Міністерство освіти і науки України
Університет митної справи та фінансів

Факультет інноваційних технологій

Кафедра комп’ютерних наук та інженерії програмного забезпечення

Кваліфікаційна робота магістра

на тему «Методи та технології створення навчального веб середовища для генерування завдань»

Виконав: студент групи К22-2м

Спеціальність 122 Комп’ютерні науки

Пінчук Андрій Васильович
(прізвище та ініціали)

Керівник к.т.н., доц. Мала Ю. А.
(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент Університет митної справи та фінансів
(місце роботи)

(посада)

(науковий ступінь, вчене звання, прізвище та ініціали)

Дніпро – 2024

АНОТАЦІЯ

Пінчук А.В. Методи та технології створення навчального веб середовища для генерування завдань.

Дипломна робота (проєкт) на здобуття освітнього ступеня магістр за спеціальністю 122 «Комп’ютерні науки». – Університет митної справи та фінансів, Дніпро, 2024.

Актуальність теми створення навчального веб-середовища для генерування завдань проявляється у зростанні вимог до ефективності та якості навчального процесу в системі дистанційної освіти України з урахуванням технічного прогресу та змін в освітній системі, яка все більше спрямовується на європейські стандарти, вимагаючи постійного удосконалення. Сучасні комп'ютерні технології відкривають можливості для розробки інноваційних систем навчання, сприяючи обміну інформацією та досвідом незалежно від фізичної віддаленості між учасниками. Тому дослідження методів та технологій, а також створення навчального веб середовища для генерування завдань з урахуванням світового досвіду, особливостей і реалій стану вітчизняної освіти є однією із актуальних і важливих наукових і практичних проблем.

Об’єктом роботи є концептуально-методологічні основи створення та використання навчальних веб середовищ.

Предметом роботи є технологія створення навчального веб середовища.

Метою дипломної роботи є створення прототипу навчального веб середовища для генерування завдань.

Ключові слова: методи, технології, інструменти розробки, навчальне веб середовище з генерування завдань, веб-додаток, прототип, дистанційна освіта, HTML, CSS, JavaScript, TypeScript, Vue, Pinia, Vite, хостинг, домен.

Список публікацій здобувача:

1. Пінчук А. В., Ульяновська Ю. В. Функціональні й нефункціональні вимоги до розробки програмного забезпечення та їх роль в

управлінні якістю. Економіко-правові та управлінсько-технологічні виміри сьогодення: молодіжний погляд: матеріали міжнар. наук.-практ. конф., м. Дніпро, 4 лист. 2022 р. Дніпро, 2022. С. 411–412.

2. Пінчук А. В., Лебідь О. Ю. Проблеми та перспективи розвитку технологій Big Data в Україні. Економіко-правові та управлінсько-технологічні виміри сьогодення: молодіжний погляд: матеріали міжнар. наук.-практ. конф., м. Дніпро, 4 лист. 2022 р. Дніпро, 2022. С. 456–458.

3. Пінчук А. В., Рудянова Т. М. Шляхи управління ризиками у процесі розробки програмного забезпечення проєкту. Інноваційні технології, моделі управління кібербезпекою ІТМК-2023: матеріали міжнар. наук. конф., м. Дніпро, 18–20 квіт. 2023 р. Дніпро, 2023. С. 16–18.

4. Пінчук А. В., Критенко О. О. Цифровізація митної справи: сучасний стан та перспективи розвитку. Цифрове суспільство: управління, фінанси та соціум: матеріали міжнар. наук.-практ. конф., м. Дніпро, 28 квіт. 2023 р. Дніпро, 2023. С. 219–221.

5. Пінчук А. В., Перепьолкін С. М. Процедура спільного транзиту переваги та стан впровадження в Україні. Український правовий вимір: пошук відповідей на глобальні міжнародні виклики: матеріали міжнар. наук.-практ. конф., м. Дніпро, 26 травня 2023 р. Дніпро, 2023. С. 144–146.

6. Леснікова І. Ю., Халіпова Н. В., Рудянова Т. М., Пінчук А. В., Дьяченко В. Д., Білоножко Д. О. Моделювання багатоетапного процесу логістичного постачання вантажів в сучасних умовах України. Grail of Science. 2023. № 27. С. 378–386.

ANNOTATION

Pinchuk A.V. Methods and technologies for creating a web-based learning environment for generating tasks.

Diploma thesis (project) for obtaining the Master's degree in specialisation 122 "Informatics". - Dnipro State University of Customs and Finance, 2024.

The relevance of the topic of creating a web-based learning environment for task generation is manifested in the increasing requirements for the efficiency and quality of the educational process in the system of distance education in Ukraine, taking into account the technological progress and changes in the educational system, which is increasingly oriented towards the European standards and requires continuous improvement. Modern computer technologies open opportunities for the development of innovative learning systems, facilitating the exchange of information and experience regardless of the physical distance between participants. Therefore, the study of methods and technologies, as well as the creation of a web-based learning environment for the generation of tasks, taking into account international experience, characteristics and realities of the state of domestic education, is one of the most relevant and important scientific and practical problems.

The subject of the work is the conceptual and methodological basis for the creation and use of web-based learning environments.

The subject of the work is the technology of creating a web learning environment.

The purpose of the thesis is to create a prototype of a web-based learning environment for task generation.

Keywords: methods, technologies, development tools, educational web environment for task generation, web application, prototype, distance education, HTML, CSS, JavaScript, TypeScript, Vue, Pinia, Vite, hosting, domain.

List of the applicant's publications:

1. Pinchuk A. V., Ulyanovskaya Y. V. Functional and non-functional requirements for software development and their role in quality management. Economic-legal and management-technological dimensions of today: a youth perspective: materials of the international scientific and practical conference, c. Dnipro, 4 November. 2022 Dnipro, 2022. C. 411–412.

2. Pinchuk A. V., Lebid O. Y. Problems and prospects for the development of Big Data technologies in Ukraine. Economic-legal and management-technological dimensions of today: a youth perspective: materials of the international scientific and practical conference, c. Dnipro, 4 November. 2022. Dnipro, 2022. C. 456–458.

3. Pinchuk A. V., Rudianova T. Ways of risk management in the process of project software development. Innovative technologies, cyber security management models ITMK-2023: materials of the international scientific conference, c. Dnipro, 18-20 April. Dnipro, 2023. C. 16–18.

4. Pinchuk A. V., Krytenko OO Digitalisation of customs: current state and prospects for development. Digital society: management, finance and society: materials of the international scientific and practical conference, c. Dnipro, 28 April. Dnipro, 2023. C. 219–221.

5. Pinchuk A. V., Perepolkin S. M. The procedure of joint transit of benefits and the state of implementation in Ukraine. Ukrainian legal dimension: searching for answers to global international challenges: materials of the international scientific and practical conference, c. Dnipro, 26 May 2023 Dnipro, 2023. C. 144–146.

6. Lesnikova I. Y., Khalipova N. V., Rudyanova T. M., Pinchuk A. V., Dyachenko V. D., Bilonozhko D. O. Modelling of the multi-stage process of logistics supply of goods in modern conditions of Ukraine. Grail of Science. 2023. № 27. C. 378–386.

ЗМІСТ

ВСТУП

На сучасному етапі розвитку освітньої системи України спостерігається інтенсивна імплементація новаторських концепцій, впровадження передових педагогічних технологій та застосування науково-методичних досягнень. Формується нова система навчання, яка використовує сучасні інформаційні та комунікаційні технології. Одним із ключових напрямків реформ в освітньому процесі є впровадження електронних освітніх видань і ресурсів, включаючи інформаційні системи навчального призначення.

Зростаюча популярність використання комп'ютерів та Інтернету у навчальних процесах свідчить про активний перехід до нових форм інтерактивного навчання, спрощуючи взаємодію між викладачем та студентом. Динамічно розвивається система дистанційної освіти, що давно стала стандартом у західних країнах. Цей розвиток сприяє використанню потужної комп'ютерної техніки та обміну досвідом через Інтернет.

Оснащеність сучасних навчальних закладів комп'ютерною технікою та доступ до Інтернету відкриває нові можливості для переходу до інформаційних технологій в оцінці знань. Застосування сучасних комп'ютерних систем у сфері освіти обіцяє значні переваги: гнучкий графік навчання, доступ до освіти для тих, хто не може скористатися традиційною формою, економія часу та коштів. Такий підхід сприяє індивідуальному розвитку та навчанню.

Актуальність теми створення навчального веб-середовища для генерування завдань проявляється у зростанні вимог до ефективності та якості навчального процесу в сучасному світі. З урахуванням технічного прогресу та змін в освітній системі, яка все більше спрямовується на європейські стандарти, матеріали та засоби навчання швидко застарівають, вимагаючи постійного удосконалення. Сучасні комп'ютерні технології відкривають можливості для розробки інноваційних систем навчання, сприяючи обміну інформацією та досвідом незалежно від фізичної віддаленості між

учасниками. Побудова ефективних комп’ютерних систем освіти з урахуванням світового досвіду, особливостей і реалій стану вітчизняної освіти є однією з актуальних і важливих наукових і практичних проблем.

