

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)
“__” _____ 2023 р.

Випускна кваліфікаційна робота бакалавра (освітній ступінь)

на тему: «Розробка веб-додатку калькуляторів»

зі спецчастиною: «Розробка модулів, структури, інтерфейсу засобами HTML, SASS, Svelte, JavaScript»

Виконав: студент 4 курсу, групи ІІЗ-19
(шифр групи)

спеціальності 121 Інженерія програмного забезпечення
(код і назва спеціальності)

освітньої програми 121 Інженерія програмного забезпечення

Машевського Д.М.
(прізвище та ініціали)

(підпис)

Керівник старший викладач кафедри ПМІ Валерій КОСТІН



(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 дипломній роботі немає
запозичень з праць інших авторів без відповідних
посилань.

Студент 
(підпис)

Луцьк – 2023р.

ДВНЗ «Донецький національний технічний університет»
Факультет комп'ютерно-інформаційних технологій та автоматизації
Кафедра прикладної математики та інформатики
Освітній ступень бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(шифр і назва)
Освітня програма 121 Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
Наталія МАСЛОВА
(підпис) (Ім'я та ПРІЗВИЩЕ)
«___» _____ 2023 року

ЗАВДАННЯ НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Машевському Даніїлу Миколайовичу
(прізвище, ім'я, по батькові)

1. Тема роботи «Розробка веб-додатку калькуляторів»
зі спецчастиною «Розробка модулів, структури, інтерфейсу засобами HTML,
SASS, Svelte, JavaScript»
керівник роботи Костін В.І., старший викладач,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом від « 5 » травня 2023 року № 175

2. Строк подання студентом роботи 08 червня 2023 року

3. Вихідні дані до роботи Результати переддипломної практики

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

- 1) аналіз предметної області та проблемних питань
- 2) опис застосовних методів та обґрунтування способів розв'язання задач
- опис процесу проектування програмного засобу та архітектурних особливостей
- 3) опис алгоритму, розробленого для рішення задачі,
структури бази даних, подробиці реалізації окремих модулів та підсистем
- 4) опис роботи розробленої програмної системи, засоби безпеки;
- 5) тестування програмної розробки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
робота містить 3 таблиць, 26 рисунків, 1 діаграм, 9 слайдів презентації та інший графічний матеріал, у кількості, достатній для демонстрації сутності роботи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Назарова І.А. доцент каф. ПМІ		 16.06.2023

7. Дата видачі завдання 5 травня 2023 року

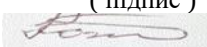
КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури з предметної області	05.05.2023 – 10.05.2023	
2	Визначення елементів розроблюємого програмного додатку	11.05.2023– 15.05.2023	
3	Розробка структури програмного забезпечення, вибір даних для обробки та експериментів.	16.05.2023 – 20.05.2023	
4	Тестування системи	21.05.2023– 01.06.2023	
5	Оформлення пояснювальної записки	01.06.2023– 07.06.2023	
6	Нормоконтроль пояснювальної записки, її перевірка антиплагіатною системою	08.06.2023- 10.06.2023	
7	Оформлення супутніх матеріалів (розробка графічного матеріалу, написання доповіді, тощо) та рецензування роботи	11.06.2023- 21.06.2023	
8	Захист роботи	23.06.2023	

Студент


(підпис)

Керівник роботи


(підпис)

Даніїл МАШЕВСЬКИЙ

(Ім'я та ПРІЗВИЩЕ)

Валерій КОСТІН

(Ім'я та ПРІЗВИЩЕ)

АНОТАЦІЯ

Машевський Д.М. Розробка веб-додатку калькуляторів/ Випускна кваліфікаційна робота на здобуття освітнього ступеня «бакалавр» за спеціальністю 121 Інженерія програмного забезпечення. – ДВНЗ ДонНТУ, Луцьк, 2023.

Об'єктом дослідження є веб-додатки.

Предметом дослідження є проектування, експлуатація та обслуговування веб-додатків.

Мета роботи полягає у проектуванні та розробленні веб-додатку калькуляторів.

Методи досліджень базуються на основних положеннях системного аналізу, аналітичних методів опрацювання інформації.

Практичне значення полягає у розробленні калькуляторів, які будуть допомагати з рішенням довгозатратних по часу задач. Результати дослідження можуть бути використані у будь-яких інших додатків в майбутньому.

Ключові слова: веб-додатки, калькулятори, розробка.

ЗМІСТ

Вступ.....	7
1 ЗАДАЧА РОЗРОБКИ WEB-ДОДАТКУ КАЛЬКУЛЯТОРІВ.....	9
1.1 Аналіз програмних продуктів-аналогів.....	10
1.2 Переваги та недоліки Web-додатку.....	13
1.2.1 Переваги створення Web-додатків.....	13
1.2.2 Недоліки створення веб-додатків.....	14
1.3 Вибір засобів реалізації.....	15
1.3.1 Загальний огляд мов програмування.....	15
1.4 Постановка задачі.....	18
1.5 Різниця між веб-сайтом та веб-додатком.....	19
2 АРХІТЕКТУРА.МОДЕЛЮВАННЯ ВЕБ-ДОДАТКУ КАЛЬКУЛЯТОРІВ.....	22
2.1 Вибір архітектури програмного комплексу.....	22
2.2 Структура веб-додатку.....	23
2.3 Клієнтська частина веб-додатку.....	23
2.4 Опис інструментів розробки.....	25
2.5 Мова програмування JS, фреймворк Svelte.js, SASS.....	26
3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	31
3.1 Архітектура веб-додатку.....	31
3.2 Структура проекту.....	32
3.3 Обґрунтування вибору програмної реалізації.....	33
3.4 Встановлення серверу.	34
3.5 Моделювання процесу роботи веб-додатку.....	35
3.5.1 Діаграма нульового рівня.Опис акторів і варіантів використання	35
3.6 Опис функцій веб-додатку.....	36
3.7 Вимоги до програмного забезпечення.....	40
4 ОГЛЯД ВЕБ-ДОДАТКУ.ТЕСТУВАННЯ.....	42
4.1 Огляд веб-додатку.....	42

4.2 Тестування.....	44
Висновки	51
Перелік джерел посилання.....	54
Додаток А Зауваження нормоконтролера.....	56
Додаток Б Матеріали презентації	57
Додаток В Фрагмент лістингу продукту	62

ВСТУП

Розробка web-додатків в даний час є одним з перспективних напрямів діяльності для багатьох компаній, зайнятих у сфері високотехнологічних цифрових та комп'ютерних технологій.

Web додаток є прикладним програмним забезпеченням, логіка якого розподілена між сервером та клієнтом, а обмін інформацією відбувається за мережі.

Клієнтська частина реалізує інтерфейс користувача, а серверна – отримує та обробляє запити від клієнта, виконує обчислення, формує веб-сторінку та відправляє її клієнту згідно з протоколом HTTP.

Даний вид додатків має низку особливостей, які впливають на процеси їх функціонування, при розробці та підтримці:

- відкриті для тестування з віддалених комп'ютерів, що оптимально для застосування гнучкої методології розробки;
- виконуються незалежно від операційної системи клієнта. При цьому на нього накладається вимога кросбраузерності, що впливає з різної реалізації браузерами стандартів HTML, CSS та DOM;
- являють собою розподілену інформаційну систему та повинні витримувати максимальну кількість звернень користувача.
- Web-додаток, створений із застосуванням сучасних методів розробки, являє собою інформаційний ресурс, завдяки якому можна здійснювати:
 - передачу будь-якої інформації про компанію, а також новинний потік для користувачів;
 - пряма взаємодія з користувачем та його інформаційну підтримку;
 - рекламу, оскільки Web-додаток здатний об'єднати у собі відеорекламу та банери.

Метою дипломної роботи є розробка Web-додатку калькуляторів.

Web додаток надасть можливість клієнту користуватися програмою, маючи при собі тільки вихід в інтернет, при цьому не буде важливо, яка операційна система встановлена у користувача і наскільки потужний у нього девайс.

Для розробки Web-програми необхідно вирішити такі завдання:

- Провести огляд існуючих інструментів та інформаційних технологій для створення Web-додатків;
- Визначити структуру та дизайн Web-додатку, що розробляється;
- Реалізувати основний функціонал Web-додатку;
- Виконати адаптацію Web-додатків під усі популярні пристрої;
- Провести функціональне тестування Web-додатку.

Кваліфікаційна робота складається із п'яти частин:

- перша частина містить постановку мети, завдання та опис предметної області.
- друга частина складається з аналізу, вибору програм для вирішення поставленого завдання, архітектури щодо розробки Web-додатка.
- третя частина складається з проектування та розробки Web-додатку.
- четверта частина містить опис розробленого Web-додатку.
- п'ята частина містить тестування Web-додатку.

1 ЗАДАЧА РОЗРОБКИ WEB-ДОДАТКУ КАЛЬКУЛЯТОРІВ

У сучасному світі, в останні роки, з розвитком сучасних технологій, поширеністю та доступністю Інтернет-комунікації відкрилися унікальні можливості для навчання. Інтернет – це не лише невичерпна скарбниця навчальної інформації, а й джерело активної інтелектуальної діяльності сучасного школяра, з необмеженими можливостями для отримання знань та вдосконалення умінь і навичок. Існує багато способів для цього: читати книги, проходити курси. Але є і інший спосіб. Виконуєте ви роботу, а там задача. Можна самому зрозуміти, прочитати і вирішити її. Але можна використати спосіб, яким багато хто нехтує. Це - калькулятори.

Багато людей думають, що це неправильно, що краще вирішити її самому, а не прибігати до таких способів. І будуть в деяком сенсі праві, але якщо подивитись на це з іншої сторони. Більшість калькуляторів, пропонують подивитись рішення задачі, а тому це як подивитись рішення у книзі, а може навіть краще, бо не буде багато теорії, а усе тільки необхідне.

Так, більшість людей просто вирішать задачу, і не будуть дивитись рішення, бо для чого це нужно. Але тут вже залежить від людини, ми не можемо навчити того, хто цього не хоче [1-2].

Зробити веб-додаток для калькуляторів було вирішено для того, щоб облегчити і прискорити різні напрямки роботи. Бо якщо ти знаєш як розв'язується та чи інша задача, чому би і не облегшити собі життя, об'єднав свої знання і програмування. І коли потрібно буде вирішити велику кількість задач, можна швидко це зробити завдяки цьому веб-додатку.

Завдання, які необхідно було вирішити: надати користувачам сучасні і необхідні калькулятори, які будуть потрібні я на роботі; надати користувачам зручний графічний інтерфейс, який може працювати на комп'ютері.

1.1 Аналіз програмних продуктів - аналогів

З кожним роком набувають все більшого поширення різноманітні веб-додатки. На їх основі вже побудовано багато різноманітного програмного забезпечення, яке вирішує дуже широкий спектр прикладних задач. Серед них можна виділити калькулятори.

Перший серед прикладів калькуляторів - веб-додаток GeoGebra, представлений на рис. 1.1.

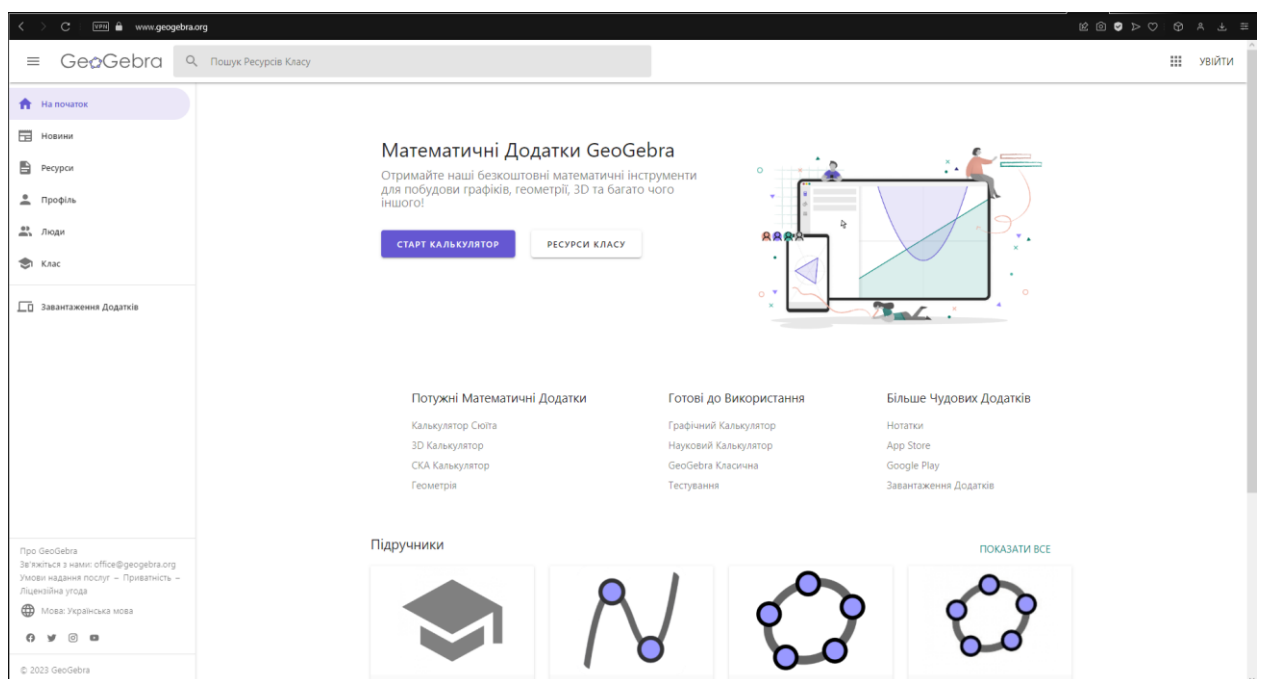


Рисунок 1.1 – Головна сторінка GeoGebra

Мінусами цього веб-додатку є невеликий спектр функцій калькулятора. Він працює лише з графіками, діаграмами і фігурами. Рішення якихось задач або рівнянь в ньому немає [3].

Перевагою є інтерфейс, який досить простий, але зручний і інтуїтивно зрозумілий.

Наступним йде веб-додаток PlanetCalc, представлений на рис. 1.2.

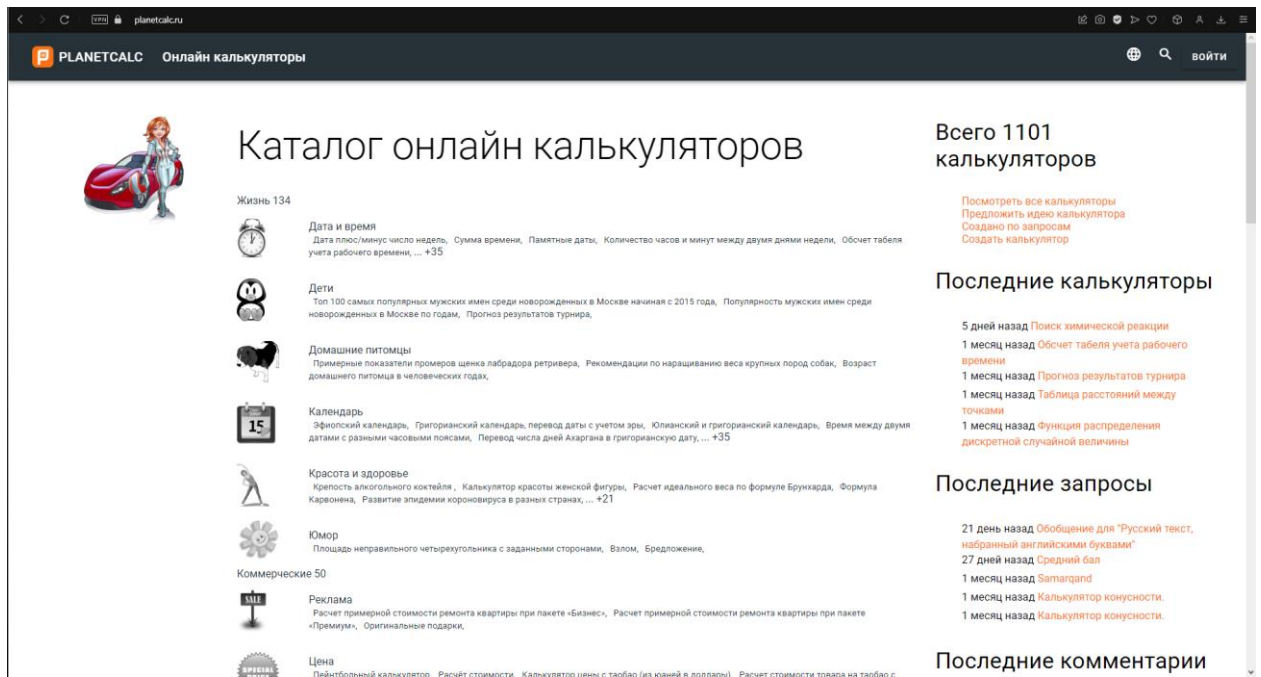


Рисунок 1.2 – Головна сторінка PlanetCalc

В цьому веб-додатку вже більший вибор калькуляторів, в ньому є такі види як:

- дата;
- час;
- фінанси;
- ціна;
- фізика і т.д.

Це здебільшого більше перевага, але я віднесу до мінусів, бо більшість з них, з моєї точки зору непотрібні. Час, дата, ціна і т.д, швидше буде самому розрахувати ніж шукати калькулятор.

Інтерфейс теж віднесу до мінусів, він простий і зрозумілий, но на мою думку, в ньому немає ніякої ізюминки, яка б відділяла його від інших.

Наступний і останній - це Microsoft Math Solver (рис. 1.3).

