн-лекції;
- добре організоване навчання, лектори багато часу приділяють методології БЕМ [15, 16].

Методологія БЕМ (англ. Block Element Modifier) - це підхід до розробки веб-інтерфейсів, який допомагає створювати легкі, гнучкі та легко змінювані CSS-стилі для веб-сайтів та веб-додатків.

Основна ідея методології БЕМ полягає в тому, щоб розділити веб-інтерфейс на блоки (англ. blocks), елементи (англ. elements) та модифікатори (англ. modifiers). Блоки є самодостатніми компонентами веб-інтерфейсу, які містять в собі інші елементи та модифікатори, і можуть бути використані

повторно в різних частинах веб-сайту. Елементи - це складові частини блоків, які не можуть бути використані окремо від блоку. Модифікатори – це зміни, які можуть бути застосовані до блоків та елементів, щоб змінити їх зовнішній вигляд або поведінку.

Використання методології БЕМ дозволяє зменшити кількість CSS-класів, необхідних для стилізації веб-інтерфейсу, та зберегти час розробки та підтримки. Також методологія БЕМ дозволяє зменшити кількість конфліктів у CSS-стилях та зберігати чистоту та читабельність коду.

Одним із основних переваг методології БЕМ є те, що вона підходить для роботи в команді, де кілька розробників можуть працювати з одним і тим же кодом.

Основні недоліки навчання з GeekBrains:

- методичні вказівки не надають достатньої інформації для виконання домашнього завдання, необхідно знаходити додаткові матеріали, враховуючи що курси платні такий підхід до надання теоретичного матеріалу є неприйнятним;

- формат онлайн-лекцій є сумнівним, із задоволенням слухаєш кваліфікованих викладачів, під час лекції складається враження, що все зрозуміло, що все зможеш використати на практиці, але враження хибні коли діло доходить до виконання домашньої роботи, розумієш що засвоїв лише невелику кількість інформації, на мою думку лекції в відео режимі з використанням анімації в якості візуалізації навчального процесу більше підходить для ефективного засвоєння матеріалів;

- недостатня візуалізація процесу побудови каскаду стилів викладачами під час онлайн-лекцій, що частково компенсується можливістю подивитися лекцію в запису, але все одно необхідно «ловити» синтаксис тегу на відео-програвачі, що вдається не завжди [15, 16].

2.3.6 Курси MateAcademy

MateAcademy [17] (дивись рисунок 2.12) - це навчальний центр, який пропонує інтенсивні курси з програмування та технічних спеціальностей для початківців і професіоналів. Курси MateAcademy мають різні рівні складності, що дозволяє студентам вибрати найбільш підходящий для них курс.



Рисунок 2.12 – Головна сторінка MateAcademy

Джерело: [17]

Навчальний процес організований у формі інтенсивних практичних занять з досвідченими викладачами, які мають багаторічний досвід у певній області. У процесі навчання студенти зможуть розвивати свої знання та навички в програмуванні, дизайні, тестуванні, проектуванні та розробці програмного забезпечення.

Кожен курс складається з ряду теоретичних занять та багатьох практичних вправ, що допомагають студентам закріпити знання та отримати практичний досвід у розробці програмного забезпечення (дивись рисунок 2.13). Також, студенти матимуть можливість працювати над реальними проектами, що дозволить їм набути цінного досвіду та відчувати себе часткою команди розробників.

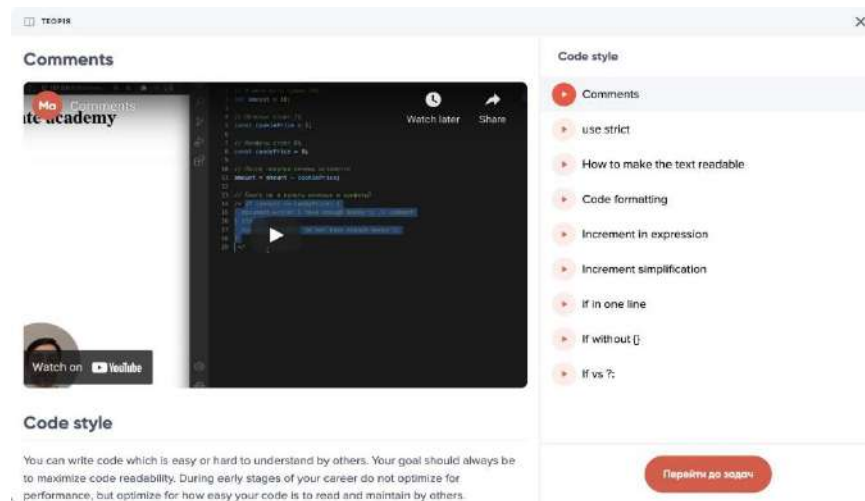


Рисунок 2.13 – Відео урок на MateAcademy

Джерело: [17]

Основні переваги навчання з MateAcademy:

- курси є україномовним ресурсом;
- широкий вибір курсів, MateAcademy пропонує широкий вибір курсів з програмування та технічних спеціальностей, що дозволяє студентам обрати курс, який відповідає їх потребам та цілям;
- практичний підхід, курси MateAcademy орієнтовані на практичне навчання, що дозволяє студентам отримати практичні навички та досвід, необхідні для успішної кар'єри в галузі програмування та технічних спеціальностей;
- гнучкий графік, MateAcademy пропонує гнучкий графік навчання, що дозволяє студентам пристосовувати свій навчальний розклад до своїх особистих потреб.

Основні недоліки навчання з MateAcademy:

- висока вартість, курси MateAcademy можуть бути досить дорогими порівняно з іншими навчальними центрами з програмування та технічних спеціальностей;
- не всі курси доступні онлайн, деякі курси MateAcademy доступні тільки в офлайн-форматі, що може стати проблемою для студентів, які не мають можливості проживати в місті, де знаходиться навчальний центр;

- відсутність індивідуального навчання, MateAcademy пропонує лише групові курси, що не підходить для кожного студента, деякі студенти можуть відчувати потребу в індивідуальному навчанні для досягнення своїх особистих навчальних цілей;

- відсутність підтримки після завершення курсів, після закінчення курсів MateAcademy, студенти не отримують достатньої підтримки від навчального центру для подальшого розвитку своєї кар'єри в галузі програмування та технічних спеціальностей;

- обмежена кількість курсів, хоча MateAcademy пропонує широкий вибір курсів, цей вибір може бути обмеженим порівняно з іншими навчальними центрами;

- основним засобом передачі інформації є відео-програвач, тобто недостатня ступінь візуалізації навчального процесу та навчальних матеріалів.

2.4 Рекомендації щодо розробки поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Проаналізувавши системи дистанційного навчання мові каскадних таблиць стилів (CSS), вивчивши виявлені переваги та недоліки таких систем, були розроблені наступні рекомендації щодо розробки системи дистанційного навчання мові каскадних таблиць стилів:

- використання сучасних засобів візуалізації навчального процесу, таких як JavaScript та інших, для анімації методичних матеріалів, завдань, супутньої анімації відео-лекцій то що, з метою збільшення ефективності засвоєння інформації;

- доступність використання для навчання, тобто можливість отримати доступ до системи навчання за допомогою комп'ютеру, або будь якого іншого гаджету;

- тривалість одного уроку не повинна перевищувати 10хв. – 20хв., довга лекція навіть у виконанні досвідченого викладача з цікавим матеріалом не сприяє ефективному засвоєнню матеріалу;

- по можливості двосторонній та швидкий контакт студента з викладача, при наявності в системі навчання двостороннього зв'язку між викладачем та студентом;

- можливість отримати зразок виконання завдання з максимально наближеним рішенням поставленої задачі, але не раніше ніж протягом часу необхідного для вирішення поставленої задачі, за умови наявності системи тестування отриманих знань у студента;

- дотримання принципу наступності, тобто неможливості перейти до наступного завдання без виконання попереднього, за умови наявності системи тестування отриманих знань у студента;

- робити упор на виконання практичних вправ, з паралельним наданням мінімально необхідної, але обов'язково достатньої кількості інформації для вирішення вправ;

- структурованість теоретичного матеріалу, тобто чіткий перехід легкого до важкого та від загального до окремого, щоб передати студентам саму суть та принцип побудови синтаксису мови програмування та формування коду;

- перевірка правильності вирішення завдання має відбуватися в максимально стислий час, при наявності системи тестування отриманих знань у студента, за умови наявності системи тестування отриманих знань у студента;

- чітка та зрозуміла постановка задачі, яка не може бути трактована по різному, за умови наявності системи тестування отриманих знань у студента;

- складність викладеного матеріалу не повинна відрізнятися від поставленого завдання, принаймні в сторону ускладнення, за умови наявності системи тестування отриманих знань у студента;

- при виникненні питань у студентів до викладача чи куратора, надавати відповідь на запитання, а не понукати студента на самостійний пошук

відповіді, при наявності в системі навчання двостороннього зв'язку між викладачем та студентом.

2.5 Висновки за розділом

В даному розділі були проаналізовані відомості про сучасне дистанційне навчання, його основні переваги та недоліки. Вивчені особливості засвоєння інформації при дистанційному навчанні.

Проведено аналіз основних сучасних систем дистанційного навчання мові каскадних таблиць стилів (CSS).

В результаті аналізу були виявлені переваги та недоліки цих систем, на основі яких були розроблені рекомендації щодо розробки поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

В наступному розділі, на основі отриманих рекомендації буде описаний процес розробки та створення поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПОЛІПШЕНОЇ СИСТЕМИ ДИСТАНЦІЙНОГО НАВЧАННЯ МОВИ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS)

3.1 Інструменти створення поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS)

В якості інструмента реалізації рекомендації щодо створення поліпшеної системи дистанційного навчання мови каскадних таблиць стилів (CSS), що були отримані при аналізі сучасних систем дистанційного навчання, буде використано текстовий редактор вихідного коду Visual Studio Code.

Поліпшена система дистанційного навчання мови каскадних таблиць стилів (CSS), буде реалізована у вигляді web-документа (web-сайту) структура якого буде написана за допомогою мови гіпертекстової розмітки HTML, зовнішній вигляд документу буде створюватися за допомогою мови каскадних таблиць CSS, рекомендації щодо візуалізації навчального процесу буде реалізовано за допомогою мови програмування JavaScript.

3.1.1 Текстовий редактор вихідного коду Visual Studio Code

Visual Studio Code [18] (VS Code) – безкоштовний текстовий редактор, розроблений компанією Microsoft для Windows, Linux та macOS у 2015 році. Являє собою редактор коду для кросплатформової розробки web та cloudy додатків. Складається з відладчика, інструментів для роботи з Git, підсвічування синтаксису та іншого (на рисунках 3.1-3.3 показані сторінки файлів HTML, CSS, JS в VSCode з підсвічуванням синтаксису).

В якості основи для VSCode використані напрацювання проекту Atom. VSCode є надбудовою над Atom Shell, що використовує браузерний «двіжок» Chromium і Node JS.

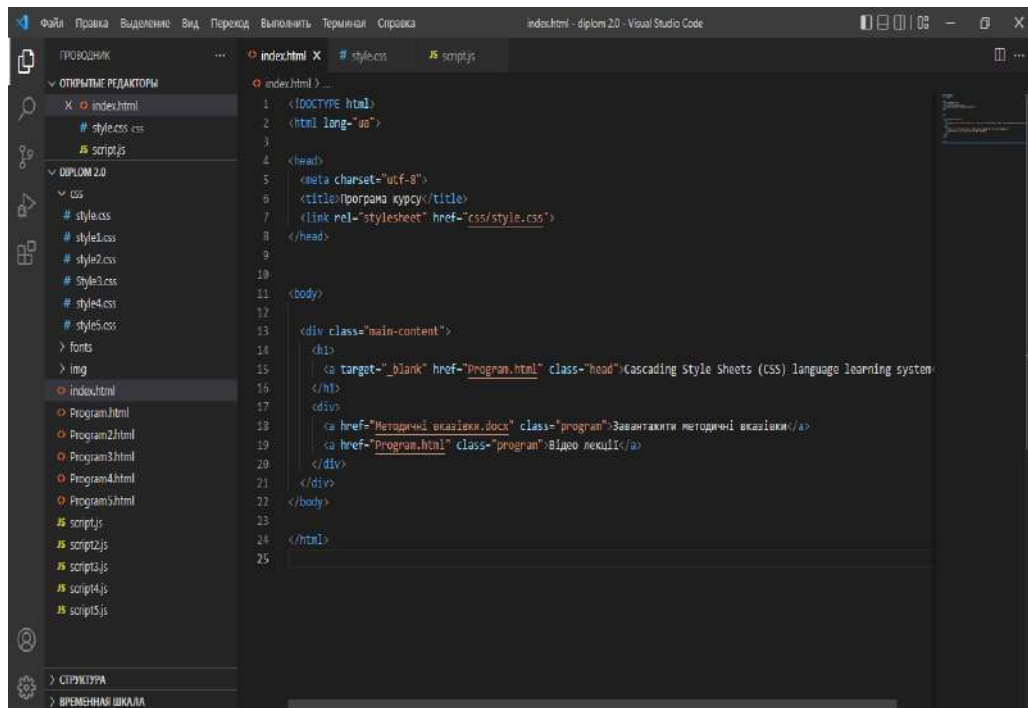


Рисунок 3.1 – Сторінка файлу HTML в VSCode з підсвічуванням синтаксису

Джерело: [Матеріал автору]

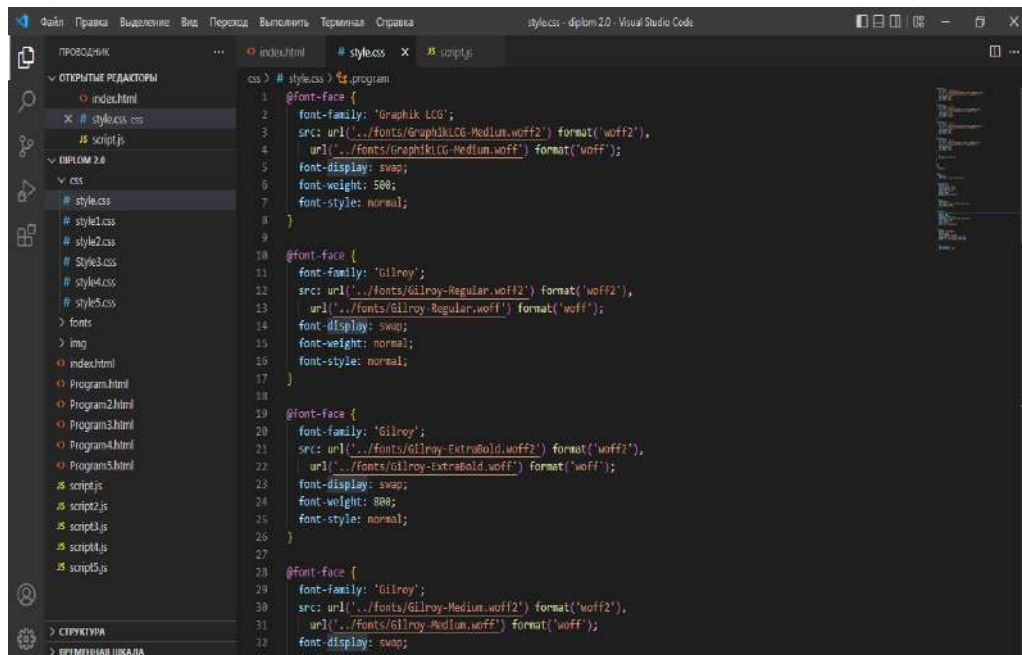


Рисунок 3.2 – Сторінка файлу CSS в VSCode з підсвічуванням синтаксису

Джерело: [Матеріал автору]

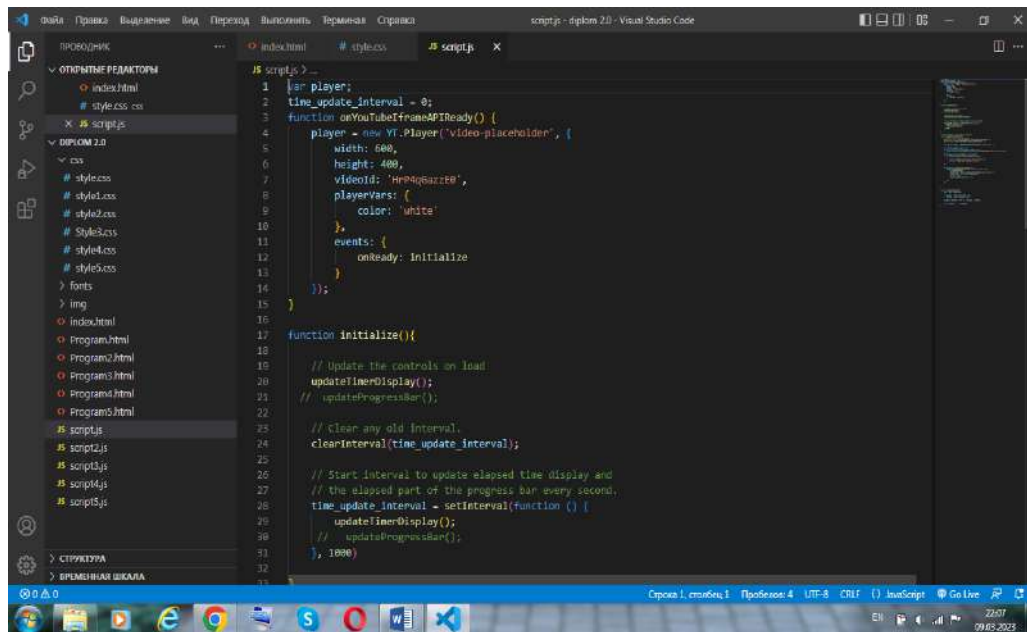


Рисунок 3.3 – Сторінка файлу JavaScript в VSCode з підсвічуванням синтаксису

Джерело: [Матеріал автору]

3.1.2 Мова гіпертекстової розмітки HTML

HTML [19] (Hyper Text Markup Language) – мова розмітки web-документів для створення web-сторінок. При використанні інструментів розробки web-сторінок, браузері отримують web-документ з кодом HTML з серверу або локального диску, далі браузері інтерпретують код в інтерфейс, який відображається на екрані монітора.