Об’єктом роботи є концептуально-методологічні основи створення та використання навчальних веб середовищ.

Предметом роботи є технологія створення навчального веб середовища.

Метою дипломної роботи є створення прототипу навчального веб середовища для генерування завдань.

Для досягнення мети дипломної роботи були поставлені такі задачі:

– аналіз предметної області;

– обґрунтування актуальності дослідження;

– постановка завдання дослідження;

– огляд існуючих розв’язків поставленої задачі;

– аналіз та вибір засобів реалізації навчального веб середовища;

– практична реалізація прототипу навчального веб середовища для генерування завдань.

Методи дослідження: теоретичні – аналіз літератури, що відносяться до програмного забезпечення для розробки веб-застосунків; емпіричні – проєктування та розробка прототипу навчального веб середовища для генерування завдань.

Практичне значення створеного навчального веб середовища для генерування завдань полягає в тому, що це середовище забезпечить генерацію випадкових варіантів завдань без необхідності створювати кожен варіант окремо, можливість додавати вкладення до курсу, тесту та окремого завдання тесту за потреби, швидке створення тестів та швидку й ефективну взаємодії між викладачем та учасниками курсу завдяки інтуїтивно зрозумілому інтерфейсу веб середовища.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. ОБГРУНТУВАННЯ АКТУАЛЬНОСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ДОСЛІДЖЕННЯ

1.1 Огляд літератури щодо створення навчальних веб середовищ для генерування завдань

Завдяки легкості доступу до різноманітних даних за допомогою комп'ютерних мереж та розвитку цифрових методів представлення інформації дистанційна освіта набирає популярність. Дистанційна освіта вважається не лише зручним, але й ефективним засобом набуття знань, представляючи собою вагому альтернативу традиційним методам навчання. Вона дозволяє вивчати матеріал в комфортних умовах, сприяючи кращому засвоєнню інформації.

Веб-застосунки є програмами, що працюють за принципом клієнт-сервер, використовуючи веб-браузери та веб-технології для виконання завдань через Інтернет. Вони мають клієнтську та серверну складові, де клієнт - це програма, яку запускає користувач, а сервер - місце зберігання інформації.

Розробка веб-застосунків включає використання різних мов програмування: серверні скрипти (Python, Java, тощо) для зберігання та отримання інформації, мова програмування клієнту JavaScript з мовою розмітки гіпертексту HTML та каскадними таблицями стилів CSS для відображення інформації користувачу.

Для роботи веб-застосунку необхідний веб-сервер для управління запитами клієнта, сервер застосунків для обробки завдань та база даних для зберігання інформації. Доступ до веб-застосунку відбувається через інтернет за допомогою веб-браузера. Основна суть роботи веб-застосунків полягає в тому, що користувач надсилає запит через інтерфейс програми на веб-сервер, веб-сервер передає запит на сервер застосунків, який обробляє запит і надсилає результати назад на веб-сервер, а потім інформація відображається на екрані користувача [1].

Веб-сервери обробляють клієнтські запити, надсилають їх та формують на основі запитів звернення до зовнішніх джерел даних. Веб-сервер працює з різними протоколами передачі даних, забезпечуючи клієнт-серверне взаємодію і не призначений для обробки користувальницьких подій без формування клієнтського запиту. Будь-яка дія користувача формується на стороні клієнта у вигляді запиту [1].

Протоколи зв'язку відіграють ключову роль у встановленні з'єднань між сервером та клієнтом. Запити клієнта передаються через протокол, для якого на сервері встановлений відповідний обробник. Наразі основним протоколом зв’язку між клієнтом та сервером в межах веб-додатку є протокол передачі гіпертекстових документів HTTP [1].

Сховища даних вирішують задачу зберігання інформації в зовнішніх, щодо програми, джерелах. Такими джерелами можуть бути різні бази даних, файлові системи і зовнішні носії інформації, які містять програмні елементи.

На рис. 1.1 схематично зображено архітектуру веб-додатку в загальному вигляді. Окремо слід звернути увагу, що зазвичай компанії при розробці веб-додатків розділяють веб-додатки на так звані Frontend та Backend сегменти, призначаючи розробників відповідно до їх спеціалізації на певному сегменті.

Основна мета побудови архітектури веб-додатку – це можливість виявити як функціональні вимоги, так і нефункціональні вимоги до системи і обробити їх для поліпшення загальної якості додатку, що в подальшому надає можливість контролювати продуктивність, масштабованість і надійність [2].

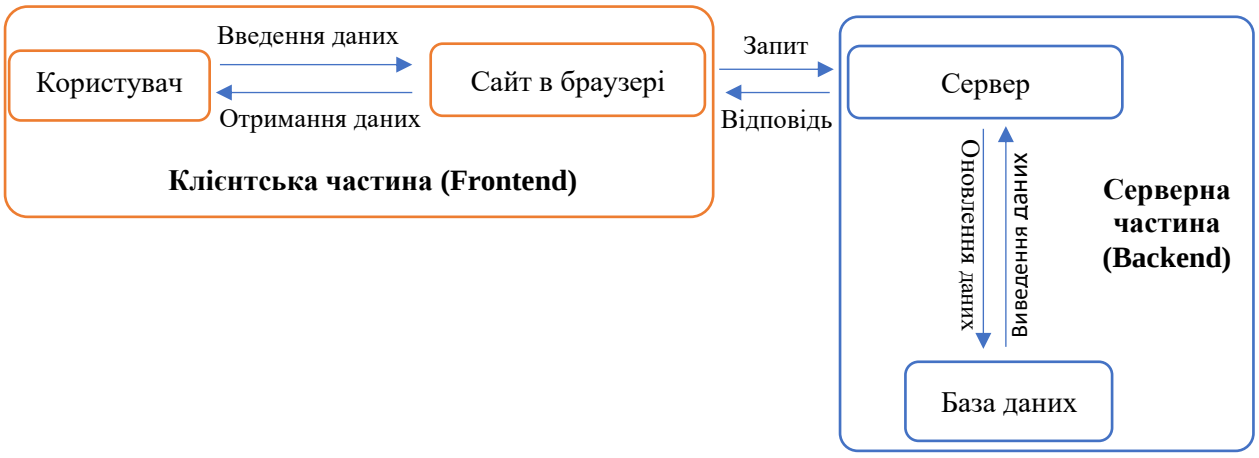


Рисунок 1.1 – Архітектура веб-застосунку

Сучасний стан розвитку освітніх веб платформ характеризується інтенсивним впровадженням цифрових технологій, що відображає глобальні тренди в освіті та відповідає сучасним вимогам суспільства. З розвитком цифрових технологій та посиленням потреб у гнучких та інноваційних підходах до навчання, освітні веб платформи стають ключовим елементом в освітньому процесі.

Освітні веб платформи, такі як Moodle, використовуються для масового дистанційного навчання, надаючи можливість для гнучкого, доступного та інтерактивного навчання. Такі платформи забезпечують ефективне управління навчальними процесами та ресурсами, підтримують розвиток неформальної освіти і сприяють самостійній роботі студентів [3]. Також зростає значення інновацій у філологічній освіті, де цифрові платформи відіграють ключову роль у модернізації та адаптації навчальних програм [4].

Розвиток освітніх веб платформ не обмежується лише технічними аспектами, але також включає розробку педагогічних стратегій та методів, які відповідають вимогам сучасної освіти [5]. Тамаркіна О.Л. підкреслює важливість інтеграції інформаційних технологій та цифрових ресурсів для підвищення ефективності самостійного навчання [6]. Дослідження Спіріна О.М. та Колос К. Р. вказують на необхідність розвитку масового дистанційного навчання, особливо в умовах карантину, що вимагає використання платформ, здатних підтримувати велику кількість користувачів [3].

Значення цифрової трансформації в освіті відзначається у роботах Бикова В. Ю., де акцентується на важливості розвитку комп'ютерно-технологічних платформ [7]. Це включає не тільки технічні аспекти, але й розвиток цифрових компетентностей та стандартів інформаційної безпеки. На закінчення, важливість цифрових освітніх веб платформ для реалізації неформальної освіти підкреслюється Острогою М., Шамонім В. та Шершенею О., де зазначається роль цих платформ у розширенні доступу до освіти [8].

Окремо слід відзначити навчальні середовища для генерування завдань, що дозволяють студентам самостійно проходити тестування по різним темам в автоматичному режимі без втручання викладача. Вони дозволяють зменшити витрати часу викладача на проведення поточного контролю або екзамену. Одним з прикладів навчального середовища з генерації завдань є середовище компанії «Mate academy», яка створена для того, щоб на початкових етапах студенти самостійно розібрались з основами програмування. В цьому середовищі присутні автоматичні тести для того, щоб перевірка правильності виконання відбувалась без втручання викладачів у цей процес. Водночас студент може задати питання в чаті до цього завдання іншим студентам або менторам. На початку 2023 року компанія підключила штучний інтелект до цього середовища генерації завдань і тепер студент може поставити питання також штучному інтелекту.