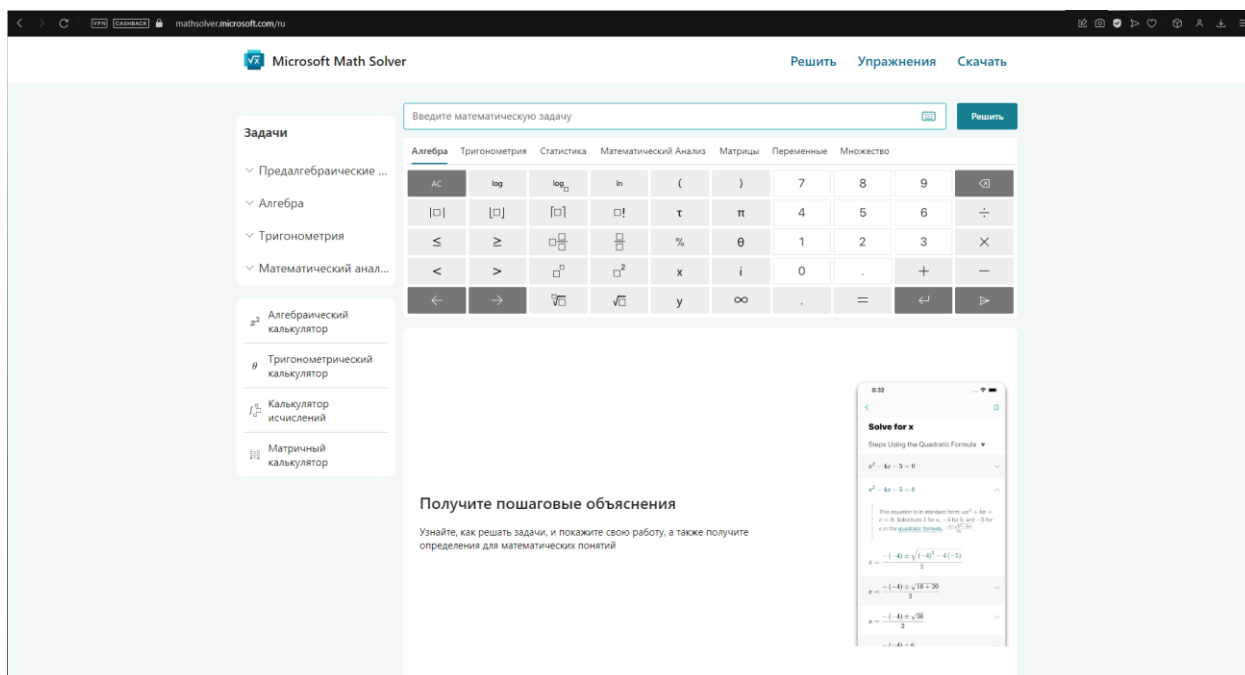


Рисунок 1.3 – Головна сторінка Microsoft Math Solver

З Microsoft Math Solver, усе куди краще ніж з іншими. Теж небагато калькуляторів, але на мою думку це перевага. Лінійні рівняння, системи рівняння, тригонометрія, усі вони необхідні, і будуть використовуватись.

З інтерфейсом теж все добре, є ізюминка, яка відрізняє від інших. Інтуїтивно зрозумілий. Можна встановити на телефон, що теж являється перевагою.[5]

Дуже багато створено і інших веб-додатків калькуляторів, але у всіх них багато проблем.

Проаналізувавши більшість з них, можна виділити основні недоліки:

- Непотрібна реклама. Дуже відталкує від додатку, коли кожна твоя дія, супроводжується рекламою;
- Зручність використання. Банальна проблема, коли бажання зробити красивий інтерфейс, руйнує зручність і людина не може на інтуїтивному рівні знайти те, що тобі треба;
- Застарілий дизайн;
- Зламани калькулятори.

Розглянемо детальніше про додатки та проведемо загальний аналіз (табл. 1.1).

Таблиця 1.1 – Аналіз розглянутих додатків калькуляторів

Критерії	Web-додаток GeoGebra	Web-додаток PlanetCalc	Web-додаток Microsoft Math Solver
Відсутність непотрібної реклами	+	+	+
Зручність використання	+	-	+
Застарілий дизайн	-	+	-
Зламани калькулятори	-	-	-
Непотрібні калькулятори	+	+	-
Інтегрованість до інших платформ	+	-	+

Отже проаналізувавши додатки, було прийнято рішення розробити додаток, який на основі аналізів таблиці 1.1, не буде мати недоліків. Дана розробка буде мати переваги даних додатків, а також сучасний і зручний інтерфейс.

1.2 Переваги та недоліки Web-додатку

1.2.1 Переваги створення Web-додатків

Веб-додатки розробляються за допомогою мов програмування, таких як HTML і CSS, які добре відомі серед ІТ-фахівців. На відміну від нативних додатків, веб-додатки можна використовувати на будь-якому пристрої. Запрограмовані для запуску на будь-якій операційній системі, веб-додатки не потребують ресурсів і не висувають жодних вимог до апаратної платформи.

Підтримка попередніх версій програм не є проблемою і не виникає проблем із сумісністю. Коли виходить нова версія нативного додатку, користувачам часто доводиться вирішувати проблеми, пов'язані з оновленням встановленої копії на своєму пристрої. Для браузерних додатків таких проблем немає. Існує лише одна версія, яка використовується всіма користувачами, і коли виходить нова, всі безвинятку переходять на неї, іноді навіть непомічаючи цього. Веб-додатки не потрібно встановлювати на пристрій. Вони запускаються у власному веб-браузері пристрою за простою URL-адресою, немає необхідності завантажувати і встановлювати їх з магазинів додатків, таких як Google Play або Apple AppStore. Це означає, що прямі посилання через веб-додатки є безкоштовними, що економить ваші гроші. Веб-додаток також можна відкрити. Це означає, що їх не потрібно оновлювати, як звичайні програми. Високий рівень розвитку мережевого підключення та надійність веб-технологій. Розробка веб-додатків - це найдешевший тип розробки додатків. Він полягає у створенні посилання або декількох посилань між додатком і URL-адресою.[2] Розробка нативних або інтерпретованих додатків коштує дорожче, але має набагато більше шансів на успіх. Веб-додатки дозволяють користувачам бути по-справжньому мобільними. Результати їхньої роботи можна зберігати на сервері, за необхідності, отримати доступ до них з будь-якої точки світу, де є інтернет-з'єднання.

1.2.2 Недоліки створення веб-додатків

Як згадувалося вище, один веб-додаток можна використовувати на всіх пристроях. Однак, звичайно, веб-додаток повинен бути запрограмований на відображення незалежно від операційної системи пристрою. Якщо веб-додаток не є адаптивним, у нього можуть виникнути проблеми з відображенням на iOS, Android і Windows Phone. Пристрій повинен бути підключений до Інтернету. В іншому випадку ви не зможете відобразити веб-додаток, і він буде марним для вас. На жаль, в нашій країні, наприклад,

інтернет доступний не скрізь. Це ускладнює роботу з веб-додатками, наприклад, під час подорожей. Кожне перезавантаження (або оновлення сторінки) вимагає встановлення HTTP-з'єднання, обробки запиту на сервері, відправки HTTP-повідомлення у відповідь по мережі і перезавантаження сторінки в браузері, що викликає помітну затримку.[7] Це призводить до переривчастого режиму роботи і сповільнює роботу з сайтом. Крім того, існують обмеження доступу до певних апаратних можливостей пристрою, на якому він працює. Існує ряд додатків, які неможливо реалізувати в Інтернеті, наприклад, неможливо створювати складні 3D-моделі в браузері.

1.3 Вибір засобів реалізації

Від створення інтерактивних розважальних додатків до серйозних бізнес-проектів, що вимагають підвищеної надійності та захисту від несанкціонованого доступу, веб-розробники сьогодні стикаються з різноманітними проблемами. Для їх вирішення їм потрібні відповідні інструменти, будь то мова програмування, фреймворк або, останнім часом, система управління контентом, що їх цікавить. На сьогоднішній день доступна ціла низка мов програмування. Переваги кожної з них можна оцінити лише в контексті поставленого завдання. Вибір мови залежить від знань програміста і від того, чи є вони достатніми для проекту.

1.3.1 Загальний огляд мов програмування

Мова програмування - це формальна символічна система, призначена для розробки комп'ютерних програм. Мова програмування визначає набір команд, синтаксичних і семантичних правил, які визначають зовнішній вигляд програми та дії, які виконує виконавець (комп'ютер) під її контролем. На наступному графіку показано поширеність мов програмування для розробки веб-додатків (рис. 1.4).

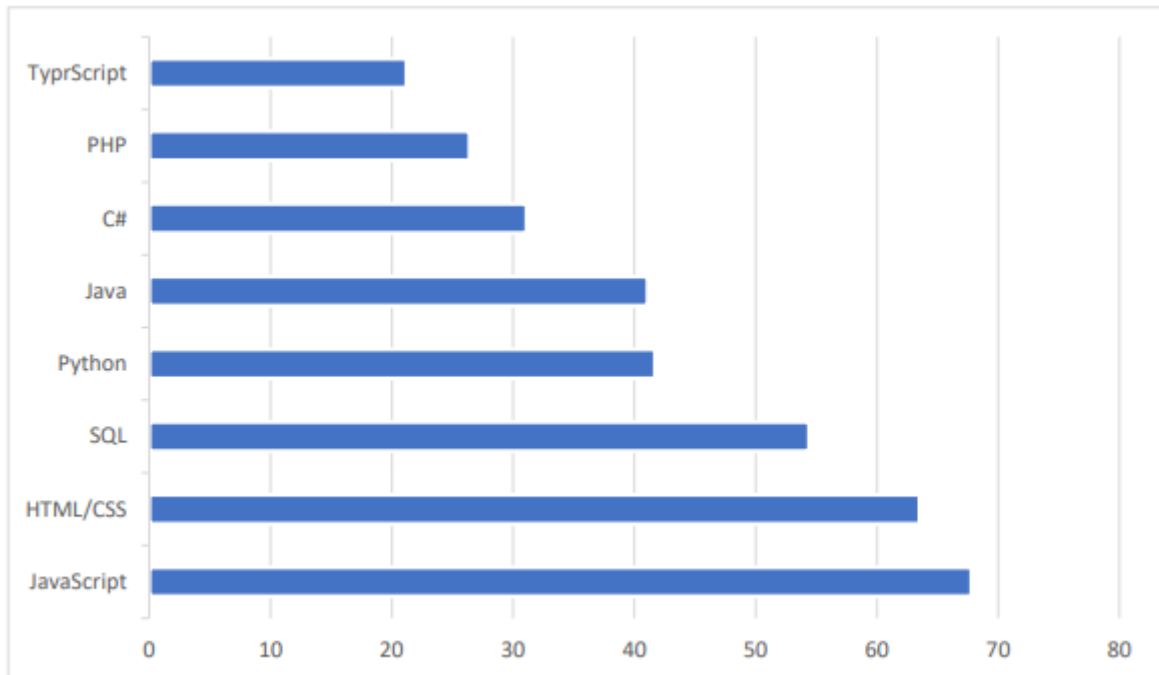


Рисунок 1.4 – Популярність мов програмування серед веб розробників

Усі мови веб-програмування можна розділити на клієнтські та серверні. Як випливає з назви, клієнтські мови пишуть клієнтські програми (веб-браузери), тоді як серверні мови пишуть програми на стороні сервера.

Найпопулярнішою мовою програмування на стороні клієнта є JavaScript; JavaScript - це мова програмування для управління сценаріями перегляду веб-сторінок. Найважливішою особливістю цієї мови є те, що при її використанні можна змінювати елементи середовища відображення під час перегляду веб-додатку, при цьому веб-сторінка не перезавантажується. Наприклад, JavaScript можна використовувати для зміни кольору фону веб-сторінки, заміни вбудованих у веб-сторінку зображень, створення нового вікна відображення або відображення повідомлення.[8]

Поряд з JavaScript використовуються також не менш популярні мови розмітки CSS і HTML, які є важливими елементами розробки веб-інтерфейсів. HTML - це мова гіпертекстової розмітки - мова веб-браузерів, на якій створюються веб-додатки а CSS - це формальна мова, яка використовується

для опису зовнішнього вигляду документів, написаних мовами розмітки. Розробники веб-додатків використовують CSS для встановлення шрифтів, розташування на сторінці та опису інших принципів того, як повинні виглядати блоки на веб-сторінці.[9]

Серверні мови веб-програмування включають популярну мову програмування SQL, яка може отримувати певні дані з великих, складних баз даних. Вона користується великим попитом у великих компаніях, таких як Microsoft, що робить її розумним вибором для розробників, які мають амбіції або потребують регулярного використання баз даних.

Порівняно з іншими мовами серверного програмування, Python дуже простий у вивченні, динамічний та універсальний. Вона набула нової популярності серед розробників, оскільки її синтаксис легко читається, простий і, перш за все, інтуїтивно зрозумілий.

Java була розроблена в 90-х роках і залишається найпопулярнішою мовою програмування; Java стала золотим стандартом для всіх областей веб-розробки в усьому світі. Вона об'єктно-орієнтована, працює на будь-якій платформі і є досить функціональною.

Java Server Page - це технологія J2EE, призначена для створення веб-додатків з використанням Java. JSP і ASP.NET мають багато спільного, і вибір між цими двома технологіями ґрунтується на перевагах розробника, а не на плюсах і мінусах цих платформ. Зазвичай так і робиться.

Сьогодні мова програмування C# є однією з найпотужніших, найбільш швидкозростаючих і популярних мов в ІТ-індустрії. Зараз вона використовується для написання додатків від невеликих десктопних програм до великих веб-порталів та веб-сервісів, які щодня обслуговують мільйони користувачів.[10]

Звичайно, найпопулярнішою мовою веб-програмування є PHP, популярна скриптова мова, яка широко використовується для розробки веб-додатків. Її основними перевагами є безкоштовне програмне забезпечення, простий синтаксис, висока швидкість і велика спільнота розробників.

PHP є динамічною мовою програмування, що означає, що немає необхідності вказувати типи даних при оголошенні змінних або самих змінних.

PHP підтримує кілька об'єктно-орієнтованих функцій, які повністю підтримуються в п'ятій редакції мови; PHP підтримує всі три основні механізми ООП: інкапсуляцію, поліморфізм і успадкування.[11] Це означає, що батьківські класи вказуються за допомогою ключового слова `extends` після імені класу. Найбільшою перевагою є те, що багато поширених CMS написані на PHP.

1.4 Постановка задачі

Кваліфікаційна робота призначена для того, щоб користувач міг з легкістю обрати для калькулятор і швидко вирішити його задачу.

Калькулятор повинен бути зроблений як веб-додаток, доступний в Інтернеті. Dodatok повинен складатись з взаємозалежних частин, а функції кожної частини повинні бути чітко розділені.

Даний додаток повинен мати наступні можливості:

- Огляд ступенів розв'язання задачі.
- Калькулятори.

Сайт повинен складатись з користувацької частини.

Користувацький інтерфейс веб-додатку повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів та сторінок.

Наявність навігації на всіх сторінках сайту. Правильна структура інформації дозволяє користувачам без проблем продовжувати дослідження додатку, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.

Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:

- Вивчити процеси реалізації задач.
- Провести аналіз додатків-аналогів для розроблення технічного завдання.
- Обрати та налаштувати інструменти реалізації.
- Визначити структуру даних та спроектувати веб-додаток.
- Реалізувати додатки та розробити інструкції користувача.

1.5 Різниця між веб-сайтом та веб-додатком

Справа в тому, що веб-сайти не обов'язково є веб-додатками, але зазвичай створюються для надання інформації. Веб-сайти можуть бути статичними або динамічними, але їхня основна мета - надавати інформацію.

Динамічні сайти.

Вміст таких сайтів не зберігається у вигляді статичних HTML-сторінок, а зберігається в базі даних і відображається безпосередньо, "на льоту", у відповідь на запити користувачів. Існує багато систем програмування та широко використовуваних і визнаних мов програмування, наприклад, PHP, Perl, ASP тощо. Ці мови дозволяють гнучко створювати веб-сайти будь-якої складності, але, звичайно, не кожен може це робити, і поріг входження в цю сферу дуже високий. Процес створення таких сайтів продовжується і ускладнюється додаванням нового учасника - програміста, який вільно володіє однією з мов програмування.

Однак одному програмісту неможливо самостійно розробити функціонал майбутнього сайту і створити переконливий дизайн, що можливо тільки разом з дизайнером. У цьому методі для розділення інформаційного та графічного наповнення ресурсу створюється спеціальний шаблон сайту, на основі якого генеруються сторінки сайту з урахуванням необхідного дизайну. Ще однією перевагою динамічного сайту є відносна простота управління ресурсом через адміністративну панель і невисока вартість розробки проекту.

Таким чином, якщо ваш сайт складається або буде складатися з багатьох сторінок, або якщо ви плануєте досить часто його оновлювати, переваги динамічної структури очевидні. Це також означає, що розробникам не доведеться повністю переробляти кожну сторінку при незначному оновленні дизайну або введенні нового розділу сайту. Відокремлення дизайну від інформаційного наповнення є головною особливістю і найбільш значущою перевагою динамічних сайтів.

Традиційні веб-сайти, засновані на html-сторінках, складні в обслуговуванні, архаїчні і непридатні для представлення складних структур даних. Динамічні сайти не тільки висувають нові вимоги до їх створення та підтримки, але й є набагато дорожчими у розробці та потребують найму нового персоналу - програмістів. Тому рішенням стало створення якісного веб-сайту за допомогою CMS.

Веб-додатки інтерактивні.