Мова HTML являє собою сукупність елементів для побудови web-сторінок, елементи HTML мають блочну структури, за допомогою блоків HTML web-сторінка отримує структуру з зображень, посилань, абзаців, заголовків то що, для інтерпретації вмісту блоків використовуються спеціальні теги (, <a>, <p></p> та інші), які виводять зміст web-документів на web-сторінку. На рисунку 3.4 показаний типовий документ зі структурою html-документу.

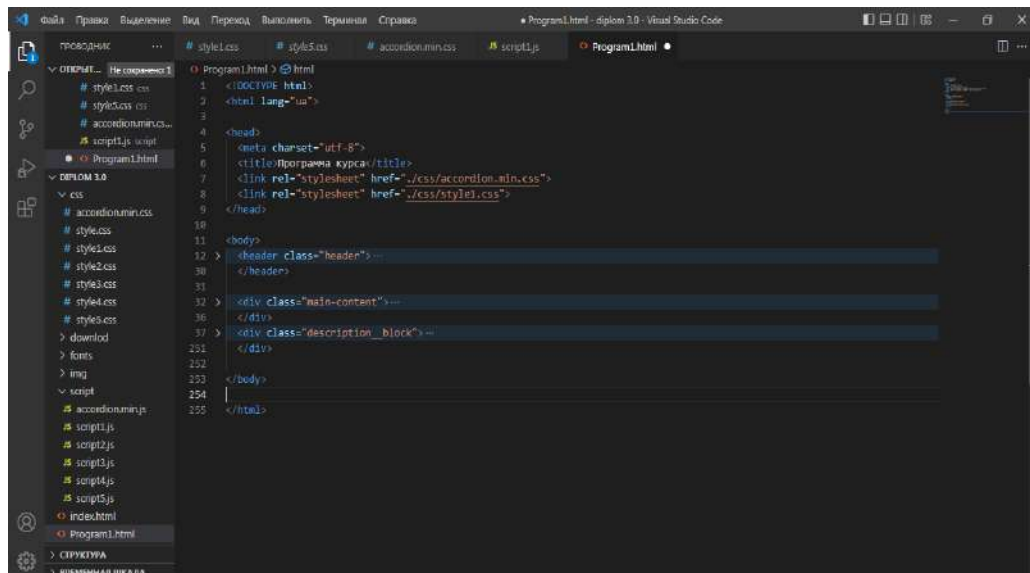


Рисунок 3.4 – Сторінка файлу в VSCode з синтаксисом html

Джерело: [Матеріал автору]

3.1.3 Мова каскадних таблиць стилів CSS

CSS (Cascading Style Sheets), каскадні таблиці стилів – це спеціальна мова (мова стилів), за допомогою якої описують вигляд документів (як і де відображати елементи веб-сторінки), написаних мовами розмітки даних. Найчастіше CSS використовується для документів, котрі створені за допомогою HTML, XHTML та XML.

CSS використовується при створенні («верстці») веб-сторінок для задання кольорів, шрифтів, розміщення і інших аспектів представлення документа.

CSS, є об'єктом вивчення даної роботи, основні відомості про CSS знаходяться в розділі 1 пункті 1.1. цього документу.

3.1.4 Мова програмування JavaScript

JavaScript [20] – динамічна, об'єктно-орієнтована прототипна мова програмування. Використовується для створення сценаріїв web-сторінок, що

дозволяє обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд web-сторінок.

Підключення JavaScript-документів до html-документів відбувається за допомогою тегу `<script>` (дивись рисунок 3.5), хоча в сучасних браузерах мова JavaScript є мовою сценаріїв за замовченням.

```
</body>
<script src="./script/accordion.min.js"></script>
<script src="./script/script1.js"></script>
</html>
```

Рисунок 3.5 – Частина коду html з підключенням JavaScript-документів

Джерело: [Матеріал автору]

На теперішній час JavaScript є найпопулярнішою мовою програмування, та за замовченням є мовою сценаріїв в більшості браузерів. На рисунку 3.6 показана структура сторінки з синтаксисом JavaScript.

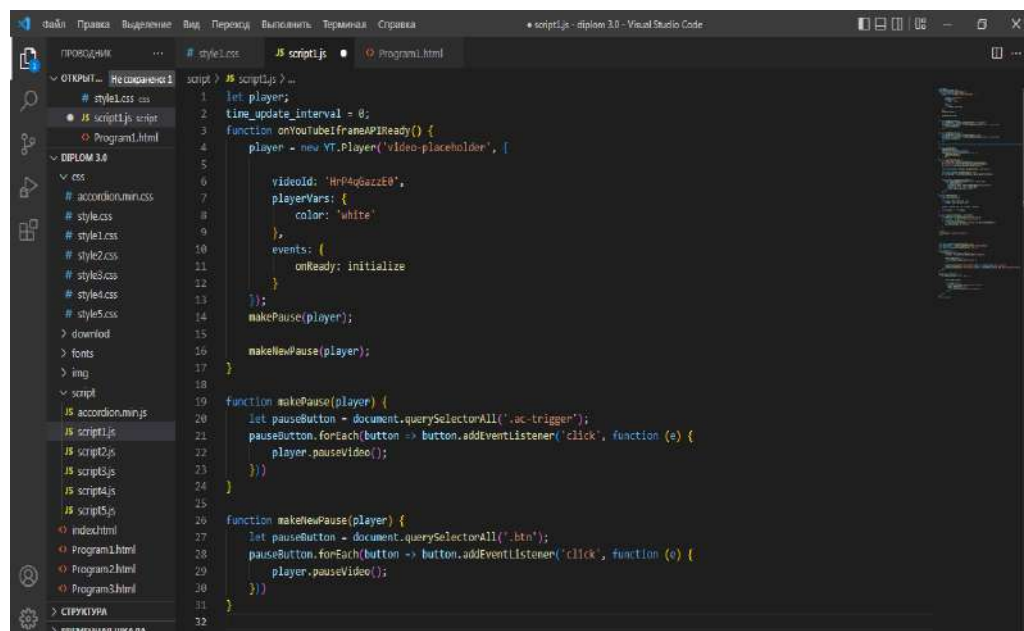


Рисунок 3.6 – Сторінка файлу в VSCode з синтаксисом JavaScript

Джерело: [Матеріал автору]

3.2 Структура та використання поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS) має вигляд web-сторінки і складається з головної сторінки з навігаційною панеллю де є можливість завантажити методичні матеріали курсу до гаджету користувача та кнопки переходу до відео-лекцій (дивись рисунок 3.7 більш детальне зображення системи знаходиться в додатку D).

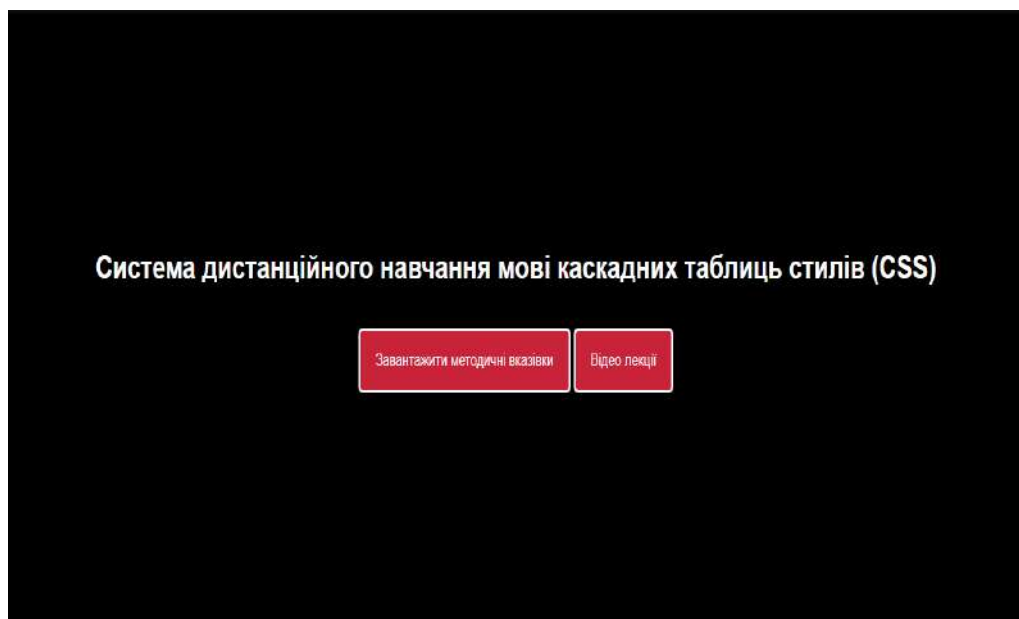


Рисунок 3.7 – Головна сторінка поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Сторінок з відео лекціями (в якості відеоматеріалу використані лекції з YouTube каналу SavchukIT.) [21]. та навігаційною панеллю за допомогою якої можна завантажити методичні матеріали за темою, перейти до наступної лекції чи повернутися до попередньої та перейти на головну сторінку (дивись рисунок 3.8 більш детальне зображення системи знаходиться в додатку B).

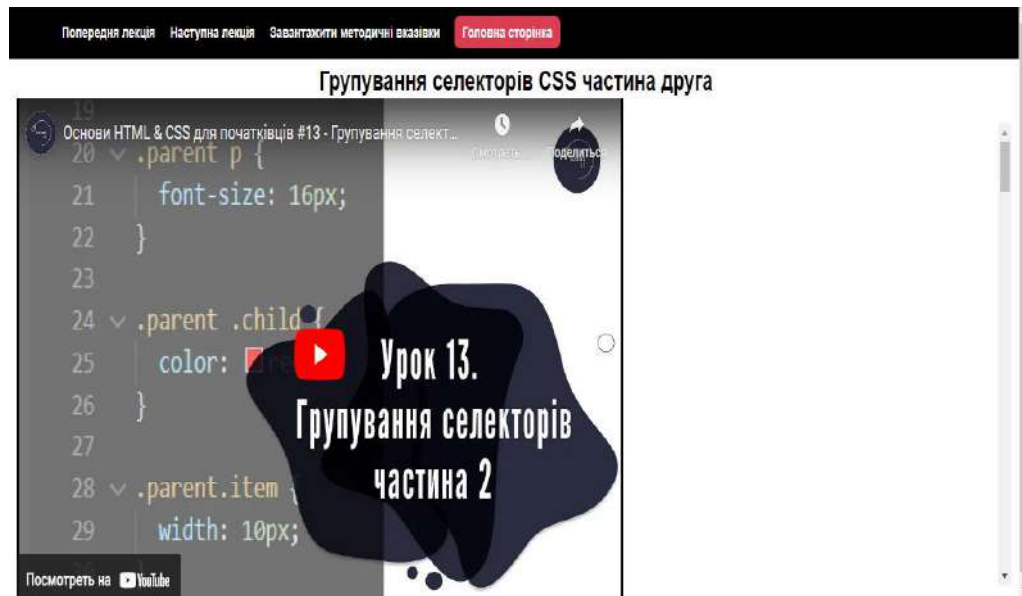


Рисунок 3.8 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Система навчання є асинхронною з модульною структурою, де блок з відео-матеріалами, блок «анімаційних вікон», «вікон з поясненнями» та «вікон імітації web-сторінок» виконані (за допомогою html, CSS та JavaScript) окремими модулями.

Модульна структура системи навчання – це підхід до організації системи, який передбачає поділ системи на окремі модулі, кожен з яких має своє призначення та функції.

Загалом система налічує п'ять сторінок з відео-лекціями. Однак концепція поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) дозволяє використовувати будь яку кількість відео-матеріалів та може бути використана для вдосконалення відео-лекцій на будь яку тематику пов'язану з мовами програмування чи засобами створення web-сторінок.

До кожної відео лекції, в якості супутньої анімації, підключено блок з «анімаційними вікнами» (загальною кількістю 46 шт.), також до кожного вікна підключаються «вікна з поясненнями» (загальною кількістю 46 шт.) та блок з

«вікнами з імітацією web-сторінки» (загальною кількістю 45шт., в четвертій відео-лекції в чотирнадцяте «анімаційне вікно», має зміст де описується селектори CSS, тобто застосувати код не має можливості).

Структура поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) представлена в таблиці 3.1.

На рисунку 3.9. приведена модель взаємодії користувача з поліпшеною системою дистанційного навчання мові каскадних таблиць стилів (CSS).

Таблиця – 3.1 Структура поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Склад web-сторінки							
Головна сторінка							
Навігаційна панель							
Кнопка завантаження методичних матеріалів курсу				Кнопка переходу до відео-лекцій			
Сторінка відео-лекції на тему: «Підключення CSS»							
Навігаційна панель							
Кнопка переходу до наступної лекції		Кнопка завантаження методичних матеріалів за темою лекції			Кнопка переходу до головної сторінки		
Блок з відео-матеріалами							
Відео-лекція з YouTube каналу SavchukIT, на тему: «Підключення CSS»							
Блок з «Анімаційними вікнами»							
Кількість та склад вікон на сторінці							
1		2		3		4	
Підключення стилю за допомогою атрибуту «style» до тегу <h1>		Підключення стилю за допомогою тегу <style> до елемента h1		Підключення стилю за допомогою CSS правила до селектору h1		Підключення CSS документу до html документу за допомогою тегу <link>	
«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»

[illegible]

[illegible]

Продовження таблиці – 3.1

Склад web-сторінки											
Сторінка відео-лекції на тему: «Devtools»											
Навігаційна панель											
Кнопка переходу до попередньої лекції			Кнопка завантаження методичних матеріалів за темою лекції			Кнопка переходу до головної сторінки					
Блок з відео-матеріалами											
Відео-лекція з YouTube каналу SavchukIT, на тему: «Devtools»											
Блок з «Анімаційними вікнами»											
Кількість та склад вікон на сторінці											
1		2		3		4		5		6	
Клавіша F12 натиснута в браузері дозволяє викликати інструмент розробника devtools		Натиснувши вкладку Elements в Devtools ми отримуємо доступ до структури web-документу		Натиснувши комбінацію клавіш Ctrl+Shift+C ми отримуємо можливість обрати конкретний елемент на сторінці		Обравши вкладку Styles ми отримуємо доступ до стилів сторінки, там же ми можемо вносити зміни в стилістичне оформлення документу, зміни діють тільки до перевантаження сторінки		Так само ми можемо вимикати чи вмикати ті чи інші стилі сторінки, зміни так само діють до першого перевантаження		При натисканні на будь який елемент сторінки правою кнопкою миші, ми можемо, обравши вкладку, Delete elements, видалити цей елемент. Діє до першого перевантаження сторінки	
«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»	«Вікно з поясненнями»	«Вікно імітації web-сторінки»



Рисунок 3.9 – Взаємодія користувача з поліпшеною системою дистанційного навчання мові дистанційного навчання мові каскадних таблиць стилів. Джерело: [Матеріал автору]

3.2.1 Навігаційна панель поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Реалізація навігаційної панелі web-сторінок поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) та функція завантаження методичних матеріалів виконана за допомогою html тегів з атрибутами та присвоєними відповідними класами (на рисунку 3.10 показана частина коду html вся структура html-документу знаходиться в додатку А). З використанням стилів CSS, псевдо класів CSS, псевдо елементів CSS та властивостей CSS, які задані до відповідних класів та тегів. Зокрема були використані інструменти Flexbox.

```
<header class="header">
  <ul class="nav-list">
    <li><a class="nav-list-link" href="./Program2.html">Наступна
лекція</a></li>
    <li><a class="nav-list-link" href="./download/МЕТОДИЧНІ МАТЕРІАЛИ
1.docx">Завантажити методичні вказівки</a></li>
    <li><a href="index.html">Головна сторінка</a></li>
  </ul>
</header>
```

Рисунок 3.10 – Частина коду html з реалізацією навігаційної панелі та функції завантаження методичних вказівок поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

CSS Flexbox [22] – це технологія (набір інструментів) створення складних, гнучких макетів web-сторінок, для розміщення елементів, як в середині блоку так і на сторінці взагалі.