Під час проходження практики на базі практики було досліджено освітню платформу та середовище генерації завдань «Dream English» На цій освітній платформі можливо, як пройти повноцінний курс з англійської мови, так і пройти підготовку для складання міжнародного екзамену «The Pearson Test of English (PTE)» з англійської мови. Одним з можливих шляхів підготовки є окреме середовище генерації завдань з англійської мови, де студент може обрати напрямок завдань (аудіювання, письмо, читання тощо) і далі в залежності від напрямку генеруються завдання. Чим більше завдань виконає студент перед проходженням екзамену, тим більше в нього шансів на успішне складання екзамену за рахунок того, що завдання схожі на ті що будуть на реальному екзамені. Вправи, які потребують перевірки голосової відповіді студента (наприклад читання) використовують сторонній сервіс «Whisper API» компанії «OpenAI» для транскрипції голосової відповіді у текст. Після отримання тексту від API на сайті здійснюється перевірка тексту на правильність відповідно до правильної текстової відповіді.

Отже, освітні веб платформи в сучасному світі перетворюються на фундаментальний елемент підготовки фахівців, забезпечуючи не тільки

теоретичні знання, але й розвиваючи практичні навички та компетентності, необхідні для успішної професійної діяльності. Адаптація до цифрових технологій, гнучкість у методиках навчання та інтеграція новітніх комунікаційних засобів в освітній процес відкривають нові можливості для здобувачів вищої освіти, водночас викликаючи потребу в постійному саморозвитку.

1.2 Аналіз методів та технологій створення навчальних веб середовищ для генерування завдань

Процесом розробки програмного забезпечення називають сукупність ряду послідовних дій, спрямованих на розробку програмного забезпечення (ПЗ). Існує кілька моделей такого процесу, кожна з яких описує свій підхід до процесу розробки програмного забезпечення. Вибір певної моделі здійснюється відповідно до обраної методології розробки програмного забезпечення. На сьогодні існують каскадна (водоспадна), ітераційна та спіральна. Каскадна модель розбиває проєкт на декілька послідовних етапів. Ітераційна та спіральна моделі пропонують ітераційний підхід до створення програмного забезпечення [9].

Незалежно від вибору моделі процесу розробки програмного забезпечення, обов’язково мають бути вимоги до програмного забезпечення. Саме за допомогою вимог визначається якість програмного забезпечення на ранніх етапах процесу розробки та існує можливість управління його якістю. Розрізняють два основних типи вимог: функціональні і нефункціональні. Функціональні вимоги описують внутрішню роботу програмного забезпечення, його поведінку: калькулювання даних, маніпулювання даними, опрацьовування даних та інші специфічні функції які повинна виконувати система [9].

Нефункціональні вимоги описують, як має працювати програмне забезпечення, у яких умовах або з якими обмеженнями та які властивості або

характеристик воно повинне мати в цілому незалежно від наявних функцій. Також ці вимоги задають критерії для оцінки якості роботи програмного забезпечення [9].

Створення веб-застосунку – це комплекс заходів і дій з планування та створення сайту в мережі Інтернет в залежності від поставлених цілей і завдань. Створення технічного завдання є одним з найважливіших етапів розробки сайтів та інших веб застосунків. Саме на цьому етапі приймаються всі найважливіші рішення. Для замовника, технічне завдання є тим документом, на який завжди можна послатися, в разі, якщо кінцевий продукт не відповідає поставленому завданню. Якщо виконавець не приділяє достатньо уваги цього етапу, то, з високою ймовірністю, результат його роботи буде далекий від того, що очікував отримати замовник. Винятком можуть бути нескладні проєкти, наприклад, прості сайти або сайти-візитки. Для таких проєктів технічне завдання є лише формальністю. Навіть для деяких сайтів середньої складності технічне завдання не завжди обов'язково [10].

Розробка структури застосунку включає все, що стосується його змісту та інформаційної стратегії, яка визначає, як повинна бути організована подача інформації, задля прискорення та полегшення її пошуку майбутніми користувачами. Створюється карта сайту, на якій наявні взаємозв'язки типових сторінок та функціональні можливості.

Дизайн, безсумнівно, один з найважливіших етапів розробки сайтів і веб застосунків. Нерідко бувають випадки, коли вирішальну роль грає упаковка, а не якість самого продукту. Саме оформлення задає відношення користувача до застосунку. Вкрай бажано мати уявлення про моду і смаки цільової аудиторії оскільки те, що подобається одній групі людей, може відштовхнути інших. Крім зовнішнього сприйняття, важливим фактором є зручність використання застосунку. Розробка інтерфейсу тісно пов'язана з дизайном проєкту і тому часто їх розглядають разом, як один етап. Розташування і зовнішній вигляд елементів навігації, інтуїтивне сприйняття цих елементів

користувачем, мінімізація дій користувача, - це ті завдання, які доводиться вирішувати дизайнеру при створенні дизайну і навігації.

Досить важливим етапом розробки веб застосунків, є верстка. Якість виконання цього етапу буде впливати на ряд параметрів застосунку. в першу чергу. Верстка впливає на швидкість завантаження веб сторінки, стабільність її роботи та кросбраузерність – здатність сайту в різних браузерах відображатися однаково. Не мало важливим є вплив верстки на обробку сайту пошуковими машинами. Верстка виконується в суворій відповідності зі специфікацією мови розмітки. Специфікація вказує браузеру, як інтерпретувати сторінку, що переглядається. Якщо вверстка не відповідає специфікації, то вона вважається невалідною і може неправильно відображатись в браузері. Для перевірки валідності верстки існує ряд відповідних веб-додатків.

Програмування є одним з основних етапів розробки веб-застосунків і сайтів. Найважливішим завданням розробника є вибір найбільш оптимального рішення для досягнення поставленої мети. Саме тому необхідно заздалегідь, в деталях, продумати весь проєкт. Інакше вже під час розробки веб-додатку може виникнути ситуація, коли обрані розробником інструменти не зможуть задовільнити потреби проєкту. Чим складніше проєкт, тим важливіше роль програмування в його реалізації, і тим вище рівень кваліфікації потрібно для його виконавця. Етап програмування сайтів і веб-застосунків можна розбити на різні етапи: проєктування баз даних, програмування API, програмування інтерфейсів клієнтської частини.

Одним з найважливіших етапів є тестування застосунку. Задача цього етапу проста – шукати помилки, які виникли при написанні коду. Метою комплексного тестування є перевірка того, що кожен модуль програмного продукту узгоджується з іншими модулями продукту. При комплексному тестуванні може використовуватися технологія обробки зверху вниз і знизу вгору, при якій кожен модуль, який є листом в дереві системи, інтегрується з наступним модулем нижчого або вищого рівня, поки не буде створено дерево

програмного продукту. Ця технологія тестування спрямована на перевірку не тільки тих параметрів, які передаються між двома компонентами, а й на перевірку глобальних параметрів і, в разі об'єктно-орієнтованого застосунку, всіх класів верхнього рівня [10].

Серед основних інструментів для створення веб-додатків виділять HTML, CSS, JavaScript.

HTML – це мова розмітки гіпертексту або стандартизована мова розмітки документів для перегляду вебсторінок у браузері. Актуальна версія цієї мови – 5. HTML документ складається з великої кількості елементів – тегів, які використовуються для відображення певних структур даних, форматування тексту або є службовими. Приклад такої розмітки відображено на рис. 1.2. Зазвичай документ розділяють на декілька основних тегів: !DOCTYPE, html, head, body.

!DOCTYPE є службовим тегом і слугує для задання документу пояснення, що наступний код використовує останню версію HTML. Без вказання цього тегу браузери можуть не зрозуміти теги, які були впроваджені лише в п’ятій версії мови.

Тег html не відображає нічого на сторінці. Він є кореневим елементом і зберігає в собі всю майбутню структуру сайту та службовий блок head. Для звернення через мову CSS до цього тегу можна використати селектор :root. Head та body належать до блоку html та відповідають за наповнення сторінки.

Head – це службовий тег, в якому вказуються службові інструкції для браузеру, такі як задання стандарту кодування, опису та ключових слів тематики сайту, підключення файлів стилю та скриптів.

