

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Навчально-науковий інститут телекомунікаційних систем

Кафедра телекомунікацій

До захисту допущено:

Завідувач кафедри

_____ Сергій КРАВЧУК

«__» _____ 2024 р.

Дипломна робота

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інженерія та програмування
інфокомунікацій»**

спеціальності 172 Телекомунікації та радіотехніка

**на тему: «Розробка веб-інтерфейсу платформи дистанційного навчання для
дисциплін телекомунікаційного спрямування.»**

Виконав:

студент IV курсу, групи ТЗ-02

Ремесло Михайло Федорович

Керівник:

Старший викладач кафедри ТК НН ІТС

Кайденко Микола Миколайович

Рецензент:

Доцент кафедри ІТТ НН ІТС, к.т.н, доцент

Правило Валерій Володимирович _____

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

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут телекомунікаційних систем
Кафедра телекомунікацій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 Телекомунікації тарадіотехніка

Освітньо-професійна програма «Інженерія та програмування інфокомунікацій»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій КРАВЧУК

«___» _____ 2024 р.

ЗАВДАННЯ

на дипломну роботу студенту

Ремесло Михайло Федорович

1. Тема роботи «Розробка веб-інтерфейсу платформи дистанційного навчання для дисциплін телекомунікаційного спрямування.», керівник роботи Кайденко Микола Миколайович, затверджені наказом по університету від «22» травня 2024 р. № 2064-с.

2. Термін подання студентом роботи “10” червня 2024 р.

3. Вихідні дані до роботи: Програмне забезпечення для реалізації frontend; підтримка відомих браузерів (Firefox, Chrome, Edge, Safari, Opera и Brave); дружній до користувача інтерфейс; авторизація користувачів; завдання, їх перевірка та оцінка.

4. Зміст роботи: Розробити веб-інтерфейс платформи дистанційного навчання для студентів які навчаються на телекомунікаційних спеціальностях та користувачів, що бажають підвищити свій рівень знань в області телекомунікацій, який інтуїтивно буде зрозумілий для користувачів різного рівня технічної підготовки та обізнаності в користуванні платформами. Забезпечити швидкий та зрозумілий доступ до функцій платформи.

5. Перелік ілюстративного матеріалу: Презентація

1. Титульний лист

2. Актуальність
3. Мета
4. Аналіз особливостей існуючих платформ дистанційного навчання
5. Вибір програмних рішень для реалізації веб-інтерфейсу
6. Реалізація розробки веб-інтерфейсу платформи дистанційного навчання
7. Дошка для проектування
8. Результат роботи
9. Реалізація
10. Висновки по роботі

6. Дата видачі завдання “1” квітня 2024 р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Визначення мети та постановка завдань дослідження	01.04.2024 - 08.04.2024	Виконано
2	Підбір та ознайомлення з технічною літературою.	09.04.2024 - 16.04.2024	Виконано
3	Аналіз існуючих рішень. Формулювання структури дослідження та розробки.	17.04.2024 - 24.04.2024	Виконано
4	Розділ 1: Аналіз особливостей існуючих платформ дистанційного навчання.	25.04.2024 - 02.05.2024	Виконано
5	Розділ 2: Дослідження та вибір програмних рішень для реалізації веб-інтерфейсу платформи дистанційного навчання для дисциплін телекомунікаційного спрямування.	09.05.2024 - 16.05.2024	Виконано
6	Розділ 3: Реалізація розробки веб-інтерфейсу платформи дистанційного навчання.	17.05.2024 - 24.05.2024	Виконано
7	Аналіз результатів дослідження та розробки.	25.05.2024 - 30.05.2024	Виконано
8	Підготовка висновків, оформлення дипломної роботи.	01.06.2024 - 08.06.2024	Виконано
9	Підготовка звітних та презентаційних матеріалів.	09.06.2024 - 16.06.2024	Виконано

Студент

Михайло РЕМЕСЛО

Керівник

Микола КАЙДЕНКО

РЕФЕРАТ

Обсяг пояснювальної записки: 65 сторінки.

Кількість ілюстрацій: 15.

Кількість таблиць: 1.

Кількість використаних джерел: 15.

Мета роботи: Розробити веб-інтерфейс платформи дистанційного навчання для студентів телекомунікаційних спеціальностей та користувачів, які бажають підвищити свій рівень знань в області телекомунікацій. Інтерфейс має бути інтуїтивно зрозумілим для користувачів з різним рівнем технічної підготовки та обізнаності у користуванні платформами, забезпечуючи швидкий та зручний доступ до функцій платформи.

Використані методи та отримані результати: Платформа написана в редакторі коду VS Code з використанням html, sass, js, node.js та json, зібрана за допомогою parcel. Інтерфейс платформи забезпечує зручне та зрозуміле користування.

Рекомендації щодо використання: Розроблену платформу можна використовувати у вищих навчальних закладах для забезпечення ефективного дистанційного навчання завдяки зрозумілому інтерфейсу та широким функціональним можливостям, які забезпечують швидкий доступ до матеріалів.

Перелік ключових слів: дистанційне навчання, веб-інтерфейс, телекомунікації, оптимізація, сумісність браузерів, програмне забезпечення, аналіз платформ, розробка, тестування, проектування, програмування

ABSTRACT

Volume of the Explanatory Note: 65 pages.

Number of Illustrations: 15.

Number of tables: 1.

Number of used sources: 15.

Objective: To develop a web interface for a distance learning platform designed for students specializing in telecommunications and users who wish to enhance their knowledge in the field of telecommunications. The interface should be intuitive for users with varying levels of technical proficiency and familiarity with using platforms, ensuring quick and easy access to the platform's features.

Methods Used and Results Obtained: The platform was developed using the VS Code editor with HTML, SASS, JavaScript, Node.js, and JSON, and was assembled using Parcel. The interface provides convenient and intuitive usage.

Usage Recommendations: The developed platform can be implemented in higher education institutions to ensure effective distance learning, thanks to its intuitive interface and extensive functionality, which provide quick access to materials.

List of Keywords: distance learning, web interface, telecommunications, optimization, cross-browser compatibility, software, platform analysis, development, testing, design, programming.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ОСОБЛИВОСТЕЙ ІСНУЮЧИХ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	10
1.1 Призначення дистанційного навчання.....	11
1.2 Особливості створення платформ для дистанційного навчання.....	12
1.3 Переваги та недоліки платформ.....	14
Висновки.....	15
РОЗДІЛ 2 ДОСЛІДЖЕННЯ ТА ВИБІР ПРОГРАМНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ІНТЕРФЕЙСУ ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ДЛЯ ДИСЦИПЛІН ТЕЛЕКОМУНІКАЦІЙНОГО СПРЯМУВАННЯ.....	16
2.1 Основні файли та папки проекту.....	17
2.2 Програми та засоби які використовуються у роботі.....	18
Висновки.....	20
РОЗДІЛ 3 РЕАЛІЗАЦІЯ РОЗРОБКИ ВЕБ-ІНТЕРФЕЙСУ ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ.....	22
3.1 Розробка структури та стилізація сайту.....	22
3.2 Розробка функціоналу платформи.....	35
3.3 API взаємодія з backend, процес інсталяції.....	56
3.4 Тестування веб-інтерфейсу платформи.....	59
Висновки.....	59
ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

ПЕРЕЛІК СКОРОЧЕНЬ

VS Code	Visual Studio Code
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheets
SASS	Syntactically Awesome Style Sheets
JS	JavaScript
Git	GitHub
JSON	JavaScript Object Notation
AR	Augmented Reality
VR	Virtual Reality
IDE	Integrated Drive Electronics
CAD	Computer-Aided Design
AI	Artificial Intelligence
SQL	Structured Query Language
API	Application Programming Interface
FTP	File Transfer Protocol
SSH	Secure Shell

ВСТУП

По мірі розвитку глобальних технологій та комунікацій, ми спостерігаємо зміни в тому, як ми навчаємось. Завдяки розвитку інтернет-технологій, студенти та викладачі почали частіше використовувати платформи для дистанційного навчання, що дозволяє мати доступ до матеріалів, комунікації та виконання робіт в будь-якому місці, де є інтернет.

Платформа дистанційного навчання надає такі ресурси, як відео, аудіозаписи та презентації. Візуальний і різноманітний навчальний контент дозволяє студентам краще навчатися, візуалізуючи складні поняття. Ці ресурси значно полегшують процес навчання для студентів, яким важко зосередитися лише на письмовому матеріалі. Крім того, більшість платформ дистанційного навчання надають можливості для особистого спілкування з викладачами.

Платформи для дистанційного навчання дозволяють студентам працювати над матеріалами у власному темпі. Це дозволяє їм продовжувати навчання, навіть якщо вони не мають доступу до аудиторії в певні періоди часу. Звичайно ж не всі платформи підходять для всіх користувачів, оскільки кожний освітній процес має свої особливості і деякі платформи не мають достатнього функціоналу для їх вирішення. Зважаючи на різноманітні особливості та потреби дистанційного навчання студентів, виникає потреба у створенні платформ, де можна вивчати більш складні спеціальності, такі як телекомунікації, програмування, інженерії. Адаптивне навчання, яке враховує індивідуальні потреби кожного студента. Деякі платформи використовують алгоритми машинного навчання, щоб пропонувати студентам матеріали та завдання, які відповідають їх рівню знань та темпу засвоєння інформації. Це дозволяє уникнути ситуацій, коли один студент відстає, а інший нудьгує через занадто простий матеріал. Такий підхід сприяє ефективнішому та продуктивнішому навчанню. Також важливою є можливість інтерактивного навчання, яке включає в себе онлайн-дискусії, форуми та розробка групових

проектів. Це допомагає студентам розвивати навички співпраці та комунікації, що є важливими у сучасному світі. Інтерактивні елементи дозволяють студентам обговорювати навчальні матеріали, ставити запитання та ділитися своїми думками, що сприяє кращому розумінню та закріпленню знань.

Існують платформи які дозволяють вивчати різні дисципліни та напрямки, але є і платформи розроблені під конкретні дисципліни або спеціальності. Такі платформи розробляють інтерфейс та функціонал під конкретні дисципліни, що спрощує та імponує їх вивченню. Платформи для вивчення програмування часто включають інтегровані середовища розробки (IDE), що дозволяють студентам писати та тестувати код безпосередньо в браузері. Вони також можуть містити інтерактивні вправи, автоматичні перевірки коду та глибокий аналіз помилок, що сприяє кращому засвоєнню матеріалу. Телекомунікаційні дисципліни також мають свої спеціалізовані платформи, які можуть включати модулі для моделювання фізичних процесів, роботи з CAD-системами та проведення віртуальних лабораторних робіт. Платформи для вивчення мов часто пропонують інтерактивні уроки з використанням аудіо та відеоматеріалів, які допомагають студентам поліпшувати свої навички слухання та вимови.

У зв'язку з зростаючою потребою в сучасних рішеннях для розвитку та вдосконаленню освіти. Світові компанії активно працюють над покращенням та розширенням можливостей своїх продуктів та залучають нових розробників для створення нових платформ. Враховуючи, розвиток технологій, платформи постійно оновлюються та вдосконалюються, додаються інтерактивні технології, а також інтеграція з іншими сервісами та платформами.

В даній дипломній роботі досліджується процес створення веб-інтерфейсу платформи дистанційного навчання для телекомунікаційних та технічних спеціальностей забезпечуючи їх необхідним функціоналом для досягнення високих результатів у навчанні та професійному розвитку, де користувачі матимуть змогу проводити та здавати онлайн лабораторні роботи та свої проекти.