CSS Flexbox дозволяють:

- вирівнювання дочірні елементи в середині батьківського блоку по вертикалі;

- оформлення усіх дочірніх елементів блоку так, щоб вони розподіляли між собою усю доступну висоту та ширину, незалежно від того скільки висоти чи ширини є в наявності;

- зробити усі колонки в макеті однаковими за шириною, незважаючи на ступінь їх заповнення.

Зовнішній вигляд навігаційної панелі реалізовано за допомогою стилів CSS, псевдо класів CSS, псевдо елементів CSS та властивостей CSS Position та Display, які задані до відповідних класів та тегів (на рисунку 3.11 показана частина CSS-документу з правилами CSS, весь CSS-документ сторінки приведено в додатку Б).

```
ul {
margin: 0;
list-style-type: none;
}

a {
text-decoration: none;
color: inherit;
}

.header {
display: flex;
align-items: center;
height: 50px;
background-color: black;
justify-content: space-between;
padding: 0 20px;
}
```

Рисунок 3.11 – Частина коду CSS з реалізацією стилів навігаційної панелі та функції завантаження методичних вказівок поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Зміна зовнішнього вигляду кнопок в навігаційній панелі при наведенні на них курсору «миші» виконана за допомогою CSS властивості position: absolute; та position: relative; (position: absolute; встановлює позицію елемента відносно найближчого батьківського елемента з position: relative; або до елемента <body>, якщо жоден такий елемент не знайдено. Коли елемент має position:

absolute;, він вилучається зі звичайного потоку сторінки, тобто інші елементи на сторінці можуть розміщуватися під ним. Координати позиції елемента встановлюються через властивості top, right, bottom та left.

position: relative; встановлює позицію елемента відносно його вихідної позиції. Коли елемент має position: relative;, він залишається в потоці сторінки, тобто інші елементи на сторінці не можуть розміщуватися під ним. Координати позиції елемента встановлюються через властивості top, right, bottom та left. на рисунку 3.12 показана частина CSS-документу з правилами CSS, весь CSS-документ сторінки приведено в додатку Б).

```
.nav-list-link {
  position: relative;
  transition: all .25s ease;
}

.nav-list-link::after {
  content: "";
  background: #17baaa;
  height: 2px;
  width: 100%;
  position: absolute;
  bottom: -8px;
  left: 0;
  transform: scaleX(0);
  transition: all .25s ease;
}
```

Рисунок 3.12 – Частина коду CSS з властивостями position: absolute; та position: relative;

Джерело: [Матеріал автору]

3.2.2 Підключення відео-лекцій до поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Підключення відео з каналу YouTube виконано за допомогою Iframe Player API [23], що дозволяє вбудовувати відео-програвач YouTube на сайт та керувати його роботою за допомогою JavaScript.

Таке рішення дозволяє використовувати відео матеріали, які зберігаються на інших серверах, що в свою чергу дає можливість скоротити об'єм пам'яті web-проекту, за рахунок відсутності відео файлів в ньому, а також зменшити час завантаження web-сторінки поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

Перед встановленням YouTube-програвача на сторінку, до html-документу необхідно підключити бібліотеку jQuery.

jQuery – це бібліотека JavaScript, яка дозволяє легко і швидко взаємодіяти з HTML-сторінками та відображеннями. Вона проста у використанні та має широку підтримку, що робить її однією з найпопулярніших бібліотек JavaScript.

За допомогою JavaScript реалізуємо програвач YouTube у JavaScript-файлі відповідної сторінки, використовуючи наступну функцію (на рисунку 3.13 показана частина коду JavaScript, весь код JavaScript сторінки приведено в додатку В).

```
let player;
time_update_interval = 0;
function onYouTubeIframeAPIReady() {
    player = new YT.Player('video-placeholder', {
        videoId: 'HrP4qGazZE0',
        playerVars: {
            color: 'white'
        },
        events: {
            onReady: initialize
        }
    });
};
```

Рисунок 3.13 – Фрагмент коду JavaScript для створення YouTube-плеєра за допомогою API Iframe

Джерело: [Матеріал автору]

Ця функція дозволяє викликати відео програвач на сторінку сайту в середині блоку <div id="ytplayer"></div>, де "ytplayer" є ідентифікатором (В HTML ідентифікатор (ID) є атрибутом, який можна встановити для будь-якого елемента на веб-сторінці. Ідентифікатор визначає унікальне ім'я для елемента,

яке використовується для зв'язку з ним з CSS та JavaScript. Ідентифікатори дозволяють легко змінювати стилі та поведінку конкретного елемента на сторінці без необхідності змінювати код всієї сторінки. Вони також дозволяють створювати якісний та читабельний код, що полегшує його зрозуміння та редагування.).

Характеристики відео програвача задаються за допомогою CSS правил в CSS-документі сторінки (на рисунку 3.14 показана частина правил CSS що надає стилі та властивості програвачу відео-лекцій, весь CSS документ сторінки надається в додатку Б).

```
.video {
    width: 120vw;
    min-height: 35vw;
    height: auto;
    -webkit-size: cover;
    -moz-size: cover;
    -o-size: cover;
    size: cover;
    margin-left: 10px;
}
```

Рисунок 3.14 – Фрагмент CSS-документу з CSS правилом для надання властивостей програвачу
Джерело: [Матеріал автору]

3.2.3 Надання методичних матеріалів поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

На головній сторінці встановлена кнопка при натисканні на яку відбувається завантаження методичних матеріалів курсу, до гаджету користувача (дивись рисунок 3.15 більш детальне зображення системи приведено в додатку Г).

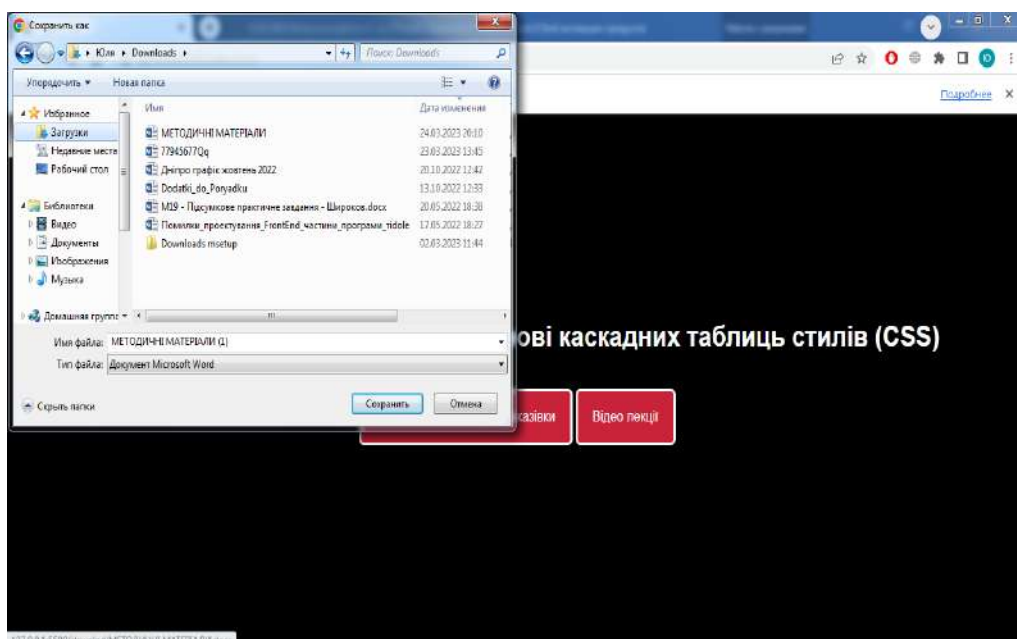


Рисунок 3.15 – Завантаження методичних матеріалів з головної сторінки поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Так само на кожній сторінці з відео-лекціями знаходиться кнопка в навігаційній панелі сторінки, що дозволяє завантажити методичних матеріалів відповідної теми, до папки «завантажень» користувача (дивись рисунок 3.16 більш детальне зображення системи приведено в додатку Г).

Такий підхід дозволяє зменшити навантаження на відео-лекції, обмеживши в них кількість теоретичних матеріалів, та зосередитися на практичній частині навчання.

Зменшення теоретичного матеріалу на відео-лекціях дозволяє більше часу виділити на практичну роботу, що може покращити навички студентів та підготувати їх до реальних завдань у майбутньому. Крім того, це може знизити вартість створення та проведення навчання, оскільки практична робота може бути більш ефективною та менш витратною в плані ресурсів.

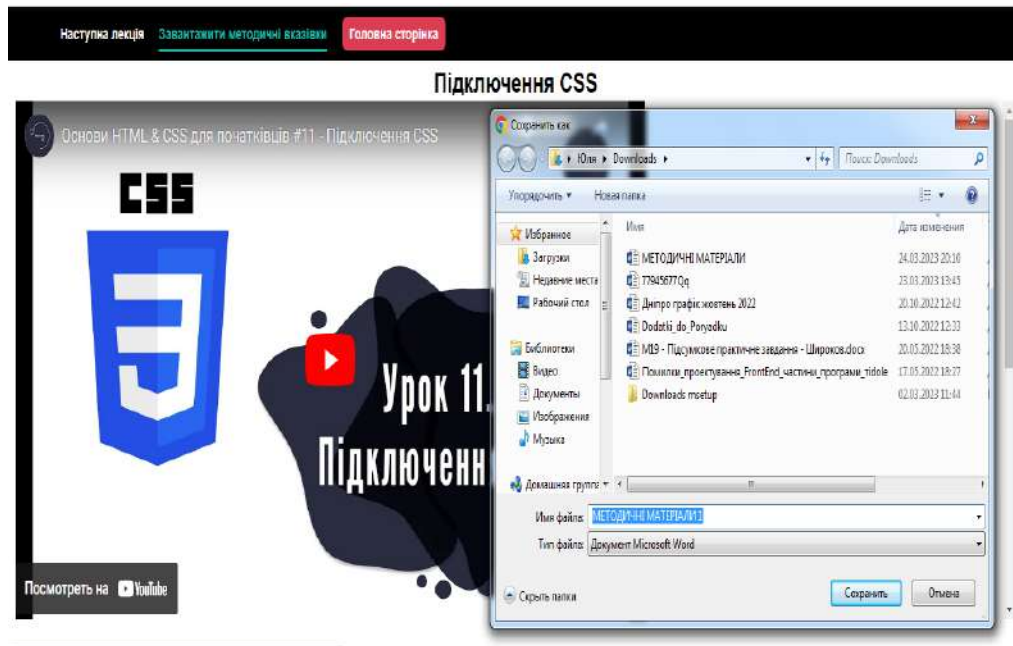


Рисунок 3.16 – Завантаження методичних матеріалів зі сторінки з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Така можливість реалізована за рахунок можливостей html тегу посилань `<a>` з атрибутом `href`, який надає можливість вказати шлях браузеру з якого сховища завантажувати матеріал (на рисунку 3.17 показана частина коду html з реалізацією посилання за яким можна завантажити методичні вказівки, вся html структура сторінки показаний в додатку А)

```
<li>
  <a class="nav-list-link" href="./download/МЕТОДИЧНІ МАТЕРІАЛИ 1.docx">
    Завантажити методичні вказівки</a>
</li>
```

Рисунок 3.17 – Частина коду html з реалізацією можливості завантажити методичні матеріали поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

3.2.4 Принцип роботи та функціонування «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Під час перегляду відео-лекцій, в обумовлений час поряд з відео-програвачем, з'являються вікна з текстом («анімаційні вікна»), що містять графічне зображення матеріалів про які розповідає автор (дивись рисунок 3.18, більш детальне зображення системи приведено в додатку Г). Таким чином відбувається візуалізація та анімація навчального матеріалу.

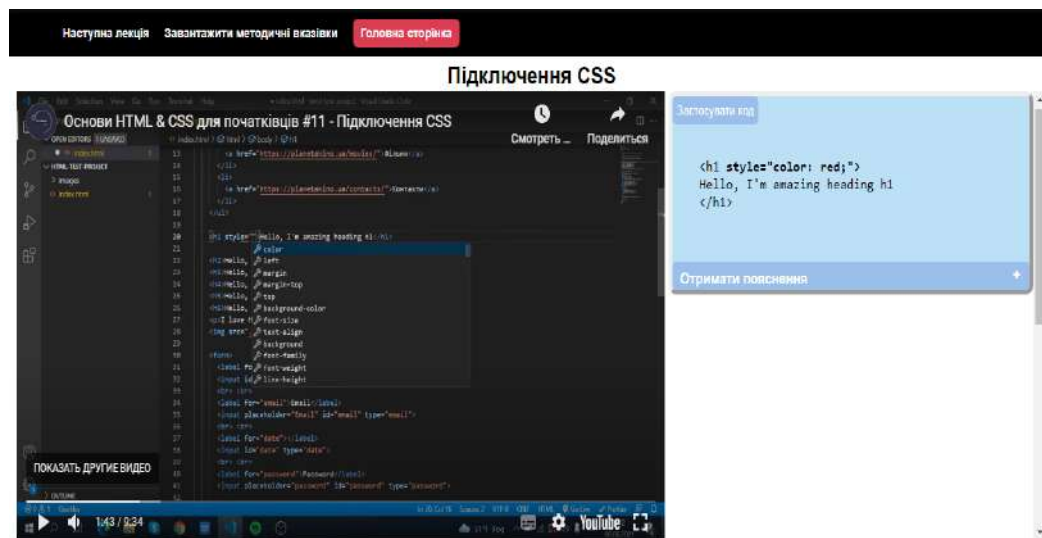


Рисунок 3.18 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з «анімаційним вікном»

Джерело: [Матеріал автору]

Вміст «анімаційних вікон» є текстом зі збереження синтаксису у залежності від змісту лекції, тобто html та css. Додається за допомогою html тегів `<pre>` (Тег `<pre>` в HTML використовується для попереднього форматування тексту, зберігаючи інтервали та переноси рядків, і виводить текст з такими самими інтервалами та переносами рядків як у вихідному коді.) з класом CSS `<pre><code class="language-css"><code></pre>` (на рисунку 3.19

показана частина коду html, вся html структура сторінки показаний в додатку А).

```
<div class="code-block" data-time-start="416">
  <pre>
    <code class="language-css">
      h1 {
        color: blue;
      }
    </code>
  </pre>
</div>
```

Рисунок 3.19 – Частина коду html з реалізацією вмісту «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Зовнішній вигляд «анімаційних вікон» створюється за допомогою стилів CSS, властивостей Flexbox, які задані до відповідних класів та тегів (на рисунку 3.20 показана частина CSS правил, всі CSS стилі показані в додатку Б).

```
.pryklady {
  flex-wrap: wrap;
  margin-left: 5px;
  margin-right: 5px;
  gap: 25px;
  width: 70%;
  height: 78vh;
  overflow-y: scroll;
  scroll-behavior: smooth;
  overflow-x: hidden;
}
.upr1_98,
.pryklady>div:nth-child(1) {
  border-color: var(--border-block-color);
  border: 5px outset;
  margin-bottom: 5px;
  background-color: rgb(188, 224, 245);
  font-size: 16px;
  border-radius: 10px;
}
```

Рисунок 3.20 – Частина коду CSS з реалізацією стилів «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Взаємодія «анімаційних вікон» з відео-програвачем YouTube відбувається за допомогою таймеру, який створено за допомогою JavaScript коду та розміщеному в JavaScript-документі сторінки (на рисунку 3.21 показано частину коду JavaScript з функцією таймеру, весь JavaScript-документ сторінки приведений в додатку В).

```
function initialize() {
    updateTimerDisplay();
    clearInterval(time_update_interval);
    time_update_interval = setInterval(function () {
        updateTimerDisplay();
    }, 1000)
}
```

Рисунок 3.21 – Частина коду JavaScript з реалізацією таймеру з таймером поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Стилі таймеру задано за допомогою CSS правила в CSS-документі (на рисунку 3.22 показана частина CSS-документа з правилом, весь CSS-документ сторінки показано у додатку Б).

```
.timer {
    display: none;
    height: 0;
    width: 0;
}
```

Рисунок 3.22 – Частина коду CSS з реалізацією стилів таймеру поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Таймер підключено до web-сторінки за допомогою тегу `<span class="timer" id="current-time">0:00</span>`, реалізація таймеру його взаємодія

з відео та «навігаційними вікнами» відбувається за допомогою JavaScript та відповідних функцій при якій що секунди відео рядок проходить відповідну перевірку, де в залежності від таймінгу властивості «анімаційних вікон» змінюються з `display: none;` на `display: block;` (на рисунку 3.23 показана частина коду JavaScript з функціями, що поєднують таймер з часом програвача, весь код JavaScript сторінки показаний в додатку В).

```
function updateTimerDisplay() {
    let currentTime = document.querySelector('#current-time');
    $('#current-time').text(formatTime(player.getCurrentTime()));
    $('#duration').text(formatTime(player.getDuration()));

    const block_examples = document.querySelectorAll('.code-block');

    const uprs = document.querySelectorAll('.upr');
    uprs.forEach(upr => visualizationCode(upr, player.getCurrentTime()))

    block_examples.forEach((block_example) => {
        block_example.addEventListener("click", function (e) {
            e.preventDefault();
            playerTime = block_example.getAttribute('data-time-start')
            console.log("click time" + playerTime);
            player.seekTo(playerTime, true);
        });
    });
}
```

Рисунок 3.23 – Частина коду JavaScript з реалізацією взаємодії «анімаційних вікон» з таймером поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

За допомогою JavaScript та відповідних функцій реалізується конвертування одиниць вимірювання відео-рядка таймера (на рисунку 3.24 показаний фрагмент коду JavaScript, що дозволяє конвертувати одиниці часу, весь код JavaScript сторінки приведено в додатку В), тобто конвертація хвилин і секунд відеоряду у секунди, до значення яких власне і підключається таймінг при якому з'являється відповідне «анімаційне вікно». Крім того в html документі відповідним блокам надані спеціальні класи з data-атрибутами <div

class="ac upr upr_98" data-time-start="98"> та <div class="code-block" data-time-start="98">

. Така зв'язка дозволяє створювати зв'язок відео-плеєра та появу «анімаційних вікон».

```
function formatTime(time) {
    time = Math.round(time);
    var minutes = Math.floor(time / 60),
        seconds = time - minutes * 60;
    seconds = seconds < 10 ? '0' + seconds : seconds;
    return minutes + ":" + seconds;
}
```

Рисунок 3.24 – Частина коду JavaScript з реалізацією конвертації одиниць вимірювання часу відео плеєра та таймера поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Data-атрибути [24] – це користувацький атрибут на елементі, назва якого починається з data-, наприклад в нашому випадку data-time-start. Дата атрибути використовуються для збереження значення на елементах в html.