Вище були описані функції, притаманні веб-сайту, однією з особливих рис, що характеризують веб-додатки, є інтерактивність. Це означає, що користувач є не просто читачем, а активним учасником роботи сайту: користувачі веб-додатків шукають потрібну їм інформацію, заповнюють форми, здійснюють платежі через особистий кабінет тощо. За рівнем складності такі рішення набагато складніші, а отже, і дорожчі за інформаційні сайти. Наприклад, "ВКонтакте" та "Однокласники" - це веб-додатки з дуже широким спектром функцій; див. домашню сторінку Amazon.com. Хоча це і гібрид, але це все ж таки інформаційний веб-сайт.

Знімок екрана вище показує багато інформації. Процесори, материнські плати, пам'ять, ціни та категорії - все це різні типи інформації. Однак на цій сторінці головна мета користувача - взаємодія. Користувач може перейти до певної категорії продуктів. Або ж він може прочитати більше про певний продукт і придбати його. Користувач може здійснити пошук або переглянути вміст кошика. Це все різні типи взаємодії.

Кожен веб-сайт представляє певну інформацію, тому вам потрібно прийняти рішення про баланс між інформацією та взаємодією. Якщо ви хочете більше читати або взаємодіяти, проведіть на сторінці 30 секунд або 10 хвилин; якщо ви витрачаєте менше часу на одну сторінку, але багато взаємодієте, вам підійде інтерактивний веб-сайт.

Звичайно, бувають і винятки. Наприклад, розважальні веб-сайти надають цікаву інформацію і в той же час пропонують користувачам інтерактивні можливості для обробки цієї інформації.

2 АРХІТЕКТУРА. МОДЕЛЮВАННЯ ВЕБ-ДОДАТКУ КАЛЬКУЛЯТОРІВ

Аналізуючи поставлену задачу та методи її вирішення, було вирішено розроблювати програмний комплекс на основі веб-технологій. Головною перевагою веб-застосунку перед іншими варіантами є його універсальність і можливість використання на будь-яких пристроях без портування на цільову операційну систему (браузер і його віртуальна машина виступає як цільова універсальна операційна система і комп'ютер).

2.1 Вибір архітектури програмного комплексу

Для реалізації поставленої задачі було вирішено використовувати двохланкову архітектуру, яка складається з таких компонентів: клієнт і сервер. Схема даної архітектури зображена на рисунку 2.1.

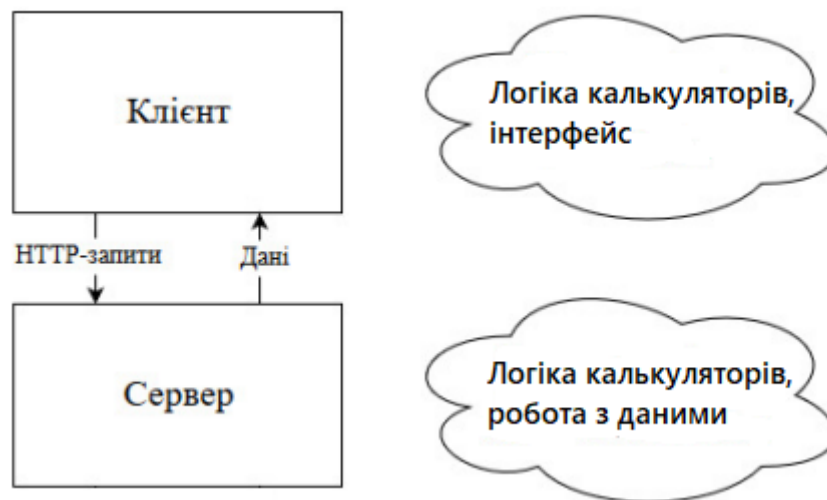


Рисунок 2.1 – Двохланкова архітектура

Під час користування програмою, користувач взаємодіє з клієнтським додатком, яким є веб-сайт в даному випадку. На рівні користувача реалізований інтерфейс, за допомогою якого відбувається налаштування програми та перегляд результатів роботи. Також на користувацькому рівні

відбувається попередня обробка даних перед відправленням на сервер і також опрацювання результатів від сервера.

2.2 Структура веб-додатку

Створений веб-додаток виступає в ролі додатку математичних калькуляторів. Структура відображена на рисунку 2.2.

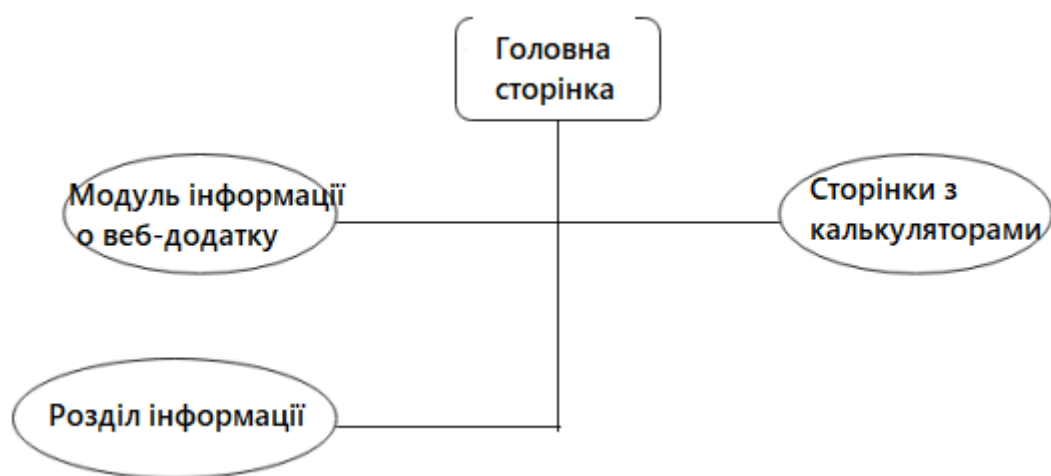


Рисунок 2.2 – Структура веб-додатку

2.3 Клієнтська частина веб-додатку

Веб-додаток складається з клієнтської та серверної частин, тим самим реалізуючи технологію "клієнт-сервер".

Клієнтська частина реалізує користувацький інтерфейс, формує запити до сервера та обробляє відповіді від цього. Серверна частина отримує запит від клієнта, виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту по мережі за допомогою протоколу HTTP. [12]

Сама веб-програма може виступати як клієнт інших служб, наприклад, бази даних або іншої веб-програми, розташованої на іншому сервері. Яскравим прикладом веб-додатку є система управління вмістом статей

Вікіпедії. Безліч її учасників можуть брати участь у створенні мережевої енциклопедії, використовуючи для цього браузері своїх операційних систем (будь Microsoft Windows, GNU/Linux або будь-яка інша операційна система) і не завантажуючи додаткових виконуваних модулів для роботи з базою даних статей. В даний час набирає популярності новий підхід до розробки веб-додатків, що називають Axios.

Axios - це бібліотека JS, яка використовується для відправки HTTP-запитів з Node.js або браузера. У документації зазначені такі можливості:

- Використання XMLHttpRequests із браузера;
- Виконання HTTP запитів із Node.js;
- Підтримка Promise API;
- Перехоплення запиту та відповіді;
- Перетворення даних запиту та відповіді;
- Скасування запитів;
- Автоматичне перетворення даних у форматі JSON;
- Клієнтська підтримка захисту від XSRF (міжсайтова підробка запитів).

Також останнім часом набирає великої популярності технологія WebSocket, яка не вимагає постійних запитів від клієнта до сервера, а створює двонаправлене з'єднання, при якому сервер може надсилати дані клієнту без запиту від останнього.

Таким чином з'являється можливість динамічно керувати контентом як реального часу. Для створення веб-застосунків на стороні сервера використовуються різноманітні технології та будь-які мови програмування, здатні здійснювати виведення у стандартну консоль.

Основою клієнтської частини є процес верстки. Цей процес вважається наступним етапом після підготовки макета дизайну Web-додатка.

Сама верстка ділиться на три основні кроки:

- нарізка макета дизайну на окремі складові;
- створення всіх необхідних файлів для проекту у форматах *.html та *.css;
- написання в них спеціальних конструкцій на мовах HTML та CSS.

Наступним етапом frontend-розробки є програмування на мовою JavaScript.

Цей процес дозволяє додати до нашого Web-додатку певну інтерактивну функціональність:

- анімацію;
- списки, що випадають;
- спрацьовування кнопок;
- плавна поява прихованих елементів;
- відображення нової інформації без перезавантаження сторінки.

Заключним етапом frontend-розробки вважається тестування працездатності всіх елементів Web-додатка та перевірка відповідності дизайну з макетом

2.4 Опис інструментів розробки

Програмний комплекс побудований за принципами двохланкової архітектури побудови програм.

Кожен рівень цієї архітектури реалізовано з використанням різних технологій і головною ціллю було створення мультиплатформного рішення з використанням відкритих технологій.

Для клієнтського рівня було використано такий набір технологій: мова програмування JavaScript, фреймворк для побудови веб-застосунків Svelte.js.

Для серверного рівня було використано такі технології: мова програмування HTML, серверна платформа Live Server.

2.5 Мова програмування JavaScript, фреймворки Svelte.js, SASS

Мова програмування JavaScript є динамічною, об'єктно-орієнтованою прототипною мовою програмування. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд вебсторінки. Мову JavaScript класифікують як прототипну (підмножина об'єктноорієнтованої), скриптову мову програмування з динамічною типізацією.[13] Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Оскільки JavaScript є інтерпретатором, без строгої типізації, і може виконуватися в різних середовищах, кожне зі своїми власними особливостями сумісності, програміст має бути дуже уважним, і повинен перевіряти, що його код виконується як очікується в широкому переліку можливих конфігурацій. Для JavaScript існують кілька фреймворків, серед яких є Vue.js, React, Svelte. Svelte (читається як "Звелт", що в перекладі означає "елегантний") - це

фреймворк. Налаштування над мовою JavaScript, що пропонує радикально інший підхід до створення веб-інтерфейсів. Svelte відрізняється від традиційніших рішень на кшталт React і Vue як із погляду організації коду, і з погляду перетворення цього коду на «готовий для вживання продукт». Svelte дозволяє створювати більш швидкі та менш ресурсомісткі програми, при цьому паралельно спрощуючи процес розробки. Звучить дуже добре, але все справді так і працює.[14] Перехід на Svelte справді не несе за собою майже жодних недоліків. У Svelte багато цікавих рішень, але я зосереджуся на фундаментальних відмінностях від конкурентів та особливостях синтаксису, що впадають у вічі тим, хто до цього не взаємодівав із подібними технологіями зовсім.

Пререндерінг.

У Svelte вбудований компілятор коду, що перетворює .svelte-файли на чистий JavaScript-код.

Підсумковий результат роботи фреймворку є кодом без натяку на цей самий фреймворк, перекладаючи всю важку роботу на етап складання програми. [20]

У випадку зі Svelte немає такого явища, як Virtual DOM і пов'язаних з ним витрат.

Якщо не знаєте, то Virtual DOM – це надбудова над деревом елементів у HTML, що використовується у фреймворках типу Vue та React для отримання інформації про зміну стану ПЗ та подальшого рендерингу оновлюваних даних на екран.

Це необхідно, щоб користувачеві не доводилося щоразу перезавантажувати програму або сторінку для отримання найбільш актуальної інформації від софту, що використовується. Проблема Virtual DOM полягає в тому, що зміни, внесені до нього, потім потрібно відобразити в реальному DOM, а це відбувається в процесі взаємодії користувача з вашим софтом, що б'є за продуктивністю та ресурсомісткістю ПЗ.

Творці Svelte змогли досягти аналогічної функціональності, але без необхідності додавати надбудови. Все торжество технологій вміщається у чистий JavaScript-код, а програмістам немає необхідності проводити додаткову роботу з оптимізації своїх проєктів.

Хоча Svelte побудований на базі JavaScript і успадковує від нього більшу частину синтаксису, тут все ж таки є ряд моментів, які значно відрізняються від звичних аспектів HTML і JS.

#if та #each блоки

Однією з нестандартних, але функціональних та корисних фішок синтаксису Svelte варто назвати if-, each- та async-блоки прямо в HTML-коді.

Допустимо, ви хочете додати до свого дерева елементів `<div>` з повідомленням «Ви успішно увійшли до системи», але вам важливо, щоб воно відображалось лише за умови, що користувач правильно ввів свої дані. Для цього потрібно створити блок `#if`, що перевіряє умову коректності логіну та пароля користувача, а в його тіло внести безпосередньо `<div>` з повідомленням.[19]

С `#each`-блоками схожа історія, тільки вони дозволяють перебирати цілі масиви даних і замінюють собою складні структури на кшталт `forEach` з наступним створенням окремих блоків через `createElement` тощо.

Отже, ви хочете перебрати масив об'єктів та розмістити їх у DOM. Для цього ви створюєте `#each`-блок, передаєте йому компоненти масиву (як окремі елементи) і вже оформляєте їх у тілі функції. Побачити код можна на рисунку

```

{#each arrayOfElements as elem, i}
<div>
  {elem} + ` – элемент номер ${i}`
</div>
{/each}

```

Рисунок 2.3 – Робота {#each} блоку

Цей код замінює щось по типу коду на рисунку 2.4.

```

arrayOfElements.forEach((i) => {
  let new element = document.createElement('div')
  element.innerHTML = `${i} – элемент номер ${indexOf(i)}`
  document.querySelector('.someDiv').appendChild(element)
})

```

Рисунок 2.4 – Код, замінюємий блоком {#each}

Що ж таке, SASS.

Sass - це розширення CSS, яке додає силу та елегантність до основної мови. Основна мета Sass – виправити недоліки CSS. Як ми всі знаємо, це не найкраща мова у світі.

Будучи досить простим для освоєння, він може швидко стати заплутаним, особливо у великих проектах. Ось тут і вступає у свою роль Sass як метамова, яка покращує синтаксис CSS, щоб надати додаткові можливості та зручні інструменти.

Тим часом Sass залишається консервативним щодо мови CSS. Мета не в тому, щоб перетворити CSS на повнофункціональну мову програмування; Sass хоче лише допомогти там, де не справляється CSS. [18]

Тому почати використовувати Sass - не складніше, ніж почати вивчати CSS: він просто додає додаткові можливості поверх CSS. Є багато способів використати ці можливості.

Якісь способи хороші, якісь не дуже, а деякі взагалі незвичайні. Цей посібник призначений для того, щоб дати послідовний та документований підхід для написання коду на Sass.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Архітектура веб-додатку

Веб-додаток буде складатись з однієї частини, а саме користувацької. Користувацький інтерфейс веб-сайту повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів.

Web-додаток повинен складатися з наступних розділів:

– головна сторінка

– містить такі сторінки калькуляторів:

- Лінійна інтерполяція;
- Елементи комбінаторики - перестановки, розміщення, поєднання;
- Розкладання функції в ряд Фур'є;
- Середнє арифметичне, Дисперсія, Варіація;
- Рішення матриць;
- Паралельність прямих на площині;
- Рішення логарифмів;
- Тригонометричні функції;
- Метод найменших квадратів;
- Сума арифметичної прогресії;

3.2 Структура проекту

Далі йде структура проекту, яку можна розглянути на рисунку 3.1.

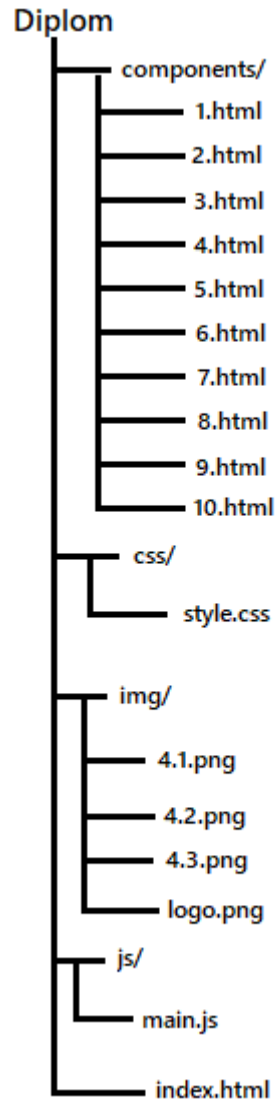


Рисунок 3.1 – Структура проекту

Проект складається з наступних каталогів:

Diplom/

Каталог в якому знаходиться головна сторінка веб-додатку, та усі інші каталоги.

Diplom/components/

Каталог в якому храняться усі сторінки, а також логіка до них. В них содержується HTML, JS коди.

Diplom/css/

Каталог з усіма стилями.

Diplom/img/

Картинки для сторінок.

Diplom/js/

Каталог з основною бібліотекою додатків.

3.3 Обґрунтування вибору програмної реалізації

При проектуванні системи було вивчено та проаналізовано предметну область.

Після ретельного аналізу було вирішено розроблювати програмний продукт, який заснований на веб-технологіях для використання за допомогою веб-браузера.

Технології, які використовуються на сервері, були обрані за принципом зручності у використанні, актуальності в наш час та можливістю виконання на будь-якій операційній системі. [15]

Мова JavaScript була обрана через те, що вона є прототипною скриптовою мовою програмування з динамічною типізацією.

Також важливою властивістю є велика швидкість роботи, надійність та кількість вбудованих можливостей.

Також для створення системи була обрана технологія AJAX через її можливість використовувати API-запит веб-клієнта (браузера) до веб-сервера за протоколом HTTP у фоновому режимі.

Дані технології в сукупності дають змогу збудувати якісний та надійний продукт, який захищений від патентних позовів з боку розробників, бо всі ці технології покриті ліцензіями, які виключають таку можливість і надають доступ до вихідних кодів даних проектів.

3.4 Встановлення серверу

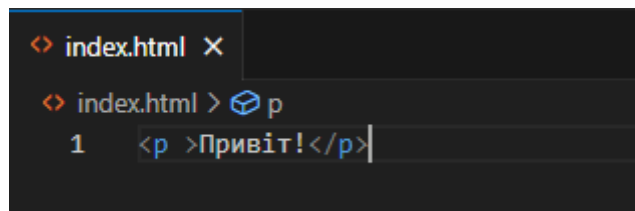
Для початку нам потрібен локальний сервер, на якому ми розмістимо наш додаток. В Open Server – це питання однієї хвилини, потрібно тільки запустити платформу і відкрити панель керування, а після натиснути «Запустити».