РОЗДІЛ 1

АНАЛІЗ ОСОБЛИВОСТЕЙ ІСНУЮЧИХ ПЛАТФОРМ ДИСТАНЦІЙНОГО НАВЧАННЯ

Платформи дистанційного навчання стають все більш функціональними та доступнішими у зв'язку з розвитком інтернету та технологій. Існує багато платформ розроблених під ті, чи інші потреби користувачів, як для проведення шкільних уроків з простішим функціоналом, так і для вивчення окремих дисциплін, професій, проведення онлайн курсів, де використовують ширший та складніший функціонал, такі платформи здебільшого пропонують підписки. Нинішні платформи дистанційного навчання розроблені з урахуванням широкого спектру потреб користувачів. Більшість з існуючих платформ дозволяють завантажувати свої роботи або матеріали для навчання, проведення відео-уроків на платформі або з використанням сторонніх програм, платформи можуть буди адаптовані під проходження різноманітних завдань або тестів.

Одна з найважливіших можливостей платформ - це змога налаштовувати курсів та матеріалів згідно з потребами аудиторії або навіть під індивідуального користувача. Можливість налаштування курсів та матеріалів з потребами конкретної аудиторії або навіть індивідуального користувача дозволить забезпечити цікаве та ефективне навчання, враховуючи рівень знань, інтересів та індивідуальних потреб студента. Ведення статистики успішності студента дає змогу слідкувати та аналізувати його прогрес у навчанні та вчасно . Використання мобільних додатків або інтегрованих веб-версій платформ для зручного доступу на мобільних девайсах.

Платформи створені для вивчення технічних спеціальностей такі, як телекомунікації, інженерія, програмування потребують відповідного функціоналу від платформ для зручного та ефективного навчання. Для вивчення програмування платформи інтегрують можливість писати код на

мовах JavaScript, Java, Python, C, C++ безпосередньо на платформах, для телекомунікацій та інженерії самого написання коду не достатньо, тому платформи створюють інтерактивні дошки або деякі навіть намагаються створити віртуальні лабораторія для взаємодії з пристроями. Також, для телекомунікаційних спеціальностей часто впроваджуються симулятори мережевих середовищ, які дозволяють студентам налаштовувати і тестувати мережеві конфігурації в реальному часі, впровадження 3D-моделювання, віртуальні симулятори фізичних процесів та аналіз даних з реальних інженерних систем. Крім того, деякі платформи інтегрують технології доповненої реальності (AR) та віртуальної реальності (VR), що дозволяє студентам взаємодіяти з віртуальними прототипами пристроїв і систем, надаючи їм можливість вивчати складні концепції у візуально привабливому і практичному форматі.

1.1 Призначення дистанційного навчання

Дистанційне навчання зазвичай адаптується до потреб та інтересів кожного студента. Дистанційні платформи дають змогу студентам обирати курси та матеріали, які відповідають їхнім потребам та інтересам. Це створює можливості для більш індивідуалізованого навчання, де кожен студент може працювати в темпі, яке відповідає його особистим можливостям та рівню знань. Це дає їм можливість використовувати більш індивідуальний підхід до вивчення предметів і більш актуальні освітні програми. Ці платформи дозволяють легко інтегрувати мультимедійні елементи — відео, подкасти та інтерактивну графіку. Ця різноманітність може зробити уроки більш цікавими та адаптованими до різних стилів навчання.

Використовуючи платформи дистанційного навчання, ви можете вибирати з широкого спектру предметів і тем, щоб навчатися бажаного в будь-який час і в будь-якому місці.

Адаптивність дистанційних платформ не менш важливий аспект, оскільки

вони роблять навчання для різноманітних груп студентів. Вони містять чіткі інструкції, організований макет і узгоджені елементи дизайну, щоб учні могли легко отримати доступ до навчального матеріалу та взаємодіяти з ним. Ще однією перевагою адаптивних платформ дистанційного навчання є сумісність пристроїв. Ці рішення створені для бездоганної роботи на таких пристроях, як ноутбуки, планшети та смартфони. Адаптування платформи для різних регіонів з можливістю обрання мови.

Розглянемо платформи які створені під вивчення конкретних курсів, дисциплін або професій. Такі платформи як Google Classroom[1], Blackboard[2] та Moodle[3] є прикладом платформ на яких можна вивчати різні курси і функціонал дозволяє проводити відео-уроки, здавати роботи та оцінювати учнів, edX та Coursera пропонують проходження курсів та надання сертифікати за успішне проходження курсу. Платформи для вивчення іноземних мов, зазвичай пропонують відео-уроки, ігрові моделі вивчення мов або правил і звичайно тести, як зазначають на інтернет платформі “Освіта UA”[4] та інтернет-виданні «Без купюр»[5].

Для технічних спеціальностей, вивчення програмування, платформи також пропонують відео-уроки, змогу виконувати завдання та писати код на самій платформі. В даній роботі буде представлено створення платформи дистанційного навчання для телекомунікаційних спеціальностей.

1.2 Особливості створення платформ для дистанційного навчання

Першим етапом розробки системної архітектури з урахування напрямку в якому буде працювати платформа та потребами користувачів. Створення дизайну, зрозумілого для різних користувачів. Спочатку платформа створюється як веб-сторінка, отже після створення дизайну буде розроблятися frontend частина сайту, платформа повинна бути інтегрована під різні пристрої та працювати з найпоширенішими браузерами та бути оптимізованою. Отримання ліцензій для використання сторонніх програм та технологій, які будуть

знаходиться на платформі. Також інтеграція новий сучасних технологій таких як штучний інтелект(AI) або віртуальну реальність(VR) та забезпечення технічного догляду за ними.

Розробка веб-інтерфейсу відноситься до області веб-розробки та комп'ютерних наук. Ця область поєдну знання з програмування, дизайну, роботою з базами даних та серверами. Для створення веб-сайту використовують frontend та backend, як представлено на Рис. 1.1.

Різниця між Frontend та Backend

Frontend - це та частина сайту яку бачить та з якою взаємодіє користувач.

Backend - це частина, яка приймає запити від користувача через сервер, обробляє їх та повертає відповідь користувачу.

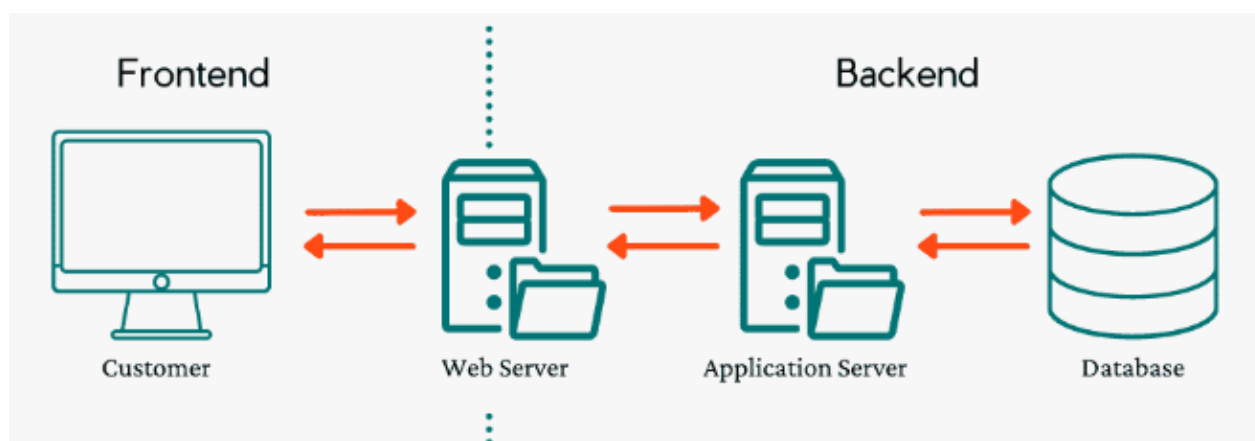


Рис. 1.1 Робота frontend та backend

Коли користувач взаємодіє з веб-сайтом, браузер відправляє запит на сервер, де запит обробляється backend. Потім він повертає відповідь у вигляді даних, які відображатимуться на екрані користувача. Для виконання цієї роботи буде використаний Frontend.

Після написання frontend частини, пишеться та розробляється backend та сервер для написання яких використовують програмування (Python, Java), фреймворки, системи управління базами даних (SQL). Ці інструменти дозволять сайту швидко обробляти великі обсяги даних та запитів від користувачів. Адміністративні аспекти включає керування користувачами та

вмістом, а також надання аналітики та звітності. Механізм управління дозволяє ефективно організувати роботу платформи та забезпечити безпеку даних користувачів.

Розробники не тільки повинні створити платформу, але й слідкувати аби вона працювала і виправляти баги або недоліки. Тестування це один з останніх етапів розробки, платформу перевіряють на баги та недоліки, тестування проводять на кожному етапі розробки. Цей фінальний етап передбачає налаштування середовища, моніторинг продуктивності та забезпечення підтримки після запуску.

1.3 Переваги та недоліки платформ

Платформи дистанційного навчання мають свої переваги та недоліки.

Переваги:

Зручність: Платформи дозволяють навчатися студентам у зручний час та у зручному місці, мати доступ до навчальних матеріалів та можливість здавати роботи з будь-якого пристрою який під'єднаний до інтернету.

Індивідуальність: Використання платформ дистанційного навчання підходить для організації індивідуального підходу. Студент сам обирає собі темп навчання, він може оперативно отримати у викладача відповіді на виникаючі питання.

Доступність: Навчальні матеріали доступні зазвичай цілодобово, дозволяючи студентам вивчати матеріал у власному темпі.

Широкий вибір: Платформи дистанційного навчання дозволяють обирати та вивчати будь-які напрямки, дисципліни чи курси.

Недоліки:

Онлайн спілкування: Не дивлячись на те, що платформи пропонують відео-уроки та деякі мають чати, онлайн спілкування не замінить оффлайн.

Зв'язок: Кожна платформа буде працювати тільки за наявності інтернету.

Неефективне використання часу: Самому інколи тяжко організувати

розклад для навчання.

Висновки:

Проаналізувавши особливості існуючих платформ, можна сказати, що цей напрямок динамічно розвивається і пропонує широкий спектр можливостей в навчанні. Вони стають все більш функціональними та доступнішими завдяки розвитку інтернету та технологій. Одна з найважливіших можливостей платформ дистанційного навчання - це змога налаштовувати курси та матеріали згідно з потребами аудиторії або навіть індивідуального користувача.

Найпоширеніші платформи, що розглядаються, Google Classroom, Blackboard, Moodle, які пропонують різноманітні можливості для навчання різних курсів та дисциплін. Додатково, існують платформи для вивчення іноземних мов, програмування та інших технічних навичок.

Платформи мають низку переваги, серед яких зручність та доступність, користувач може використовувати її з любого пристрою та у зручний час, широкий вибір курсів. Однак платформи мають і недоліки, платформа не працюватиме без інтернет з'єднання і звичайно обмежене живе спілкування.

Зокрема, для вивчення телекомунікацій та інженерії починають створювати інтерактивні дошки, віртуальні лабораторії та симулятори мережових середовищ, які дозволяють студентам практично застосовувати знання та взаємодіяти з віртуальними пристроями і системами, що значно підвищує ефективність навчання.