Для чого за допомогою атрибуту data-time-start (на рисунку 3.25 показана частина коду html, вся html структура сторінки показаний в додатку А) відбувається «прив'язка» таймеру, відео та «анімаційного вікна».

```
<div class="code-block" data-time-start="98">
    <pre>
        <code class="language-css">
            &lt;h1 <b>style="color: red;"&gt;</b>
            Hello, I'm amazing heading h1
            &lt;/h1&gt;
        </code>
    </pre>
</div>
```

Рисунок 3.25 – Частина коду html з реалізацією взаємодії «анімаційних вікон» з таймером поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Блок появи «анімаційних вікон» виконано окремим та за допомогою властивостей CSS overflow, заданий скролінг блоку при його заповненні, за допомогою властивості CSS flex, задано можливість елементам в блоці появи «анімаційних вікон» заповнювати собою весь заданий простір, також відбувається фіксація скролінгу на блок, що з'являється за допомогою властивості CSS scroll-behavior (на рисунку 3.26 показана частина CSS документу, весь CSS-документ сторінки приведено в додатку Б).

```
.pryklady {
  flex-wrap: wrap;
  margin-left: 5px;
  margin-right: 5px;
  gap: 25px;
  width: 70%;
  height: 78vh;
  overflow-y: scroll;
  scroll-behavior: smooth;
  overflow-x: hidden;
}
```

Рисунок 3.26 – Частина коду CSS з реалізацією властивостей CSS, блоку появи «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Для того щоб фіксація на «анімаційному вікні» відбувалася тільки в момент його появи за допомогою JavaScript, задана відповідна перевірка активності «анімаційного вікна» (на рисунку 3.27 показана частина коду JavaScript, весь код JavaScript сторінки приведено в додатку В).

```
function visualizationCode(upr, currentTime) {
    let timeStart = upr.getAttribute('data-time-start');
    if (currentTime > timeStart) {
        if (!upr.classList.contains('upr-active')) { upr.scrollIntoView();
    }
        upr.classList.add('upr-active');
    }
}
```

Рисунок 3.27 – Частина коду JavaScript з реалізацією перевірки активності, блоку появи «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Такий підхід до візуалізації навчального матеріалу дозволяє виконувати всі дії викладача паралельно з ним без постійних переходів меж режимами перегляду відео (дивись рисунок 3.28 більш детальне зображення системи приведено в додатку Г).

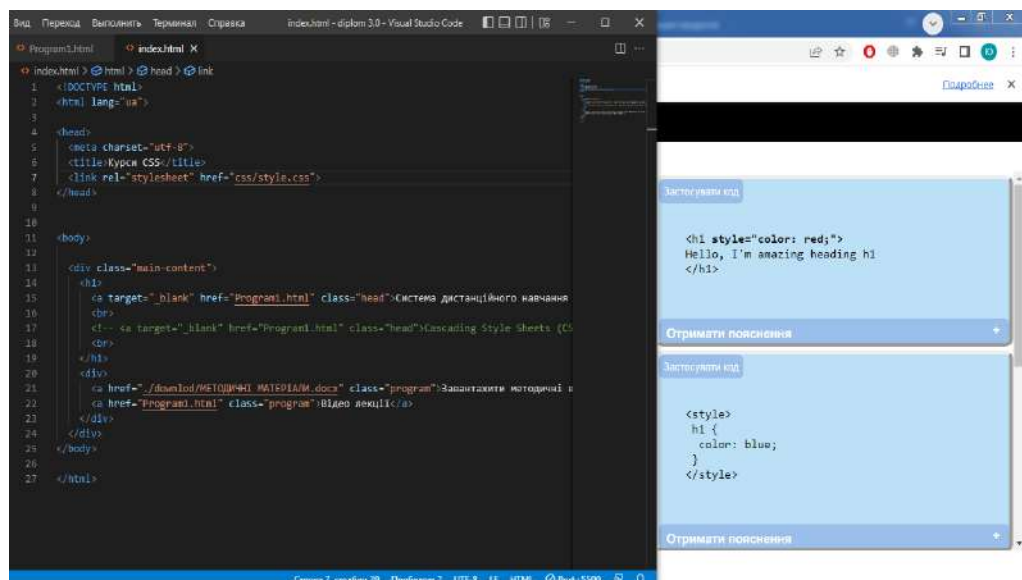


Рисунок 3.28 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) та інструменту створення web-сторінок VSCode

Джерело: [Матеріал автору]

Також текст з анімаційних вікон можна скопіювати (дивись рисунок 3.29 більш детальне зображення системи приведено в додатку Г) відразу до інструменту створення web-документів, зі збереженням синтаксису матеріалів (дивись рисунок 3.30).

Така можливість обумовлена вбудованими властивостями браузеру, за умовою того що текст на web-сторінці не захищений від копіювання. Сам текст web-сторінці збережено в форматі html, який є мовою розмітки, за замовченням коли відбувається виділення та копіювання тексту з web-сторінки у буфер обміну, браузер копіює просто текстову інформацію, а не html-код сторінки.

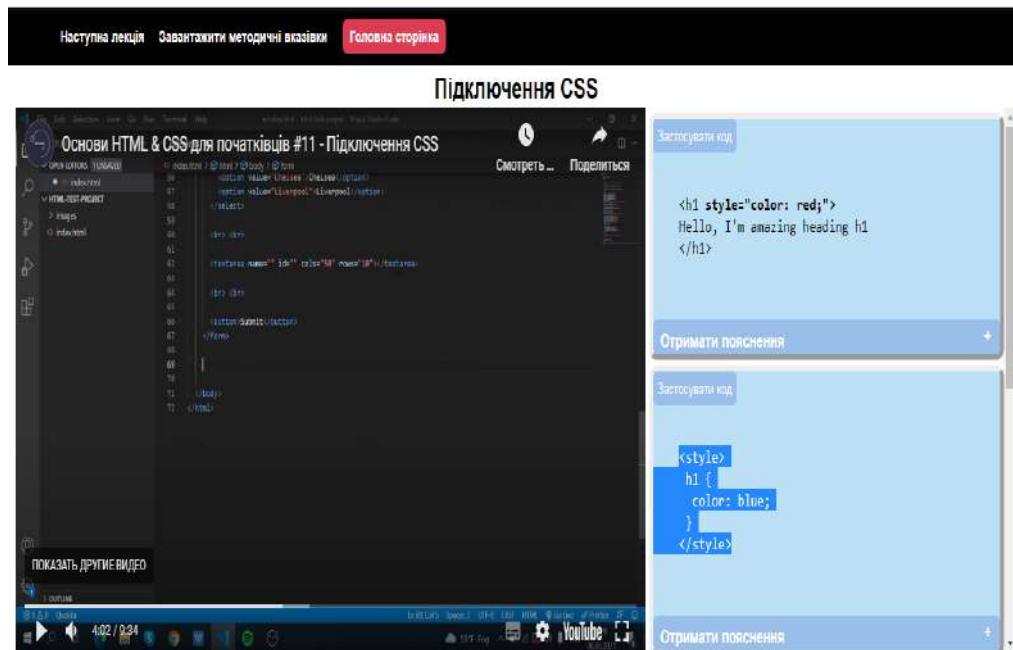


Рисунок 3.29 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з виділенням тексту анімаційного вікна для копіювання

Джерело: [Матеріал автору]

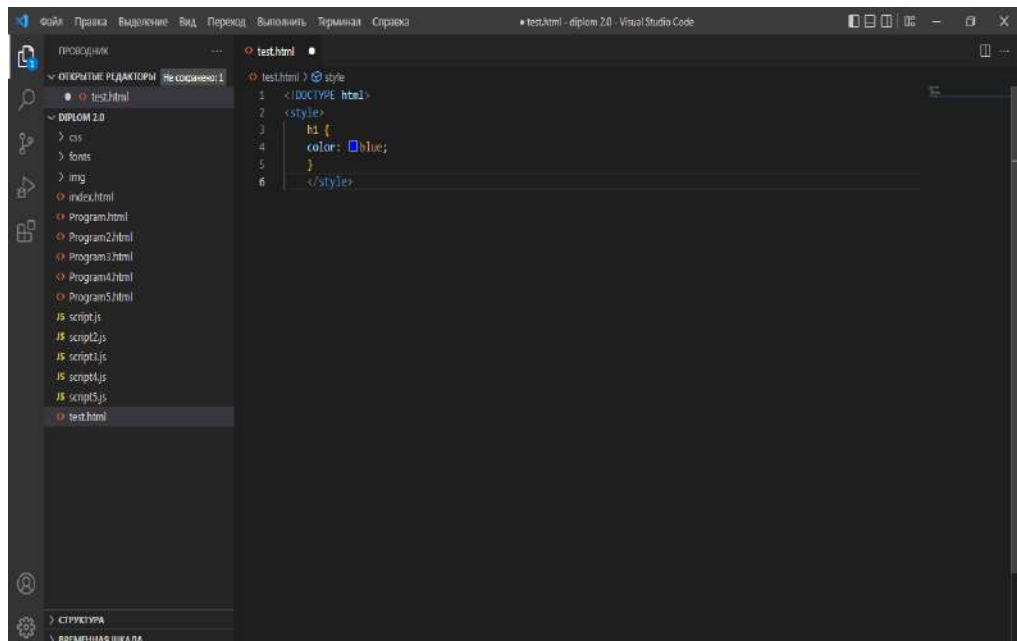


Рисунок 3.30 – HTML файл з тегами, що були скопійовані з анімаційного вікна поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

3.2.5 Навігація по відео-лекції за допомогою «анімаційних вікон» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Кожне анімаційне вікно взаємодіє з таймінгом відео-програвача, що дозволяє при натисканні на вікно, перевести відео рядок до місця в якому автор викладає відповідний навчальний матеріал (дивись рисунок 3.31 більш детальне зображення системи приведено в додатку Г). Таким чином можна здійснювати перехід між викладеними навчальними матеріалами точно та без витрати часу.

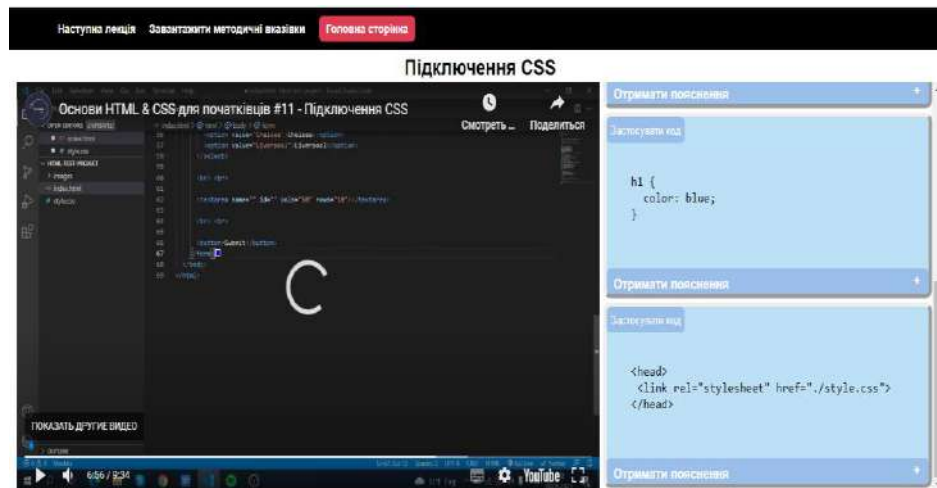


Рисунок 3.31 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з завантаженням відео-лекції при натисканні на «анімаційне вікно»

Джерело: [Матеріал автору]

Реалізація такої можливості здійснена за допомогою документа JavaScript, в якому реалізована перевірка взаємодії «анімаційного вікна» з таймінгом програвача, вище на рисунку 3.31 показана взаємодія «анімаційного вікна» та програвача, щодо появи вікна в заздалегідь обумовлений час, використавши цю функцію, було додано можливість переводити відео-рядок програвача на місце появи «анімаційного вікна», при натисканні на нього (на рисунку 3.32 показана частина коду JavaScript, весь код JavaScript сторінки приведено в додатку В).

```
block_examples.forEach((block_example) => {
    block_example.addEventListener("click", function (e) {
        e.preventDefault();
        playerTime = block_example.getAttribute('data-time-start')
        console.log("click time" + playerTime);
        player.seekTo(playerTime, true);
    });
});
```

Рисунок 3.32 – Частина коду JavaScript з реалізацією можливості переходити до тієї частини відео в якій була озвучена інформація зі змістом «анімаційного вікна», при натисканні на нього

Джерело: [Матеріал автору]

3.2.6 Організація роботи та функціонування «вікон з поясненнями» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

До кожного «анімаційного» вікна було додано можливість отримати пояснення змісту вікна, та інформації яку передає викладач. При натисканні на кнопку «Отримати пояснення», з'являється вікно з поясненнями змісту «анімаційного вікна» та роз'яснення інформації лектору, «вікно з поясненнями» розгортається у вигляді «вертикального акордеону», при натисканні на кнопку «Отримати пояснення» відео-програвач стає на паузу (дивись рисунок 3.33 більш детальне зображення системи приведено в додатку Б).

Такий підхід дозволяє отримувати додаткові навчальні матеріали без відриву від процесу навчання в системі.

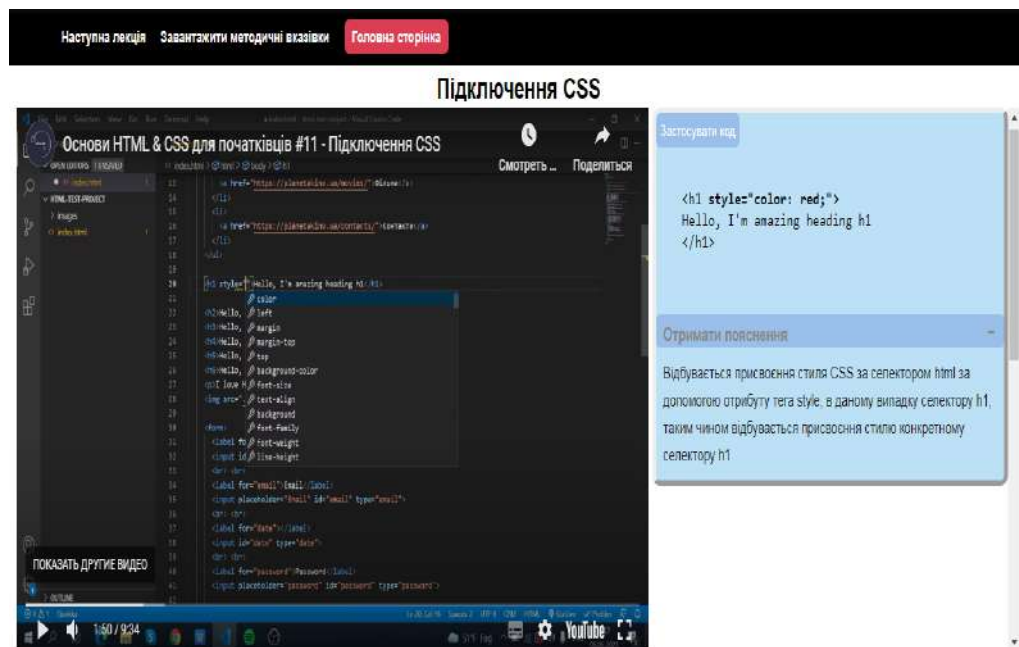


Рисунок 3.33 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з відкритим змістом «вікна з поясненнями»

Джерело: [Матеріал автору]

Елемент візуалізації Accordion в JavaScript (25) [24]– це вертикально складений набір заголовків. Кожен з яких може бути розгорнутий, для того щоб показати вміст, пов’язаної з ним інформації. Цей елемент отримав таку назву через схожість з однойменним музичним інструментом.