Далі потрібно створити директорію на сервері щоб розмістити там проект. Для цього у папці Domains, що розташована в дерикторії Open Server створюємо папку diplom, що і буде доменом додатку на локальному сервері

Далі потрібно перезапустити сервер.

Оскільки налаштування серверу не змінювалися, то початково, при посиланні на домен, відкриється файл index.html.

Для перевірки роботи виведемо на екран привітання: «Привіт!»



```
< index.html X
< index.html > p
1  <p>Привіт!</p>
```

Рисунок 3.3 – Код привітання

Отже тепер перевіримо працездатність серверу та домену, ввівши у адресний рядок браузера «constructor».

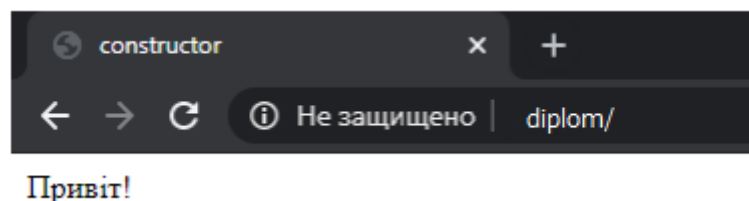


Рисунок 3.4 – Перевірка роботи серверу

3.5 Моделювання процесу роботи веб-додатку

3.5.1 Діаграма нульового рівня. Опис акторів і варіантів використання

Діаграма нульового рівня розглядається як функціональний блок із усіма відповідними робочими та керуючими об'єктами. [17]

Ця діаграма також відображає всі необхідні дані та вхідну інформацію, яка використовується для замовлення одягу.

Діаграма нульового рівня відображена на рис. 3.5.

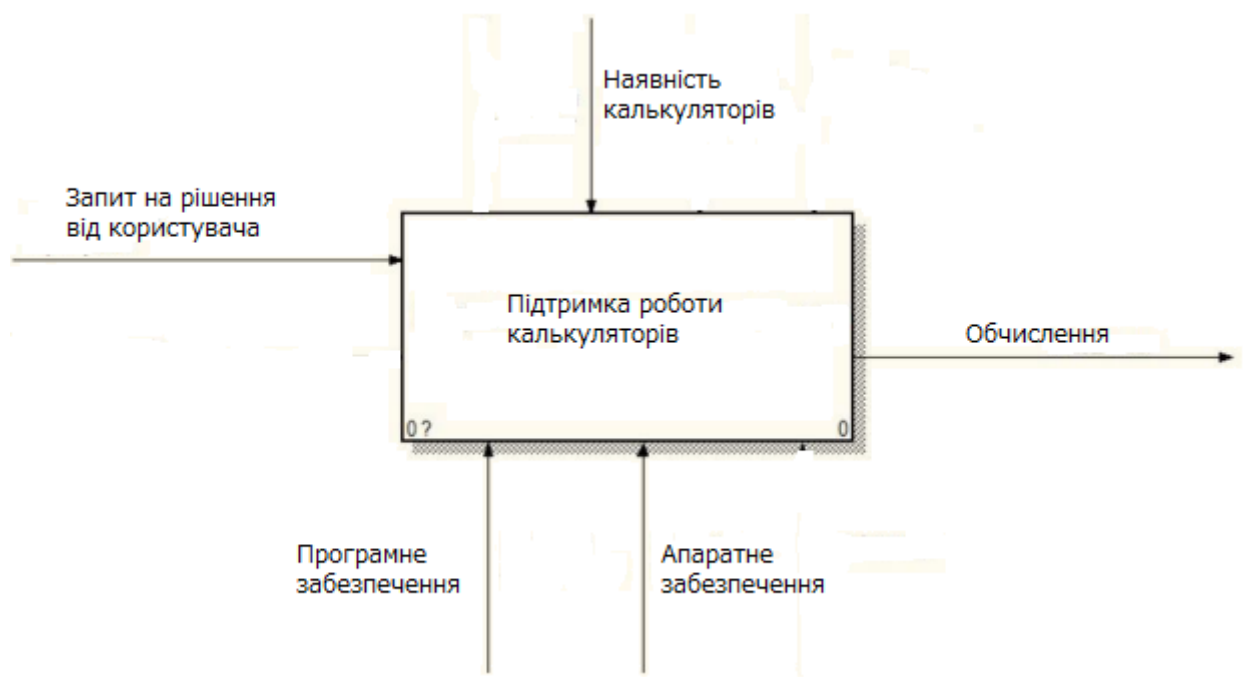


Рисунок 3.5 – Діаграма нульового рівня

Таблиця 2.1 – Опис акторів

Назва	Опис
Актори-користувачі	
Користувач	Користувач калькуляторів.

Таблиця 2.2 – Опис варіантів використання

Назва	Опис
Перегляд калькуляторів	Користувач має можливість переглянути калькулятори у веб додатку.
Редагування даних	Користувач має можливість редагувати дані калькуляторів.
Обчислення	Користувач має можливість обчислити дані калькуляторів.

3.6 Опис функцій веб-додатку

У додатку Г наведен код сторінок веб-додатку.

Розберемося з функціями кожної з них:

Лінійна інтерполяція (1.html)

Обчислює лінійну інтерполяцію

Функція `checknum`:

Відповідає за перевірку усіх значень, щоб вони були правильні, і якщо все добре, то воно зберігає їх для наступного обчислення.

Якщо, допустимо замість значення будуть дорівнюватися “NaN”, то функція повертає значення між 0 та `dd.length-1`.

Функція `compute`:

Відповідає вже за само обчислення. Сканує дані 5 інпутів по `id`.

І вичислює через структуру `temp`, у якої срок життя обмежується одним запитом. І також через `id` інпута, повертає відповідь.

Функція `clearvalues`:

Відповідає за очищення значень. По `id` кожного інпута очищає їх.

Елементи комбінаторики – перестановки, розміщення, поєднання (2.html).

Обчислює комбінаторику – перестановки, розміщення, сполучення.

Функція `reshenie`:

Відповідає за рішення. Створюємо декілька змінних, після чого зчитуємо k, n з інпутів для вводу. Якщо $k > n$ то з'являється помилка. Округлюємо їх. Після чого виконується обчислення і значення приходить в інпут по `id`.

Функція `factorial`:

Відповідає за факторіал. Звичайна функція для легкості написання. Завдяки неї можна не писати кожен раз обчислення факторіалу, а визвати на прикладі:

`P = factorial(n);`

Розкладання функції в ряд Фур'є (3.html)

Розкладує функції в ряд Фур'є

Функція `mkFourier`:

Відповідає за розклад функції. Створюється чотири змінні:

- `pa` – позитивне ціле число,
- `pz` – час,
- `dn` – $2p/T$,
- `htm` – відповідь.

Після, перевіряє введені значення. З чого починається, з мінуса чи чередується. У вигляді суми чи різності. І тут допомагає швидка перевірка, яка виконується у вигляді:

`let result = условие ? значение1 : значение2;`

Після, `htm` виводиться у інпут `out`.

Середнє арифметичне, Дисперсія, Варіація (4.html).

Знаходить середнє арифметичне, дисперсію, варіацію

Функція `myRound`:

Відповідає також за облегчення обчислення. Щоб не писати кожен раз одне і теж. Створив цю функцію.

Функція `myRoundA`:

Аналогічно з попередньою.

Функція Vubor:

Створюємо масив a_i . Після чого підкидуємо усі 30 значень масиву. А потім через цикли і умови обчислюємо задачі.

Рішення матриць (5.html)

Дії над матрицями.

Содержить купу назв кнопок, подробиць до дій, помилок.

Також купу функцій, які з самого початку повинні були бути поодинці як різні сторінки, але сміг об'єднати усе в одно велике.

Отже отримали не декілька калькуляторів, а один великий.

Розберемося в декількох його функціях.

Функція transformExponentiation:

Використовує власно ручно створені функції

BigInteger для exponentiation або на українському піднесення до степеня.

Паралельність прямих на площині (6.html)

Обчислює паралельність прямих

Функція calculate:

Створюється два нових масиви, у які записуються значення x і y . А далі у свої інпути записуються формули, кут похилу, пересічення з y , формула для паралельної прямої, формула для перпендикулярної прямої.

Рішення логарифмів (7.html)

Вирішує логарифми

Функція round:

Створена для округлення чисел.

Функція disable:

Натискаючи на radio інпут, цілі, вона виключається. І стає недоступною для змінення. А також це означає що тепер буде обчислювати це значення.

Функція logCalc:

Перевіряє який інпут виключений, аби обчислити його.

Тригонометричні функції (8.html).

Вирішує тригонометричні функції

Функція alg:

Виконується первинне обчислення, для застосування в інших функціях.

Функція get_alg1:

Виконується обчислення кута в градусах x_1 , через число Π , та математичні функції Math. Значення також переводяться із градусного заходу в радіану і навпаки. А також обчислюються основні тригонометричні функції.

Чотири функції задають активність відповідних полів

Функція get_alg2:

Аналогічно, тільки обчислює кут в градусах x_2 .

Функція get_algr1:

Обчислює кут в радіанах x_1 .

Функція get_algr2:

Обчислює кут в радіанах x_2 .

Функція clear:

Видаляє усі значення усіх інпутів.

Метод найменших квадратів (9.html).

Метод найменших квадратів.

Функція chesnum:

Використовується для інпутів. Забороняє вводити букви, чи якісь непотрібні символи.

Функція disp:

Використовується для створення нових пар x і y .

Функція calc:

Використовується для обчислення кутового коефіцієнту m , у пересічення, рівняння лінії y .

Також видає помилку, якщо одна із змінних не введена.

Функція `rst`:

Видаляє всі значення із інпутів.

Сума арифметичної прогресії (10.html).

Сума арифметичної прогресії

Функція `reshenie`:

Відповідає за обчислення значень. А також за відображення помилок і результатів.

Функція `Power`:

Возводить в степінь.

3.7 Вимоги до програмного забезпечення

Програмне забезпечення клієнтської частини повинне задовольняти наступним вимогам:

- Веб-браузер: Internet Explorer 7.0 і вище, або Firefox 7.5 і вище, або Opera;
- 9.5 і вище, або Safari 6.1 і вище, або Chrome 7 і вище.

4 ОГЛЯД ВЕБ-ДОДАТКУ. ТЕСТУВАННЯ

4.1 Огляд веб-додатку

При відкритті додатку в браузері відображається головна сторінка, на шапці якої розташований логотип, який є кнопкою до головної сторінки (рис. 4.1).

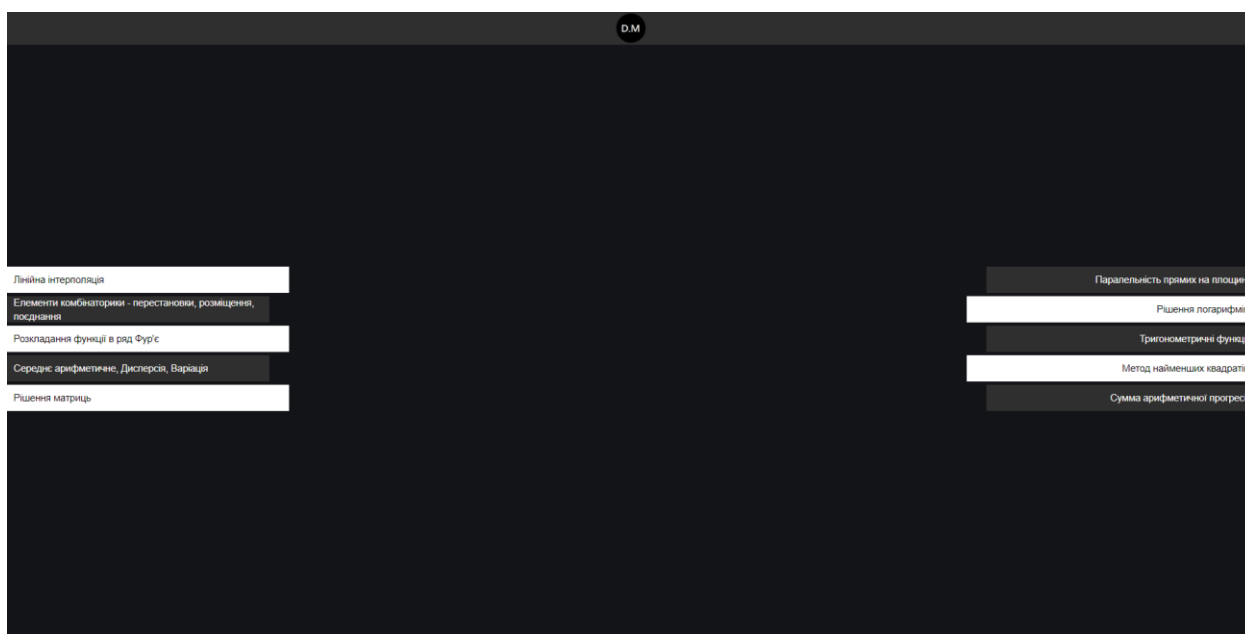


Рисунок 4.1 – Головна сторінка додатку

По праву та ліву сторону розташована навігаційна панель, яка оформлена в різних стилях, також при наведенні різні ефекти. Побачити це можна на рисунках 4.2 і 4.3.

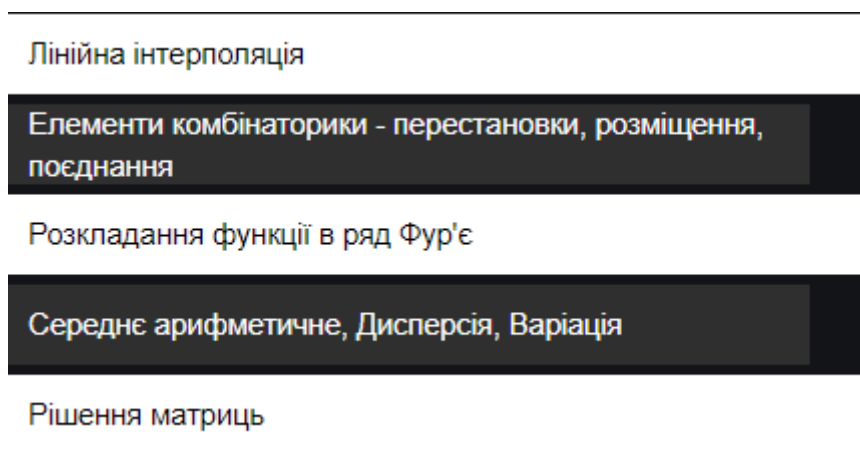


Рисунок 4.2 – Ефект при наведенні на сірий блок

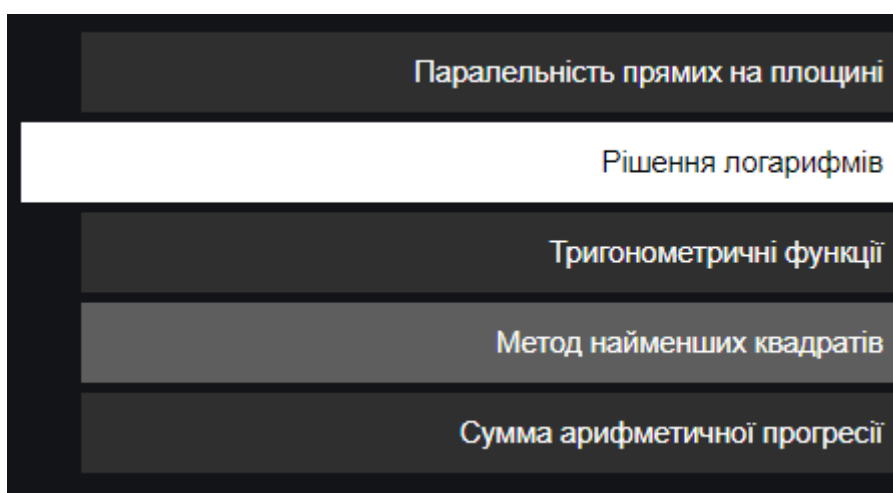


Рисунок 4.3 – Ефект при наведенні на білий блок

Самі ж калькулятори опущені небагато нижче, аби при листанні вниз, вони були на весь екран (рис. 4.4, 4.5).



Рисунок 4.4 – Калькулятор матриць

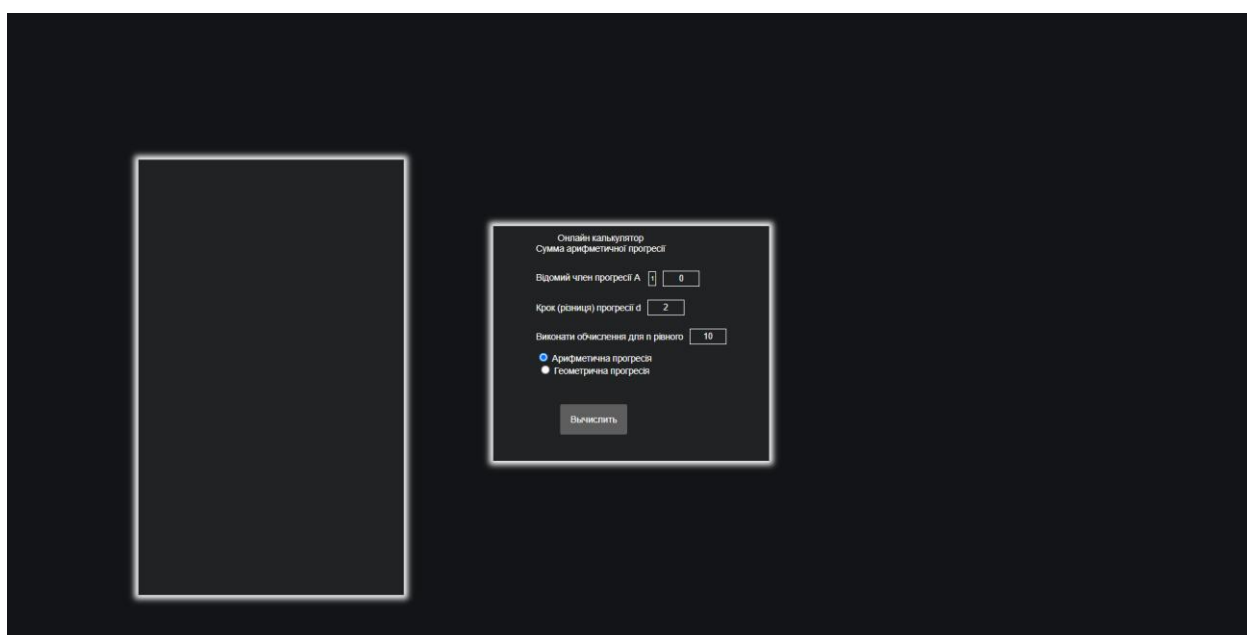


Рисунок 4.5 – Калькулятор арифметичної прогресії

4.2 Тестування

Тепер почнемо тести. Виберемо на головній сторінці калькулятор логарифму і перейдемо на його сторінку.