Розробка власної платформи дозволяє врахувати унікальні потреби та вимоги для функціоналу, що відповідає специфіці навчання в певній сфері.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ТА ВИБІР ПРОГРАМНИХ РІШЕНЬ ДЛЯ РЕАЛІЗАЦІЇ ВЕБ-ІНТЕРФЕЙСУ ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ ДЛЯ ДИСЦИПЛІН ТЕЛЕКОМУНІКАЦІЙНОГО СПРЯМУВАННЯ

Розробка платформи складатиметься з декількох етапів. Розробка проводитиметься на комп'ютері на операційній системі Windows. Обрано програмне забезпечення VS Code[6] для написання коду, збереження та редагування. Оскільки дає змогу створювати та редагувати код, працює з найпоширенішими браузерями, підтримує безліч розширень які спрощують та допомагають в розробці.

Структура сайту яка міститиме текстову частину, картинки, блоки, елементи, посилання та інші сторінки сайту буде написана на html. Для написання стилів, анімації, форматування та оформлення веб-сторінки використовують css, але для цієї роботи буде використано препроцесор css, такий як scss або sass, він використовує схожий синтаксис та структурувати файли під окремі розділи і потім імпортувати в main файл. JS використовуватиметься для написання логіки, функціоналу та взаємодії сайту.

У середовищі Git[7] створено репозиторій в який за допомогою терміналу в VS Code завантажено папку проекту. Після редагування коду потрібно оновлювати код на Git, використовуючи термінал та команду `git push` та `git commit`. Репозиторій є публічним і кожен хто має посилання зможе переглянути код. Завантажуємо Node.js та пакетний менеджер npm, після встановлення будуть створені папки `.parcel-cache`, `node_modules`. Вводимо команду в термінал:

```
npm install
```

Додатково завантажуюмо пакувальник для веб'додатків parcel з сайту Parcel[8], інсталюємо використовуючи термінал та команду:

```
npm install --save-dev parcel
```

Після інсталяції буде створено файли `package-lock.json` та `package.json` це файли конфігурації проекту в Node.js у форматі JSON. Тепер ми можемо налаштувати npm для використання, вписавши код:

```
"scripts": {
  "start": "parcel --open",
  "build": "parcel build --public-url=/src/"
},
```

ця частина коду дозволяє збирати і запускати всі файли роботи використавши команду: `npm build` та `npm start`, після успішного збирання веб-сторінка автоматично відкриється в браузері і буде створена папка `dist`.

2.1 Основні файли та папки проекту

Папка проекту складається:

Папка “`src`”: основна папка, яка включає файли з кодом, стилями, картинками, допоміжними папками та файлами. Включає папки `images`, `styles`, `blocks`.

Папка “`images`”: всі ка стилі які написані під конкретні елементи, розділи та блоки.

Папка “`scripts`”: містить всі файли js, які відповідають за функціонал та логіку сайту. ртинки які використовуються у проекті зберігаються тут.

Папка “`styles`”: містить усі написані стилі.

Папка “`blocks`”: містить стилі розподілені по елементам.

Папка “`node_modules`”: містить пакети і всі залежності проекту, які встановлені за допомогою npm.

Папка “`dist`”: містить скомпільований, готовий до розгортання або публікації код, цей файл був створений після встановлення `parcel`.

Папка “`.parcel-cache`”: містить кешовані дані, пов’язані зі збіркою проекту використовуючи `parcel`.

Файл “`package.json`”: файл містить інформацію про проект, назву, версію,

автора, залежності, дозволяє керувати проектом, встановлювати пакети, запуск скриптів.

Файл “package.json”: містить точні версії кожної залежності, яка встановлена у проєкті, URL-адреси репозиторіїв для кожного пакету, який встановлено, хеші файлів кожного завантаженого модуля та конфігураційні дані npm.

Файл “index.html”: містить структуру сайту, текстовий контент, блоки, елементи, посилання.

Файл “main.scss”: основний файл стилів в який імпортуються дані з інших файлів стилів, посилання на цей файл знаходиться в index.html.

Файл “README.md”: містить основні дані про проєкт, такі, як назва, призначення, автор.

2.2 Програми та засоби які використовуються у роботі

Microsoft Windows — це операційна система, розроблена та випущена корпорацією Microsoft. Вона є однією з найпоширеніших операційних систем для персональних комп'ютерів та інших пристроїв, таких як сервери і мобільні пристрої. Windows забезпечує різноманітні можливості для роботи з комп'ютером, включаючи запуск програм, роботу з файлами та доступ до Інтернету. Під час виконання роботи використовується Windows 11.

Visual Studio Code - це безкоштовний текстовий редактор, розроблений компанією Microsoft. Він призначений для редагування коду і підтримує багато мов програмування, таких як JavaScript, TypeScript, Python, інші. VS Code має ряд корисних функцій, таких як підтримка розширень, вбудована підтримка Git, інструменти для відлагодження коду, автодоповнення, а також інші можливості, які допомагають розробникам працювати більш ефективно. Для виконання роботи використовуються HTML, CSS, SCSS та JavaScript.

HTML - це стандартна мова розмітки для створення та представлення веб-сторінок в Інтернеті. HTML використовується для створення структури і

змісту веб-сторінок, визначення елементів таких як заголовки, параграфи, списки, таблиці, посилання та інші. Він використовує теги, які вказують браузеру, як відображати вміст кожного елементу.

CSS - це мова стилів, яка використовується для оформлення і форматування веб-сторінок, написаних на HTML. CSS дозволяє встановлювати різноманітні стилі для елементів веб-сторінки, такі як шрифти, розміри, кольори, відступи, ефекти, рамки, а також розташовування та взаємодія з елементів на сторінці.

SCSS - це препроцесор CSS, який допомагає зменшити повторення з CSS і, в свою чергу, економить час. Він вважається більш стабільним і потужним в порівнянні з CSS. Він описує стиль документа дуже чітко та структурно. Це допомагає швидше і якісніше виконувати роботу, що дає йому перевагу над звичайним CSS. Він також підтримує логічне вкладення, яке не має CSS. SCSS дозволяє користувачеві вкладений код, який можна вставити один у одного.

БЕМ — методологія в основі якої лежить домовленість про те, як розбивати інтерфейс на певні незалежні блоки. Розшифровується аббревіатура як «Блок Елемент Модифікатор». Цей підхід до веб-розробки дозволяє швидко створювати сайти з гнучкою архітектурою.

JavaScript - це мова програмування, що використовується для створення динамічного змісту та інтерактивних елементів на веб-сторінках. Вона широко використовується для розробки клієнтських скриптів у веб-додатках, які запускаються в браузерах користувачів. JavaScript дає можливість додавати різноманітні функції, анімацію, перевірку валідності даних, змінювати вміст сторінок та взаємодіяти з користувачем.

GitHub - це веб-платформа для зберігання, спільного використання та управління проектами програмного забезпечення, що ґрунтується на системі керування версіями Git. На GitHub користувачі можуть завантажувати свої проекти, створювати репозиторії (репо), працювати з іншими користувачами над спільними проектами, вносити зміни в код через систему контролю версій та вести документацію.

Parcel - це пакувальник для веб-додатків для розробників з різним досвідом. Він пропонує чудову швидку роботу з використанням багатоядерної обробки і не вимагає налаштувань.

Node.js - це вільна, відкрита платформа для виконання JavaScript коду на сервері. Вона дозволяє використовувати JavaScript для написання серверного коду, що раніше було можливим тільки за допомогою мов, таких як PHP, Python або Ruby. Node.js використовує необхідний JavaScript-двигун для виконання коду, наприклад, V8, який також використовується в браузері Google Chrome.

npm - це стандартний пакетний менеджер для Node.js, який використовується для управління залежностями проекту та установки зовнішніх пакетів JavaScript. npm дозволяє розробникам легко встановлювати, оновлювати та видаляти пакети з централізованого репозиторію.

Figma - це веб-платформа для дизайну і прототипування, яка дозволяє командам створювати та редагувати графічні проекти в реальному часі.

Висновки:

Для розробки веб-інтерфейсу платформи дистанційного навчання, було обрано операційну систему Windows 11 та редактор коду VS Code, оскільки він є лідером на ринку. Структура сайту розробляється за допомогою HTML, препроцесора SASS і JavaScript для логіки та взаємодії. Проект розподілений по папкам, що забезпечить зручне орієнтування в файлах, кожний елемент та сторінка сайту має свій окремий файл html, scss та js які під'єднані до основних файлів, що забезпечить легкість в управлінні кодом, процесі розробки та підтримки, а також підвищить масштабованість і організованість проекту. Така структура дозволяє легко інтегрувати нові функції та модулі без необхідності значних змін у існуючому коді. Це сприяє гнучкості у розвитку проекту та швидкому реагуванню на вимоги користувачів.

Використання SASS дозволяє створювати стилі більш структуровано та

підтримувати їх легше завдяки можливості використання змінних та вкладених селекторів. Це значно спрощує процес оновлення та редагування стилів у майбутньому. Parcel збиратиме проект та відкриватиме його в обраному браузері та створює папку `dist` для подальшого розгортання на сервері.

Репозиторій на Git дозволяє зберігати та оновлювати код проекту, що робить його доступним для перегляду всіма, хто має посилання. Інсталяція пакувальника Parcel дозволяє автоматизувати процес збірки і запуску проекту. Команди `npm build` та `npm start` дозволяють швидко зібрати проект та запустити його у браузері, забезпечуючи зручність і швидкість розробки.

Отже, обраний підхід забезпечує гнучкість та ефективність розробки веб-платформи, що дозволяє швидко створювати, редагувати та тестувати веб-сторінку.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ РОЗРОБКИ ВЕБ-ІНТЕРФЕЙСУ ПЛАТФОРМИ ДИСТАНЦІЙНОГО НАВЧАННЯ.

Визначення функціональних вимог до системи, вивчення потреб користувачів і основних функцій, які виконуватиме інтерфейс. Архітектура платформи створена з урахуванням масштабованості та ефективності. Для реалізації веб-інтерфейсу були обрані технології, які відповідають потребам проекту. Для написання коду використовується програма Visual Studio Code, яка є найпопулярнішим безкоштовним редактором коду. VS Code підтримує багато мов програмування, має зручний та зрозумілий інтерфейс та має вбудований інструмент Git. Код складається з написання структури сайту за допомогою html, тобто текстовий контент та задання елементів. Написання стилів, таких як кольори, шрифти розміри, ефекти та розташування для елементів сайту використовуючи CSS або SCSS. Функціонал, логіка та анімація будуть написані на мові програмування JavaScript.

3.1 Розробка структури та стилізація сайту

Головна сторінка платформи складається з шапки сайту та основної частини. В шапці сайту користувачу буде запропоновано дві опції, створити аккаунт або увійти, якщо він був створений раніше. Після обрання функції, з'явиться вікно в якому користувач для реєстрації повинен буде вказати свою електронну адресу, придумати пароль та вписати своє ім'я, для користувачів які обрали "Вхід" у вікні потрібно буде вказати поштову адресу за якою вони зареєструвалися та пароль. Іконка меню буде відкривати додаткову панель, яка слугуватиме основною навігацією сайту і матиме п'ять посилань:

- Класи, на сторінку, де знаходяться ваші класи до яких ви приєдналися або створили.
- Проекти, веде на сторінку, де знаходяться домашні роботи також можна приєднати свій аккаунт GitHub і здавати посилання або написати код на самій платформі.
- Бали, веде на сторінку, де відображатимуться ваші бали.
- Профіль, веде на сторінку, де користувач зможе змінити аватарку, пароль, встановити нік та групу.
- Сторінка побудови схеми, користувачам представлена інтерактивна дошка на якій можна обирати пристрої та з'єднувати їх.