Для реалізації елементу візуалізації «вертикальний акордеон» була використана бібліотека Accordion.JS [26]. Accordion.JS – це бібліотека JavaScript, яка дозволяє створювати елементи візуалізації Accordion без написання власного коду. Вона базується на jQuery і має простий та легкий для використання інтерфейс.

До основного html документу сторінки Program1.html було додано JavaScript документ accordion.min.js з реалізацією Accordion (дивись рисунок 3.34).

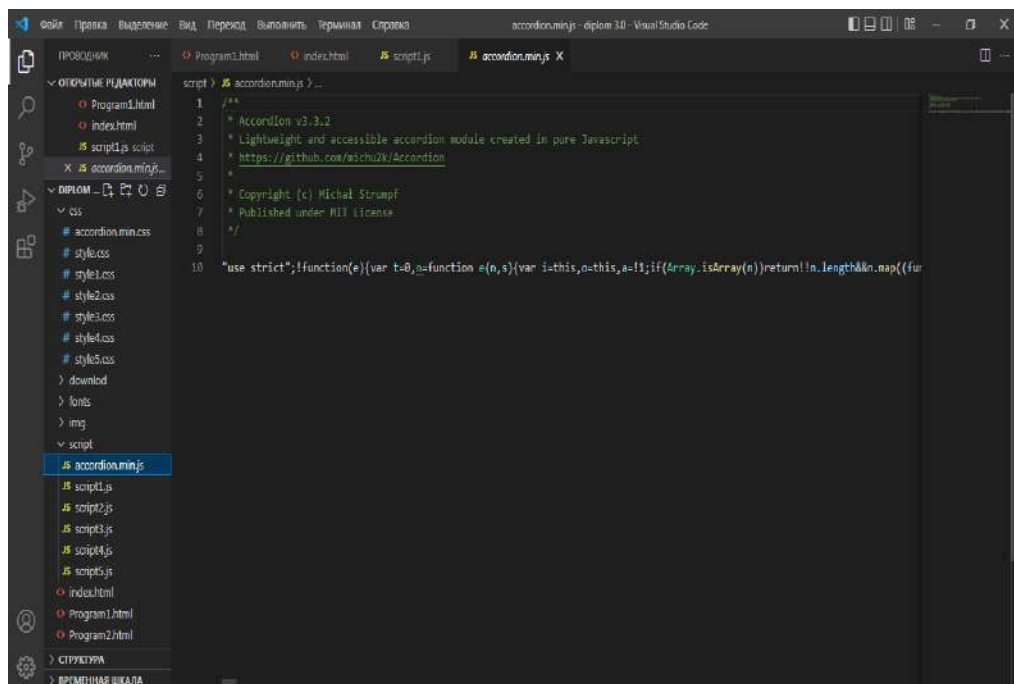


Рисунок 3.34 – Вміст JavaScript документ accordion.min.js з реалізацією Accordion

Джерело: [Матеріал автору]

Взаємодія основного JavaScript документ сторінки script1.js відбувається за допомогою функції ініціалізації `new Accordion('.accordion-container');`, що

була додана до основного JavaScript документ сторінки script1.js (на рисунку 3.35 приведена частина коду JavaScript, весь код JavaScript приведено в додатку В).

```
//begin
new Accordion('.accordion-container');
//end
```

Рисунок 3.35 – Частина коду JavaScript з функцією ініціалізації документу з бібліотеки Accordion.JS

Джерело: [Матеріал автору]

Також з бібліотеки Accordion.JS було додано CSS документ accordion.min.css, підключивши який були отримані стилі елементу візуалізації Accordion (дивись рисунок 3.36).

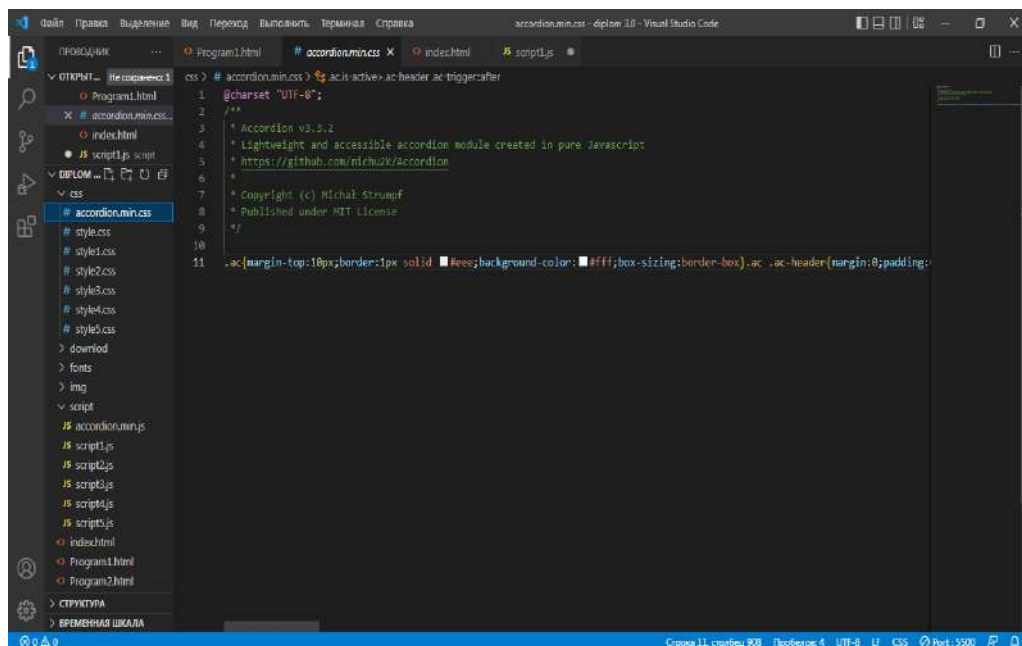


Рисунок 3.36 – Вміст CSS документу accordion.min.css зі реалізацією стилів для Accordion

Джерело: [Матеріал автору]

Вміст «вікна з поясненнями» задається в html документі сторінки Program1.html, у вигляді тексту, що прописується всередині тегу <p>, з класом ac-text, сам тег <p>, обернуто в тег <div> з класом ac-panel, далі в блоці на одному рівні з тегом <div> з класом ac-panel доданий тег заголовку другого рівня <h2> з класом ac-header в який помістили тег <button>(кнопка) з класом ac-trigger, через який реалізована можливість виклику «вікна пояснення». Усі вищеперераховані елементи коду було «обернуто» в тег <div> з класом pryklady accordion container. Через класи ac, реалізовано взаємодію JavaScript функцій, структури web-сторінки в html-документі та стилями web-сторінки в CSS-документі (на рисунку 3.37 приведена частина коду html з реалізацією структури «вертикального акордеону», вся html структура сторінки приведена в додатку А, на рисунку 3.38 приведена частина CSS правил, що до стилів елементу «вертикального акордеону», весь CSS-документ сторінки приведено в додатку Б).

```

<!-- акордеон -->
    <h2 class="ac-header">
        <button type="button" class="ac-trigger">Отримати
пояснення</button>
    </h2>
    <div class="ac-panel">
        <p class="ac-text">
            Відбувається присвоєння стиля CSS за селектором html за
            допомогою атрибуту тега style, в даному випадку селектору h1, таким чином
            відбувається присвоєння стилю конкретному селектору h1
        </p>
    </div>
</div>
<!-- акордеон -->

```

Рисунок 3.37 – Частина коду html зі структурою елементу Accordion

Джерело: [Матеріал автору]

```

.ac{
    margin-top:10px;
    border:1px solid #eee;
    background-color:#fff;
    box-sizing:border-box
}
.ac .ac-header{
    margin:0;
    padding:0
}
.ac .ac-trigger{
    font:bold 16px Arial,sans-serif;
    color:#111;
    text-align:left;
    width:100%;padding:10px 10px 10px 10px;
    display:block;
    cursor:pointer;
    background-color:transparent;
    transition:color .25s ease;
    position:relative;
    text-decoration:none;
    margin:0;
    border:0
}

```

Рисунок 3.38 – Частина коду CSS з правилами для елемента Accordion

Джерело: [Матеріал автору]

3.2.7. Принцип роботи та функціонування «вікон імітації web-сторінок» поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Для реалізації принципу ефективного засвоєння навчального матеріалу, до кожного «анімаційного вікна», була додана можливість отримати наглядну демонстрацію наслідків використання тих чи інших дії викладача, на «вікні імітації web-сторінки». Натиснувши на «анімаційному вікні» кнопку «Застосувати код», замість web-програвача з'являється модальне вікно, що імітує web-сторінку (на рисунку 3.39, зліва від блоку «анімаційних вікон», ми бачимо «вікно імітації web-сторінки», що з'явилося замість web-програвача, більш детальне зображення роботи «анімаційних вікон» приведено в додатку Г).



Рисунок 3.39 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з «вікном імітації web-сторінки»

Джерело: [Матеріал автору]

Реалізація «вікон імітації web-сторінок» виконано за допомогою елемента візуалізації модальне вікно.

Модальне вікно [27] – це елемент інтерфейсу, яке візуально представляє собою вікно, що впливає, та відображається над рештою сторінки. В момент коли вікно з'являється на сторінці, решта елементів на сторінці стають недоступними. Для того щоб звернути «вікно імітації web-сторінки» необхідно кликнути по будь-якій частині сторінки, крім самого вікна.

Виклик модального вікна може бути прив'язаний до різноманітних подій на сторінці, або до кнопки, в нашому випадку до кнопки на «Застосувати код».

Модальні вікна поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS), створені за допомогою JavaScript-документа, в якому прописані відповідні функції (на рисунку 3.40 показана частина коду JavaScript з реалізацією «модальних вікон», весь JavaScript-документ сторінки показаний в додатку В).

```

const btns = document.querySelectorAll('.btn');
const modalOverlay = document.querySelectorAll('.modal-overlay');
const modals = document.querySelectorAll('.modal');
btns.forEach((el) => {
  el.addEventListener('click', (e) => {
    let path = e.currentTarget.getAttribute('data-path');
    modals.forEach((el) => {
      el.classList.remove('modal--visible');
    });
    document.querySelector(`[data-target="${path}"]`).classList.add('modal--visible');
    document.querySelector(`[data-target="${path}"]`).parentNode.classList.add('modal-
overlay--visible');
  });
});
modalOverlay.forEach((el) => {
  el.addEventListener('click', (e) => {
    console.log(e.target);
    if (e.target == el) {
      el.classList.remove('modal-overlay--visible');
      modals.forEach((el) => {
        el.classList.remove('modal--visible');
      });
    }
  });
});
});

```

Рисунок 3.40 – Частина коду JavaScript з реалізацією «модального вікна імітації web-сторінки» поліпшеної системи навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Оформлення «вікон імітації web-сторінок» здійснюється за допомогою стилів CSS, що прописані в CSS-документі сторінки. Також за допомогою класів відбувається взаємодія JavaScript та html (на рисунку 3.41 показана частина CSS-документу з CSS-правилами для реалізації «модальних вікон», весь JavaScript-документ сторінки показано в додатку Б).

```

.btn {
  display: inline-block;
  align-items: center;
  background-color: #92b8e9d0;
  height: 30px;
  border-radius: 6px;
  font-family: 'Raleway', Helvetica, Arial, sans-serif;
  color: white;
  border: 0;
}
.modal-overlay {
  position: fixed;
  left: 0;
  top: 0;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  width: 100%;
  opacity: 0;
  visibility: hidden;
  transition: all 0.3s ease-in-out;
}

```

Рисунок 3.41 – Частина коду CSS зі стилями «модального вікна імітації web-сторінки» поліпшеної системи навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

За допомогою data-атрибутів були додані значення на html елементи сторінок, в даному випадку data-path та data-target. Також за допомогою html, де за допомогою тегів з відповідними класами, а також тегів, що дублюють структуру сторінки викладача, були створена структура «вікон імітації web-сторінок» (на рисунку 3.42 показана частина html структури з «модальним вікном» та їх наповненням для імітації web-сторінки автору, весь JavaScript-документ сторінки приведено в додатку В).

```

<!-- імітація сторінки -->
<button class="btn" data-path="one">Застосувати код</button>
<div class="modals">
  <div class="modal-overlay">
    <div class="modal modal--1" data-target="one">
      <ul>
        <li><a href="Зозклад"></a>Зозклад</li>
      </ul>
      <ul>
        <li><a href="Фільми"></a>Фільми</li>
      </ul>
      <ul>
        <li><a href="Контакти"></a>Контакти</li>
      </ul>
      <h1 style="color: red;">Hello, I'm amazing heading h1</h1>
      <h2>Hello, I'm amazing heading h2</h2>
      <h3>Hello, I'm amazing heading h3</h3>
      <h4>Hello, I'm amazing heading h4</h4>
      <h5>Hello, I'm amazing heading h5</h5>
      <h6>Hello, I'm amazing heading h6</h6>
      <p>I love HTML</p>
      
    </div>
  </div>
</div>
<!-- імітація сторінки -->

```

Рисунок 3.42 – Частина коду html з реалізацією структури «модального вікна імітації web-сторінки» поліпшеної системи навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

Стилі, які демонструє викладач на своїй сторінці, додаються до відповідних html елементів за допомогою класів, або id в CSS-документі, що дублюють стилі та властивості, які додає викладач до свого проекту (на рисунку 3.43 показано частину CSS правил, які імітують стилі на web-сторінці автору, весь CSS-документ сторінки приведено в додатку Б).

```

/* study */
.test {
    color: red;
}
#heading {
    color: red;
}
#test {
    color: red;
}
/* study */

```

Рисунок 3.43 – Частина коду CSS зі стилями що надають властивості структурі html, для імітації дії викладача в «модальному вікні імітації web-сторінки» поліпшеної системи навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

3.2.8. Взаємодія «вікон з поясненнями» та «вікон імітації web-сторінок» з відео-програвачем поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

Для зручності користування поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS), кожному «вікну з поясненнями» та «вікну імітації web-сторінок», була додана функція завдяки якій відбувається постановка відео-лекції на паузу при натисканні відповідних кнопок на «анімаційному вікні», тобто кнопок: «Отримати пояснення» та «Застосувати код». Як бачимо на рисунку 3.44, при відкритому «вікні пояснень» відео-лекція стала на паузу, про що свідчить відповідна позначка на панелі відео-плеєра. Так само на рисунку 3.45 бачимо, що при відкритому «вікні імітації web-сторінки» відео-лекція так само стоїть на паузі.

Такий підхід дозволяє зосередитися на вивченні додаткових матеріалів, без відриву від відео-лекцій.