Ключові слова та опис сайту задають відображення сайту в різних пошукових системах та SEO оптимізують сайт. Одним з способів задати певні налаштування – це передати їх у вигляді атрибутів тегу meta. Наприклад, для того щоб задати стандарт кодування необхідно додати до тегу head <meta charset=”utf-8”>.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <link rel="icon" href="/favicon.svg">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <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=Inter&family=Roboto:wght@400;500;700&display=swap" rel="stylesheet">
    <title>Examinate</title>
  </head>
  <body>
    <div id="app"></div>
    <script type="module" src="/src/main.ts"></script>
  </body>
</html>
```

Рисунок 1.2 – Базова структура HTML документу

Тег body відображається у браузері та є контейнером, у якому знаходиться уся розмітка веб-сторінки. В HTML5 була створене поняття семантика за рахунок нових структурних елементів. Раніше, при створенні веб-сторінок розробники надавали блокам певне значення та логіку за допомогою додавання та відповідного найменування класів до елементів.

Стилі найменування блоків в класах у розробників різні. Для створення єдиного рішення щодо семантики були створені відповідні теги (header, nav, main, footer та ін.). Пошукові роботи орієнтуються на ці теги та виконують певну оптимізаційну роботу за аналізом та структурою цих тегів. Семантика спростила СЕО оптимізацію та процеси, які пов’язані з просуванням веб-додатків та сайтів у пошукових системах.

CSS – каскадні таблиці стилів, задають візуальну складову сайту. Файл підключається через тег link в тег head. Кожен тег має атрибути – певні параметри, які конкретизують роботу тегу та можуть передавати певну інформацію до елементу. Для того щоб звернутися до елементу треба скористатись одним з селекторів: ідентифікатор, клас, атрибут або тег.

Ідентифікатор – це спеціальне значення елементу, яке має бути одним на весь документ, задається в документі CSS зі знаком # (рис. 1.3). Серед інших селекторів має найвищу специфіку – це означає, що властивості описані в даному селекторі будуть використовуватись в першу чергу та перекривати інші властивості.

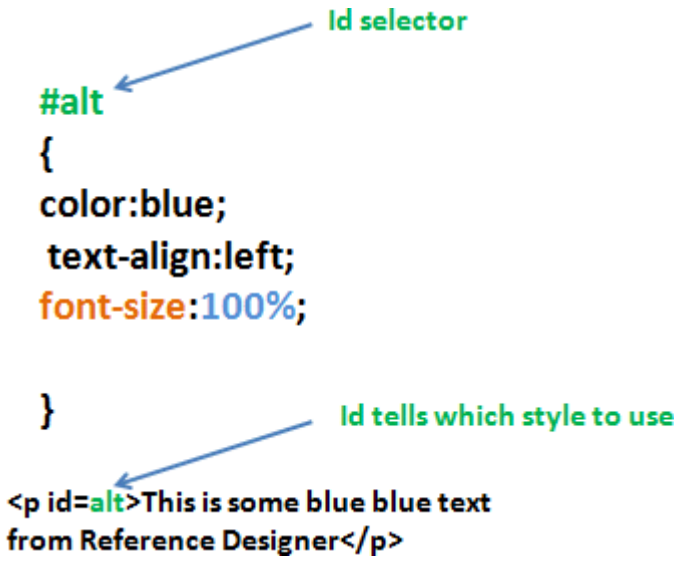


Рисунок 1.3 – Приклад використання ідентифікаторів

Клас задається за допомогою спеціального знаку «.» перед ім'ям, використовується для створення певного набору властивостей, які будуть використовуватись багатьма схожими елементами (рис. 1.4).

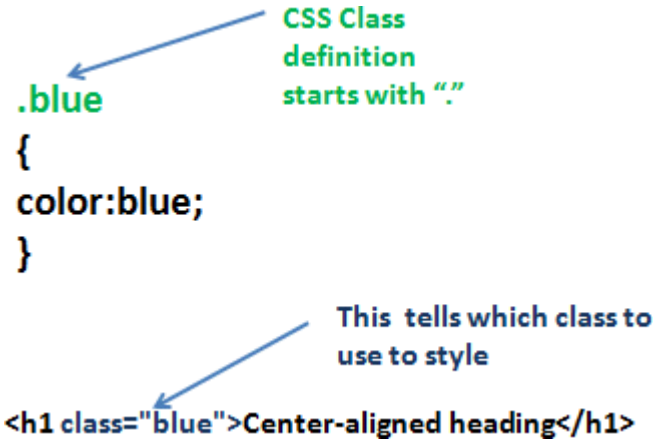


Рисунок 1.4 – Приклад використання класів

SCSS – це препроцесор, який спрощує роботу з CSS та мінімізує його недоліки. Основними особливостями препроцесору є можливість створення вкладеності селекторів, створення змінних, імітація циклів, міксінів та розширень, які допомагають використовувати велику кількість разів блоки властивостей для інших елементів, розділення документу на частини не впливаючи на швидкість роботи веб-додатку.

CSS має свій функціонал для імпорту (виконується за допомогою аналогічної функції @import). Приклад роботи імпорту відображено на рис. 1.6. Різниця цих способів в швидкодії. Оригінальний функціонал @import у CSS при кожній ін'єкції файлів створює новий http запит. На практиці для створення зручної структури файлів мова йде про десятки імпортів. @import в SCSS виправив цей недолік.

Рисунок 1.5 – Приклад роботи @import в SCSS

Дуже спрощує роботу розробника така особливість як наявність змінних (рис. 1.6). Змінні в препроцесорі SCSS можуть зберігати будь-які інформацію – від назви css властивостей до певних рядків тексту, які можуть повторюватись в коді. Часто в проєктах створюють окремий файл з змінними, які зберігають кольори, текст, значення властивостей.

Рисунок 1.6 – Приклад застосування змінних в SCSS

Для можливості використання набору властивостей з певним значенням в різних частинах коду можна застосувати міксини (рис. 1.7). Часто застосовують для задання певної властивості з врахуванням усіх вендорних префіксів. Це гарантує, що вендорні префікси не забудуться розробником і будуть гарантовано прописані в коді. Це необхідно для того, щоб забезпечити Інша ситуація в якій використовують міксини – це створення модульності стилів. За допомогою міксину створюється набір властивостей для певного блоку, який можна модифікувати за допомогою вхідних параметрів і аналогічним чином застосовується в потрібних частинах коду.

```scss
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}
.box { @include transform(rotate(30deg)); }
```

```scss
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

Рисунок 1.7 – Приклад застосування міксинів в SCSS

Схожий функціонал мають розширення (рис. 1.8). Розширення – це можливість використовувати класи знову і знову за допомогою спеціального функціоналу препроцесора. На практиці існує такий підхід для створення модифікацій: створюється клас з властивостями (властивості повинні бути ті, які повторюються в кожній з модифікацій (блочні особливості, розміри і т.д.), далі створюється клас з модифікацією та за допомогою @extend додається клас з базовими властивостями. Також є можливість створювати класи, які будуть наслідуватися та не будуть відображені в результуючому файлі CSS якщо не використовувались за допомогою знаку % перед класом.

```scss
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

.success {
  @extend %message-shared;
  border-color: green;
}

.error {
  @extend %message-shared;
  border-color: red;
}

.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

```scss
.message, .success, .error, .warning {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

Рисунок 1.8 – Приклад застосування розширень в SCSS

SCSS має можливість створювати вкладеність властивостей. Це спрощує читабельність та розуміння коду. Проте бувають ситуації, коли необхідно вивести певні властивості в корінь, зберігши при цьому розуміння до якого блоку властивостей цей код належить. Це можна виконати за допомогою @at-root (рис. 1.9). Працює даний функціонал для вкладеностей першого рівня.

```scss
.parent {
  ...
  @at-root {
    .child1 { ... }
    .child2 { ... }
  }
  .step-child { ... }
}
```

Рисунок 1.9 – Приклад застосування @at-root в SCSS

JS – це функціональна мова розробки веб-додатків, основна мова, яка використовується для створення сценаріїв та логіки для сайтів та інших

можливих видів веб-додатків. Основні задачі для яких вона використовується: маніпуляції з елементами веб-сторінки, реакція на події комп’ютерної миші, клавіатури. Особливості мови JavaScript, які ми розглянемо в цьому розділі це нетипізованість мови, типи даних, експорт та імпорт модулів, асинхронність, DOM-дерево, інтерфейсні події.

Особливість цієї мови, яка виділяє цю мову серед інших це нетипізованість змінних. Це означає, що при оголошенні змінних ми не повинні задавати тип даних. Перевага такої особливості в тому, що ми можемо використовувати одну змінну для різних даних і на виході ми не побачимо помилок. Недоліком є те, що на певних етапах та в певних функціях є необхідним робота з певним типом даних. А завдяки нетипізованості мови та її внутрішнім технологіям таким як приведення типів, результат роботи деяких функцій може бути непередбачуваним, що призведе до небажаних наслідків.

JavaScript налічує 7 типів даних:

– number, створений для роботи з будь-якими числами (цілі та з плаваючою крапкою;

– string, створений для рядків та текстових символів;

– boolean, має значення true або false;

– null, використовується для змінних, коли їх значення було задане і позначене невідоме;

– undefined використовується для змінних, коли їх значення не було задане;

– object є складним типом даних. На основі цього типу існують функції, масиви, об’єкти;

– symbol створює унікальні ідентифікатори.