Також там наявні посилання на соціальні мережі та меседжери, де можна зв'язатися з підтримкою для вирішення питань.

На головній сторінці доступна можливість створити свій клас або приєднатися до існуючого. Після натискання на посилання “Створити”, у вікні буде надана можливість створити свій клас, у вікні потрібно буде ввести назву предмету та ім'я викладача. Після натискання на посилання “Приєднатися”, у спливаю чому в вікні вам потрібно буде ввести ID класу до якого хотіли б доєднатися, ви будете автоматично додані до обраного класу.

На сторінці “Класи”, користувач зможе заходити в свої класи, де матиме змогу вивчати наданий йому матеріал, який може бути у вигляді відео-лекції, також він матиме перелік домашніх завдань клацнувши на яке перейде до сторінки, де він зможе завантажити свій файл або надати посилання. На сторінці обраного класу є можливість доєднатися до уроку перейшовши за посиланням “Зустріч”. На сторінці “Проекти” є функції здачі домашніх завдань та проектів. Сторінка “Бали” показуватиме табличку, де відображатимуться назви класів і відповідні бали до них. Кожна з перерахованих сторінок матиме функцію повернення до головної сторінки платформи. Дизайн для кожного елемента розроблений в середовищі Figma.

Для виконання дизайну було обрано середовище Figma[9], оскільки воно не потребує витрат, має гарну репутацію та зручне у використанні. Платформа

буде доступна для використання на двох пристроях таких, як комп'ютер та планшет, для цього сайт буде адаптований під ці пристрої. Користувачі зможуть використовувати весь функціонал сайту на обох пристроях. Додана можливість для користувачів обирати мову (англійська, українська) відображення контенту на платформі.

Для написання структури сайту використовувався html з використання методології БЕМ, що забезпечить зручне написання та в майбутньому орієнтацію в стилях та елементах.

Головна сторінка сайту index.html код:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Study Bridge</title>
  <link rel="stylesheet" href="styles/main.scss">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">
  <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200">
</head>
```

`<!DOCTYPE html>`: Цей рядок оголошує тип документа, що використовується, у даному випадку HTML5.

`<html lang="en">`: Відкриває тег html і встановлює атрибут lang в значення "en", що вказує, що мова контенту сторінки — англійська.

`<head>`: Відкриває тег head, який містить метадані для документа.

`<meta charset="UTF-8">`: Встановлює кодування символів для документа

у UTF-8, що дозволяє відображати більшість символів з різних мов.

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`:

Встановлює налаштування області відображення для мобільних пристроїв, що забезпечує коректне масштабування і розмір сторінки. `width=device-width` встановлює ширину області перегляду рівною ширині екрану пристрою, а `initial-scale=1.0` встановлює початковий рівень масштабування.

`<title>Study Bridge</title>`: Встановлює заголовок документа, який відображається в заголовку вкладки браузера або у вікні.

`<link rel="stylesheet" href="styles/main.scss">`: Підключає файл стилів SCSS до документа. Проте зазвичай браузери не обробляють SCSS напряму, цей файл має бути скомпільований у CSS. Вірогідно, ваш інструмент для розробки автоматично виконує цю компіляцію.

`<link rel="preconnect" href="https://fonts.googleapis.com">`: Виконує попереднє з'єднання з сервісом Google Fonts для прискорення завантаження шрифтів.

`<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>`: Виконує попереднє з'єднання з сервером Google Fonts, який безпосередньо постачає шрифти. Атрибут `crossorigin` вказує, що цей запит може включати інформацію про крос-доменні запити.

`<link href="https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">`: Підключає шрифт Roboto з різними стилями і вагами (товщиною) з Google Fonts. Параметр `display=swap` вказує браузеру використовувати тимчасовий запасний шрифт до завантаження шрифту Roboto.

`<link href="https://fonts.googleapis.com/css2?family=Material+Symbols+Rounded:opsz,wght,FILL,GRAD@20..48,100..700,0..1,-50..200">`: Підключає шрифт Material Symbols Rounded з Google Fonts з налаштуваннями оптичного розміру (`opsz`),

ваги (wght), заливки (FILL) і градієнта (GRAD). Ці параметри налаштовують зовнішній вигляд і стиль символів.

Шрифт було взято з Google Fonts[10], картинки та іконки з SVG Repo[11] ці платформи безкоштовні і їх продукти можна використовувати.

Код для шапки сайту:

```
<header class="header">
  <button class="header__buttons header__buttons-register" data- translate-key="header__buttons-
register">
    РЕЄСТРАЦІЯ
  </button>
  <button class="header__buttons header__buttons-logIn" data-   translate-key="header__buttons-
logIn">
    УВІЙТИ
  </button>
  <div class="header__change">
    <div class="header__change__flag header__change__flag-   ukraine"></div>
    <a href="#" class="header__change__languages   header__change__languages-
ukranian">укр</a>
    <div class="header__change__flag header__change__flag-   usa"></div>
    <a href="#" class="header__change__languages   header__change__languages-
english">eng</a>
  </div>
  <button class="header__icon header__icon__calendar"></button>
  <a href="#menu" class="header__icon header__icon__menu"></a>
</header>
```

`<header class="header">`: Відкриває тег header, який містить інформацію про шапку сторінки Рис. 3.1. Атрибут `class="header"` задає клас header для цього елемента, який можна використовувати для стилізації за допомогою SCSS.

`<button class="header__buttons header__buttons-register" data-translate-key="header__buttons-register">`: Створює кнопку з класами `header__buttons` і

header__buttons-register.

`<div class="header__change">`: Створює контейнер `div` з класом `header__change`, який містить елементи для зміни мови.

`<a href="#" class="header__change__languages header__change__languages-ukranian">укр</a>`: Створює посилання з класами `header__change__languages` і `header__change__languages-ukranian`, яке представляє українську мову. Текст посилання — "укр". Атрибут `href="#"` вказує, що посилання не веде на іншу сторінку, використовується для обробки зміни мови через JavaScript.

`<a href="#" class="header__change__languages header__change__languages-english">eng</a>`: Створює посилання з класами `header__change__languages` і `header__change__languages-english`, яке представляє англійську мову. Текст посилання — "eng".

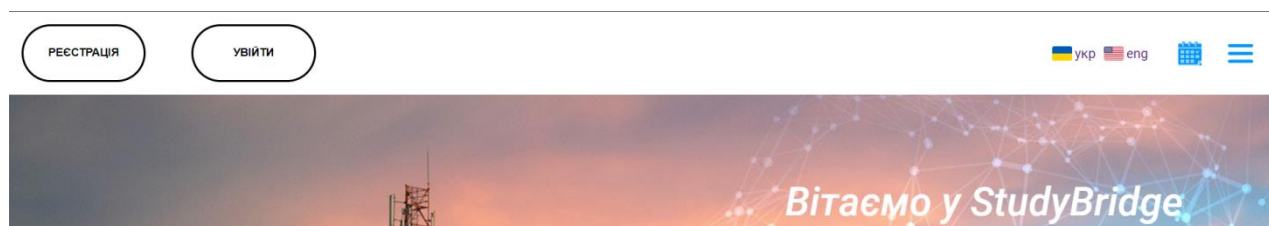


Рис. 3.1 Шапка сайту

Код структури головної сторінки:

```
<main class="main">
```

```
  <h1 class="main__name" data-translate-key="main__name">Вітаємо у StudyBridge</h1>
```

```
  <div class="main__picture"></div>
```

```
  <div class="main__buttons">
```

```
    <button class="main__button main__button-create" data-translate-key="main__button-create">
```

```
      СТВОРИТИ
```

```
    </button>
```

```
    <button class="main__button main__button-join" data-translate-key="main__button-join">
```

ПРИЄДНАТИСЯ

</button>

</div>

</main>

Всі картинки та іконки які використовуюються знаходяться в папці images, оскільки використовується SCSS для написання стилів, то посилання на картинку: url(../images/image.svg);

Код стилів для головної сторінки:

```
.main {
  font-family: 'Roboto', sans-serif;
  margin: 100px 0 0;
  padding: 0;
  background-image: url(../images/sec.jpg);
  background-size: cover;
  background-repeat: no-repeat;
  width: 100%;
  height: 595px;
  position: static;
  z-index: -1;

  &__name {
    position: absolute;
    margin: 100px 120px 0 0;
    right: 0;
    font-size: 48px;
```

```
font-weight: 500;

font-style: italic;

color: $main-color-white;

z-index: 0;

}
```

```
&__buttons {

    position: absolute;

    right: 0;

    left: 0;

    display: flex;

    flex-direction: row;

    justify-content: space-between;

    margin: 400px 150px 0 150px;

}
```

```
&__button {

    cursor: pointer;

    margin: 37px 40px 0 37px ;

    width: 230px;

    height: 80px;

    border: 5px solid $main-color-black;

    background-color: $main-color-buttons-menu;

    border-radius: 50px;

    display: flex;

    flex-direction: row;
```

```

justify-content: center;

align-items: center;

color: $main-color-black;

font-size: 24px;

font-weight: 600;

text-decoration: none;

transition: transform 400ms ease;

&:hover {

    transform: scale(1.2);

}

}

&__link {

    text-decoration: none;

}

}

```

Символ “&” є однією з особливостей SCSS. Символ “&” використовується для референції до батьківського селектора, це описано в w3schools[12]. У цьому випадку &__name означає .main__name, тобто елемент із класом __name, що є нащадком елемента з класом main, який відповідає за стилі до головної сторінки Рис. 3.2. Окремо для стилів створений файл “variables”, який містить змінні які визначають кольори, цей файл імпортований перший в main.scss для того аби браузер прочитав спочатку кольори, а потім використовував їх на сторінці.

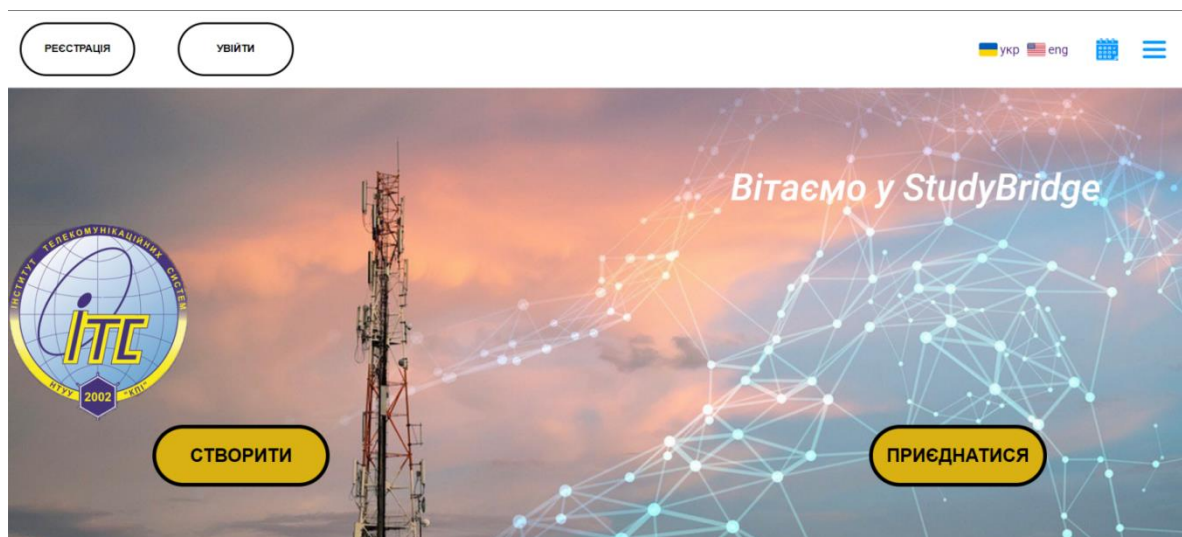


Рис. 3.2 Головна сторінка сайту

Календар Рис. 3.3 на головній сторінці показує точну дату, клікнувши на будь яку дату у вікні можна вписати нотатки.

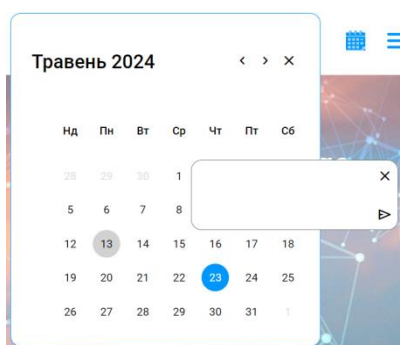


Рис. 3.3 Календар

Вікна реєстрації Рис. 3.4 та входу, які відкриваються на головній сторінці написані з використанням тегу `aside`, який взято з Mozilla Developer Center[13] та відповідними класами.

Регістрація

Пошта

Пароль

Ім'я користувача

Відправити

Рис. 3.4 Вікно реєстрації

Меню Рис. 3.5 відповідає за навігацію сайту, має поле пошуку в якому можна надрукувати запит і якщо елемент присутній на сторінці то одразу перейти на нього, або використовувати звичайну навігацію по сторінках.

Пошук

Курси

Проекти

Бали

Профіль

Складання схеми

Контакти

Instagram, Facebook, Twitter, WhatsApp icons

Рис. 3.5 Меню та навігація сайту

Код для навігації виконаний використовуючи JS:

Створивши масив рядків