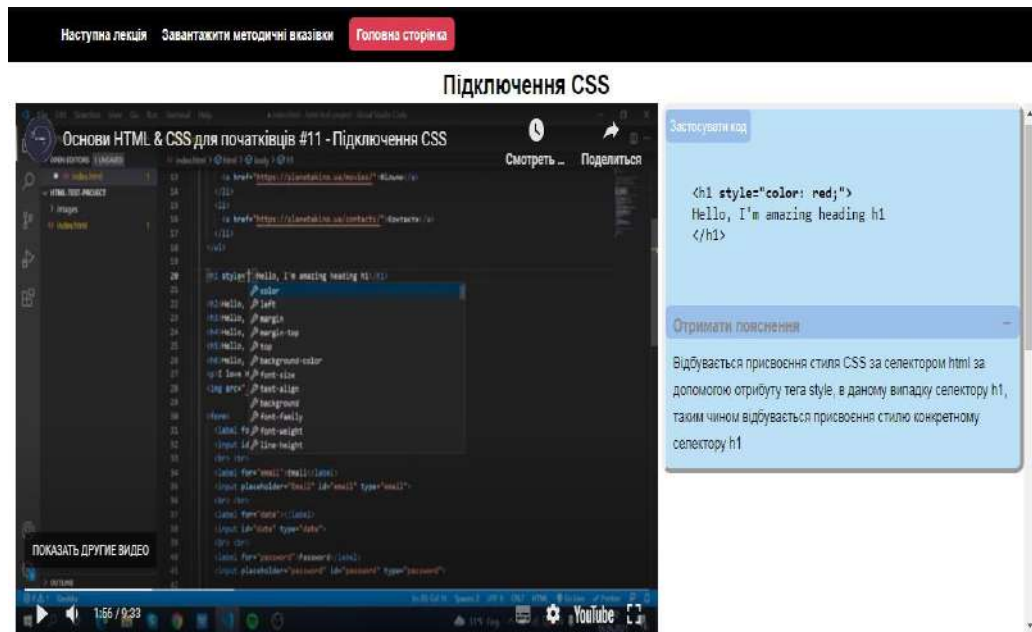


Рисунок 3.44 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з відкритим «вікном пояснення», відео стоїть на паузі

Джерело: [Матеріал автору]

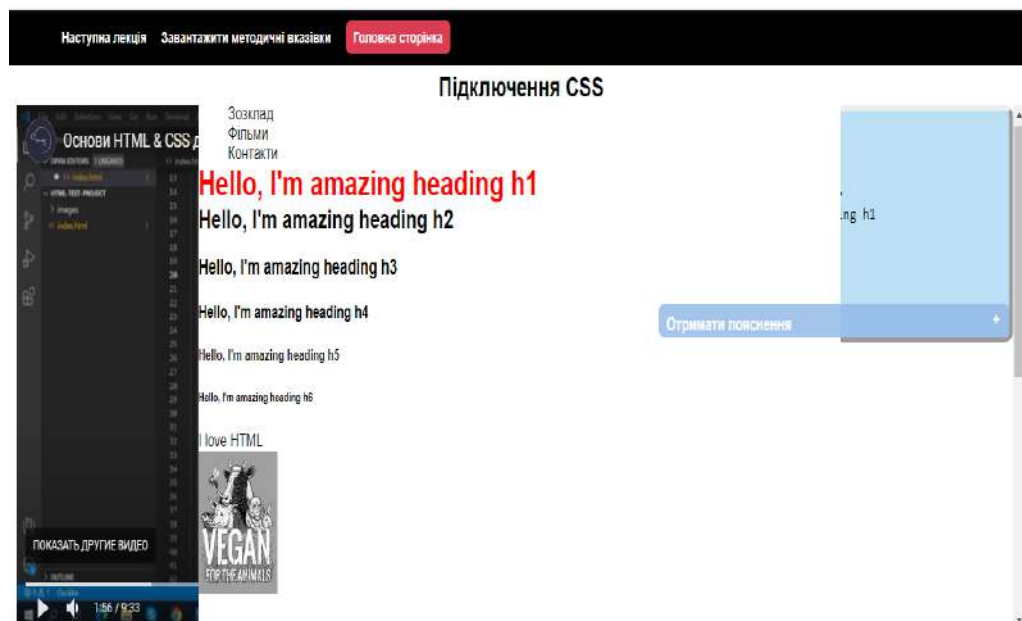


Рисунок 3.45 – Сторінка з відео-лекцією поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) з відкритим «вікном імітації web-сторінки», відео стоїть на паузі

Джерело: [Матеріал автору]

Реалізація можливості зупиняти відео-лекцію при натисненні кнопок: «Отримати пояснення» та «Застосувати код», відбувається за допомогою функції JavaScript, які прописані до JavaScript-документу сторінки (на рисунку 3.46 показана частина коду JavaScript з реалізацією постановки на паузу відео, весь JavaScript-документ приведено в додатку В).

```
let player;
time_update_interval = 0;
function onYouTubeIframeAPIReady() {
    makePause(player);
    makeNewPause(player);
}
function makePause(player) {
    let pauseButton = document.querySelectorAll('.ac-trigger');
    pauseButton.forEach(button => button.addEventListener('click', function (e) {
        player.pauseVideo();
    }));
}
function makeNewPause(player) {
    let pauseButton = document.querySelectorAll('.btn');
    pauseButton.forEach(button => button.addEventListener('click', function (e) {
        player.pauseVideo();
    }));
}
```

Рисунок 3.46 – Частина коду JavaScript з реалізацією можливості поставити на паузу відео при натисканні відповідних кнопок поліпшеної системи навчання мові каскадних таблиць стилів (CSS)

Джерело: [Матеріал автору]

3.3. Переваги поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS)

До переваг поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS) відноситься:

- наявність та активне використання анімації та візуалізації при викладанні навчального матеріалу («анімаційні вікна», «вікна пояснень», «вікна імітації web-сторінки»);
- наявність «анімаційних вікон» дозволяє бачити дії викладача, та копіювати його текст, таким чином ми можемо повторювати усі шаги викладача у своєму редакторі коду паралельно з ним, не має необхідності переходити між режимами роботи екрану програвача;
- наявність «вікон з поясненнями» дозволяє отримувати теоретичні знання без відриву від відео-лекцій;
- наявність «вікон з імітацією web-сторінки» дозволяє порівняти результати роботи викладача зі своїми результатами;
- модульна структура системи навчання – робить її більш гнучкою та ефективною кожен модуль може бути змінений, оновлений та вдосконалений, незалежно від інших модулів.;
- доступність системи для користувачів, оскільки поліпшена система навчання мові каскадних таблиць стилів (CSS) являє собою web-сторінку, доступ до неї може отримати будь який користувач з можливістю використання гаджета, що дозволяє використання браузера, та можливість переглядати відео-матеріали;
- тривалість відео-лекцій не перевищує 20 хвилин.

3.4 Висновки за розділом

Цей розділ присвячено розробці та створенню поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

В розділі описані інструменти створення системи, надана структура поліпшеної системи та представлена схема взаємодії користувача з нею.

Описані функціональні можливості системи та представлені засоби та рішення за допомогою яких ці можливості здійснюються.

Описані засоби надання навчальних матеріалів та представлені фрагменти коду HTML, CSS, JavaScript за допомогою яких побудована структура системи, надані стилі та візуальні особливості системи, представлені фрагменти коду за допомогою яких здійснюється функціонування супутньої анімації навчальних матеріалів.

Визначені основні переваги поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

ВИСНОВКИ

В ході виконання роботи був проведений порівняльний аналіз існуючих систем дистанційного навчання мові каскадних таблиць стилів (CSS), були визначені переваги та недоліки цих систем.

З урахуванням переваг та недоліків існуючих систем дистанційного навчання мові каскадних таблиць стилів (CSS), були розроблені рекомендації для побудови поліпшеної системи дистанційного навчання мові каскадних таблиць стилів (CSS).

На основі рекомендацій була розроблена та створена поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS), яка представлена в даній роботі.

Поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS) створена у вигляді web-сайту, де в якості супутньої анімації використовуються «анімаційні вікна», «вікна з поясненнями» та «вікна з імітацією web-сторінки».

Сама поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS) має модульну структуру.

Такий підхід робить систему навчання більш гнучкою та ефективною: кожен модуль може бути змінений, оновлений та вдосконалений, незалежно від інших модулів. В окремий модуль з відео-лекцією, можна помістити будь-який лекційний матеріал відповідного формату. Кількість сторінок з відео-лекціями в системі навчання не обмежена, оскільки до кожної сторінки з відео-лекцією належать власні html, CSS, JavaScript документи, ми можемо змінювати зміст цих документів в залежності від теми лекції, змінюючи тільки id відео. В модулі з анімацією відео матеріалів необхідно буде змінити тільки кількість та зміст «анімаційних вікон», «вікон з поясненнями» та «вікон імітації web-сторінки», загальна структура документів зміни не потребує та

буде функціонувати з будь-якою кількістю «анімаційних вікон» та будь якими відео-лекціями по темах пов'язаними з розробкою web-сторінок.

Враховуючи вищевикладене можна зробити висновок, що створена в даній роботі поліпшена система дистанційного навчання мові каскадних таблиць стилів (CSS) відповідає розробленим рекомендаціям, та позбавлена недоліків, що притаманні сучасним системам дистанційного навчання мові каскадних таблиць стилів (CSS).

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Університет, Одеський Національний Політехнічний. Каскадні таблиці стилів. StudFiles. [Електроний ресурс] Університет, Одеський Національний Політехнічний. <https://studfile.net/preview/5172671/>.
2. Що таке CSS. css.in.ua. [Електроний ресурс] https://css.in.ua/article/shcho-take-html_10.
3. Дистанційне навчання. Wikipedia . [Електроний ресурс] https://uk.wikipedia.org/wiki/%D0%94%D0%B8%D1%81%D1%82%D0%B0%D0%BD%D1%86%D1%96%D0%B9%D0%BD%D0%B5_%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F
4. Выбор и адаптация систем дистанционного обучения. e-learning. [Електроний ресурс] https://sites.google.com/site/varart78/home/lms_compare.
5. Усвоение знаний: как правильно донести информацию. Unicraft. [Електроний ресурс] <https://www.unicraft.org/blog/1341/kak-uluchshit-process-usvoenia-znani/>.
6. М.В., Ежова Ю.В. Пац. YouTube как обучающий ресурс. CYBERLENINCA. [Електроний ресурс] <https://cyberleninka.com/article/n/youtube-kak-obuchayuschiy-resurs-inostranny-yazyk-neyazykovoy-vuz/viewer>.
7. YouTube. WikipediA. [Електроний ресурс] <https://uk.wikipedia.org/wiki/YouTube>.
8. Обзор популярных школ программирования: личный опыт, плюсы, минусы и фейлы. Хекслет. [Електроний ресурс] <https://com.hexlet.io/blog/posts/obzor-populyarnyh-shkol-programmirovaniya-lichnyy-opyt-plyusy-minusy-i-feily#kak-ya-uchilsya-na-freecodecamp>.
9. Learn to code — for free. freeCodeCamp. [Електроний ресурс] <https://www.freecodecamp.org/>.
10. freeCodeCamp. WikipediA. [Електроний ресурс] <https://en.wikipedia.org/wiki/FreeCodeCamp>.

11. Бесплатные курсы образования. Code Basics. [Электроний ресурс]
<https://code-basics.com/com>.
12. Code Basics: Тренинги. EDUZORRO. [Электроний ресурс]
https://eduzorro.com/online_courses/code-basics/.
13. Skillbox . WikipediA. [Электроний ресурс]
<https://ua.wikipedia.org/wiki/Skillbox>.
14. SkillBox: Веб-разработка. SkillBox. [Электроний ресурс]
<https://skillbox.com/courses/>.
15. GeekBrains. WikipediA. [В Интернете] <https://ua.wikipedia.org/wiki/GeekBrains#:>.
16. GeekBrains - образование в ИТ. GeekBrains. [Электроний ресурс]
<https://gb.com/>.
17. MateAcademy – Онлайн академія ІТ професій. MateAcademy. [Электроний ресурс]
[https://mate.academy/?utm_source=google&utm_medium=cpc&utm_term=
=
mate%20academy&utm_content=587604768452&utm_campaign=gs_brand_ua_m
xd&gad=1&gclid=CjwKCAjwrpOiBhBVEiwA_473dAOHb9FWB1J5cI7Y_0XV
X4VCbmMqR-2Cb_nhGWXEQJ3hi7VAafau0hoCA_IQAvD_BwE](https://mate.academy/?utm_source=google&utm_medium=cpc&utm_term=&utm_content=587604768452&utm_campaign=gs_brand_ua_mx&gad=1&gclid=CjwKCAjwrpOiBhBVEiwA_473dAOHb9FWB1J5cI7Y_0XVX4VCbmMqR-2Cb_nhGWXEQJ3hi7VAafau0hoCA_IQAvD_BwE).
18. Visual Studio Code. WikipediA. [Электроний ресурс]
https://uk.wikipedia.org/wiki/Visual_Studio_Code.
19. HTML. WikipediA. [Электроний ресурс]
<https://uk.wikipedia.org/wiki/HTML>.
20. JavaScript. WikipediA. [Электроний ресурс]
[https://uk.wikipedia.org/wiki/Java Script](https://uk.wikipedia.org/wiki/Java_Script).
21. Савчук, Віталій. SavchukIT. YouTube. [Электроний ресурс]
<https://www.youtube.com/@savchukit1454>.
22. Flexbox. MDN Plus. [Электроний ресурс] [https://developer.
mozilla.org/com/docs/Learn/CSS/CSS_layout/Flexbox](https://developer.mozilla.org/com/docs/Learn/CSS/CSS_layout/Flexbox).
23. YouTube IFrame Player API. YouTube Player API Reference for iframe Embeds. [Электроний ресурс] <https://developers.google.com/youtube>

[/iframe_api_reference?hl=com#:~:text=API%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B3%D1%80%D1%8B%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F%20IFrame%20%D0%BF%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D1%8F%D0%B5%D1%82%20%D0%B2%D1%81%D1%82%D1%80%D0%B.](#)

24. Dataset Простейший способ хранить данные в HTML и читать их из JavaScript. doka.guide. [Электронный ресурс] <https://doka.guide/js/element-dataset/>.

25. Как создать вертикальный аккордеон для сайта. ИТШЕФ. [Электронный ресурс] <https://itchief.ru/javascript/accordion>.

26. Github. [Электронный ресурс] <https://github.com/michu2k/Accordion>.

27. ИТШЕФ. Модальное окно для сайта на чистом CSS и JavaScript. [Электронный ресурс] <https://itchief.com/javascript/modal-window>.

Додаток А

Зміст html-документу однієї зі сторінок з відео-лекціями поліпшеної системи
дистанційного навчання мові каскадних таблиць стилів (CSS)

```
<!DOCTYPE html>
<html lang="ua">

<head>
  <meta charset="utf-8">
  <title>Програма купса</title>
  <link rel="stylesheet" href="./css/accordion.min.css">
  <link rel="stylesheet" href="./css/style1.css">
</head>

<body>
  <header class="header">
    <ul class="nav-list">
      <li>
        <a class="nav-list-link" href="./Program2.html">Наступна лекція</a>
      </li>
      <li>
        <a class="nav-list-link" href="./downlod/МЕТОДИЧНІ МАТЕРІАЛИ
1.docx">Завантажити методичні вказівки</a>
      </li>
      <li>
        <a href="index.html">Головна сторінка</a>
      </li>
    </ul>
  </header>

  <div class="main-content">
    <h1>
      <a href="" class="second">Підключення CSS</a>
    </h1>
  </div>
  <div class="description__block">
    <div id="ytplayer"></div>
    <div id="video-placeholder" class="video"></div>
    <script src="https://www.youtube.com/iframe_api"></script>
    <span class="timer" id="current-time">0:00</span>

    <div class="pryklady accordion-container">
      <!-- БІКНО -->
      <div class="ac upr upr_98" data-time-start="98">
        <!-- імітація сторінки -->
        <button class="btn" data-path="one">Застосувати код</button>
        <div class="modals">
          <div class="modal-overlay">
```

```

<div class="modal modal--1" data-target="one">
  <ul>
    <li><a href="Зозклад"></a>Зозклад</li>
  </ul>
  <ul>
    <li><a href="Фільми"></a>Фільми</li>
  </ul>
  <ul>
    <li><a href="Контакти"></a>Контакти</li>
  </ul>

  <h1 style="color: red;">Hello, I'm amazing heading h1</h1>
  <h2>Hello, I'm amazing heading h2</h2>
  <h3>Hello, I'm amazing heading h3</h3>
  <h4>Hello, I'm amazing heading h4</h4>
  <h5>Hello, I'm amazing heading h5</h5>
  <h6>Hello, I'm amazing heading h6</h6>
  <p>I love HTML</p>
  
</div>
</div>
</div>
<!-- імітація сторінки -->
<!-- окно -->
<div class="code-block" data-time-start="98">
  <pre>
    <code class="language-css">
&lth1 <b>style="color: red;"&gt;</b>
Hello, I'm amazing heading h1
&lt;/h1&gt;
    </code>
  </pre>
</div>
<!-- вікно -->
<!-- акордеон -->
<h2 class="ac-header">
  <button type="button" class="ac-trigger">Отримати
пояснення</button>
</h2>
<div class="ac-panel">
  <p class="ac-text">
    Відбувається присвоєння стиля CSS за селектором html за
    допомогою атрибуту тега style, в даному випадку селектору h1, таким чином
    відбувається присвоєння стилю конкретному селектору h1
  </p>
</div>
</div>
<!-- акордеон -->
<!-- ВІКНО -->
<div class="ac upr upr_242" data-time-start="242">
  <button class="btn" data-path="two">Застосувати код</button>