Експорт та імпорт даних надає можливість розділяти код на різні файли і таким чином структурувати код та полегшувати його читабельність. За допомогою службового слова export ми даємо знати, що цей модуль буде використовуватись в інших файлах. Для того щоб додати модуль в інший документ необхідно використати інше службове слово import, вказати ім’я, за

яким цей елемент буде використовуватись в цьому файлі та вказати шлях до модуля за допомогою from ‘path’. Є випадки коли один файл експортує декілька функціональних елементів. В такому випадку для імпорту необхідно написати import { element1, element2 } from ‘path’.

Асинхронність – це можливість відкласти виконання певної частини коду на певний час або продовжити виконання наступного коду не затримуючи і не перевантажуючи систему виконанням коду.

Document Object Model або DOM-дерево – це представлення HTML документу у вигляді дерева тегів. Корневим або батьківським елементом усіх вкладених елементів є тег <html>. Таким чином, маючи доступ к об’єктам DOM-дерева ми можемо маніпулювати HTML сторінкою. За допомогою інтерфейсних подій можна додавати певний інтерактив на сторінку веб-додатку

Розробка сучасних освітніх веб платформ вимагає інтеграції різноманітних технологій, щоб забезпечити гнучкість, масштабованість, інтерактивність та адаптивність навчальних процесів. Особливу увагу слід приділити технологіям, які дозволяють створювати динамічні навчальні середовища з можливістю генерування завдань [11].

Першим кроком у розробці освітної веб платформ є аналіз вимог. Це включає вивчення цільової аудиторії, визначення освітніх цілей, та врахування специфіки предметних областей. Планування також повинно охоплювати структуру курсів, типи завдань, а також методи їх оцінювання.

Рішення про вибір технологій та платформ залежить від багатьох факторів, включаючи бюджет, доступність ресурсів, та необхідні функціональні можливості. Популярними мовами програмування для розробки освітніх веб платформ є JavaScript (frontend) та Python або Java (backend). Розглядаються також хмарні платформи для хостингу та зберігання даних.

Інтерфейс користувача має бути інтуїтивно зрозумілим та зручним для широкої аудиторії користувачів. Важливо забезпечити легкий доступ до

основних функцій, зрозумілість навігації, та ефективне представлення навчальних матеріалів.

Для навчальних середовищ, які включають генерацію завдань, розробляються спеціалізовані алгоритми. Це може включати використання шаблонів для різних типів завдань, а також алгоритми штучного інтелекту для створення унікальних та адаптивних завдань.

Ефективність навчального процесу може бути підвищена за допомогою інтеграції аналітичних інструментів. Це дозволяє вчителям та адміністраторам відстежувати прогрес студентів, оцінювати ефективність навчальних матеріалів, та вчасно вносити корективи в освітній процес.

Окрім розробки, обов’язково потрібно продумати етап тестування. Воно включає перевірку функціональності, виконання, безпеки, та сумісності на різних пристроях та браузерах та забезпечення відповідності платформи освітнім стандартам та нормативам.

Освітні веб платформи зберігають велику кількість персональних даних та інформації про навчальний процес. Тому важливо впровадити сучасні методи шифрування, захисту даних, та забезпечити дотримання норм конфіденційності та захисту інформації.

Для клієнтської частини (frontend) доцільно обрати технології HTML, CSS, JavaScript. Оскільки, як було зазначено вище – це стандартна основа для створення користувацьких інтерфейсів. Також можливо підключити фреймворки та бібліотеки JavaScript, такі як React, Angular або Vue.js, що дозволяють створювати динамічні та інтерактивні веб-сторінки для покращення якості, швидкості розробки та покращення користувацького досвіду. Також можливе використання CSS фреймворків, таких як Bootstrap, Tailwind для уніфікації стилів на проєкті, що дозволить зменшити час на розробку.

Для хостингу можливо обрати хмарні платформи AWS, Azure, Google Cloud, DigitalOcean, Render.com. Вони надають широкий спектр сервісів для

хостингу, зберігання даних, машинного навчання, та аналітики, що забезпечує масштабованість та гнучкість освітніх веб платформ.

Для забезпечення високого рівня безпеки користувачів необхідно використовувати https з'єднання з надійним SSL/TLS сертифікатом для шифрування зв'язку між браузером користувача та сервером. Окрім цього при аутентифікації користувача слід дотримуватись стандарту OAuth 2.0.

Розробка освітніх веб платформ, особливо тих, що включають генерування завдань, є складним процесом, який вимагає глибокого розуміння освітніх потреб, вибору відповідних технологій, та забезпечення високої якості та безпеки. Застосування цих методів та підходів допомагає створювати ефективні та інноваційні освітні рішення, що відповідають вимогам сучасного освітнього процесу.

1.3 Особливості впровадження комп'ютерних технологій в дистанційну систему освіти

Процес впровадження засобів нових комп'ютерних технологій в систему освіти називають інформатизацією. Інформатизація освіти є одним з пріоритетних напрямів процесу її розвитку. Це зробить можливим:

– вдосконалення механізмів управління системою освіти на основі використання банків даних науково-педагогічної інформації, інформаційно-методичних матеріалів, а також комунікаційних мереж;

– вдосконалення стратегії відбору змісту, методів і організаційних форм навчання, що відповідають завданням розвитку особистості учня в сучасних умовах інформатизації суспільства;

– створення методичних систем навчання, орієнтованих на розвиток інтелектуального потенціалу учня, на формування умінь самостійно придбавати знання, здійснювати інформаційно-учбову, експериментально-дослідницьку діяльність, різноманітні види самостійної діяльності по обробці інформації;

– створення і використання комп'ютерних тестуючих, діагностуючих, контролюючих і оцінюючих систем.

Рівень розвитку країни значною мірою визначається рівнем розвитку інформаційних технологій та послуг. Тому система освіти повинна швидко й адекватно реагувати на потреби суспільства, позбавляючись шляхом проведення кардинальних реформ притаманного теперішній освіті консерватизму [11].

Розвиток рівня цифровізації держави, створення нових інформаційних технологій та надання нових інформаційних послуг не може обмежуватись лише системою освіти. Це стосується усіх сфер життєдіяльності країни. Наприклад в сфері надання державних послуг значного покращення рівня цифровізації потребує Державна митна служба України у зв’язку з тим, що у в сфері митних послуг існує багато не пов’язаних між собою інформаційних систем, що призводить до значного погіршення якості та швидкості надання митних послуг [12].

Одним із важливих чинників реформування освіти є її інформатизація. Побудова ефективних систем інформатизації освіти з урахуванням світового досвіду, особливостей і реалій стану вітчизняної освіти – одна із актуальних і важливих наукових і практичних проблем [13].

Серед основних стратегічних цілей розвитку інформаційного суспільства в Україні, зокрема, названі:

– прискорення розробки та впровадження новітніх комп’ютерних технологій в усі сфери суспільного життя;

– забезпечення комп’ютерної та інформаційної грамотності населення шляхом створення системи освіти, орієнтованої на використання новітніх комп’ютерних технологій;

– створення загальнодержавних інформаційних систем у різних сферах науки [14].

Властивості сучасних інформаційних технологій:

– можливість передавати величезні обсяги інформації з будь-яких галузей знань;

– наявність в Інтернеті спеціальних навчальних курсів із різних дисциплін, кількість яких постійно збільшується;

– можливість доступу до інформаційних ресурсів у будь-який час і в будь- якому місці;

– можливості дистанційного навчання дозволяють кожній людині навчатися, самостійно вибираючи бажану галузь і траєкторію навчання.

Основною сутністю інформатизації освіти є використання комп'ютерних технологій у різних видах діяльності, які здійснюються в системі освіти [15]. Досліджуючи систему освіти як об'єкт інформатизації, основну увагу необхідно приділити дослідженню цих видів діяльності, визначити критерії їх класифікації, виходячи з психолого-педагогічних та інформаційних характеристик і класифікувати їх за визначеними критеріями.

Впровадження сучасних інформаційно-комунікаційних технологій у заклади освіти регулюється наступними документами:

– Законом України "Про освіту".

– Законом України "Про загальну середню освіту".

– Указом Президента України "Про невідкладні заходи щодо забезпечення функціонування та розвитку в Україні".

Необхідно підкреслити, що багато проблем, пов'язаних з впровадженням у навчально-виховний процес загальноосвітніх навчальних закладів інформаційно-комунікаційних технологій, залишаються нерозв'язаними: слабка матеріально-технічна база; не ефективно використовуються вже наявні електронні інформаційні ресурси, існує проблема технічного обслуговування і ремонту сучасних інформаційно-комунікаційних засобів у навчальних закладах; використовується морально застаріла комп'ютерна техніка; незадовільний стан підключення до Інтернету; недостатній рівень підготовки вчителів загальноосвітніх навчальних закладів

з питань використання інформаційно-комунікаційних технологій в навчально-виховному процесі тощо.

Академік В. Ю. Биков у своєму дослідженні, присвяченому сучасним проблемам відкритої освіти підкреслював, що «сучасні завдання системи освіти передбачають розвиток змісту освіти та педагогічних технологій, що застосовуються в навчально-виховному процесі» і одним із основних чинників, що мають сприяти розв'язанню цих завдань, називає інформатизацію освіти, «що відповідає цілям і завданням формування інформаційного суспільства і, в даному контексті, передбачає створення єдиного інформаційного освітнього простору – змістово-предметної, комп'ютерно- технологічної та інформаційно-комунікаційної платформи інтеграції і демократизації освіти» [7].