```
let availableKeywords = [
  'Курси',
  'Проекти',
  'Бали',
  'Профіль',
  'Інше',
  'Комп'ютерні мережі',
  'Англійська мова',
  'Технології мобільного зв'язку',
  'Лекція',
  'Завдання',
  'Meet',
];
```

Функцією відображення результату пошуку

```
function display(result) {
  const content = result.map((list) => {
    return `<li><a href="${getLinkForKeyword(list)}" onclick="selectInput(this)">${list}</a></li>`;
  });

  menuResult.innerHTML = "<ul>" + content.join("");
}
```

Та функцією яка матиме посилання та перемикатиме на нього згідно запитів користувача

```
function getLinkForKeyword(keyword) {
  switch(keyword) {
    case 'Курси':
```

```
    return 'http://localhost:1234/lessons.html';

case 'Проекти':

    return 'http://localhost:1234/projects.html';

case 'Бали':

    return 'http://localhost:1234/points.html';

case 'Профіль':

    return 'http://localhost:1234/profil.html';

case 'Складання схеми':

    return '#';

case 'Комп'ютерні мережі':

    return 'http://localhost:1234/computersience.html';

case 'Англійська мова':

    return 'http://localhost:1234/english.html';

case 'Математика':

    return 'http://localhost:1234/math.html';

case 'Лекція':

    return 'http://localhost:1234/lecture.html';

case 'Завдання':

    return 'http://localhost:1234/task.html';

case 'Складання схеми':

    return 'http://localhost:1234/network.html';

case 'Meet':

    return 'http://localhost:1234/meet.html';

default:

    return '#';

}
```

}

3.2 Розробка функціоналу платформи

Сторінка “Курси” представляє всі дисципліни обрані або створені. AI помічника, який може відповідати на запитання студентів та викладачів, рекомендувати матеріали та завдання на основі особистих потреб студентів. Перевіряти завдання надані студентом, шляхом аналізу відповідей студента і порівнюючи їх з правильними або критеріями оцінювання. Для уникнення академічної нечесності буде впроваджено обмеження, які унеможливають використання AI для автоматичного вирішення завдань замість студентів.

Також буде доступним VR, який створюватиме віртуальне середовище, де студенти можуть експериментувати з мережевими налаштуваннями та пристроями безпосередньо в VR, що надає їм можливість отримати практичний досвід у безпечному і контрольованому середовищі, доступ до 3D-моделей мережових конфігурацій та пристроїв. Можливість відтворювати реальні сценарії телекомунікаційних проблем у віртуальному середовищі, де студенти зможуть вирішувати їх.

Для створення AI помічника Рис. 3.6 було взято посилання та ключ доступу до OpenAI API[14].

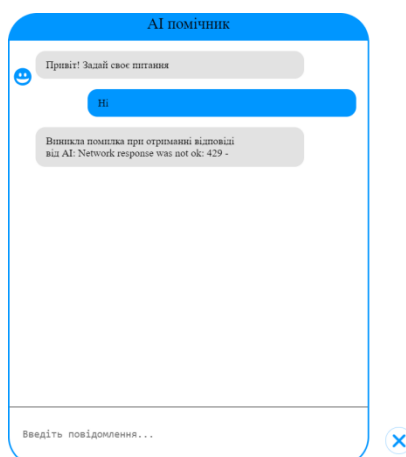


Рис. 3.6 Чат AI помічника

Повідомлення “Виникла помилка при отриманні відповіді від AI: Network response was not ok: 429 -” означає, що сервер відхилив запит користувача через обмежений доступ до API.

```
let API_URL = "https://api.openai.com/v1/chat/completions";
```

Ця адреса використовується для доступу до OpenAI's Chat Completions API, вона вказує, що запити надіслані на це посилання будуть оброблені серверами Open AI для виконання функцій чату за допомогою моделі GPT-3.5-turbo.

```
let API_KEY = "sk-proj-09w0Tken.....";
```

API_KEY ключ доступу до OpenAI's Chat Completions API після оплати, він є конфіденційним, забезпечує, що запити авторизовані і що ви маєте дозвіл на використання ресурсів OpenAI. Ключ дозволяє серверу визначити, хто надсилає запити і перевірити чи цей користувач має право доступу до серверів.

Код функцій для AI:

```
document.addEventListener('DOMContentLoaded', function() {
```

```
    const AIButton = document.querySelector('.lessons__AI');
```

```
    const AIWindow = document.querySelector('.AI');
```

```
    const closeAiWindow = document.querySelector('.AIconclose');
```

```
    const closeAi = document.querySelector('.AIconclose');
```

```
document.addEventListener('DOMContentLoaded', () => {
```

```
    const AItyping = document.querySelector('.AI__typing__textarea');
```

```
    const sendAiBtn = document.querySelector('.AI__sendIcon');
```

```
    const chatBox = document.querySelector('.AI__chatBox');
```

Функція створення повідомлення для чату, створює новий `<li>` елемент із зазначеними стилями і з введеним текстовим повідомленням

```
const createChatLi = (message, className) => {

  const chatLi = document.createElement('li');

  chatLi.classList.add('AI__incomeText', className);

  chatLi.innerHTML = `<p class="AI__text">${message}</p>`;

  return chatLi;

};
```

Функція `getAIResponse` асинхронна, вона надсилає запит до Open AI для отримання відповіді від сервера. Вона приймає введене повідомлення від користувача надсилає до API а потім повертає відповідь

```
const getAIResponse = async (userMessage) => {

  try {

    const response = await fetch(API_URL, {

      method: 'POST',

      headers: {

        'Content-Type': 'application/json',

        'Authorization': `Bearer ${API_KEY}`

      },

      body: JSON.stringify({

        model: "gpt-3.5-turbo",

        messages: [{ role: "user", content: userMessage }],

        max_tokens: 150

      })

    });
```

```

});

    if (!response.ok) {

        throw new Error(`Network response was not ok: ${response.status} -
        ${response.statusText}`);

    }

    const data = await response.json();

    if (!data.choices || data.choices.length === 0) {

        throw new Error('No choices returned from API');

    }

    const aiMessage = data.choices[0].message.content;

    return aiMessage;

} catch (error) {

    console.error('Error fetching AI response:', error);

    return `Виникла помилка при отриманні відповіді від AI: ${error.message}`;

}

};

```

Функція відповідає за обробку взаємодії користувача з чатом, вона приймає текст користувача перевіряє чи не порожній і відправляє до сервера, поки відповідь не відправилась з сервера користувачам буде показуватись напис “Думаю...” потім отримує відповідь та відображає в чаті з створеним новим `<li>` елементом із зазначеними стилями.

```

const handleAI = async () => {

    const userMessage = AItyping.value.trim();

    if (!userMessage) {

```

```

    return;
}

const newChatLi = createChatLi(userMessage, 'AI__incomeText-outcome');
chatBox.appendChild(newChatLi);

AItyping.value = "";

const thinkingChatLi = createChatLi("Думаю...", 'AI__incomeText-income');
chatBox.appendChild(thinkingChatLi);

const aiMessage = await getAIResponse(userMessage);

chatBox.removeChild(thinkingChatLi);

const aiChatLi = createChatLi(aiMessage, 'AI__incomeText-income');
chatBox.appendChild(aiChatLi);
};

sendAiBtn.addEventListener('click', handleAI);
});

```

Платформа представлена на двох мовах українська за замовчуванням, та англійська яку можна обрати на головній сторінці сайту. Сторінка завантажується на мові з локального сховища браузера або встановлює мову за замовчуванням, якщо відсутня. Потім він прив'язує обробники подій до кнопок зміни мови, які змінюють мову інтерфейсу та оновлюють всі елементи сторінки, які мають атрибут `data-translate-key` або `data-translate-placeholder`,

замінюючи їх вміст або атрибут на відповідні тексти з масиву перекладів translations. Обрана мова також зберігається у локальному сховищі для подальшого використання.

Код для зміни мови сторінки:

```
document.addEventListener('DOMContentLoaded', function () {

  const ukrBtn = document.querySelector('.header__change__languages-ukranian');

  const engBtn = document.querySelector('.header__change__languages-english');


  const translations = {

    ukr: {

      'header__buttons-register': 'РЕЄСТРАЦІЯ',

      'header__buttons-logIn': 'УВІЙТИ',

      'registerWin__title': 'Реєстрація',

      'registerWin__names-email': 'Пошта',

      'registerWin__names-password': 'Пароль',

      'registerWin__names-user': "Ім'я користувача",

      'buttonSend': 'Відправити',

      'logInWin__title': 'Увійти',

      'logInWin__names-email': 'Пошта',

      'logInWin__names-password': 'Пароль',

      'menu__search__input': 'Пошук',

      'menu__block-courses': 'Курси',

      'menu__block-projects': 'Проекти',

      'main__name': 'Вітаємо у StudyBridge',

      'menu__block-points': 'Бали',

      'menu__block-profile': 'Профіль',
```

```

'menu__block-other': 'Складання схеми',

'menu__contact__text': 'Контакти',

'main__button-create': 'СТВОРИТИ',

'main__button-join': 'ПРИЄДНАТИСЯ',

'createWin__title': 'Створити',

'createWin__names': 'Назва дисципліни',

'createWin__names-admin': 'Адмін',

'buttonNext-create': 'Далі',

'joinWin__title': 'Приєднатися',

'joinWin__names-code': 'Код дисципліни',

'join__text': 'Введіть код, який отримали від адміністратора дисципліни',

'buttonNext-join': 'Далі',

'calendar__currentData': 'Травень 2024',

'calendar__weeks': ['Нд', 'Пн', 'Вт', 'Ср', 'Чт', 'Пт', 'Сб'],

'lessons__main__text': 'Курси',

'lessons__disciplines': 'Мої дисципліни',

'lessons__courses-computerscience': "Комп'ютерні мережі",

},

eng: {

'header__buttons-register': 'REGISTER',

'header__buttons-logIn': 'LOG IN',

'registerWin__title': 'Registration',

'registerWin__names-email': 'Email',

'registerWin__names-password': 'Password',

'registerWin__names-user': 'Username',

'buttonSend': 'Send',

```

'logInWin__title': 'Log In',
'logInWin__names-email': 'Email',
'logInWin__names-password': 'Password',
'menu__search__input': 'Search',
'menu__block-courses': 'Courses',
'menu__block-projects': 'Projects',
'main__name': 'Welcome StudyBridge',
'menu__block-points': 'Points',
'menu__block-profile': 'Profile',
'menu__block-other': 'Building',
'menu__contact__text': 'Contacts',
'main__button-create': 'CREATE',
'main__button-join': 'JOIN',
'createWin__title': 'Create',
'createWin__names': 'Discipline Name',
'createWin__names-admin': 'Admin',
'buttonNext-create': 'Next',
'joinWin__title': 'Join',
'joinWin__names-code': 'Discipline Code',
'join__text': 'Enter the code received from the discipline admin',
'buttonNext-join': 'Next',
'calendar__currentData': 'May 2024',
'calendar__weeks': ['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat'],
'lessons__main__text': 'Courses',
'lessons__disciplines': 'My Disciplines',
'lessons__courses-computerscience': 'Computer Networks',