Для початку обчислемо задачу без використання додатку.

$$\log_2 8 = 3$$

Так, як $2^3 = 8$

$$\log_7 49 = 2$$

Так, як $7^2 = 49$

Тепер подивимось на веб-додаток.

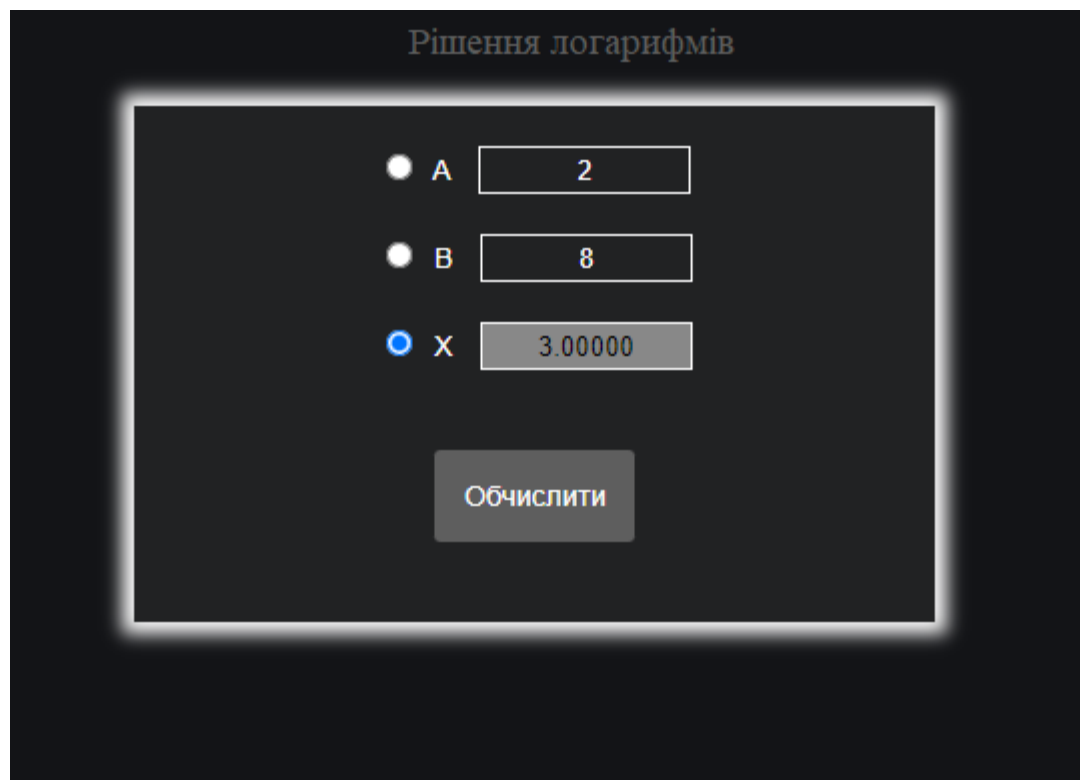


Рисунок 4.6 – Рішення першого логарифму

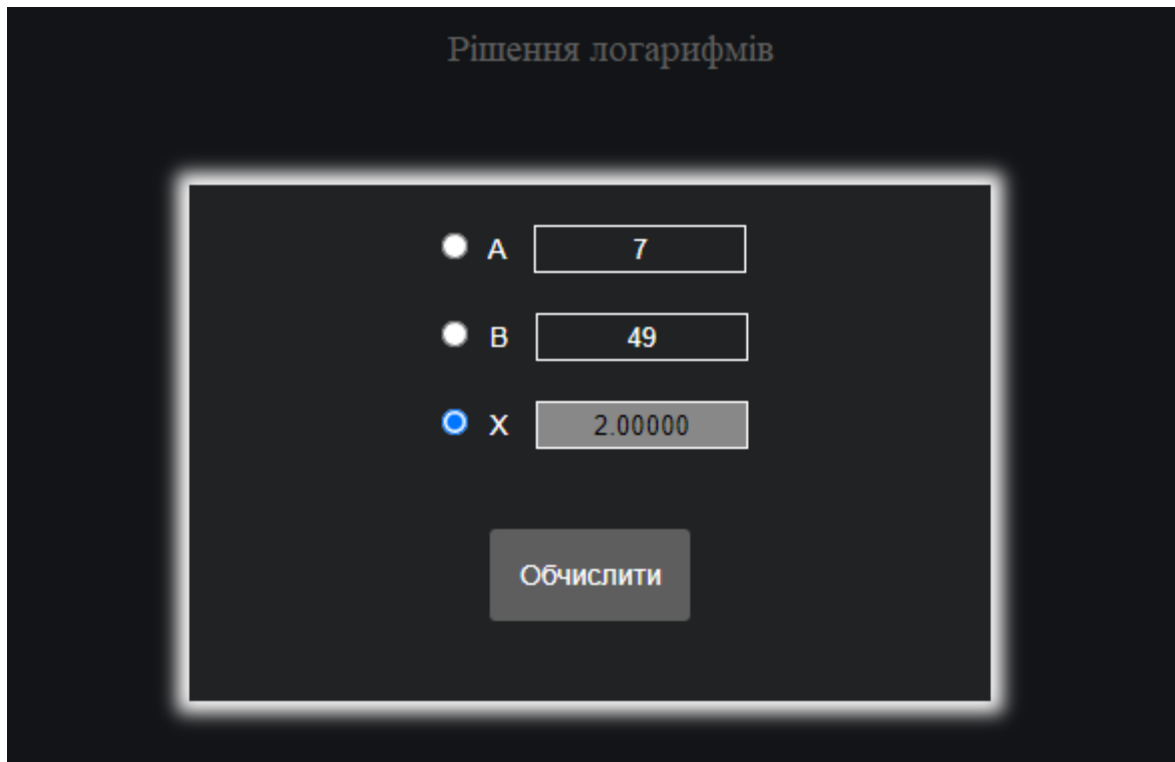


Рисунок 4.7 – Рішення другого логарифму

Як можна побачити, цей калькулятор, працює правильно.

Перевіримо лінійну інтерполяцію.

Приклад.

Необхідно знайти коефіцієнт переходу - від ваги снігового покриву на горизонтальній поверхні землі, до нормативного навантаження на покриття, для кута нахилу покрівлі - 35° . Відомо, що для кута нахилу покрівлі 30° коефіцієнт $C = 1$, а для кута нахилу покрівлі 60° коефіцієнт $C = 0$.

$$y = ?$$

$$x = 35$$

$$x_1 = 30$$

$$x_2 = 60$$

$$y_1 = 1$$

$$y_2 = 0$$

$$\text{Отримуємо відповідь: } 1 + \left(\frac{(35-30)}{60-30} * (0 - 1) \right) = 0,8333.$$

Тепер перевіряємо додаток.

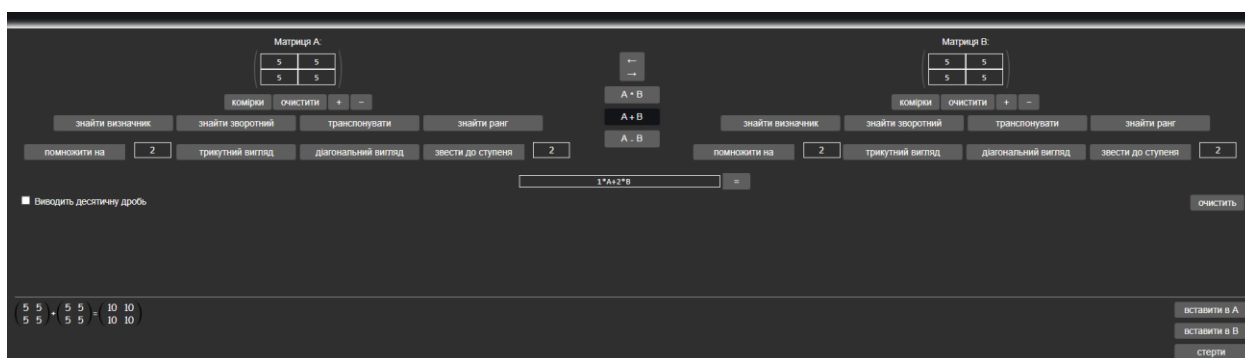


Рисунок 4.10 – Додавання матриць

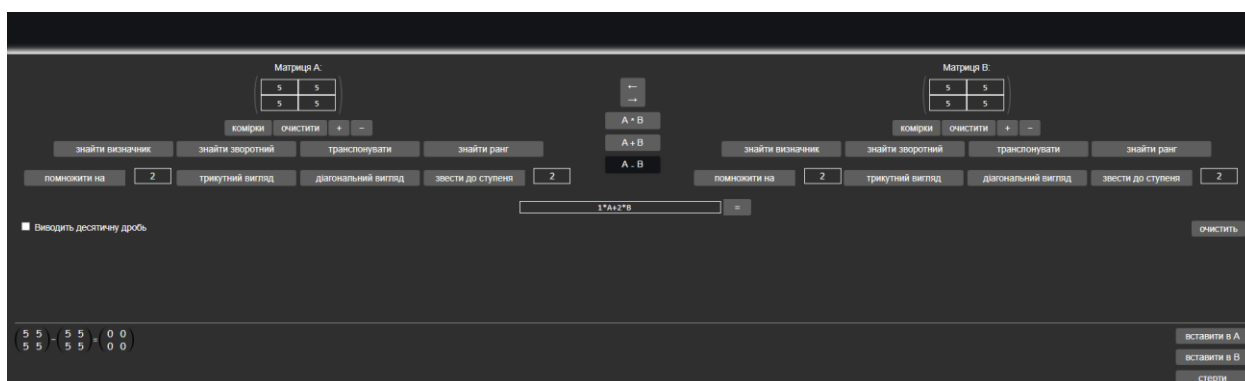


Рисунок 4.11 – Віднімання матриць

Як можна побачити, з цими легкими діями, додток впорався. Пройдемося тепер по більш тяжким функціям.

Знайдемо визначник матриці:

$$\begin{bmatrix} 12 & 7 \\ 21 & 5 \end{bmatrix} = 12 * 5 - 21 * 7 = 60 - 147 = -87$$

Подивимось як з цим впорався додток.

Матриця A:

12	7
21	5

комірки очистити + -

знайти визначник знайти зворотний транспонувати знайти ранг

помножити на 2 трикутний вигляд діагональний вигляд звести до ступеня 2

☐ Виводить десятичну дробь

$\begin{vmatrix} 12 & 7 \\ 21 & 5 \end{vmatrix} = -87$
[► Подорожчії](#)

Рисунок 4.12 – Визначник матриці

Як можна побачити, така ж сама відповідь.

Йдемо далі, тепер перевіримо комбінаторику.

Приклад:

Скількома способами можна розставити 15 томів на книжковій полиці, якщо вибрати їх з наявних 30-ти книг?

Визначимо загальну кількість розміщень із 30 елементів по 15 за формулою:

$$A_{30}^{15} = 30 * 29 * 28 * \dots * (30 - 15 + 1) = 30 * 29 * 28 * \dots * 16$$

$$= 202843204931727360000$$

Перевіримо додаток.

k
 n

 Число перестановок $P_{30} = 2.6525285981219103e+32$
 Число розміщень $A_{30}^{15} = 202843204931727330000$
 Число поєднань $C_{30}^{15} = 155117520$

Рисунок 4.13 – Число розміщень комбінаторики

Як можемо побачити, все у нормі. На останнє перевіримо число поєднань.

Приклад:

$$C_3^1 = \frac{3!}{2!*1!} = \frac{6}{2*1} = 3$$

А зараз додаток.

к

п

Обчислити

Число перестановок $P_3 = 6$
Число розміщень $A_3^1 = 3$
Число поєднань $C_3^1 = 3$

Рисунок 4.14 – Число поєднань комбінаторики

Нехай, подивились не усі калькулятори, але інші працюють не гірше.
То що, усі тести вдалі, непогані результати.

ВИСНОВКИ

За результатами аналізу наукової літератури було детально розібрано поняття веб-додатку.

Розглянуто питання стосовно різниці веб-сайта та веб-додатка. В висновку до даного питання можна сказати, що веб-додаток це той же динамічний сайт, але з більшим нахилом на інтерактивність ніж на інформативну частину. Було визначено принципи роботи прикладних програм та веб-додатків.

Веб-додатки є веб-орієнтованими програмами, а тому вони не прив'язані до певної обчислювальної машини, чи до певної операційної системи, на відміну від ПЗ, вони можуть працювати практично на всіх пристроях без прив'язки до певного місця.

Був створений повноцінний веб-додаток. В практичній частині реалізовано всі поставлені задачі: створено вебдодаток калькуляторів. Реалізовано на даний момент 10 штук працюючих моделей калькуляторів:

- Лінійна інтерполяція;
- Елементи комбінаторики – перестановки, розміщення, поєднання;
- Розкладання функції в ряд Фур'є;
- Середнє арифметичне, Дисперсія, Варіація;
- Рішення матриць;
- Паралельність прямих на площині;
- Рішення логарифмів;
- Тригонометричні функції;
- Метод найменших квадратів;
- Сума арифметичної прогресії.

В лінійній інтерполяції було використовувана функція `parseFloat` яка преобразує строку в число:

```
function compute()
{
    var x1=parseFloat(document.getElementById('x1').value);
    var x2=parseFloat(document.getElementById('x2').value);
    var x3=parseFloat(document.getElementById('x3').value);
    var y1=parseFloat(document.getElementById('y1').value);
    var y2=parseFloat(document.getElementById('y2').value);
    var y3,temp;
    temp=((x3-x1)*(y2-y1))/(x2-x1);
    y3=temp+y1;
    document.getElementById('y').value=y3;
}
```

Для прив'язки інпутів до якогось значення, я використав властивість form документа:

```
function calculate() {
    var y = new Array();
    y[0] = null;
    y[1] = document.form.y1.value;
    y[2] = document.form.y2.value;
    var x = new Array();
    x[0] = null;
    x[1] = document.form.x1.value;
    x[2] = document.form.x2.value;
    var m = (y[2] - y[1]) / (x[2] - x[1]);
    var b = y[1]-(m * x[1]);
    var formula = "y="+m+"x + "+b;
    var rand = parseInt(Math.random() * 7);
    var parf = "y="+m+"x + "+rand;
    var newm = m / -1;
    var perf = "y="+newm+"x + "+b;
    document.form.formula.value = formula;
    document.form.slope.value = m;
    document.form.yint.value = b;
    document.form.parf.value = parf;
    document.form.perf.value = perf;
}
```

Щоб облегчити вичислення елементів комбінаторики, я написав додаткову функцію для факторіалів, коли треба було провести дію над факторіалом,

використовував її:

```
function factorial(G){  
    var FCT=1;  
    if (G==0) {return FCT;}  
    for(ii=1; ii<=G; ii++)  
        {FCT=ii*FCT;}  
    return FCT;  
}
```

Відповідно до практичної частини можна зробити висновок, що маючи певний набір універсальних програмних модулів та компонентів, чи таких, що легко змінюються під потреби тієї чи іншої предметної області, можна швидко та якісно розробляти програмні комплекси в певній галузі.

Одержані результати дослідження дають підстави вважати, що завдання реалізовані, мета досягнута. Але є деякі сумніви про дизайн, хотілось би зробити краще, але на дизайнера не вчився.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кращі мови програмування 2020 року, які варто вивчати. URL: <https://merehead.com/ru/blog/popular-programming-languages-2020/>.
2. Що таке CSS. URL: <http://phpist.com.ua/css/5-whatcss/>.
3. Кращі CMS платформи для запуску веб-сайту в 2020 році. URL: <https://www.hostinger.com.ua/rukovodstva/luchshie-cms-platformy-2019/>.
4. Діаграми UML. URL: <https://planerka.info/item/diagrammy-kommunikacij-uml/>.
5. The 2020 Roadmap To Fullstack Web Development. URL: <https://codingthesmartway.com/the-2020-roadmap-to-fullstack-web-development/>
6. Що таке CMS? URL: <https://wiki.rookee.ru/cms/>.
7. How To Become A Web Developer — Everything You Need To Know. URL: <https://careerfoundry.com/en/blog/web-development/what-does-it-take-to-become-a-web-developer-everything-you-need-to-know-before-getting-started/>.
8. Крокфорд Дуглас. JavaScript. Сильні сторони, 2016. 176 с.
9. Хавербеке Марейн. Виразний JavaScript. 2 видання, 2015. 745 с.
10. Платформа як послуга. URL: <http://www.softline.kiev.ua/ua/khmarni-poslugi/poslugi-ukhmarakh/platforma-yak-posluga.html> .
11. svitla.com URL: <https://svitla.com/blog/web-application-architecture>
12. tutorialspoint.com URL: https://www.tutorialspoint.com/css/what_is_css.htm
13. Веб застосунок. URL: <http://uk.wikipedia.org/wiki/Веб-застосунок>— Веб застосунок.
14. Платформа як послуга. URL: <http://www.softline.kiev.ua/ua/khmarni-poslugi/poslugi-ukhmarakh/platforma-yak-posluga.html> — Платформа як послуга.