Сучасні комп'ютерні технології дозволяють вивчати будь-які предмети за допомогою спеціального програмного забезпечення (електронного посібника, веб-курсу тощо). На сьогоднішній день велика увага приділяється комп'ютерному супроводу навчання в сучасних освітніх установах. Впроваджуються навчальні і тестуючі системи з різних дисциплін. Застосування мультимедійних засобів на уроках дозволяє підвищити інтерес до навчального процесу, прокращити засвоєння матеріалу, здійснювати самоконтроль. Комп'ютерні засоби навчання дають можливість застосовувати в них сучасні способи подання інформації, включати інтерактивні засоби контролю знань. Крім того, в умовах недостатньої забезпеченності підручниками, програму легко розмістити у мережі Інтернет чи на сервері школи та користуватися нею в домашніх умовах.

На сьогоднішній день існує два шляхи використання ПК – створення комп'ютерних навчальних лабораторій та використання можливостей мережі Інтернет. Щодо першого напрямку, то в навчальних закладах усього світу активно вирішується проблема створення комп'ютерних класів з можливістю дистанційного управління учнівськими станціями та вільної передачі й отримання аудіо- й відеоінформації. Така мультимедійна лабораторія має бути

оснащена персональними комп’ютерами та спеціальним апаратним або програмно-апаратним забезпеченням [16].

Використання для навчання персонального комп’ютера з мережею Інтернет дозволяє учневі навчатися у зручному середовищі та в оптимальному темпі, дає можливість інтерактивного спілкування з викладачем та поточного контролю, допомагає досягти кращих результатів в опануванні дисципліною. Звичайно, такий спосіб організації навчального процесу вимагає від викладача інноваційних методичних підходів до викладання і можливий за наявності відповідного програмного забезпечення [16].

Застосування сучасних технологій для розвитку системи освіти дасть наступні можливості:

– використання педагогами банків даних науково-педагогічної інформації, інформаційно-методичних матеріалів, доступних в мережі Інтернет, а також суспільних мереж для успішнішої організації процесу навчання;

– відбір змісту, методів і організаційних форм навчання, що відповідають сучасним вимогам;

– створення методичних систем навчання, орієнтованих на розвиток інтелектуального потенціалу учня, на формування умінь самостійно придбавати знання, здійснювати інформаційно-учбову, експериментально-дослідницьку діяльність, різноманітні види самостійної діяльності по обробці інформації;

– створення і використання комп'ютерних тестуючих, діагностуючих, контролюючих і оцінюючих систем.

В Україні дуже динамічно розвивається система дистанційної освіти. Така практика вже давно функціонує у західних країнах та користується великою популярністю [13]. Розвитку дистанційної освіти сприяє оснащення закладів потужною комп’ютерною технікою, а також перейняття досвіду колег з інших країн за допомогою спілкування у мережі Інтернет. Оснащеність сучасних шкіл комп'ютерною технікою, наявність власних локальних мереж,

доступ до мережі Інтернет дозволяє перейти від традиційних методів оцінки отриманих знань до нових інформаційних технологій.

Дистанційне навчання – це форма навчання з використанням комп'ютерних і телекомунікаційних технологій, які забезпечують інтерактивну взаємодію викладачів та студентів на різних етапах навчання і самостійну роботу з матеріалами [13].

Досліджуючи погляди науковців, можна визначити, що дистанційне навчання – це нова, специфічна форма навчання, дещо відмінна від звичних форм очного або заочного навчання. Вона передбачає інші засоби, методи, організаційні форми навчання, іншу форму взаємодії викладача і учня, учнів між собою. Дистанційна форма навчання обумовлена специфікою використовуваної технологічної основи (наприклад, тільки комп'ютерних засобів, комп'ютерних засобів у комплексі з друкованими засобами, компакт-дисками, ін.) [3, 6, 17, 18].

Не слід ототожнювати заочне та дистанційне навчання. Їх головна відмінність у тому, що при дистанційному навчанні забезпечується систематична і ефективна інтерактивність. Слід розглядати дистанційне навчання як нову форму навчання і, відповідно, дистанційну освіту як нову форму освіти [13].

Дистанційна форма навчання передбачає можливість подання лекційного матеріалу в придатній формі для відображення в інтернеті з використанням комп'ютерної графіки, анімації тощо, що покращує сприйняття матеріалу і надає студентам можливість набувати знання в комфортних умовах та у зручному для себе темпі, можливість обговорення лекційного матеріалу шляхом організації аудіо й відео конференцій за умови віддаленості викладача та студентів [4].

Дистанційна освіта має такі переваги [4]:

– актуальність – впровадженя новітніх інформаційних, технологічних та педогогічних розробок;

– зручність – можливість доступу до курсу у зручний час з будь-якого місця, відсутність часових обмежень;

– інтерактивність – активна взаємодія вчителя з учнем;

– застосування нових засобів контролю;

– гнучкість – урахування особливостей підготовки та рівня учнів;

– модульність – можливість засвоєння завершених тем окремим учнем чи групою.

В дистанційній формі навчання застосовують різні форми: електронні підручники та статті, тестування, опитування, відправка виконаних завдань викладачеві на пошту чи в спеціальну мережу. Розвиток інформаційних технологій дає широку можливість для винаходу нових методів та методик в освіті і тим самим підвищує її якість.

На даний час широко використовуються такі програмні продукти, як Moodle, Human School, Google Classroom, Edmodo, Coogle Meet, Zoom, MS Word Pad, MS Excel, Statistika, MathCAD, MATLAB,MS Power Point, AutoCAD, ArchiCAD, Speaking Mouse, Education Games,Virtual Worlds [18].

Сучасні освітні веб платформи та навчальні середовища стають все більш популярними та важливими в дистанційні формі навчання. Вони пропонують різноманітні можливості для навчання, але, як і будь-які інноваційні технології, мають свої переваги та недоліки.

Серед основних переваг освітніх веб платформ слід відзначити:

– однією з основних переваг освітніх веб платформ є їх доступність. Студенти можуть навчатися в будь-який час і з будь-якого місця, що робить освіту більш інклюзивною та доступною для широкого кола людей;

– багато платформ використовують алгоритми, що адаптуються до рівня знань та швидкості навчання кожного учня, що дозволяє створити персоналізований навчальний план. Широкий спектр навчальних матеріалів, від лекцій до інтерактивних завдань, дозволяє учням вибрати найбільш підходящий для них спосіб навчання;

– сучасні освітні веб платформи часто включають інтерактивні елементи, такі як відео, ігри, симуляції, що робить навчання більш захоплюючим та ефективним.

– навчальні середовища, що пропонують генерацію завдань, дозволяють створювати унікальні та різноманітні задачі, що підтримують інтерес та виклик у процесі навчання. Також, такі навчальні середовище дозволяють студентам отримувати миттєвий зворотній зв’язок, що сприяє швидшому розвитку та вдосконаленню навичок.

– більшість платформ надають можливість відстежувати навчальний прогрес, що допомагає студентам та вчителям у плануванні та коригуванні процесу навчання.

Серед основних недоліків освітніх веб платформ слід відзначити:

– одним з головних недоліків дистанційного навчання є відсутність безпосереднього спілкування між студентами та вчителями, що може вплинути на якість освіти та мотивацію;

– для ефективного використання освітніх веб платформ потрібне надійне інтернет-з'єднання та сучасні гаджети, що може бути проблемою в менш розвинених регіонах;

– велика кількість доступних ресурсів може викликати відчуття перенавантаження та ускладнити процес вибору найбільш ефективних матеріалів для навчання;

– дистанційне навчання вимагає високого рівня самодисципліни та організованості, що може бути викликом для деяких студентів;

– деякі навички та дисципліни вимагають практичних занять, які важко або неможливо відтворити в онлайн-форматі.

Однією з проблем в сучасній дистанційній формі навчання є відсутність універсальних веб-застосунків для генерації завдань. Багато навчальних закладів в світі та в Україні вже використовують Google Classroom для організації дистанційної освіти [18]. Google Classroom дозволяє викладачам та студентам спілкуватися, ділитися матеріалами та завданнями. Однак, в силу

своєї спрощеної природи, він також має обмеження у функціональності щодо генерації та оцінювання завдань.

Наприклад, використання Google Forms має свої недоліки, такі як відсутність можливості додавати вкладення до завдань формули, таблиці тощо. Для створення різних варіантів завдань для всіх студентів потрібно окремо створювати кожний варіант, що забирає дуже багато часу. Також, якщо необхідно створити тест, в якому на одному екрані буде одне питання, а не всі одразу – це теж додаткові витрати часу для виклачада.