```

    }
};

function changeLanguage(lang) {

    document.querySelectorAll('[data-translate-key]').forEach(element => {

        const key = element.getAttribute('data-translate-key');

        if (translations[lang][key]) {

            element.textContent = translations[lang][key];

        }

    });

    const placeholders = document.querySelectorAll('[data-translate-placeholder]');

    placeholders.forEach(element => {

        const key = element.getAttribute('data-translate-placeholder');

        if (translations[lang][key]) {

            element.placeholder = translations[lang][key];

        }

    });

    localStorage.setItem('selectedLanguage', lang);

}

const savedLanguage = localStorage.getItem('selectedLanguage') || 'ukr';

changeLanguage(savedLanguage);

if (ukrBtn) ukrBtn.addEventListener('click', () => changeLanguage('ukr'));

if (engBtn) engBtn.addEventListener('click', () => changeLanguage('eng'));

});

```

На сторінці курси представлені всі ваші дисципліни обрані або створені,

AI помічник та VR кімната. Перейшовши в обрану дисципліну користувач бачитиме список лекцій та завдань, додатково можна створювати нові та редагувати їх. Сторінка лекції Рис. 3.7 дозволяє користувачам які створили дисципліни змінювати назву лекції, опис та додавати відеоматеріали.

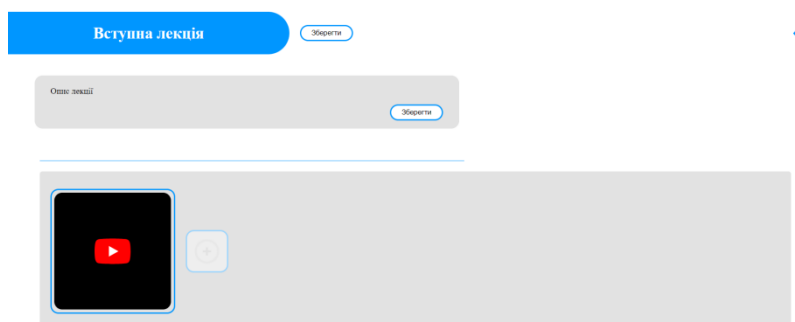


Рис. 3.7 Сторінка лекції

На сторінці завдання Рис. 3.8 функціонал про зміни назви та опису завдання аналогічний, додано варіанти здачі або вирішення завдання, завантаження файлу з вирішенням, написання на платформі текстом або якщо необхідно виконати завдання з програмування можна перейти у вікно “Coding” та написати код на платформі, якщо потрібно виконати проектування мережі, системи, схеми можна перейти на сторінку “Проектування” і виконати завдання там. Після виконання завдання потрібно натиснути кнопку “Здати” і на сторінці Бали колір статусу завдання зміниться на зелений, що означає, що завдання здано і надіслано керівнику.

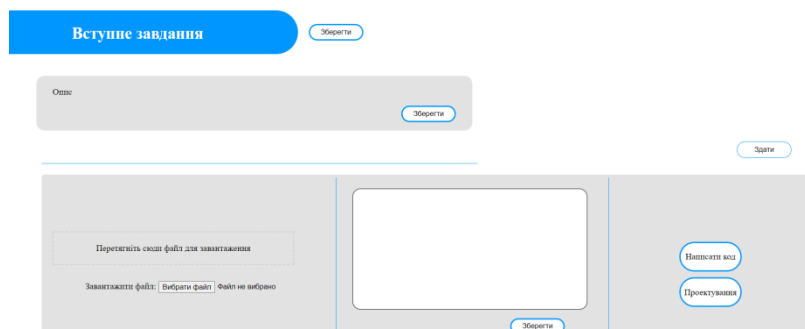


Рис. 3.8 Сторінка завдання

Сторінка написання коду(IDE) Рис. 3.9 складається з чотирьох вікон, написання html, scss, js і вікна отримання результату. В подальшому з додаванням серверної частини, можна додати більше можливостей для написання коду на різних мовах.



Рис. 3.9 Сторінка кодування

Коли користувач введе код HTML, CSS та JavaScript, він відображатиметься у вікні виведення, де HTML обробляється як вміст сторінки, CSS додається як внутрішній стиль, а JavaScript виконується. Крім того, код автоматично зберігається у локальному сховищі браузера, щоб забезпечити його збереження між сеансами користування. Коли сторінка завантажується, відбувається завантаження збереженого коду з локального сховища та автоматичний запуск функції `run()`, що оновлює вміст вікна виведення збереженим кодом.

Код для IDE:

```
function run() {
    let htmlCode = document.getElementById("html-code").value;
    let cssCode = document.getElementById("css-code").value;
```

```

let jsCode = document.getElementById("js-code").value;

let outputCode = document.getElementById("output-code");

outputCode.contentDocument.body.innerHTML = htmlCode + "<style>" + cssCode + "</style>";
outputCode.contentWindow.eval(jsCode);


saveToLocalStorage("htmlCode", htmlCode);
saveToLocalStorage("cssCode", cssCode);
saveToLocalStorage("jsCode", jsCode);
}


function saveToLocalStorage(key, value) {
    localStorage.setItem(key, value);
}


document.getElementById("html-code").addEventListener("keyup", function() {
    saveToLocalStorage("htmlCode", this.value);
});

document.getElementById("css-code").addEventListener("keyup", function() {
    saveToLocalStorage("cssCode", this.value);
});

document.getElementById("js-code").addEventListener("keyup", function() {
    saveToLocalStorage("jsCode", this.value);
});


window.addEventListener("load", function() {

```

```

document.getElementById("html-code").value = localStorage.getItem("htmlCode") || "";
document.getElementById("css-code").value = localStorage.getItem("cssCode") || "";
document.getElementById("js-code").value = localStorage.getItem("jsCode") || "";

run();

});

```

Всі завдання розміщені на сторінці Проекти Рис. 3.10, також там доступні посилання на розв’язання завдань методом кодування або проектування і посилання на GitHub, якщо до прикладу завдання розраховане на команду, то студенти зможуть завантажувати та обмінюватися роботами використовуючи це середовище.



Рис. 3.10 Сторінка проекти

Встановлення строків здачі завдання, виставлення балів та статус завдання можна переглянути на сторінці Бали Рис. 3.11, також викладачам доступна функція коментувати завдання.

Бали	Статус	Потрібно здати	Бали	Коментар
Завдання	Зробити	29.05.2024 17:13	0 / 10	Написати коментар
Завдання	Зробити	дд.мм.рррр --:--	0 / 10	Написати коментар
Завдання	Зробити	дд.мм.рррр --:--	0 / 10	Написати коментар
Завдання	Зробити	дд.мм.рррр --:--	0 / 10	Написати коментар
Завдання	Зробити	дд.мм.рррр --:--	0 / 10	Написати коментар

Рис. 3.11 Сторінка бали

На платформі доступно два види профілів Рис. 3.12, для викладачів та студентів. Викладачі які створюють курси та надають код доступу мають право редагувати їх та додавати завдання, виставляти строки здачі та бали, створювати конференції та лекції. Студенти також можуть створювати свої дисципліни, але якщо студент не є адміністратором дисципліни, то функціонал обмежений в зміні назви, опису, балів та строків. Адміністратор буде збережений та прив'язаний до дисципліна за допомогою серверу.

Студентський Профіль

Вибрати фото

Вибрати аватар

Ім'я користувача

Група

Електронна пошта

Завантажити пароль

Новий пароль

Підтвердження пароля

Зберегти

Профіль

Вибрати фото

Вибрати аватар

Ім'я користувача

Електронна пошта

Завантажити пароль

Новий пароль

Підтвердження пароля

Зберегти

Рис. 3.12 Профілі користувачів

Дошка для проектування Рис. 3.13 використовується для вирішення завдань або ж викладачем під час лекції для пояснення матеріалу студентам, для проектування мереж та систем використовується візуалізація архітектури програмного забезпечення або мережі, що допомагає в плануванні, розробці та підтримці системи.

При проектуванні нових телекомунікаційних мереж або вдосконаленні

існуючих, візуалізація архітектури допомагає краще планувати розташування обладнання. Телекомунікаційні системи, складаються з безлічі компонентів, включаючи сервери, маршрутизатори, комутатори, мережеві пристрої, програмне забезпечення та протоколи зв'язку. Дошка дозволяє візуально представити всі ці елементи та їхні взаємозв'язки, що полегшує розуміння та аналіз системи. Завдяки дошці для проектування, можна створювати деталізовані схеми та моделі, які допомагають у майбутньому обслуговуванні та модернізації системи. Це важливо не тільки на етапі створення, але й під час експлуатації системи, коли виникає потреба у внесенні змін або усуненні несправностей.

Функціонал сторінки дозволяє обирати пристрої з вікна після натискання на кнопку “Обрати пристрій”, після обрання пристрою він автоматично з'явиться на дошці, обмежень по кількості обраних пристроїв не має, їх можна переміщувати за допомогою миші і розташовувати на дошці як користувач захоче, також є можливість з'єднувати їх кольоровими лініями різної довжини.

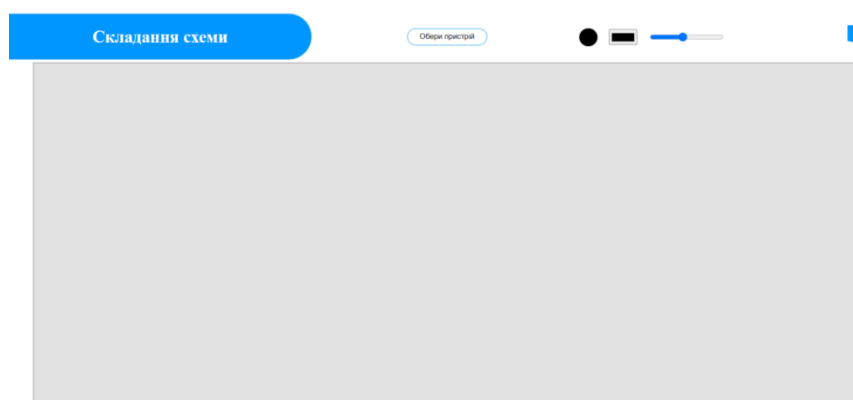


Рис. 3.13 Сторінка для проектування

Функціонал сторінки проектування був реалізований за допомогою JS:

```
document.addEventListener('DOMContentLoaded', () => {
  const button = document.querySelector('.chosedevise__button');
  const deviceWindow = document.querySelector('.chosedevise__chosedevise');
  const desk = document.querySelector('.chosedevise__desk');
```

```
const devices = document.querySelectorAll('.chosedevise__devise');
const drawingBoard = document.getElementById('drawingBoard');
const colorPicker = document.getElementById('color');
const sizePicker = document.getElementById('chosedevise__size');
const chosenColor = document.getElementById('chosenColor');
```

Ініціалізуємо змінні та стани для функції малювання.