15. Реферат на тему: «Засоби створення Web-додатків». URL:
<http://ifreestore.net/2267>
16. В чем отличие сайта от веб-приложения? URL:
<http://evriqum.ru/work/web-applications> – В чем отличие сайта от веб-приложения?
17. Всеукраїнська науково-практична Internet-конференція. Автоматизація та комп'ютерно-інтегровані технології у виробництві та освіті: стан, досягнення, перспективи розвитку. URL:
http://conference.ikto.net/pub/akit_2015_16-20march.pdf
18. Сайт URL: 66 <https://ru.wikipedia.org/wiki/>
19. От Web-сайтов к Web-приложениям: Часть 1. Web-сайт или Webприложение? URL: <http://www.ibm.com/developerworks/ru/library/wa-websiteapp/> – От Websайтов к Web-приложениям: Часть 1. Web-сайт или Web-приложение?
20. Автоматизація. URL: <http://uk.wikipedia.org/wiki/Автоматизація> – Автоматизація.

Додаток Б

Матеріали презентації

ПОСТАНОВКА ЗАДАЧІ

- Дипломний проект призначений для того, щоб користувач міг з легкістю обрати для калькулятор і швидко вирішити його задачу.
- Калькулятор повинен бути зроблений як веб-додаток, доступний в Інтернеті. Додаток повинен складатись з взаємозалежних частин, а функції кожної частини повинні бути чітко розділені.
- Даний додаток повинен мати наступні можливості:
 - Огляд ступенів розв'язання задачі.
 - Калькулятори.
- Сайт повинен складатись з користувацької частини.
- Користувацький інтерфейс веб-додатку повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів та сторінок.
- Наявність навігації на всіх сторінках сайту. Правильна структура інформації дозволить користувачам без проблем продовжувати дослідження додатку, залишаючись впевненими в тому, що вони завжди зможуть без зусиль повернутися до раніше переглянутих сторінок.
- Для досягнення мети кваліфікаційної роботи потрібно вирішити такі задачі:
 - – Вивчити процеси реалізації задачі.
 - – Провести аналіз додатків-аналогів для розроблення технічного завдання.
 - – Обрати та налаштувати інструменти реалізації.
 - – Визначити структуру даних та спроектувати веб-додаток.
 - – Реалізувати додатки та розробити інструкції користувача.

АРХІТЕКТУРА

- Веб-додаток буде складатись з однієї частини, а саме користувацької. Користувацький інтерфейс веб-сайту повинен бути зрозумілим, інтуїтивно представляти структуру розміщеної на ньому інформації та швидко та логічно переходити до розділів.
- Веб-додаток повинен складатись з наступних розділів:
 - – Головна сторінка
 - – містити такі сторінки калькуляторів:
 - Лінійна інтерполяція;
 - Елементи комбінаторики - перестановки, розміщення, поєднання;
 - Розкладання функції в ряд Фур'є;
 - Середнє арифметичне, Дисперсія, Варіація;
 - Рішення матриць;
 - Паралельність прямих;
 - Рішення логарифмів;
 - Тригонометричні функції;
 - Метод найменших квадратів;
 - Сума арифметичної прогресії;
 - – сторінка з інформацією о калькуляторах
 - – сторінка з інформацією о веб-додатку

ОГЛЯД

При відкритті додатку в браузері відображається головна сторінка, на шапці якої розташований логотип, який є кнопкою до головної сторінки, побачити можна на рисунку 4.1.

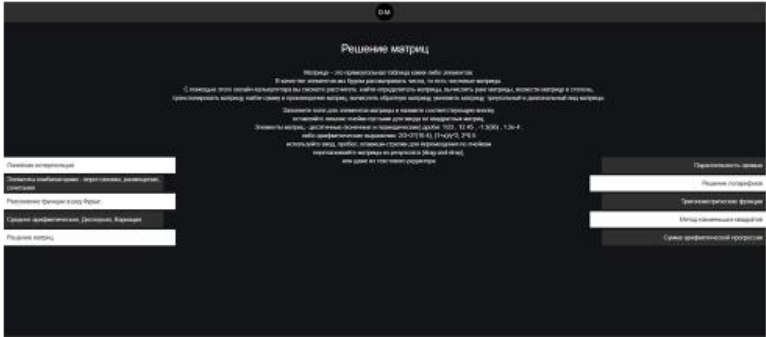
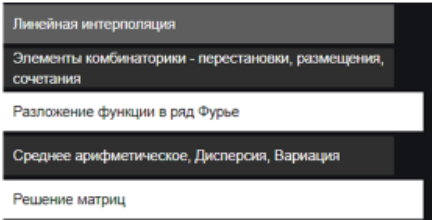
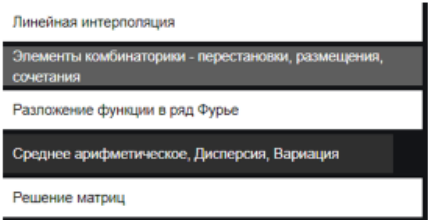
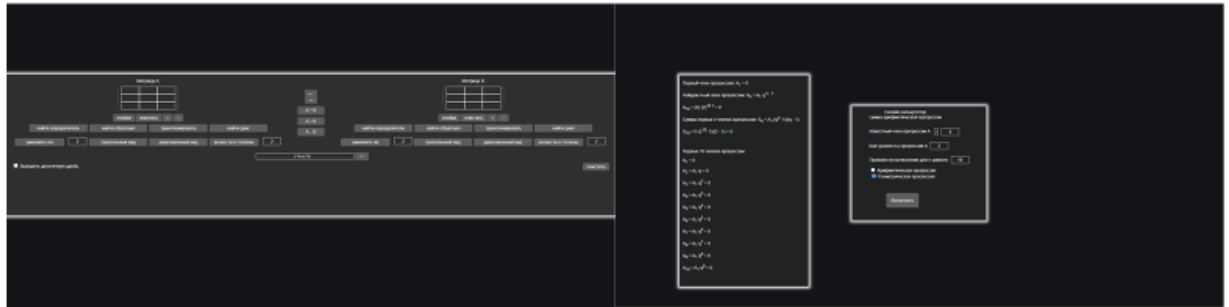


Рисунок 4.1 – Головна сторінка додатку

- По праву, та ліву сторону розташована навігаційна панель, яка оформлена в різних стилях, також при наведенні, різні ефекти.



- Самі ж калькулятори опущені небагато нижче, аби при листанні вниз, вони були на весь екран.



ТЕСТУВАННЯ

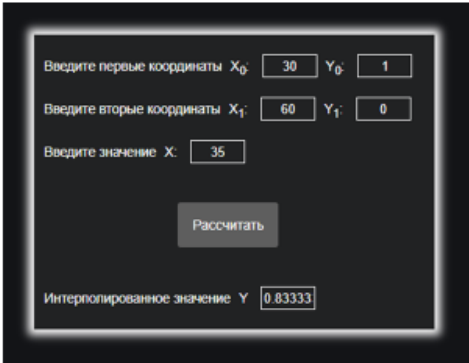
- Якби калькулятори не виглядали, але ж вони повинні гарно і правильно працювати, зараз це і протестуємо.
- Почнемо з логарифмів:
-
- $\log_2 8 = 3$
- Так, як $2^3 = 8$
-
- $\log_7 49 = 2$
- Так, як $7^2 = 49$

☐ A
☐ B
☒ X

☐ A
☐ B
☒ X

- Перевіримо лінійну інтерполяцію.
- Приклад:
- Необхідно знайти коефіцієнт переходу - від ваги снігового покриву на горизонтальній поверхні землі, до нормативного навантаження на покриття, для кута нахилу покрівлі - 35° . Відомо, що для кута нахилу покрівлі 30 коефіцієнт $C = 1$, а для кута нахилу покрівлі 60 коефіцієнт $C = 0$.
- $y = ?$
- $x = 35$
- $x_1 = 30$
- $x_2 = 60$
- $y_1 = 1$
- $y_2 = 0$
- Отримуємо відповідь: $1 + \left(\frac{(35-30)}{60-30} * (0 - 1) \right) = 0,8333$

Тепер перевіряємо додаток.



The image shows a software interface for linear interpolation. It has a dark background with white text and input fields. The interface includes labels for inputting two points (X_0, Y_0) and (X_1, Y_1) , a value X for interpolation, a 'Рассчитать' (Calculate) button, and the resulting interpolated value Y .

Input	Value
Введите первые координаты X_0	30
Y_0	1
Введите вторые координаты X_1	60
Y_1	0
Введите значение X	35
Интерполированное значение Y	0.83333

Додаток В

Фрагмент лістингу програми

Index.html

```

<html>
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Інтернет Додаток Калькулятор</title>
    <link rel="stylesheet" href="../css/style.css">
  </head>
  <body>
    <div class="calculator__wrapper calculator__wrapper-index">
      <div class="calculator__header-wrapper">
        <div class="calculator__header">
          <a href="index.html">
            
          </a>
        </div>
      </div>
      <div class="calculator">
        <div class="calculator__menu-wrapper">
          <div class="calculator__left-menu">
            <a href="../components/1.html" class="calculator__left-
menu-items-selected">
              <p class="calculator__left-menu-items-title">
                Лінійна інтерполяція
              </p>
            </a>
            <a href="../components/2.html" class="calculator__left-
menu-items">
              <p class="calculator__left-menu-items-title">
                Елементи комбінаторики - перестановки, розміщення,
поєднання
              </p>
            </a>
            <a href="../components/3.html" class="calculator__left-
menu-items-selected">
              <p class="calculator__left-menu-items-title">
                Розкладання функції в ряд Фур'є
              </p>
            </a>
            <a href="../components/4.html" class="calculator__left-
menu-items calculator__left-menu-items">
              <p class="calculator__left-menu-items-title">
                Середнє арифметичне, Дисперсія, Варіація
              </p>

```

```

        </a>
        <a href="../../components/5.html" class="calculator__left-
menu-items-selected">
            <p class="calculator__left-menu-items-title">
                Рішення матриць
            </p>
        </a>
    </div>
    <div class="calculator__right-menu">
        <a href="../../components/6.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Паралельність прямих на площині
            </p>
        </a>
        <a href="../../components/7.html" class="calculator__right-
menu-items-selected">
            <p class="calculator__right-menu-items-title">
                Рішення логарифмів
            </p>
        </a>
        <a href="../../components/8.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Тригонометричні функції
            </p>
        </a>
        <a href="../../components/9.html" class="calculator__right-
menu-items-selected">
            <p class="calculator__right-menu-items-title">
                Метод найменших квадратів
            </p>
        </a>
        <a href="../../components/10.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
</div>
</body>
</html>

```

1.html

```

<!doctype html>
<html lang="ru">

```

```

<head>
  <title>Лінійна інтерполяція | Онлайн калькулятор</title>
  <link type="text/css" rel="stylesheet" media="all"
href="../css/style.css">
</head>
<script type="text/javascript">
  function checnum(as)
  {
    var dd = as.value;

    if(isNaN(dd))
    {
      dd = dd.substring(0,(dd.length-1));
      as.value = dd;
    }
  }
  function compute()
  {
    var x1=parseFloat(document.getElementById('x1').value);
    var x2=parseFloat(document.getElementById('x2').value);
    var x3=parseFloat(document.getElementById('x3').value);
    var y1=parseFloat(document.getElementById('y1').value);
    var y2=parseFloat(document.getElementById('y2').value);
    var y3,temp;
    temp=((x3-x1)*(y2-y1))/(x2-x1);
    y3=temp+y1;
    document.getElementById('y').value=y3;

  }
  function clearvalues()
  {
    document.getElementById('x1').value='';
    document.getElementById('x2').value='';
    document.getElementById('x3').value='';
    document.getElementById('y1').value='';
    document.getElementById('y2').value='';
    document.getElementById('y').value='';

  }
</script>

<body>
  <div class="calculator__wrapper">
    <div class="calculator__header-wrapper">
      <div class="calculator__header">
        <a href="../index.html">
          
        </a>

```



```

        </div>
    </div>
    <div class="calculator">
        <div class="calculator__menu-wrapper">
            <div class="calculator__left-menu">
                <a href="../components/1.html" class="calculator__left-
menu-items-selected">
                    <p class="calculator__left-menu-items-title">
                        Лінійна інтерполяція
                    </p>
                </a>
                <a href="../components/2.html" class="calculator__left-
menu-items">
                    <p class="calculator__left-menu-items-title">
                        Елементи комбінаторики - перестановки,
розміщення, поєднання
                    </p>
                </a>
                <a href="../components/3.html" class="calculator__left-
menu-items-selected">
                    <p class="calculator__left-menu-items-title">
                        Розкладання функції в ряд Фур'є
                    </p>
                </a>
                <a href="../components/4.html" class="calculator__left-
menu-items calculator__left-menu-items">
                    <p class="calculator__left-menu-items-title">
                        Середнє арифметичне, Дисперсія, Варіація
                    </p>
                </a>
                <a href="../components/5.html" class="calculator__left-
menu-items-selected">
                    <p class="calculator__left-menu-items-title">
                        Рішення матриць
                    </p>
                </a>
            </div>
            <div class="calculator__right-menu">
                <a href="../components/6.html" class="calculator__right-
menu-items calculator__right-menu-items">
                    <p class="calculator__right-menu-items-title">
                        Паралельність прямих на площині
                    </p>
                </a>
                <a href="../components/7.html" class="calculator__right-
menu-items-selected">
                    <p class="calculator__right-menu-items-title">
                        Рішення логарифмів
                    </p>
                </a>
            </div>
        </div>
    </div>

```

```

        <a href="../components/8.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Тригонометричні функції
            </p>
        </a>
        <a href="../components/9.html" class="calculator__right-
menu-items-selected">
            <p class="calculator__right-menu-items-title">
                Метод найменших квадратів
            </p>
        </a>
        <a href="../components/10.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
</div>
<div class="calculator__body" id="content">
    <p class="calculator__body-title" id="cat-200">
        Лінійна інтерполяція
    </p>
    <div class="calculator__body-content">
        <div class="calculator__body-description">
            <p>
                Інтерполяція в загальному розумінні - це метод
обчислення певних проміжних значень будь-якої досліджуваної величини з набору
відомих величин.

                <br />
                Якщо досліджуваний процес можна описати лінійною
функцією, процедура обчислення невідомих параметрів в порівнянні з іншими
випадками обчислення може бути істотно спрощена.

                <br />
                Математичне моделювання різних виробничих ситуацій
інженерної та наукової практики методами лінійної інтерполяції передбачає
можливість

                <br />
                математичне прогнозування шляхом виявлення значення
інтерпольованої координати Y заданим параметром координати X при відомих
координатах двох точок лінійної функції.
            </p>
        </div>
        <div class="calculator__body-description">
            <p>
                Для успішного управління потрібно передбачити, як
поведе себе та чи інша система в рамках існуючого процесу, що описується
відповідною лінійною функцією.