Для подолання цих обмежень потрібно розробити або використовувати більш розширені веб-застосунки для генерації завдань, які дозволяють додавати вкладення, налаштовувати різні варіанти для студентів та забезпечують зручний інтерфейс для викладачів та учнів. Важливо продовжувати дослідження та впровадження сучасних технологій в дистанційну освіту з метою поліпшення якості та ефективності навчального процесу.

Можна виділити наступні конкретні шляхи покращення освітніх веб платформ за рахунок покращення навчальних середовищ для генерування завдань:

1. Адаптація завдань до індивідуального рівня знань та темпу навчання учня дозволить забезпечити більш ефективне засвоєння матеріалу та збільшить вірогідність успішного завершення курсу.

2. Збір та аналіз даних про успішність та переваги учнів може допомогти у формуванні більш персоналізованих завдань. Це може включати в себе використання штучного інтелекту для аналізу відповідей та адаптації завдань під конкретного учня.

3. Інтеграція мультимедійних та інтерактивних елементів може зробити процес навчання більш цікавим. Це може бути відео, аудіо, інтерактивні діаграми та ігрові елементи, які допомагають утримувати увагу студентів та покращувати засвоєння матеріалу.

4. Розвиток гнучких методів оцінювання, які враховують індивідуальні особливості навчання та прогрес учнів, підвищить мотивацію студентів та покращить об'єктивність оцінювання.

5. Створення відкритого (open source) навчального середовища, де викладачі можуть самостійно генерувати завдання за обраним предметом, відкриє значні можливості для розвитку дистанційного навчання:

– залучення великої спільноти розробників, викладачів та студентів для спільної роботи над платформою;

– залучення студентів до спільноти, де вони можуть ділитися знаннями, вирішувати задачі в групах, або отримувати допомогу від однолітків та вчителів, що підвищить їх мотивацію та залученість у навчанні;

– можливість додавання нових функцій або типів завдань, розроблених спільнотою;

– можливість створення необмеженої кількості курсів для самостійного навчання будь-якої спеціальності. Створення рейтингової системи курсів дозволить виокремити найкращі серед них та зрозуміти, що хочуть бачити студенти;

– можливість налаштувати навчальне середовище під сучасні потреби на основі пропозицій спільноти;

1.4 Постановка завдання дослідження

Метою роботи є дослідження методів та технологій створення прототипу навчального веб середовища для генерування завдань, яке покликано вирішити існуючі технічні обмеження освітніх платформ. Це веб середовище буде забезпечувати взаємодію між студентом та викладачем шляхом реалізації інноваційних механізмів навчального процесу, таких як генерування випадкових варіантів тестових завдань для кожного студенту, їх перевірка та оцінювання, можливість додати вкладення не лишу до тесту в цілому, а й до кожного завдання, що дозволить максимально швидко додавати

будь-які формули та таблиці в зручному вигляді для студентів та викладачів. Веб середовище дозволить викладачам створювати тестові завдання різних типів та з будь-якою конфігурацією, а також редагувати їх.

Для реалізації вищезазначеного навчального веб середовища сформульовані наступні задачі:

– провести аналіз наявних інструментів веб-додатків;

– розробити архітектуру проєкту та схеми функціональних можливостей користувачів з різними ролями в застосунку;

– визначити необхідні сторінки веб-додатку на основі архітектури та схеми можливостей користувачів;

– створити дизайн для кожної сторінки;

– створити логіку функціонування веб-додатку та викласти в мережі Інтернет;

1.5 Висновки до першого розділу

Отже, завдяки дослідженню наявних проблем у сучасній дистанційні освіті в Україні, зокрема у наявних освітніх платформах та навчальних середовищах та огляду наукової літератури було виявлено, що в начальних середовищах, які є невід’ємними компонентами освітніх платформ, які в свою чергу є невід’ємними компонентами системи дистанційної освіти, існує ряд невирішених проблем, як технічних так і нетехнічних. Серед технічних було виділено:

– обмеженість функціоналу освітніх платформ, що присутні в системі дистанційної освіти в Україні. На сьогодні в межах цих платформ не можливо створити тестові завдання, що можуть включати формули, таблиці та вкладення до кожного завдання та неможливо генерувати випадкові варіанти для студентів, що зобов’язує викладачів для кожного окремого варіанту створювати окремі тестові завдання, що веде до значних витрат у часі;

– відсутність універсальних навчальних веб середовищ поза межами освітніх платформ, які б дозволили будувати тестові завдання з урахуванням вищезазначених наявних проблем в освітніх платформах. В Україні вже існують просунуті веб середовища, які мають дійсно вражаючий набір функцій, проте на сьогодні вони існують лише в межах комерційних освітніх платформ та прив’язані до контексту курсів, що вже існують в межах цих платформ та в них немає можливості створення тестових завдань користувачами.

Метою дослідження є розробка створення навчального веб середовища для генерування завдань. Для досягнення мети дослідження необхідно здійснити наступні кроки:

– огляд існуючих розв’язків поставленої задачі;

– аналіз та вибір засобів реалізації навчального веб середовища;

– практична реалізація прототипу навчального веб середовища для генерування завдань.

Для практичної реалізації прототипу навчального веб середовища для генерації завдань були сформульовані наступні задачі:

– провести аналіз наявних інструментів веб-додатків;

– розробити архітектуру проєкту та схеми функціональних можливостей користувачів з різними ролями в застосунку;

– визначити необхідні сторінки веб-додатку на основі архітектури та схеми можливостей користувачів;

– створити дизайн для кожної сторінки;

– створити логіку функціонування веб-додатку та викласти в мережі Інтернет;

РОЗДІЛ 2 АНАЛІЗ ЗАСОБІВ РЕАЛІЗАЦІЇ НАВЧАЛЬНОГО ВЕБ СЕРЕДОВИЩА

2.1 Огляд існуючих розв’язків поставленої вище задачі

Серед основних вже реалізованих веб-додатків, що можуть виконувати функції навчального веб середовища та генерувати завдання відповідно до поставлених умов, слід виділити Schoology та Eliademy.

Schoology – це послуга соціальних мереж та віртуальне середовище навчання для повних шкільних та вищих навчальних закладів, яка дозволяє користувачам створювати, керувати та ділитися навчальним вмістом. Також відома як система управління навчанням (LMS) або система управління курсами (CMS), хмарна платформа надає інструменти, необхідні для управління онлайн-аудиторією. Schoology може допомогти викладачам зв’язуватися з студентами з домашніми завданнями тощо. Вони можуть розміщувати щоденні нагадування або оновлення. Вони можуть надсилати повідомлення студентам, керувати календарем завдань і ставити нові завдання.

Послуга включає в себе записи про відвідування, онлайн-зошит, тести та вікторини та папки для домашніх завдань. Функції соціальних медіа полегшують співпрацю між класом чи групою. Система може бути інтегрована до існуючих навчальних систем звітності та інформації, а також забезпечує безпеку, фільтри та підтримку, необхідних округам.

Основний продукт можуть використовувати приватні особи, вчителі, викладачі та інші безкоштовно. Додаткову плата відбувається за рахунок преміальних додатків, таких як індивідуальні брендинги, пакети підтримки, збільшене сховище, єдиний вхід та інтеграція даних із існуючими інформаційними системами для студентів (SIS).

Платформа Schoology була розроблена Джеремі Рейдом, Райаном Хвангом і Тімом Тринідадом, ще будучи студентами університету Вашингтона в Сент-Луїсі, штат Міссурі. Спочатку призначений для обміну

нотами, Schoology вийшла на ринок у серпні 2009 року. Schoology забезпечила свій перший інституційний раунд фінансування венчурного капіталу в розмірі 1,25 млн. Доларів від венчурного капіталу Meakem Becker Venture Capital після чергової інвестиції ангела у 2009 році від неназваного інвестора. Станом на 2010 рік в службі було понад 2400 шкіл по всій країні, і планували розробити інтерактивний контент, який викладачі можуть використовувати для підтримки навчальних матеріалів та надання більшого доступу до батьків. До розширень включено сповіщення про текстові повідомлення, мобільні додатки для iOS та Android, інтеграція з Google Drive та іншими службами, бібліотека спільних ресурсів та імпортер запитань для тестів та вікторин. Станом на 2023 рік повідомлялося, що послуга використовується у понад 60 000 школах.

Eliademy є безкоштовною онлайн-аудиторією, яка дозволяє викладачам та студентам створювати, обмінюватися та керувати онлайн-курсами за допомогою дискусій у реальному часі та управління завданнями. Eliademy базується на Moodle (підтримка імпорту у форматі Moodle), Twitter Bootstrap та інших технологіях з відкритим кодом. Eliademy була оприлюднена в лютому 2013 року CBTec. Eliademy доступна 32 мовами. Клієнт Android Eliademy (також сумісний з Moodle LMS) – це проєкт з відкритим кодом, доступний у GitHub. Він був запущений за підтримки Tekes – Фінляндського фінансового агентства з технологій та інновацій. Eliademy є частиною ініціативи Digile, Фінляндії, створеної з метою створення нових руйнівних цифрових послуг у сфері освіти. Eliademy базується на віртуальному навчальному середовищі Moodle з відкритим кодом. Сайт локалізований більш ніж на 19 мовах (включаючи латинську), розроблений для мобільного використання.