```
let isDrawing = false;
let drawingEnabled = false;
let color = '#000000';
let size = 5;
let startX = 0;
let startY = 0;
```

Встановлюємо початковий колір для малювання, потім колір який обраний користувачем та змінюємо товщину пензля. Увімкнути або вимкнути режим малювання можна натиснувши на елемент поточного кольору, після того як використали пензлик необхідно натиснути на елемент в якому показується поточний колір аби вимкнути малювання.

```
chosenColor.style.backgroundColor = color;

colorPicker.addEventListener('input', (e) => {
  color = e.target.value;
  chosenColor.style.backgroundColor = color;
});

sizePicker.addEventListener('input', (e) => {
```

```

    size = e.target.value;

});

chosenColor.addEventListener('click', () => {

    drawingEnabled = !drawingEnabled;

    chosenColor.classList.toggle('active');

});

button.addEventListener('click', () => {

    deviceWindow.classList.toggle('choseDevise__chose--hidden');

});

```

Додаємо можливість обирати пристрої із відкритого вікна і додавання його на дошку.

В даній роботі кожен пристрій виконаний та стилізований як окремий елемент, тому їх кількість обмежена і для того аби додавати будь-які інші елементи потрібно прописувати кожен окремо.

```

devices.forEach(device => {

    device.addEventListener('click', () => {

        const clonedDevice = device.cloneNode(true);

        desk.appendChild(clonedDevice);

        deviceWindow.classList.add('choseDevise__chose--hidden');

        makeDraggable(clonedDevice);

    });

});

```

Функція яка дозволяє розміщувати пристрої в довільному порядку на дошці, за межами дошки пристрої функція не працює. Функція `makeDraggable` дозволяє перетягувати елементи. При натисканні миші всередині елемента

розраховуються відступи від поточної позиції миші до лівого та верхнього країв елемента. Під час руху миші обчислюються нові координати елемента з врахуванням меж дошки, щоб елемент не виходив за її межі. Нові координати обмежуються розмірами дошки, після чого елемент розміщується на новій позиції. При відпусканні миші обробники подій для руху та відпускання миші видаляються, а курсор повертається до стандартного вигляду.

```
function makeDraggable(element) {
  let offsetX, offsetY;

  element.addEventListener('mousedown', (e) => {
    offsetX = e.clientX - element.getBoundingClientRect().left;
    offsetY = e.clientY - element.getBoundingClientRect().top;
    element.style.cursor = 'grabbing';

    const onMouseMove = (e) => {
      const deskRect = desk.getBoundingClientRect();
      let newX = e.clientX - deskRect.left - offsetX;
      let newY = e.clientY - deskRect.top - offsetY;

      newX = Math.max(0, Math.min(newX, deskRect.width - element.clientWidth));
      newY = Math.max(0, Math.min(newY, deskRect.height - element.clientHeight));

      element.style.left = `${newX}px`;
      element.style.top = `${newY}px`;
    };
  });
}
```

```

const onMouseUp = () => {

  document.removeEventListener('mousemove', onMouseMove);

  document.removeEventListener('mouseup', onMouseUp);

  element.style.cursor = 'grab';

};

document.addEventListener('mousemove', onMouseMove);

document.addEventListener('mouseup', onMouseUp);

});
}

```

```

drawingBoard.addEventListener('mousedown', (e) => {

  if (!drawingEnabled) return;

  isDrawing = true;

  const rect = drawingBoard.getBoundingClientRect();

  startX = e.clientX - rect.left;

  startY = e.clientY - rect.top;

  if (!e.shiftKey) {

    const point = createPoint(e);

    drawingBoard.appendChild(point);

  }

});

```

Функція, що дає змогу малювати прямі лінії, якщо користувач натискає та тримає клавішу shift.

```

drawingBoard.addEventListener('mouseup', (e) => {

  if (!isDrawing) return;

```

```

isDrawing = false;

if (e.shiftKey) {

    const rect = drawingBoard.getBoundingClientRect();

    const endX = e.clientX - rect.left;

    const endY = e.clientY - rect.top;

    drawLine(startX, startY, endX, endY);

}

});

```

Малювання на дошці реалізоване без використання `<canvas>`, оскільки тоді не можна взаємодіяти з приладами на дошці. Отже було створено точковий елемент який малюватиме на дошці. При пересуванні миші, якщо режим малювання увімкнено та не натиснуто клавішу Shift, створюється і додається нова точка на дошку, використовуючи функцію `createPoint`. Ця функція отримує координати миші відносно дошки, створює новий `<div>` елемент з класом `drawing-point`, встановлює його розмір, колір та позицію відповідно до вибраного кольору і розміру.

Якщо ж під час малювання натиснута клавіша Shift, то при відпусканні миші викликається функція `drawLine`, яка малює лінію від точки початку до точки завершення. Вона обчислює довжину та кут нахилу лінії, створює `<div>` елемент з класом `drawing-line`, встановлює його розмір, колір, позицію і обертання, щоб намалювати лінію на дошці.

```

drawingBoard.addEventListener('mousemove', (e) => {

    if (!isDrawing || e.shiftKey) return;

    const point = createPoint(e);

    drawingBoard.appendChild(point);

});

function createPoint(e) {

```

```

const rect = drawingBoard.getBoundingClientRect();

const x = e.clientX - rect.left;

const y = e.clientY - rect.top;

const point = document.createElement('div');

point.className = 'drawing-point';

point.style.width = `${size}px`;

point.style.height = `${size}px`;

point.style.backgroundColor = color;

point.style.position = 'absolute';

point.style.left = `${x - size / 2}px`;

point.style.top = `${y - size / 2}px`;

return point;
}

function drawLine(x1, y1, x2, y2) {

  const length = Math.sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2);

  const angle = Math.atan2(y2 - y1, x2 - x1) * 180 / Math.PI;

  const line = document.createElement('div');

  line.className = 'drawing-line';

  line.style.position = 'absolute';

  line.style.width = `${length}px`;

  line.style.height = `${size}px`;

  line.style.backgroundColor = color;

  line.style.transformOrigin = '0 50%';

  line.style.transform = `rotate(${angle}deg)`;

  line.style.left = `${x1}px`;

```

```

line.style.top = `${y1 - size / 2}px`;

drawingBoard.appendChild(line);

}

});

```

3.3 API взаємодія з backend, процес інсталяції

Для повноцінної роботи сайту необхідно буде під'єднати серверну частину, яка оброблятиме запити від користувачів, зберігатиме та керуватиме даними. Серверна частина має надавати API Рис. 3.14, через яке frontend зможе здійснювати запити на авторизацію або реєстрацію, створення або приєднання до курсів, збереження лекцій та завдань. Кожен запит до API повинен бути безпечним, з використанням токенів для аутентифікації та авторизації користувачів. Інтеграція з backend дозволить забезпечити динамічність сайту, зберігання та обробку даних, а також масштабованість платформи для майбутніх оновлень та розширень.

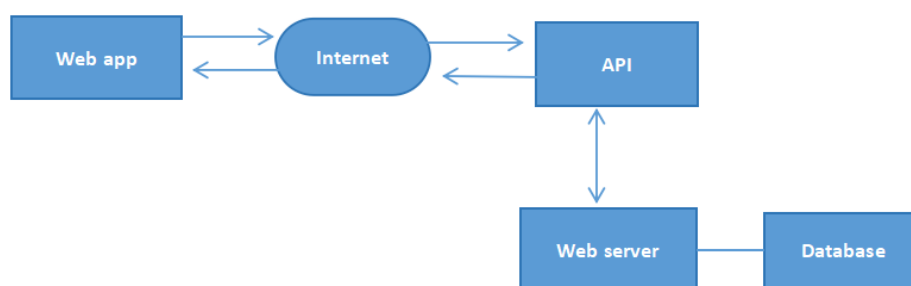


Рис. 3.14 Взаємодія з API

Функції входу та реєстрації забезпечують аутентифікацію користувачів, що є необхідним для доступу до ресурсів та даних платформи. На одну електронну адресу можна створити тільки один аккаунт.

Користувач заповнює поля реєстрації на платформі, що включають логін(електронна пошта), пароль та ім'я, потім frontend відправляє POST-запит з інформацією про нового користувача:

```
{
"username": "newUser",
"password": "newPassword",
"email": "newUser@example.com"
}
```

Backend повертає повідомлення про успішну реєстрацію або помилку, якщо користувач з даним логіном вже зареєстрований на платформі, дані зберігаються в базі. Користувач може обрати як він хоче бути зареєстрованим на платформі, якщо як викладач і він планує тільки створювати дисципліни він матиме один профіль, якщо як студент, то інший і матиме можливість створювати свої дисципліни.