```

```

        <br />
        Перша точка лінійної функції має координати X0,Y0,
        друга - X1,Y1, обчислена ефективна інтерпольована координата <b>Y</b> із заданого
        значення координати X обчислюється за формулою  $((X - X0) \times (Y1 - Y0) / (X1 - X0)) + Y0$ .<b></b><b></b>
        <br />
        У практиці розрахунків іноді виникає необхідність
        порівняння отриманого значення параметрів третьої точки (X, Y) по прямій з деяким
        граничним значенням <b>X lim, Y lim</b>.<b></b>
        <br>
    </p>
</div>
<form class="calculator__body-calculator-wrapper" name="form"
style="min-width:390px">
    <table class="my_table">
        <tr>
            <td class="calculator__body-calculator-title">
                Введіть перші координати
                <span class="calculator__body-calculator-
title-decoration">
                    X<sub>0</sub>:
                </span>
                <input class="calculator__body-calculator-
input" type="text" id="x1" onkeyup="checnum(this)" size="4">
                <span class="calculator__body-calculator-
title-decoration">
                    Y<sub>0</sub>:
                </span>
                <input class="calculator__body-calculator-
input" type="text" id="y1" onkeyup="checnum(this)" size="4">
            </td>
        </tr>
        <tr>
            <td class="calculator__body-calculator-title">
                Введіть другі координати
                <span class="calculator__body-calculator-
title-decoration">
                    X<sub>1</sub>:
                </span>
                <input class="calculator__body-calculator-
input" type="text" id="x2" onkeyup="checnum(this)" size="4">
                <span class="calculator__body-calculator-
title-decoration">
                    Y<sub>1</sub>:
                </span>
                <input class="calculator__body-calculator-
input" type="text" id="y2" onkeyup="checnum(this)" size="4">
            </td>
        </tr>
    </tr>

```

```

        <td class="calculator__body-calculator-title">
            Введіть значення
            <span class="calculator__body-calculator-
title-decoration">

                X:
            </span>
            <input class="calculator__body-calculator-
input" type="text" id="x3" onkeyup="checnum(this)" size="4">
        </td>
    </tr>
    <tr>
        <td class="calculator__body-calculator-button-
wrapper" colspan="3">
            <input class="calculator__body-calculator-
button" type="button" value="Рассчитать" onclick="compute()">
        </td>
    </tr>
    <tr>
        <td class="calculator__body-calculator-title"
id="my_color">
            Интерполяване значення
            <span class="calculator__body-calculator-
title-decoration">

                Y
            </span>
            <input class="calculator__body-calculator-
input" type="text" id="y" size="12" readonly>
        </td>
    </tr>
</table>
</form>
</div>
</div>
</div>
</body>
</html>

```

2.html

```

<!doctype html>
<html lang="ru">
    <head>
        <title>Елементи комбінаторики - перестановки, розміщення, поєднання | Онлайн
калькулятор</title>
        <link type="text/css" rel="stylesheet" media="all" href="../../css/style.css">
    </head>
    <script>
        function reshenie() {

            var k,n,C,A,P,rnd;

```

```

    rnd=10000;
    var outputstring = "<br />";
    k=parseFloat(document.getElementById('inK').value);
    n=parseFloat(document.getElementById('inN').value);
    k=Math.round(k);
    n=Math.round(n);

    if (k>n)
    {alert("к не повинно бути більше n!");
    return;}

    P=factorial(n);
    A=P/factorial(n-k);
    C=P/factorial(k)/factorial(n-k);

    outputstring+="Число перестановок  $P_{n+}$  =
"+Math.round(rnd*P)/rnd+"<br />";
    outputstring+="Число розміщень  $A_{n+}^{k+}$  =
"+Math.round(rnd*A)/rnd+"<br />";
    outputstring+="Число поєднань  $C_{n+}^{k+}$  =
"+Math.round(rnd*C)/rnd+"<br />";

    document.getElementById('output').innerHTML = outputstring;
    return;
}

function factorial(G){
    var FCT=1;
    if (G==0) {return FCT;}
    for(ii=1; ii<=G; ii++)
    {FCT=ii*FCT;}
    return FCT;
}
</script>

<body>
    <div class="calculator__wrapper">
        <div class="calculator__header-wrapper">
            <div class="calculator__header">
                <a href="../index.html">
                    
                </a>
            </div>
        </div>
        <div class="calculator">
            <div class="calculator__menu-wrapper">
                <div class="calculator__left-menu">

```

```

        <a href="../components/1.html" class="calculator__left-menu-
items-selected">
            <p class="calculator__left-menu-items-title">
                Лінійна інтерполяція
            </p>
        </a>
        <a href="../components/2.html" class="calculator__left-menu-
items">
            <p class="calculator__left-menu-items-title">
                Елементи комбінаторики - перестановки, розміщення,
поєднання
            </p>
        </a>
        <a href="../components/3.html" class="calculator__left-menu-
items-selected">
            <p class="calculator__left-menu-items-title">
                Розкладання функції в ряд Фур'є
            </p>
        </a>
        <a href="../components/4.html" class="calculator__left-menu-
items calculator__left-menu-items">
            <p class="calculator__left-menu-items-title">
                Середнє арифметичне, Дисперсія, Варіація
            </p>
        </a>
        <a href="../components/5.html" class="calculator__left-menu-
items-selected">
            <p class="calculator__left-menu-items-title">
                Рішення матриць
            </p>
        </a>
    </div>
    <div class="calculator__right-menu">
        <a href="../components/6.html" class="calculator__right-menu-
items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Паралельність прямих на площині
            </p>
        </a>
        <a href="../components/7.html" class="calculator__right-menu-
items-selected">
            <p class="calculator__right-menu-items-title">
                Рішення логарифмів
            </p>
        </a>
        <a href="../components/8.html" class="calculator__right-menu-
items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Тригонометричні функції
            </p>

```

```

        </a>
        <a href="../../components/9.html" class="calculator__right-menu-
items-selected">
            <p class="calculator__right-menu-items-title">
                Метод найменших квадратів
            </p>
        </a>
        <a href="../../components/10.html" class="calculator__right-
menu-items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
</div>
<div class="calculator__body" id="main">
    <h1 class="calculator__body-title" id="cat-200">
        Елементи комбінаторики - перестановки, розміщення, поєднання
    </h1>
    <div class="calculator__body-content">
        <p class="calculator__body-description">
            Нехай необхідно виконати послідовно k
            дій. Якщо першу дію можна виконати n1
            <br />
            способами, другу – n2
            способами і так далі до k
            - ї дії, яку можна виконати nk
            способами, то всі k
            <br />
            дій можна виконати  $n1 \cdot n2 \cdot \dots \cdot nk$ 
            способами.
        </p>
        <p class="calculator__body-description">
            Вибором об'єктів і їх розташуванням в тому чи іншому порядку
            доводиться займатися практично у всіх сферах
            <br />
            людська діяльність, наприклад, дизайнер, який розробляє
            розподіл сільськогосподарських культур на декількох полях.
            <br />
            Цей онлайн-калькулятор обчислює наступні комбінаторні
            значення:
        </p>
        <div class="calculator__body-description">
            <ul>
                <li>
                    Число перестановок з n елементів: P
                    <sub>n</sub>
                    = n!
                </li>
            </ul>
        </div>
    </div>

```

```

<li>
    Число розміщень из n по k: A
    <sub>n</sub>
    <sup>k </sup>
    = n
    <span style="-webkit-border-horizontal-spacing: 2px;
-webkit-border-vertical-spacing: 2px" class="Apple-style-span">
        .
    </span>
    (n-1)
    <span style="-webkit-border-horizontal-spacing: 2px;
-webkit-border-vertical-spacing: 2px" class="Apple-style-span">
        .
    </span>
    (n-2)
    <span style="-webkit-border-horizontal-spacing: 2px;
-webkit-border-vertical-spacing: 2px" class="Apple-style-span">
        .
    </span>
    ...
    <span style="-webkit-border-horizontal-spacing: 2px;
-webkit-border-vertical-spacing: 2px" class="Apple-style-span">
        .
    </span>
    (n-k+1) = n!/(n-k)!
</li>
<li>
    Число поєднань з n по k: C
    <sub>n</sub>
    <sup>k</sup>
    = n! / [k!
    <span style="-webkit-border-horizontal-spacing: 2px;
-webkit-border-vertical-spacing: 2px" class="Apple-style-span">
        .
    </span>
    (n-k)!]
</li>
</ul>
<p style="text-align: center"></p>
<div id="rslokadv" style="position:absolute; margin-top:75px;
width: 360px;display:none;">
</div>
<form class="calculator__body-calculator-wrapper
calculator__body-calculator-wrapper-komb" name="form">
    <table align="center" class="my_table">
        <tr>
            <td>
                k

```



```

        <input class="calculator__body-calculator-
input" type="text" id='inK' size="8" maxlength="10" value="1" onfocus='if
(this.value=="1") this.value="";' onblur='if (this.value=="") this.value="1";'>
    </td>
</tr>
<tr>
    <td>
        n
        <input class="calculator__body-calculator-
input" type="text" id='inN' size="8" maxlength="10" value="5" onfocus='if
(this.value=="5") this.value="";' onblur='if (this.value=="") this.value="5";'>
    </td>
</tr>
<tr>
    <td>
        <input class="calculator__body-calculator-
button calculator__body-calculator-button-komb" type='button' value="Вычислить"
name="B1" onclick="reshenie();">
    </td>
</tr>
<tr>
    <td class="calculator__body-calculator-output">
        <span id="output">
        </span>
    </td>
</tr>
</table>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```

3.html

```

<!doctype html>
<html lang="ru">
    <head>
        <title>Розкладання функції в ряд Фур'є | Онлайн калькулятор</title>
        <link type="text/css" rel="stylesheet" media="all" href="../css/style.css">
    </head>
    <script type="text/javascript">
        function mkFourier(f){
            var htm = '';
            var na = parseInt(f.FOU_NA.value), nz = parseInt(f.FOU_NZ.value), dn =
            parseFloat(f.FOU_DN.value);
            var sgn = f.FOU_SGN[2].checked? -1 : f.FOU_SGN[3].checked? 1 : 0;

```

```

    for (var i = na; i <= nz; i += (dn || nz))
    {
        htm += ((i == na)? ' ' : (sgn < 0 || !sgn && f.FOU_SGN[1].checked)? '-' :
        '+' ) + f.FOU_TERM.value.replace(/s/g, ' ').replace(/sin/g, 'sin').replace(/n/gi,
        ''+i).replace(/sim/g, 'sin');
        sgn = -sgn;}
    f.FOU_OUT.value = htm;}
</script>
<body>
    <div class="calculator__wrapper">
        <div class="calculator__header-wrapper">
            <div class="calculator__header">
                <a href="../index.html">
                    
                </a>
            </div>
        </div>
        <div class="calculator">
            <div class="calculator__menu-wrapper">
                <div class="calculator__left-menu">
                    <a href="../components/1.html" class="calculator__left-menu-items-
selected">
                        <p class="calculator__left-menu-items-title">
                            Лінійна інтерполяція
                        </p>
                    </a>
                    <a href="../components/2.html" class="calculator__left-menu-items">
                        <p class="calculator__left-menu-items-title">
                            Елементи комбінаторики - перестановки, розміщення, поєднання
                        </p>
                    </a>
                    <a href="../components/3.html" class="calculator__left-menu-items-
selected">
                        <p class="calculator__left-menu-items-title">
                            Розкладання функції в ряд Фур'є
                        </p>
                    </a>
                    <a href="../components/4.html" class="calculator__left-menu-items
calculator__left-menu-items">
                        <p class="calculator__left-menu-items-title">
                            Середнє арифметичне, Дисперсія, Варіація
                        </p>
                    </a>
                    <a href="../components/5.html" class="calculator__left-menu-items-
selected">
                        <p class="calculator__left-menu-items-title">
                            Рішення матриць
                        </p>
                    </a>

```

```

    </div>
    <div class="calculator__right-menu">
        <a href="../../components/6.html" class="calculator__right-menu-items
calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Паралельність прямих на площині
            </p>
        </a>
        <a href="../../components/7.html" class="calculator__right-menu-items-
selected">
            <p class="calculator__right-menu-items-title">
                Рішення логарифмів
            </p>
        </a>
        <a href="../../components/8.html" class="calculator__right-menu-items
calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Тригонометричні функції
            </p>
        </a>
        <a href="../../components/9.html" class="calculator__right-menu-items-
selected">
            <p class="calculator__right-menu-items-title">
                Метод найменших квадратів
            </p>
        </a>
        <a href="../../components/10.html" class="calculator__right-menu-items
calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
<div class="calculator__body">
    <h1 class="calculator__body-title" id="cat-200">
        Розкладання функції в ряд Фур'є
    </h1>
    <div class="calculator__body-content">
        <p class="calculator__body-description">
            У цьому розділі пропонується рішення такої задачі, як розкладання
функції на ряд Фур'є.
        <br />
        Якщо ви самостійно розкладете функцію в ряд Фур'є, то це,
безсумнівно, займе у вас багато часу, але за допомогою калькулятора ви зможете
зробити це всього в кілька натискань кнопок.
        <br />
        Причому ви отримаєте не просто готове рішення, а його приклади і
серії.

```

```

</p>
<p class="calculator__body-description">
    Практично будь-яка функція зі значенням періоду  $T$  ( $f(t)$ ) може
    прийняти суму косинусів і синусів з аргументів  $nwt$  (ряд Фур'є),
    <br />
    де значення  $n$  – додатнє ціле число,  $t$  – час, а  $w$  дорівнює кутовій
    частоті  $2\pi/T$ . Кожна складова ряду Фур'є називається гармонікою.
    <br />
    При цьому важливо розуміти, що абсолютно будь-яку парну функцію можна
    розкласти на ряди Фур'є, які будуть складатися з синусів і косинусів.
    <br />
    Тоді як непарну функцію можна розкласти лише на ряди синусів.
</p>
<form class="calculator__body-calculator-wrapper calculator__body-
calculator-wrapper-fure" name="form">
    <table cellpadding="5" class="my_table">
        <tr>
            <td align="center">
                <p class="calculator__body-calculator-title">
                     $n$  от
                    <input class="calculator__body-calculator-input"
style="width:40px" type="text" value="1" name="FOU_NA">
                    до
                    <input class="calculator__body-calculator-input" type="text"
maxlength="4" size="4" value="10" name="FOU_NZ">
                    ширина кроку
                    <input class="calculator__body-calculator-input" type="text"
style="width:40px" value="1" name="FOU_DN">
                    <br>
                    У вигляді суми
                    <input class="calculator__body-calculator-input" checked=""
type="radio" name="FOU_SGN">
                    , разности
                    <input class="calculator__body-calculator-input" type="radio"
name="FOU_SGN">
                    .
                    <br>
                    Або чередувати знаки, починая з плюса
                    <input class="calculator__body-calculator-input" type="radio"
name="FOU_SGN">
                    , с минуса
                    <input class="calculator__body-calculator-input" type="radio"
name="FOU_SGN">
                    <br>
                    Функция
                    <input class="calculator__body-calculator-input" type="text"
style="width:300px" value="0.5^n * sin(n*x)/3" name="FOU_TERM">
                    <input class="calculator__body-calculator-button"
onclick="mkFourier(this.form)" type="button" value="Вычислить">
                </p>

```

```

        </td>
      </tr>
      <tr>
        <td>
          <p class="calculator__body-calculator-area-wrapper">
            <textarea class="calculator__body-calculator-area"
name="FOU_OUT" style="width:400px" rows="10" readonly></textarea>
          </p>
        </td>
      </tr>
    </table>
  </form>
</div>
</div>
</div>
</body>
</html>

```

4.html

```

<!doctype html>
<html lang="ru">
  <head>
<title>Середнє арифметичне, Дисперсія, Варіація | Онлайн калькулятор</title>
<link type="text/css" rel="stylesheet" media="all" href="../css/style.css">
</head>
<script>
  var ai=new Array();
  var as = 0;

  function myRound(n)
  {
    n = n * 100;
    return Math.round(n)/100;
  }

  function myRoundA(n)
  {
    n = n * 100000;
    return Math.round(n)/100000;
  }

  function Vubor()
  {
    var ai=new Array();
    var as = 0;
    var n1 = 0;
    var ds = 0;
    var sigma =0; var ma=0; var me=0; var asimet=0; var eksces=0; var lina=0;

```

```

        ai[1] = document.form.a1.value; ai[11] = document.form.a11.value; ai[21]
= document.form.a21.value;
        ai[2] = document.form.a2.value; ai[12] = document.form.a12.value; ai[22]
= document.form.a22.value;
        ai[3] = document.form.a3.value; ai[13] = document.form.a13.value; ai[23]
= document.form.a23.value;
        ai[4] = document.form.a4.value; ai[14] = document.form.a14.value; ai[24]
= document.form.a24.value;
        ai[5] = document.form.a5.value; ai[15] = document.form.a15.value; ai[25]
= document.form.a25.value;
        ai[6] = document.form.a6.value; ai[16] = document.form.a16.value; ai[26]
= document.form.a26.value;
        ai[7] = document.form.a7.value; ai[17] = document.form.a17.value; ai[27]
= document.form.a27.value;
        ai[8] = document.form.a8.value; ai[18] = document.form.a18.value; ai[28]
= document.form.a28.value;
        ai[9] = document.form.a9.value; ai[19] = document.form.a19.value; ai[29]
= document.form.a29.value;
        ai[10] = document.form.a10.value; ai[20] = document.form.a20.value;
ai[30] = document.form.a30.value;

    for (i=1; i<31; i++)
    {
        if(myRoundA(ai[i])==" ") {}
        else
        {
            as = as + myRoundA(ai[i]);
            n1 = n1 + 1;
        }
    }
    as = as / n1;
    document.form.a.value = myRoundA(as);

    for (i=1; i<31; i++)
    {
        if(myRoundA(ai[i])==" ") {}
        else{
            ds = ds + (myRoundA(ai[i])-as)*(myRoundA(ai[i])-as);}
    }
    ds = ds / (n1-1);
    document.form.d.value = myRoundA(ds);

    for (i=1; i<31; i++)
    {
        if(myRoundA(ai[i])==" ") {}
        else{
            sigma = sigma + (myRoundA(ai[i])-as)*(myRoundA(ai[i])-as);}
    }
    sigma = Math.sqrt(sigma / (n1-1));
    document.form.s.value = myRoundA(sigma);

```

```

document.form.v.value = myRound(sigma/as*100)+"%";

for (i=1; i<31; i++)
{
    if(myRoundA(ai[i])==" ") {}
    else{
        asimet = asimet + Math.pow(myRoundA(ai[i])-as,3)}
    }
ma = Math.sqrt(6*(n1-1)/(n1+1)/(n1+3));
asimet = asimet/ma/n1/Math.pow(sigma,3);
document.form.asim.value = myRoundA(asimet);

for (i=1; i<31; i++)
{
    if(myRoundA(ai[i])==" ") {}
    else{
        eksces = eksces + Math.pow(myRoundA(ai[i])-as,4)}
    }
eksces = eksces/n1/Math.pow(sigma,4) - 3;

me = Math.sqrt(24*n1*(n1-2)*(n1-3)/(n1-1)/(n1-1)/(n1+3)/(n1+5));
eksces = eksces/me;
document.form.eksc.value = myRoundA(eksces);

for (i=1; i<31; i++)
{
    if(myRoundA(ai[i])==" ") {}
    else{
        lina = lina + Math.abs(myRoundA(ai[i])-as)}
    }
lina = lina/n1;
document.form.linotkla.value = myRoundA(lina);
}

```

```
</script>
```

```
<body>
```

```
<div class="calculator_wrapper calculator_wrapper-arifm">
```

```

<div class="calculator__header-wrapper">
  <div class="calculator__header">
    <a href="../index.html">
      
    </a>
  </div>
</div>
<div class="calculator">
  <div class="calculator__menu-wrapper">
    <div class="calculator__left-menu">
      <a href="../components/1.html" class="calculator__left-menu-
items-selected">
        <p class="calculator__left-menu-items-title">
          Лінійна інтерполяція
        </p>
      </a>
      <a href="../components/2.html" class="calculator__left-menu-
items">
        <p class="calculator__left-menu-items-title">
          Елементи комбінаторики - перестановки, розміщення,
поєднання
        </p>
      </a>
      <a href="../components/3.html" class="calculator__left-menu-
items-selected">
        <p class="calculator__left-menu-items-title">
          Розкладання функції в ряд Фур'є
        </p>
      </a>
      <a href="../components/4.html" class="calculator__left-menu-
items calculator__left-menu-items">
        <p class="calculator__left-menu-items-title">
          Середнє арифметичне, Дисперсія, Варіація
        </p>
      </a>
      <a href="../components/5.html" class="calculator__left-menu-
items-selected">
        <p class="calculator__left-menu-items-title">
          Рішення матриць
        </p>
      </a>
    </div>
    <div class="calculator__right-menu">
      <a href="../components/6.html" class="calculator__right-menu-
items calculator__right-menu-items">
        <p class="calculator__right-menu-items-title">
          Паралельність прямих на площині
        </p>
      </a>

```



```

        <a href="../components/7.html" class="calculator__right-menu-
items-selected">
            <p class="calculator__right-menu-items-title">
                Рішення логарифмів
            </p>
        </a>
        <a href="../components/8.html" class="calculator__right-menu-
items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Тригонометричні функції
            </p>
        </a>
        <a href="../components/9.html" class="calculator__right-menu-
items-selected">
            <p class="calculator__right-menu-items-title">
                Метод найменших квадратів
            </p>
        </a>
        <a href="../components/10.html" class="calculator__right-menu-
items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
<div class="calculator__body" id="main">
    <h1 class="calculator__body-title">
        Середнє арифметичне, Дисперсія, Варіація
    </h1>
    <div class="calculator__body-content">
        <div class="calculator__body-description">
            <p>
                Он-лайн калькулятор, який обчислює середнє арифметичне,
                дисперсію, варіацію, стандартне відхилення,
                <br />
                Простий у використанні. Для отримання правильного
                результату необхідно правильно заповнити запропоновану таблицю.
            </p>
        </div>
        <div class="calculator__body-description">
            <p>
                Такий онлайн-калькулятор стане незамінним помічником для
                математичних розрахунків.
                <br />
                Після вказівки потрібних значень потрібно просто
                натиснути кнопку «Розрахунок» і відразу з'явиться правильний результат.
                <br />

```

Простота використання робить калькулятор незамінним і популярним. Його легко використовувати на роботі, вдома, під час навчання.