```

```

<div class="modals">
  <div class="modal-overlay">
    <div class="modal modal--2" data-target="two">
      <ul>
        <li><a href="Зозклад"></a>Зозклад</li>
      </ul>
      <ul>
        <li><a href="Фільми"></a>Фільми</li>
      </ul>
      <ul>
        <li><a href="Контакти"></a>Контакти</li>
      </ul>
      <h1 style="color: blue;">Hello, I'm amazing heading h1</h1>
      <h2>Hello, I'm amazing heading h2</h2>
      <h3>Hello, I'm amazing heading h3</h3>
      <h4>Hello, I'm amazing heading h4</h4>
      <h5>Hello, I'm amazing heading h5</h5>
      <h6>Hello, I'm amazing heading h6</h6>
      <p>I love HTML</p>
      
    </div>
  </div>
</div>
<div class="code-block" data-time-start="242">
  <pre>
    <code class="language-css">
<style>
  h1 {
    color: blue;
  }
</style>
    </code>
  </pre>
</div>
<h2 class="ac-header">
  <button type="button" class="ac-trigger">Отримати
пояснення</button>
</h2>
<div class="ac-panel">
  <p class="ac-text">
    Відбувається присвоєння стиля CSS за тегом style, в даному
    випадку селектору h1, таким чином відбувається присвоєння стилю усім
    селекторам h1 в документі</p>
  </div>
</div>

<div class="ac upr upr_416" data-time-start="416">
  <button class="btn" data-path="three">Застосувати код</button>
  <div class="modals">
    <div class="modal-overlay">
      <div class="modal modal--3" data-target="three">

```

```

    <ul>
      <li><a href="Зозклад"></a>Зозклад</li>
    </ul>
    <ul>
      <li><a href="Фільми"></a>Фільми</li>
    </ul>
    <ul>
      <li><a href="Контакти"></a>Контакти</li>
    </ul>
    <h1>Hello, I'm amazing heading h1</h1>
    <h2>Hello, I'm amazing heading h2</h2>
    <h3>Hello, I'm amazing heading h3</h3>
    <h4>Hello, I'm amazing heading h4</h4>
    <h5>Hello, I'm amazing heading h5</h5>
    <h6>Hello, I'm amazing heading h6</h6>
    <p>I love HTML</p>
    
  </div>
</div>
</div>
<div class="code-block" data-time-start="416">
  <pre>
    <code class="language-css">
h1 {
  color: blue;
}
    </code>
  </pre>
</div>

  <h2 class="ac-header">
    <button type="button" class="ac-trigger">Отримати
пояснення</button>
  </h2>
  <div class="ac-panel">
    <p class="ac-text">
      Відбувається присвоєння стиля за допомогою CSS документа до
селектора html, в даному випадку селектору h1, такий спосіб підключення
стилю також присвоює стиль усім селекторам h1 в документі. спрацьовує такий
спосіб тільки при підключенні відповідного css документа, до html
документа.</p>
    </div>
  </div>

<div class="ac upr upr_478" data-time-start="478">
  <button class="btn" data-path="four">Застосувати код</button>
  <div class="modals">
    <div class="modal-overlay">
      <div class="modal modal--4" data-target="four">
        <ul>
          <li><a href="Зозклад"></a>Зозклад</li>

```

```

    </ul>
    <ul>
      <li><a href="Фільми"></a>Фільми</li>
    </ul>
    <ul>
      <li><a href="Контакти"></a>Контакти</li>
    </ul>
    <h1 style="color: blue;">Hello, I'm amazing heading h1</h1>
    <h2>Hello, I'm amazing heading h2</h2>
    <h3>Hello, I'm amazing heading h3</h3>
    <h4>Hello, I'm amazing heading h4</h4>
    <h5>Hello, I'm amazing heading h5</h5>
    <h6>Hello, I'm amazing heading h6</h6>
    <p>I love HTML</p>
    
  </div>
</div>
</div>
<div class="code-block" data-time-start="478">
  <pre>
    <code class="language-css">
&lt;thead&gt;
  &lt;link rel="stylesheet" href="./style.css"&gt;
&lt;/thead&gt;
    </code>
  </pre>
</div>
<h2 class="ac-header">
  <button type="button" class="ac-trigger">Отримати
пояснення</button>
</h2>
<div class="ac-panel">
  <p class="ac-text">
    Таким чином відбувається підключення CSS документа зі стилями
до html документу, після підключення ми бачимо що стиль css підключається
до селектора h1 </p>
  </div>
</div>
</div>
</div>
</body>
<script src="https://code.jquery.com/jquery-3.6.3.min.js"
  integrity="sha256-pvPw+upLPUjgMXy0G+800xUf+/Im1MZjXxxgOcBQBXU="
crossorigin="anonymous"></script>
<script src="./script/accordion.min.js"></script>
<script src="./script/script1.js"></script>
</html>

```

Додаток Б

Зміст CSS-документу однієї зі сторінок з відео-лекціями поліпшеної системи
дистанційного навчання мові каскадних таблиць стилів (CSS)

```
:root {
  --bg-color: rgb(37, 185, 185);
  --bg-block: honeydew;
  --border-block-color: ghostwhite;
}

body {
  margin: 0;
  font-family: 'Raleway', Helvetica, Arial, sans-serif;
}

ul {
  margin: 0;
  list-style-type: none;
}

a {
  text-decoration: none;
  color: inherit;
}

.header {
  display: flex;
  align-items: center;
  height: 50px;
  background-color: black;
  justify-content: space-between;
  padding: 0 20px;
}

.nav-list {
  display: flex;
  align-items: center;
}

.nav-list li {
  padding: 0 10px;
  color: white;
  font-weight: bold;
  font-size: 14px;
}

.nav-list li:last-of-type {
  padding-right: 0px;
}
```

```

}

.nav-list-link {
  position: relative;
  transition: all .25s ease;
}

.nav-list-link::after {
  content: "";
  background: #17baaa;
  height: 2px;
  width: 100%;
  position: absolute;
  bottom: -8px;
  left: 0;
  transform: scaleX(0);
  transition: all .25s ease;
}

.nav-list-link:hover {
  color: #17baaa;
}

.nav-list-link:hover::after {
  transform: scaleX(1);
}

.nav-list li:last-of-type a {
  display: flex;
  align-items: center;
  background-color: #dd3d53;
  height: 30px;
  padding: 0 10px;
  border-radius: 6px;
  transition: all 250ms ease;
}

.nav-list li:last-of-type a:hover {
  background-color: #c9233a;
}

.nav-list li:last-of-type a svg {
  width: 1em;
}
p,
h1,
h2 {
  margin: 0;
  bottom: 0;
}

```

```
.main-content {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  height: auto;
}

.pryklady {
  flex-wrap: wrap;
  margin-left: 5px;
  margin-right: 5px;
  gap: 25px;
  width: 70%;
  height: 78vh;
  overflow-y: scroll;
  scroll-behavior: smooth;
  overflow-x: hidden;
}

.upr {
  width: 100%;
  margin: 5px;
}

.second {
  padding-left: 5px;
  font-size: 25px;
  color: black;
}

.btn {
  display: inline-block;
  align-items: center;
  background-color: #92b8e9d0;
  height: 30px;
  border-radius: 6px;
  font-family: 'Raleway', Helvetica, Arial, sans-serif;
  color: white;
  border: 0;
}

.modal-overlay {
  position: fixed;
  left: 0;
  top: 0;
  display: flex;
  align-items: center;
  justify-content: center;
  height: 100vh;
  width: 100%;
```

```

    opacity: 0;
    visibility: hidden;
    transition: all 0.3s ease-in-out;
}

.modal {
    width: 860px;
    height: 480px;
    display: flex;
    align-items: center;
    justify-content: center;
    display: none;
    background-color: white;
    font-family: 'Raleway', Helvetica, Arial, sans-serif;
    margin-right: 500px;
    margin-top: 40px;
}

.modals--visible {
    display: block;
}

.modal-overlay--visible {
    opacity: 1;
    visibility: visible;
    transition: all 0.3s ease-in-out;
}

.modal--visible {
    display: block;
}

.ac .ac-trigger {
    display: inline-block;
    align-items: center;
    background-color: #92b8e9d0;
    height: 30px;
    border-radius: 6px;
    font-family: 'Raleway', Helvetica, Arial, sans-serif;
    color: white;
    border: 0;
}

.upr1_98,
.pryklady>div:nth-child(1) {
    border-color: var(--border-block-color);
    border: 5px outset;
    margin-bottom: 5px;
    background-color: rgb(188, 224, 245);
    font-size: 16px;
    border-radius: 10px;
}

```

```

}

.upr2_242,
.pryklady>div:nth-child(2) {
  border-color: var(--border-block-color);
  border: 5px outset;
  margin-bottom: 5px;
  background-color: rgb(188, 224, 245);
  font-size: 16px;
  border-radius: 10px;
}

.upr3_416,
.pryklady>div:nth-child(3) {
  border-color: var(--border-block-color);
  border: 5px outset;
  margin-bottom: 5px;
  background-color: rgb(188, 224, 245);
  font-size: 16px;
  border-radius: 10px;
}

.upr4_478,
.pryklady>div:nth-child(4) {
  border-color: var(--border-block-color);
  border: 5px outset;
  margin-bottom: 5px;
  background-color: rgb(188, 224, 245);
  font-size: 16px;
  border-radius: 10px;
}

[class*="upr"] {
  opacity: 0;
  transition: opacity 0.4s ease-in-out;
}

.upr-active {
  opacity: 1;
  transition: opacity 1.0s ease-in-out;
}

.description__block {
  display: flex;
  flex-direction: row;
  align-items: center;
}

.video {
  width: 120vw;
  min-height: 35vw;
}

```

```
height: auto;
-webkit-size: cover;
-moz-size: cover;
-o-size: cover;
size: cover;
margin-left: 10px;
}

.timer {
display: none;
height: 0;
width: 0;
}
```

Додаток В

Зміст JavaScript-документу однієї зі сторінок з відео-лекціями поліпшеної системи
дистанційного навчання мові каскадних таблиць стилів (CSS)

```

let player;
time_update_interval = 0;
function onYouTubeIframeAPIReady() {
    player = new YT.Player('video-placeholder', {
        videoId: 'HrP4qGazzE0',
        playerVars: {
            color: 'white'
        },
        events: {
            onReady: initialize
        }
    });

    makePause(player);

    makeNewPause(player);
}

function makePause(player) {
    let pauseButton = document.querySelectorAll('.ac-trigger');
    pauseButton.forEach(button => button.addEventListener('click', function
(e) {
        player.pauseVideo();
    })))
}

function makeNewPause(player) {
    let pauseButton = document.querySelectorAll('.btn');
    pauseButton.forEach(button => button.addEventListener('click', function
(e) {
        player.pauseVideo();
    })))
}

function initialize() {
    updateTimerDisplay();
    clearInterval(time_update_interval);
    time_update_interval = setInterval(function () {
        updateTimerDisplay();
    }, 1000)
}

function updateTimerDisplay() {
    let currentTime = document.querySelector('#current-time');
    $('#current-time').text(formatTime(player.getCurrentTime()));
}

```

```

$('#duration').text(formatTime(player.getDuration()));

const block_examples = document.querySelectorAll('.code-block');

const uprs = document.querySelectorAll('.upr');
uprs.forEach(upr => visualizationCode(upr, player.getCurrentTime()))

block_examples.forEach((block_example) => {
    block_example.addEventListener("click", function (e) {
        e.preventDefault();
        playerTime = block_example.getAttribute('data-time-start')
        console.log("click time" + playerTime);
        player.seekTo(playerTime, true);
    });
});

}

function formatTime(time) {
    time = Math.round(time);
    var minutes = Math.floor(time / 60),
        seconds = time - minutes * 60;
    seconds = seconds < 10 ? '0' + seconds : seconds;
    return minutes + ":" + seconds;
}

function visualizationCode(upr, currentTime) {
    let timeStart = upr.getAttribute('data-time-start');
    if (currentTime > timeStart) {
        if (!upr.classList.contains('upr-active')) { upr.scrollIntoView();
    }
        upr.classList.add('upr-active');
    }
}

new Accordion('.accordion-container');
const btns = document.querySelectorAll('.btn');
const modalOverlay = document.querySelectorAll('.modal-overlay');
const modals = document.querySelectorAll('.modal');

btns.forEach((el) => {
    el.addEventListener('click', (e) => {
        let path = e.currentTarget.getAttribute('data-path');
        modals.forEach((el) => {
            el.classList.remove('modal--visible');
        });
        document.querySelector(`[data-
target="${path}"]`).classList.add('modal--visible');
        document.querySelector(`[data-
target="${path}"]`).parentNode.classList.add('modal-overlay--visible');
    });
});

```

```
modalOverlay.forEach((el) => {  
  el.addEventListener('click', (e) => {  
    console.log(e.target);  
    if (e.target == el) {  
      el.classList.remove('modal-overlay--visible');  
      modals.forEach((el) => {  
        el.classList.remove('modal--visible');  
      });  
    }  
  });  
});
```

ДОДАТОК Г

ПРЕЗЕНТАЦІЯ ПОЛІПШЕНОЇ СИСТЕМА ДИСТАНЦІЙНОГО НАВЧАННЯ

МОВІ КАСКАДНИХ ТАБЛИЦЬ СТИЛІВ (CSS)

Система дистанційного навчання мові каскадних таблиць стилів (CSS)

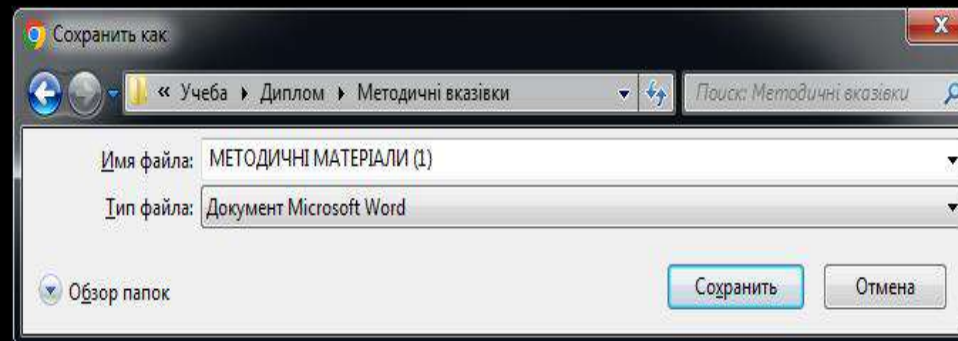
[Завантажити методичні вказівки](#)

[Відео лекції](#)

Система дистанційного навчання мові каскадних таблиць стилів (CSS)

Завантажити методичні вказівки

Відео лекції



[Наступна лекція](#)[Завантажити методичні вказівки](#)[Головна сторінка](#)

Підключення CSS

Основи HTML & CSS для початківців #11 - Підключення CSS

Смотреть... Поделиться

Урок 11. Підключення CSS

ПОКАЗАТЬ ДРУГИЕ ВИДЕО

0:02 / 9:33

YouTube

[Застосувати код](#)

```
<h1 style="color: red;">  
Hello, I'm amazing heading h1  
</h1>
```

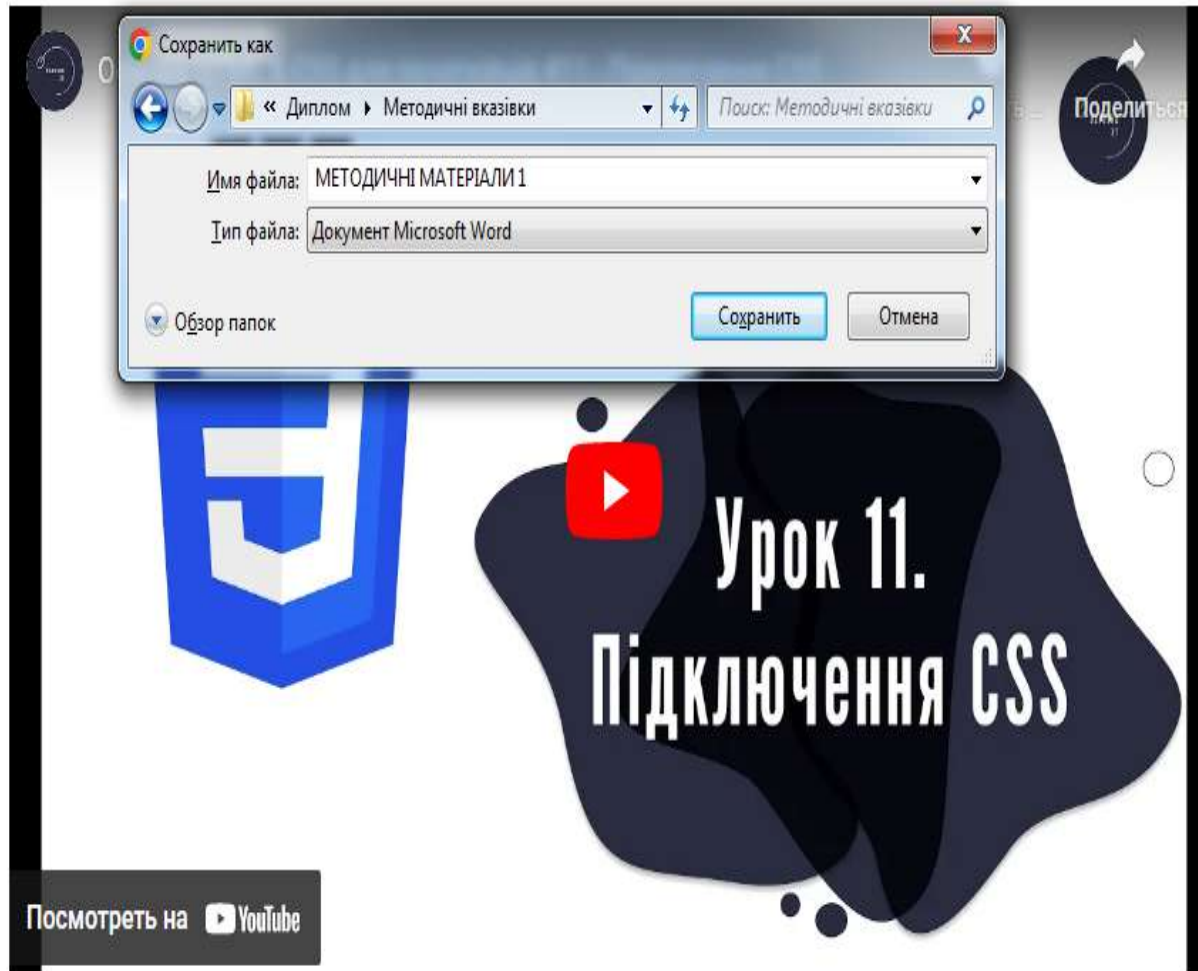
[Отримати пояснення](#)[Застосувати код](#)

```
<style>  
h1 {  
  color: blue;  
}  
</style>
```