```
{
"message": "User registered successfully"
}
```

Для входу на платформу користувач заповнює поля , що включають логін(електронна пошта), пароль, потім frontend відправляє POST-запит з інформацією про користувача і backend, якщо такий користувач є в базі, то його авторизують, якщо ні, то запропонують зареєструватись, цей процес створюють за допомогою мови програмування Python, фреймворку Flask та базою даних, яка працює з SQL.

Функціонал створення та приєднання до дисциплін працює наступним чином, користувач вводить дані назви дисципліни та ім'я, інформація передається на сервер де зберігається і повертає посилання на створену дисципліну та згенерований код для доступу. Коли користувач хоче

приєднатися до дисципліни, йому необхідно ввести код доступу, який йому видасть викладач, після відправлення коду на сервер, він перевіряє чи існує такий код, якщо так, то повертає посилання на дисципліну, але обмежує доступ до певних функцій та даних для даної дисципліни оскільки користувач не є викладачем. Студент може мати доступ лише для перегляду матеріалів і виконання завдань, але не може вносити зміни у вміст дисципліни. Якщо код не правильний сервер поверне помилку.

Лекції та завдання додані в дисципліні також зберігатимуться на сервері, адміністратор дисципліни матиме змогу створювати, редагувати назву та опис лекції та завдання, додавати туди відео матеріали. Студентам такі функції не доступні, вони можуть тільки переглядати викладений матеріал та взаємодіяти з функціоналом платформи для вирішення завдання, їх завдання після натискання кнопки “Здати” відправляються на сервер і статус завдання змінюється. Функціонал для виконання завдань на платформі такі, як друкування відповідей, завантаження файлів, написання коду або проектування на дошці після здачі завдання відправлятимуться на сервер та зберігатимуться і потім з’являтиметься у викладача для перевірок. Бали виставлені викладачем також відправлятимуться на сервер де зберігатимуться під кожного окремого користувача та завдання і будуть доступні для перегляду студентам.

Для інсталяції сайту потрібно спочатку налаштувати збір середовища це може включати встановлення програмного забезпечення, такого як Node.js, npm. Потім завантаження всіх файлів на GitHub-репозиторій. Прописання команди для встановлення всіх залежностей, вказаних у файлі package.json, за допомогою npm. Розгорнути створений frontend на сервері можна за допомогою FTP або SSH. Обидва методи мають свої переваги та недоліки. FTP зазвичай простіший у використанні, але менш безпечний, оскільки дані передаються у відкритому вигляді. У той же час, SSH є більш безпечним методом, оскільки використовує шифрування даних під час передачі. Після розгортання всіх frontend файлів на сервері, важливо встановити необхідні залежності в “package.json”, після цих кроків frontend готовий до використання з backend.

3.4 Тестування веб-інтерфейсу платформи

Проект завантажений на GitHub-репозиторій, окрім файлів пов'язаних з OpenAI URL та KEY, їх не можна туди завантажувати, оскільки вони є приватними, це правило платформи. Збірка проекту відбувається через файл “package.json”, командою npm start. Основний браузер для роботи та тестування Google Chrome, функціонал реєстрації, входу, створення або приєднання до дисципліни працює без передачі даних на сервер, календар показує точну дату та дає змогу вписувати нотатки, меню також працює і пошуковий рядок також працює.

AI помічник приймає повідомлення, але оскільки сервер працює за схемою орендування, то повідомлення не повертається користувачу, VR в демо версії і видає повідомлення, що пристрій не під'єднаний.

Всі посилання на сторінки працюють, назви та описи лекцій та завдань не зберігаються локально. Функціонал для вирішення завдання, написання текстової частини не зберігається локально, написання коду працює і зберігається, проектування на дошці працює. Здача завдань та встановлення строків працюють, але оскільки завдання не зберігається на сервері, то статус зміниться. Мовний функціонал працює, користувач може обрати українську чи англійську мову інтерфейсу.

Платформа протестована декількома браузерами, такими як, Google Chrome та Microsoft Edge, платформа працює справно.

Висновки:

Розроблено веб-інтерфейс платформи дистанційного навчання для дисциплін телекомунікаційного спрямування, використовуючи редактор коду VS Code, HTML для написання структури сайту, SASS для стилізації, JavaScript для логіки та функціоналу, Node.js та JSON для управління залежностями

проекту та створені файли конфігурації для легкого керування налаштуваннями.

Шрифт було взято з Google Fonts, в роботі використано шрифт 'Roboto', картинки та іконки з SVG Repo. Використання SVG іконок забезпечує високу якість графіки незалежно від роздільної здатності екрану, що покращує візуальне сприйняття сайту. Дизайн сайту орієнтований на мінімалістичний підхід, що робить інтерфейс інтуїтивно зрозумілим та зручним для користувачів.

Вікна для реєстрації, входу, створення або приєднання до дисципліни прописані тегом `aside`, відповідними стилями, їх функціонал написаний на JS. Для кожної сторінки сайту створені окремі файли, які логічно пов'язані між собою для забезпечення цілісності та зручності навігації. Для реалізації AI помічника було взято URL та KEY з платформи OpenAI для можливої взаємодії з їхнім сервером. Це дозволяє користувачам отримувати додаткову підтримку та консультації в режимі реального часу, підвищуючи ефективність навчання.

Дошка для проектування використовується для вирішення завдань або ж викладачем під час лекції для пояснення матеріалу студентам, реалізована функція обрання пристроїв з вікна яке відкривається при натисканні на кнопку "Обрати пристрій", після обрання він з'явиться на дошці, на ній також можна малювати, використовуючи мишку, що дозволяє ілюстративно взаємодіяти з елементами. Інтерфейс дошки розроблений таким чином, щоб забезпечити максимальну простоту та зручність використання навіть для користувачів без технічного досвіду.

Проект був успішно завантажений до GitHub-репозиторію[15] для подальшої інтеграції з серверами. Для перевірки функціоналу платформи було проведено тестування в браузерях Google Chrome та Microsoft Edge.

ЗАГАЛЬНІ ВИСНОВКИ ПО РОБОТІ

Проаналізовано існуючі платформи дистанційного навчання для різних спеціальностей, виявлено їхні переваги та недоліки. Під час аналізу було виявлено, що багато платформ мають недостатньо гнучкі інтерфейси, що ускладнює процес навчання для користувачів з різним рівнем підготовки. Крім того, не всі платформи забезпечують вивчення складних технічних дисциплін, такий як програмування, телекомунікації, інженерія. Платформи які інтегрували функціонал для вивчення таких дисциплін здебільшого розроблені для платних курсів. Поєднання інтуїтивного інтерфейсу користувача з потужними інструментами для створення та розв'язання завдань, а також забезпечення адаптивності для різних пристроїв та середовищ використання. Було особливо важливо зробити платформу доступною як для новачків, так і для досвідчених користувачів, надаючи всім рівні можливості для навчання.

Враховуючи сучасні тенденції веб-розробки було прийнято використовувати HTML, SASS для стилізації, JavaScript для функціональної логіки, а також Node.js та JSON для збору проекту та інтеграції з серверної частини. Було створено структуру проекту, що дозволяє легко розширювати та підтримувати код. Використання SASS дозволило створити структурований та зрозумілий код стилів, який легко підтримувати та оновлювати. Кожен компонент був створений з урахуванням принципів модульності, це зручно орієнтуватися в коді. Платформа включає основний функціонал, такі як реєстрація, управління курсами і завданнями та методи вирішення завдань, такі як написання текстової відповіді, написання коду, завантаження сторонніх файлів або використання інтерактивної дошки.

Особлива увага була приділена інтерактивній дошці для проектування, яка є важливим інструментом для студентів телекомунікаційних спеціальностей. Вона дозволяє викладачам наочно пояснювати складні концепції та моделі, а

студентам - активно взаємодіяти з матеріалом під час лекцій та виконання завдань. Також платформа забезпечує адаптивність і мобільну доступність, що дозволяє студентам використовувати її веб-версію на різних пристроях, включаючи комп'ютери та планшети. Це робить навчальний процес більш зручним та доступним незалежно від місця перебування користувачів.

Проект завантажений на GitHub, включаючи приватні файли, пов'язані з OpenAI URL та KEY для подальшої інтеграції серверної частини з frontend, це важливою для забезпечення функціональності, безпеки та масштабованості платформи. Впровадження API дозволить ефективно обробляти запити користувачів, керувати даними та забезпечувати аутентифікацію і авторизацію.

Подальша інтеграція з іншими навчальними ресурсами через API, що дозволяє розширити функціонал платформи без значних змін в основному коді. Створення і приєднання до дисциплін, збереження лекцій та завдань, а також обробка і зберігання відповідей на завдання. Вибір технологій для реалізації серверної частини та процесу реєстрації і авторизації гарантує безпеку та зручність використання. Завдяки правильному налаштуванню середовища та розгортанню frontend і backend, платформа буде готова до майбутніх оновлень і розширень, що сприятиме її подальшому розвитку та вдосконаленню.

Збірка проекту здійснюється через файл "package.json" за допомогою команди npm start, платформа успішно протестована у найпопулярніших браузерях Google Chrome та Microsoft Edge, функціонал працює бездоганно. Розробка веб-інтерфейсу дистанційного навчання успішно завершена і платформа готова до використання.

Здобуті результати можуть бути використані для практичного застосування у вищих навчальних закладах та компаніях, які розробляють платформи для дистанційного навчання для інтеграції нового функціоналу та поліпшення процесу навчання. Аналізування ефективності використання різних технологій у навчальних процесах та їхній вплив на результативність навчання.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Google Classroom — вебсервіс, створений Google для навчальних закладів з метою спрощення створення, поширення і класифікації завдань безпаперовим шляхом [Електронні ресурси] – Режим доступу до ресурсу: <https://classroom.google.com/>

2. Blackboard - це платформа для управління навчанням та електронного навчання, яка широко використовується в навчальних закладах та установах по всьому світу [Електронні ресурси] – Режим доступу до ресурсу: <https://www.blackboard.com/>

3. Moodle — навчальна платформа, призначена для об'єднання педагогів, адміністраторів і учнів в одну надійну, безпечну та інтегровану систему для створення персоналізованого навчального середовища [Електронні ресурси] – Режим доступу до ресурсу: <https://moodle.org/?lang=uk>

4. Освіта UA авторитетне українське медіа про систему освіти в Україні та за кордоном [Електронні ресурси] – Режим доступу до ресурсу: Режим доступу до ресурсу: https://osvita.ua/news/89291/#google_vignette

5. Інтернет-видання «Без купюр» [Електронні ресурси] – Режим доступу до ресурсу: <https://www.kypur.net/innovacijni-mozhlyvosti-platform-dlya-dystancijnogo-navchannya-interaktyvnisti-ta-adaptyvnisti/>

6. Visual Studio Code редактор вихідного коду, який можна використовувати з різними мовами програмування, включаючи C, C#, C++, Fortran, Go, Java, JavaScript, Node. js, Python, Rust [Електронні ресурси] – Режим доступу до ресурсу: <https://code.visualstudio.com/>

7. GitHub один з найбільших вебсервісів для спільної розробки програмного забезпечення. Існують платні та безплатні тарифні плани користування. Базується на системі керування версіями Git і розроблений на Ruby on Rails і Erlang компанією GitHub [Електронні ресурси] – Режим доступу

до ресурсу: <https://github.com/>

8. Parcel — це пакувальник для веб-додатків для розробників з різним досвідом. Він пропонує чудову швидку роботу з використанням багатоядерної обробки і не вимагає налаштувань [Електронні ресурси] – Режим доступу до ресурсу: <https://parceljs.org/>

9. Figma — векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи, що розробляється однойменною компанією [Електронні ресурси] – Режим доступу до ресурсу: <https://www.figma.com/>

10. Google Fonts бібліотека понад 950 вільно розповсюджуваних шрифтів, інтерактивний каталог для їх перегляду і прикладні програмні інтерфейси для використання вебшрифтів за допомогою CSS і на Андроїд [Електронні ресурси] – Режим доступу до ресурсу: <https://fonts.google.com/>

11. SVG Repo джерело безкоштовних картинок та іконок [Електронні ресурси] – Режим доступу до ресурсу: <https://www.svgrepo.com/>

12. W3Schools — це безкоштовний освітній вебсайт для навчання кодуванню онлайн. Спочатку випущений у 1998 році, він отримав свою назву від Всесвітньої павутини, але не пов'язаний з консорціумом W3. Пропонує курси охоплюючі усі аспекти веброботи, також публікує безкоштовні шаблони HTML [Електронні ресурси] – Режим доступу до ресурсу: <https://www.w3schools.com/>

13. MDN Web Docs, раніше Mozilla Developer Network, Mozilla Developer Center, є сховищем документації та навчальним ресурсом для веброботників. Його започаткувала Mozilla у 2005 році як єдине місце для документації про відкриті вебстандарти, власні проекти Mozilla та посібники для розробників [Електронні ресурси] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/>

14. OpenAI — це лабораторія досліджень штучного інтелекту, що складається з комерційної корпорації OpenAI LP та її материнської компанії, неприбуткової OpenAI Inc [Електронні ресурси] – Режим доступу до ресурсу:

<https://platform.openai.com/api-keys>

<https://platform.openai.com/docs/api-reference/introduction>

15. GitHub репозиторій [Електронні ресурси] – Режим доступу до ресурсу: https://github.com/misharemm/my_diploma_work