```
<br />
```

Вона не зажадає від вас виконання складних дій. Ви зможете зробити всі розрахунки для всіх необхідних величин.

</p>

</div>

```
<form class="calculator__body-calculator-wrapper  
calculator__body-calculator-wrapper-arifm" method="post" action="" name="form"  
style="min-width:900px">
```

```
<table cellpadding="0" cellspacing="0" align="center"
border="0" class="my_table" style="margin-left: 140px">
```

```
<tr class="calculator__body-calculator-title"
id="my_color">
```

```
<td></td>
```

```
<td width="121">Досліджені<br> значення  
a<sub>i</sub></td>
```

```
<td></td>
```

 Досліджені^і значення a_i |

```
<td></td>
```

 Досліджені^і значення a_i |

</tr>

```
<tr class="calculator body-calculator-title">
```

```
<td height="28">1</td>
```

```
<td height="28">
```

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" value="120.48" name="a1"></td>
```

 11 |

```
<td height="28">
```

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" name="a11"></td>
```

 21 |

```
<td height="28">
```

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" name="a21"></td>
```

```
<tr class="calculator body-calculator-title">
```

```
<td height="20">2</td>
```

 |

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" value="100.14" name="a2"></td>
```

 12 | |

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" name="a12"></td>
```

 22 | |

```
<input class="calculator__body-calculator-input" type="text" style="width: 140px;" name="a22"></td>
```



```

        document.form.parf.value = parf;
        document.form.perf.value = perf;
    }
</script>
</head>
<body>
    <div class="calculator__wrapper">
        <div class="calculator__header-wrapper">
            <div class="calculator__header">
                <a href="../index.html">
                    
                </a>
            </div>
        </div>
        <div class="calculator">
            <div class="calculator__menu-wrapper">
                <div class="calculator__left-menu">
                    <a href="../components/1.html" class="calculator__left-menu-
items-selected">
                        <p class="calculator__left-menu-items-title">
                            Лінійна інтерполяція
                        </p>
                    </a>
                    <a href="../components/2.html" class="calculator__left-menu-
items">
                        <p class="calculator__left-menu-items-title">
                            Елементи комбінаторики - перестановки, розміщення,
поєднання
                        </p>
                    </a>
                    <a href="../components/3.html" class="calculator__left-menu-
items-selected">
                        <p class="calculator__left-menu-items-title">
                            Розкладання функції в ряд Фур'є
                        </p>
                    </a>
                    <a href="../components/4.html" class="calculator__left-menu-
items calculator__left-menu-items">
                        <p class="calculator__left-menu-items-title">
                            Середнє арифметичне, Дисперсія, Варіація
                        </p>
                    </a>
                    <a href="../components/5.html" class="calculator__left-menu-
items-selected">
                        <p class="calculator__left-menu-items-title">
                            Рішення матриць
                        </p>
                    </a>
                </div>
            </div>
        </div>
    </div>

```



```

        <div class="calculator__right-menu">
            <a href="../../../components/6.html" class="calculator__right-menu-
items calculator__right-menu-items">
                <p class="calculator__right-menu-items-title">
                    Паралельність прямих на площині
                </p>
            </a>
            <a href="../../../components/7.html" class="calculator__right-menu-
items-selected">
                <p class="calculator__right-menu-items-title">
                    Рішення логарифмів
                </p>
            </a>
            <a href="../../../components/8.html" class="calculator__right-menu-
items calculator__right-menu-items">
                <p class="calculator__right-menu-items-title">
                    Тригонометричні функції
                </p>
            </a>
            <a href="../../../components/9.html" class="calculator__right-menu-
items-selected">
                <p class="calculator__right-menu-items-title">
                    Метод найменших квадратів
                </p>
            </a>
            <a href="../../../components/10.html" class="calculator__right-menu-
items calculator__right-menu-items">
                <p class="calculator__right-menu-items-title">
                    Сума арифметичної прогресії
                </p>
            </a>
        </div>
    </div>
    <div class="calculator__body" id="main">
        <h1 class="calculator__body-title" id="cat-200">
            Паралельність прямих на площині
        </h1>
        <div class="calculator__body-content">
            <div class="calculator__body-description">
                <p>
                    Пряма, паралельна прямий, у багатьох випадках інженерних
                    розрахунків обраховується за формулою прямий  $y = mx + b$ , в якій при
                    збереженні незмінним значення кутового коефіцієнта  $m$  зсув прямий, що
                    спостерігається для досліджуваної прямий по осі  $Y$ , забирається -  $b = 0$ .
                </p>
            </div>
            <div class="calculator__body-description">
                <p>

```

У результаті формула прямої перетворюється для паралельної їй прямої $y = mx + 0$.

Хоча паралельних прямих ліній на площині і в просторі можна провести скільки завгодно багато.

Формула паралельної прямої обчислюється для різних конкретних випадків (на площині), у тому числі і для характерного випадку $y = mx + 0$.

Паралельність отриманої розрахунком нової прямої з вихідною прямою можна перевірити шляхом з'ясування наявності пропорційності коефіцієнтів

при змінних x та y .

```

</p>
<p>
    Паралельність отриманої розрахунком нової прямої
з вихідною прямою можна перевірити шляхом з'ясування наявності
коефіцієнтів

    <br />
    при змінних <b>x</b> та <b>y</b>.
</p>
</div>
</div>
<form class="calculator__body-calculator-wrapper
calculator__body-calculator-wrapper-pryam" name="form" style="min-width:690px">
    <table border="0" class="my_table">
        <tr style="display: flex; align-items: center;
margin-left: 380px;">
            <td class="calculator__body-calculator-title">
                X1
            </td>
            <td>
                <input class="calculator__body-calculator-
input" type="text" name="x1" style="width:100px">
            </td>
        </tr>
        <tr style="display: flex; align-items: center;
margin-left: 380px;">
            <td class="calculator__body-calculator-title">
                Y1
            </td>
            <td>
                <input class="calculator__body-calculator-
input" type="text" name="y1" style="width:100px">
            </td>
        </tr>
        <tr style="display: flex; align-items: center;
margin-left: 380px;">
            <td class="calculator__body-calculator-title">
                X2
            </td>
            <td>
                <input class="calculator__body-calculator-
input" type="text" name="x2" style="width:100px">

```

```

        </td>
    </tr>
    <tr style="display: flex; align-items: center;
margin-left: 380px;">
        <td class="calculator__body-calculator-title">
            Y2
        </td>
        <td>
            <input class="calculator__body-calculator-
input" type="text" name="y2" style="width:100px">
        </td>
    </tr>
    <tr align="center">
        <td colspan="2">
            <input class="calculator__body-calculator-
button" type="button" value="Вычислить" onclick="calculate();">
        </td>
    </tr>
    <tr id="my_color" align="left">
        <td class="calculator__body-calculator-title">
            Формула
        </td>
        <td>
            <input class="calculator__body-calculator-
input" type="text1" name="formula" style="width:380px">
        </td>
    </tr>
    <tr id="my_color" align="left">
        <td class="calculator__body-calculator-title">
            Кут нахилу
        </td>
        <td>
            <input class="calculator__body-calculator-
input" type="text1" name="slope" style="width:380px">
        </td>
    </tr>
    <tr id="my_color" align="left">
        <td class="calculator__body-calculator-title">
            Перетин з віссю
        </td>
        <td>
            <input class="calculator__body-calculator-
input" type="text1" name="yint" style="width:380px">
        </td>
    </tr>
    <tr id="my_color" align="left">
        <td class="calculator__body-calculator-title">
            Формула для паралельної
        </td>
        <td>

```

```

        <input class="calculator__body-calculator-
input" type="text1" name="parf" style="width:380px">
    </td>
</tr>
<tr id="my_color" align="left">
    <td class="calculator__body-calculator-title">
        Формула для перпендикулярної
прямої
    </td>
    <td>
        <input class="calculator__body-calculator-
input" type="text1" name="perf" style="width:380px">
    </td>
</tr>
</table>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```

7.html

```

<!doctype html>
<html lang="ru">
    <head>
<title>Рішення логарифмів | Онлайн калькулятор</title>
<link type="text/css" rel="stylesheet" media="all" href="../../css/style.css">
</head>
<script language="JavaScript">
    function round(n,dec)
    {
        X = n * Math.pow(10,dec);
        X= Math.round(X);
        return (X / Math.pow(10,dec)).toFixed(dec);
    }

    function disable() {
        var controlPairs = new Array();
        controlPairs['F1'] = 'A';
        controlPairs['F2'] = 'B';
        controlPairs['F3'] = 'X';
        var disabledClass = "calculator__body-calculator-input--disabled";
        var enabledClass = "calculator__body-calculator-input";

        for (var i in controlPairs) {
            var field = document.getElementById(controlPairs[i]);
            var toggle = document.getElementById(i);

```

```

        field.disabled = toggle.checked;
        field.className = toggle.checked ? disabledClass : enabledClass;
    }
}

function logCalc() {
    var target, answer;
    if (document.getElementById('F1').checked) {
        target = document.getElementById("A");
        answer =
round(Math.pow(getFieldFloatValue('B'),1/getFieldFloatValue('X')), 5);
    }
    else if (document.getElementById('F2').checked) {
        target = document.getElementById("B");
        answer =
round(Math.pow(getFieldFloatValue('A'),getFieldFloatValue('X')), 5);
    }
    else if (document.getElementById('F3').checked) {
        target = document.getElementById("X");
        answer =
round(Math.log(getFieldFloatValue('B'))/Math.log(getFieldFloatValue('A')), 5);
    }

    target.value = isNaN(answer) ? '-' : answer;
}

function getFieldFloatValue(fieldId) {
    return
parseFloat(document.getElementById(fieldId).value.replace("\",", "."));
}
</script>
<body>
    <div class="calculator__wrapper">
        <div class="calculator__header-wrapper">
            <div class="calculator__header">
                <a href="../index.html">
                    
                </a>
            </div>
        </div>
        <div class="calculator">
            <div class="calculator__menu-wrapper">
                <div class="calculator__left-menu">
                    <a href="../components/1.html" class="calculator__left-menu-
items-selected">
                        <p class="calculator__left-menu-items-title">
                            Лінійна інтерполяція
                        </p>
                    </a>

```

```

        <a href="../../../components/2.html" class="calculator__left-menu-
items">
        <p class="calculator__left-menu-items-title">
            Елементи комбінаторики - перестановки, розміщення,
поєднання
        </p>
        </a>
        <a href="../../../components/3.html" class="calculator__left-menu-
items-selected">
        <p class="calculator__left-menu-items-title">
            Розкладання функції в ряд Фур'є
        </p>
        </a>
        <a href="../../../components/4.html" class="calculator__left-menu-
items calculator__left-menu-items">
        <p class="calculator__left-menu-items-title">
            Середнє арифметичне, Дисперсія, Варіація
        </p>
        </a>
        <a href="../../../components/5.html" class="calculator__left-menu-
items-selected">
        <p class="calculator__left-menu-items-title">
            Рішення матриць
        </p>
        </a>
    </div>
    <div class="calculator__right-menu">
        <a href="../../../components/6.html" class="calculator__right-menu-
items calculator__right-menu-items">
        <p class="calculator__right-menu-items-title">
            Паралельність прямих на площині
        </p>
        </a>
        <a href="../../../components/7.html" class="calculator__right-menu-
items-selected">
        <p class="calculator__right-menu-items-title">
            Рішення логарифмів
        </p>
        </a>
        <a href="../../../components/8.html" class="calculator__right-menu-
items calculator__right-menu-items">
        <p class="calculator__right-menu-items-title">
            Тригонометричні функції
        </p>
        </a>
        <a href="../../../components/9.html" class="calculator__right-menu-
items-selected">
        <p class="calculator__right-menu-items-title">
            Метод найменших квадратів
        </p>
    </div>

```

```

        </a>
        <a href="../../../components/10.html" class="calculator__right-menu-
items calculator__right-menu-items">
            <p class="calculator__right-menu-items-title">
                Сумма арифметичної прогресії
            </p>
        </a>
    </div>
</div>
</div>
<div class="calculator__body" id="main">
    <h1 class="calculator__body-title" id="cat-200">
        Рішення логарифмів
    </h1>
    <div class="calculator__body-content">
        <div class="calculator__body-description">
            <p>
                Для пояснення терміну логарифму числа розбір легше
                провести з прикладу.
            <br />
                Математичні вирази  $X = \log_5 15$  та
                ( $5^X = 15$ ) ідентичні.
            <br />
                Звідси логарифм числа ( $X$ ) є показником ступеня, в
                який необхідно звести інше число (основа ступеня в нашому прикладі « $5$ »)
            <br />
                для отримання цього числа ( $15$ ).
            </p>
        </div>
        <div class="calculator__body-description">
            <p>
                Вираз ( $5^X = 15$ ) є показовим
                рівнянням.
            <br />
                Логарифми використовуються у складних розрахунках,
                особливістю їх застосування є можливість заміни таких дій як множення більш
                простим підсумовуванням.
            <br />
                Якщо числа замінювати їх логарифмами у розрахунках можна
                операцію поділу замінити на віднімання.
            </p>
            <p>
                Замість вилучення коренів можна здійснювати поділ,
                операція зведення у ступінь може замінюватись множенням.
            <br />
                Серед логарифмів найчастіше використовуються в практичних
                розрахунках десяткові логарифми  $\lg$  (у них основою
                є число « $10$ »)
            <br />

```

та натуральні логарифми $\ln$ (з основою числа $e = 2,718 \dots$).

```

    </p>
  </div>
</div>
<form class="calculator__body-calculator-wrapper calculator__body-calculator-wrapper-log" name="form" style="min-width:380px">
  <table class="my_table" style="display: flex; justify-content: center;">
    <tr>
      <td>
        <div class="L-td L">
          <input class="css-radio" type="radio" id="F1"
name="find" onclick="disable();">
          <label for="F1" class="calculator__body-calculator-title">&nbsp;A</label>
          <input class="calculator__body-calculator-input"
id="A" name="A" value="5" type="text" style="width: 100px;" onclick="this.value =
'';logCalc();">
        </div>
      </td>
    </tr>
    <tr>
      <td>
        <div class="L-td L">
          <input class="css-radio" type="radio" id="F2"
name="find" onclick="disable();">
          <label for="F2" class="calculator__body-calculator-title">&nbsp;B</label>
          <input class="calculator__body-calculator-input"
id="B" name="B" value="15" type="text" style="width: 100px;" onclick="this.value
= '';logCalc();">
        </div>
      </td>
    </tr>
    <tr>
      <td>
        <div class="row">
          <div class="L-td L">
            <input class="css-radio" type="radio" id="F3"
name="find" onclick="disable();" checked="checked">
            <label for="F3" class="calculator__body-calculator-title">&nbsp;X</label>
            <input class="calculator__body-calculator-input--disabled" id="X" name="X" value="0" type="text" style="width: 100px;"
disabled="" onclick="this.value = '';logCalc();">
          </div>
        </div>
      </td>
    </tr>
  </table>
</form>

```



```
        <tr>
            <td colspan="4" align="center">
                <input class="calculator__body-calculator-button"
type="button" style="width:100px" value="Вычислить" name="gdi"
onclick="logCalc()">
            </td>
        </tr>
    </table>
</form>
</div>
</div>
</body>
</html>
```