[Отримати пояснення](#)

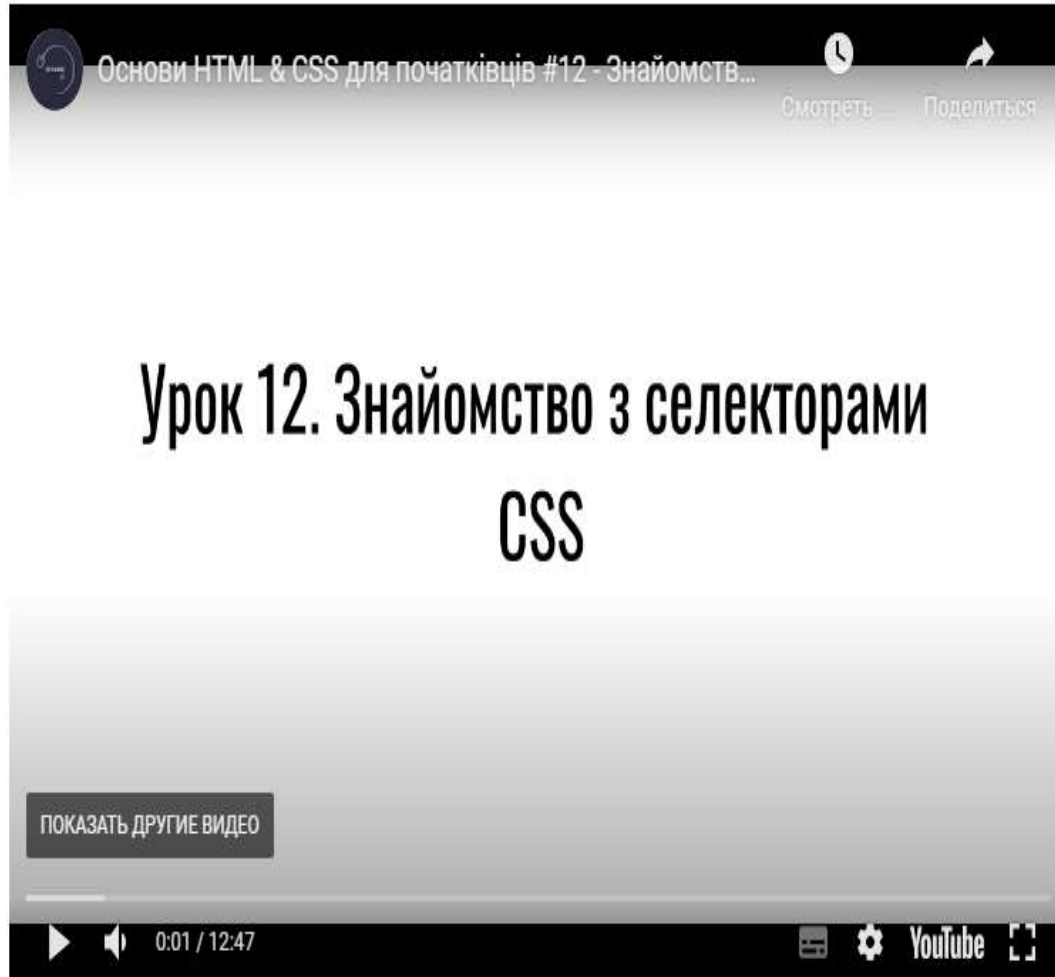
[Наступна лекція](#)[Завантажити методичні вказівки](#)[Головна сторінка](#)

Підключення CSS



[Попередня лекція](#) [Наступна лекція](#) [Завантажити методичні вказівки](#) [Головна сторінка](#)

Знайомство з CSS селекторами



Застосувати код

```
<h3 class="test">Hello, I'm amazing heading h3</h3>
<input class="test" id="name" placeholder="name" type="text">
<button class="test">submit</button>
```

Отримати пояснення

Застосувати код

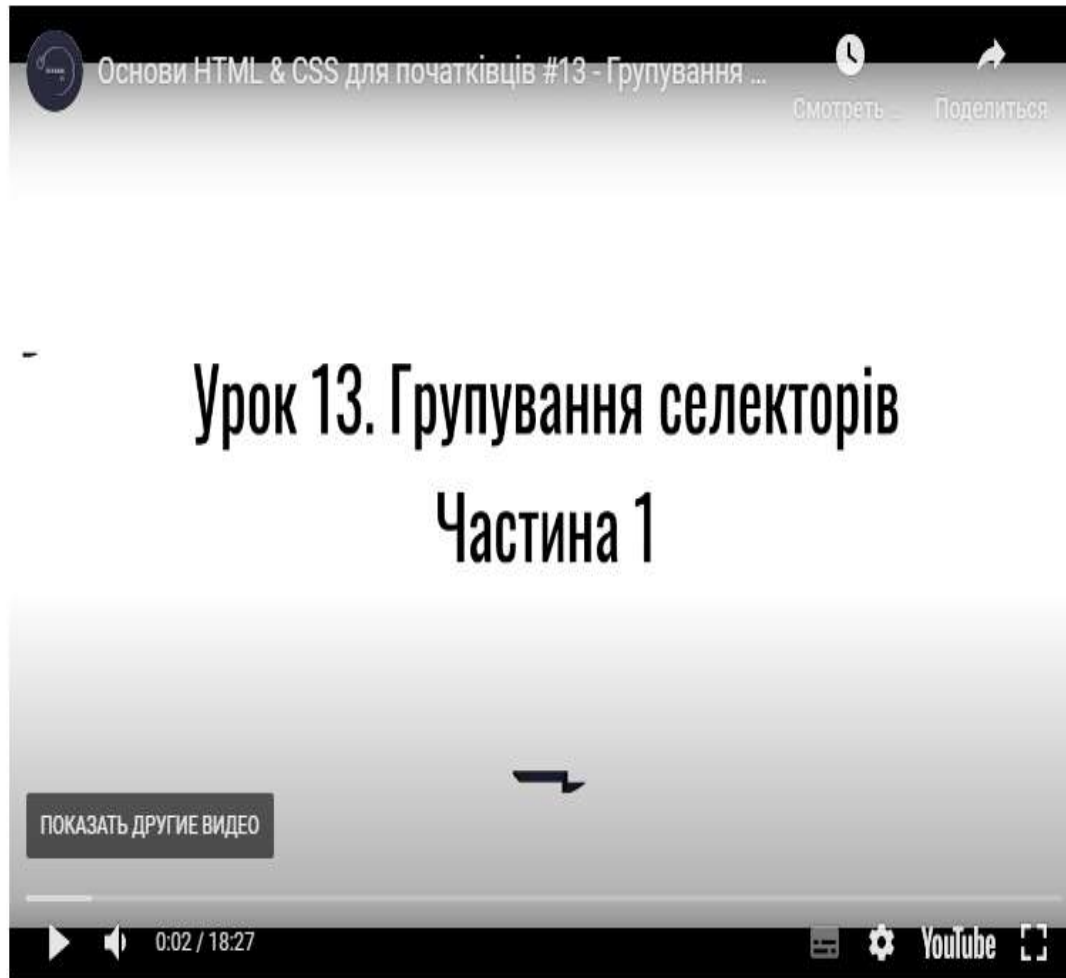
```
/* .test {
  color: red;
} */

#test {
  color: red;
}
```

Отримати пояснення

[Попередня лекція](#)[Наступна лекція](#)[Завантажити методичні вказівки](#)[Головна сторінка](#)

Групування селекторів CSS частина перша

[Застосувати код](#)

```
.div-1 ul {  
  color: white;  
  font-size: 22px;  
}
```

[Отримати пояснення](#)[Застосувати код](#)

```
<p class="div-1-paragraph">Hello from the outor content</p>  
.div-1 .div-1-paragraph {  
  color: white;  
  font-size: 22px;  
}
```

[Отримати пояснення](#)

[Попередня лекція](#) [Наступна лекція](#) [Завантажити методичні вказівки](#) [Головна сторінка](#)

Групування селекторів CSS частина друга

Основи HTML & CSS для початківців #13 - Групування селекто...
Смотреть ... Поделиться

Урок 13. Групування селекторів Частина 2

ПОКАЗАТЬ ДРУГИЕ ВИДЕО

0:02 / 12:52

YouTube

Застосувати код

```
.parent-1 ~ p {  
  color: red;  
}
```

Отримати пояснення

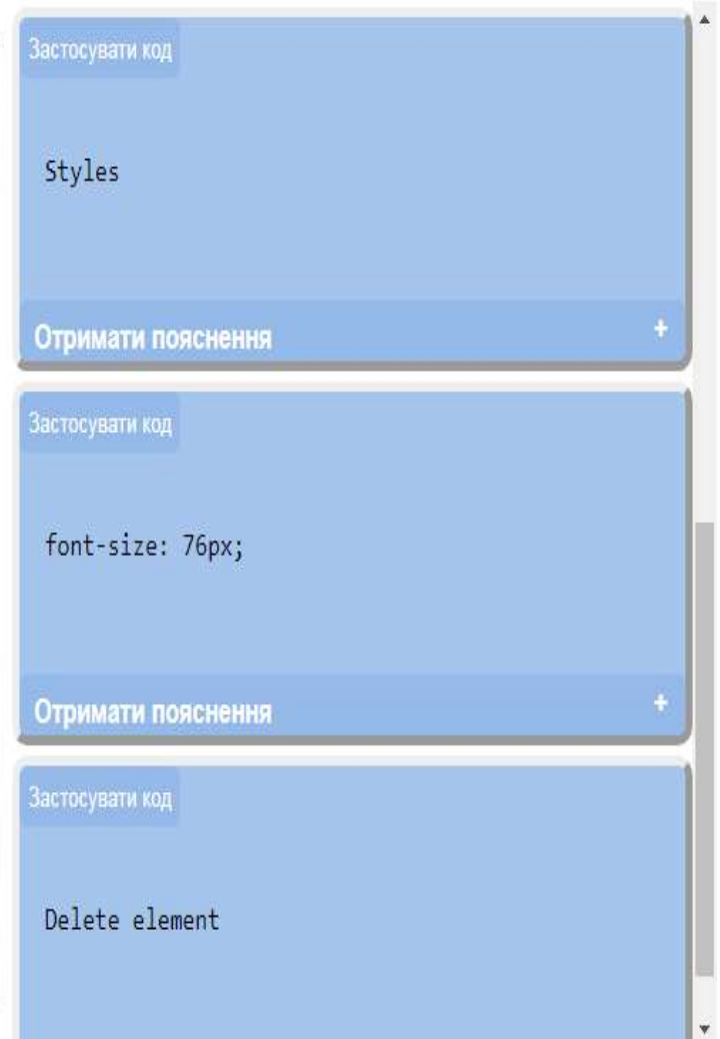
Застосувати код

```
<div>  
  <p>Hello2</p>  
  <p>Hello2</p>  
  <p>Hello2</p>  
  <p>Hello2</p>  
  <p>Hello2</p>  
</div>
```

Отримати пояснення

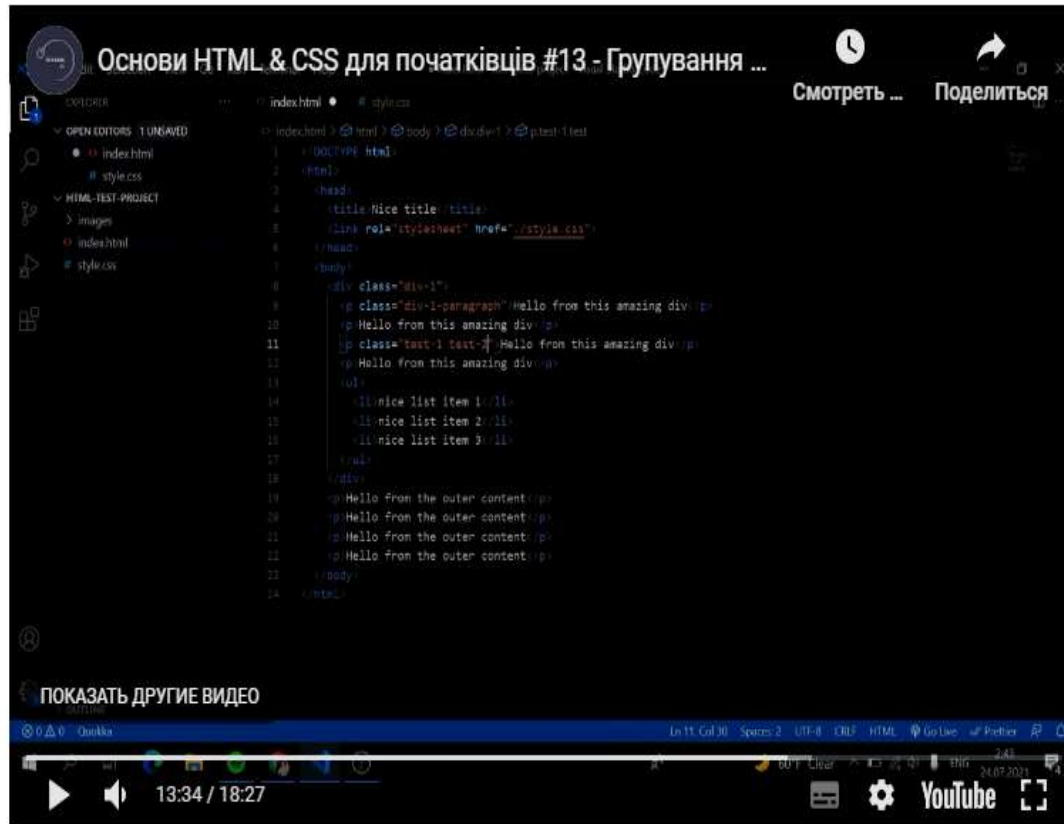
[Попередня лекція](#)[Завантажити методичні вказівки](#)[Головна сторінка](#)

Devtools



[Попередня лекція](#)
[Наступна лекція](#)
[Завантажити методичні вказівки](#)
[Головна сторінка](#)

Групування селекторів CSS частина перша


[Застосувати код](#)

```

<p class="div-1-paragraph">Hello from the utor content</p>
.test-1.test-2 {
  color: yelow;
}
<p class="test-1 test-2">Hello from the utor content</p>
  
```

[Отримати пояснення](#)

Таким чином ми "достукуємося" до тегу з двома класами, тобто записавши селектори `.test-1.test-2`, в css документі без пробілу, ми звернулися до тегу `p`, який має обидва класи, тобто `class="test-1 test-2"`

Групування селекторів CSS частина перша

Hello from this amazing div

Hello from this amazing div

Hello from this amazing div

Hello from this amazing div

- nice list item 1

- nice list item 2

- nice list item 3

Hello from the outor content

Hello from the outor content

Hello from the outor content

Hello from the outor content

Застосувати код

```
<p class="div-1-paragraph">Hello from the outor content</p>
.test-1.test-2 {
  color: yelow;
}
<p class="test-1 test-2">Hello from the outor content</p>
```

Отримати пояснення

Таким чином ми "достукуємося" до тегу з двома класами, тобто записавши селектори `.test-1.test-2`, в css документі без пробілу, ми звернулися до тегу `p`, який має обидва класи, тобто `class="test-1 test-2"`

Додаток Д

Зауваження нормоконтролера

Таблиця Д.1 – Зауваження нормоконтролера

Сторінка	Позначення	Зміст зауваження
ПЗ		Нещільне заповнення сторінок
Розділ 3		Продовження таблиць