Існуючі розв'язки, такі як Schoology і Eliademy, демонструють ефективність віртуальних навчальних середовищ і платформ для організації процесу навчання та обміну інформацією між викладачами та студентами. Проте, існує обгрунтована необхідність створення нового та власного

навчального веб середовища для генерування завдань. Нижче розглянемо основні аспекти такої необхідності:

– Існуючі розв'язки не повністю відповідають специфічним потребам користувача або навчального заклад. Створення власного навчального веб середовища дозволить налаштовувати функціональність та інтерфейс відповідно до конкретних вимог.

– Якщо навчальні завдання мають специфічні вимоги, які не враховані в існуючих платформах, створення власного середовища може дати можливість генерувати завдання точно відповідно до заданих критеріїв.

– Якщо в навчальній установі вже використовуються певні інформаційні системи, створення власного середовища дозволить легше інтегрувати його з існуючими системами, забезпечуючи зручну і неперервну роботу.

– Деякі навчальні заклади можуть мати специфічні вимоги, такі як конкретні методи оцінювання, типи завдань, чи особливості звітності. Створення власної платформи дозволяє легше враховувати такі унікальні потреби.

– Платформа, розроблена внутрішньо, може бути легше підтримувана і розвиватися відповідно до змін у вимогах навчального процесу. Внутрішній розвиток також може забезпечити більш ефективну та швидку підтримку користувачів.

– Для деяких навчальних установ важливо мати повний контроль над збереженням та захистом навчальних матеріалів і персональних даних студентів. Створення внутрішньої платформи дозволяє більше можливостей для управління безпекою і конфіденційністю.

Отже, створення власного навчального веб середовища є обгрунтованим стратегічним рішенням для задоволення усіх унікальних потреб та вимог навчального процесу, особливо коли існуючі розв'язки не можуть повністю врахувати ці параметри.

2.2 Вибір основних інструментів розробки

Для розробки клієнтської частини навчального веб середовища для генерування завдань будуть використані наступні інструменти:

– HTML5. Остання версія найпопулярнішої мови розмітки [19, 20]

– CSS3 (SASS). Остання версія каскадних таблиць стилів CSS. Для реалізації даного проєкту було прийнято рішення не використовувати додаткові бібліотеки, зокрема Bootstrap, для уникнення додавання зайвих залежностей в проєкт, що можуть сповільнити роботу застосунку [19, 21, 22].

– JavaScript. Це одна з мов програмування, яка дозволяє створити динамічно оновлюваний контент [19, 20, 23].

– TypeScript. Це мова програмування, що розширює JavaScript, додаючи строгу статичну типізацію та інші функціональності для поліпшення роботи з великими кодовими базами. TypeScript допомагає виявляти та усувати помилки на етапі розробки, забезпечуючи більшу надійність та зручність у процесі програмування [24, 25].

– Vue.js – це прогресивний фреймворк для створення інтерфейсів користувачів. На відміну від інших монолітних фреймворків, Vue.js покладає свій акцент на рівень уявлення (view) та легко інтегрується з іншими бібліотеками та проєктами. Його унікальність полягає в тому, що він спрощує розробку інтерфейсів користувачів, а також відмінно підтримує створення складних односторінкових додатків (SPA) в поєднанні з сучасними інструментами та бібліотеками.

2.3 Вибір фреймворку та додаткових інструментів

Для розробки клієнтської частини було обрано фреймворк Vue.js. Цей фреймворк, розроблений з використанням JavaScript із відкритим вихідним кодом, легко інтегрується в різноманітні проєкти, використовуючи інші

JavaScript-бібліотеки. Vue.js може функціонувати як ефективний веб фреймворк для створення односторінкових програм у реактивному стилі.

Vue.js, створений Еваном Ю і ще двома стами тридцяти чотирма ентузіастами, набрав більше ста двадцяти однієї тисячі зірок на GitHub. Він включає доступну кореневу бібліотеку, яка в першу чергу вирішує завдання рівня уявлення, і екосистему додаткових бібліотек, що дозволяє створювати складні і об'ємні односторінкові додатки (Single-Page Applications) [26].

Vue.js відзначається своєю легкістю в освоєнні й для його використання достатньо базових знань JavaScript і HTML. Він має одну з найкращих офіційних документацій, в тому числі українською мовою, що ще більше полегшує його вивчення [27].

Vue.js можливо успішно використовувати з Typescript. Фреймворк втілює шаблон MVVM та надає можливість прив'язки даних на рівні JavaScript, що дозволяє безпосередньо пов'язувати виведення та введення даних з джерелом даних. Це дозволяє уникнути ручного визначення даних в HTML-DOM, і враховується автоматично. Важливою особливістю Vue.js є можливість робити реактивні елементи звичайними JavaScript-змінними без необхідності в додаткових анотаціях, як, наприклад, в Knockout.js. Під час розробки Vue.js, команда Евана Ю брала кращі практики з React та Angular. Саме з React було запозичено ідею віртуального DOM. Цей підхід виключає пряму взаємодію з вузлами інтерфейсу. Початкова робота ведеться з його virtual DOM. І тільки після внесення змін та оновлення virtual DOM йде порівняння virtual DOM з реальним DOM і перемалювання лише тих вузлів реального DOM, що були змінені [28, 29, 30, 31].

У Vue.js доступний ряд модулів, що реалізують сучасний підхід до розробки веб застосунків. Зокрема слід виділити бібліотеку для контролю загального стану додатку – Vuex. Вона повністю запозичує ідеї з Redux, що був розроблений спеціально для React, але ступінь інтеграції цієї бібліотеки з Vue набагато вище, ніж у випадку з React і Redux [32].

Головна альтернатива Vuex – це бібліотека керування станом веб застосунку Pinia. Обидві бібліотеки, Pinia та Vuex, призначені для керування станом в Vue.js додатках, але вони мають свої відмінності, які можуть вплинути на вибір для конкретного проєкту:

– Pinia орієнтована на TypeScript, що робить її привабливою для проєктів, які активно використовують TypeScript, в той час коли Vuex також підтримує TypeScript, але це не є її основним фокусом.

– Pinia славиться своєю простотою використання та декларативним синтаксисом та робить акцент на продуктивності та оптимізаціях для великих та складних додатків, а Vuex може вимагати додаткових оптимізацій для великих проєктів.

– Pinia має більш компактний розмір в порівнянні з Vuex, що може бути важливим для проєктів із підвищеним вимогами до об'єму завантаження сторінки.

Враховуючи вищезазначені порівняння, в проєкті буде використовуватись саме Pinia.

Однією з найновітніших розробок від Евана Ю є інструмент збірки веб-додатків Vite, що призначений для прискорення швидкості розгортання додатків та покращення досвіду розробки клієнтської частини веб-додатку, використовуючи Vue.js, React або навіть чистий JavaScript без використання фреймворків.

Головною альтернативою Vite наразі є Vue CLI, що по суті є додатковою обгорткою для стандартного інструмента збірки веб-додатків Webpack. Vue CLI широко використовується в індустрії з великою активною спільнотою, має велику екосистему плагінів для розширення функціоналу, а також використовує потужність Webpack для забезпечення широких можливостей оптимізації та управління модулями. В свою чергу, Vite забезпечує нативну сумісність з Vue, використовує ESBuild для швидкого збирання та забезпечує інноваційний підхід до розробки з гарячою заміною коду та використанням dev-сервера, що робить процес розробки швидшим та ефективнішим. Саме

тому в дипломному проєкті буде використовуватися інструмент збірки веб-додатків Vite.

Один з важливих етапів при плануванні реалізації веб застосунку на Vue.js – це вибір API, який буде використовуватися для взаємодії між різними компонентами фреймворку Vue.js. На сьогодні основними API у Vue.js є:

- Composition API;
- Options API;
- Class API;

Options API, Class API та в Vue.js. Порівняння основних переваг та сценаріїв використання приведено в таблиці 2.1.

Таблиця 2.1

Порівняльна характеристика Options API, Class API та Composition API

Характеристики	Основні переваги	Сценарії використання
Options API	1. Простота та прозорість у використанні. 2. Зручний для менших та менш складних компонентів. 3. Легко зрозуміти для розробників, звиклих до цього підходу.	1. Підходить для невеликих та менш складних компонентів. 2. Зручний для розробників, які вже використовують цей підхід та звикли до його синтаксису.
Class API	1. Використання об'єктно-орієнтованого підходу з класами. 2. Дозволяє створювати компоненти як екземпляри класів.	1. Зручний для розробників, які звикли до об'єктно-орієнтованого програмування. 2. Є альтернативою для використання класів у Vue-компонентах.
Composition API	1. Модульність та легше виокремлення функціональності. 2. Зручне використання функціональних блоків (Composition Functions). 3. Дозволяє зменшити дублювання логіки та полегшити її повторне використання.	1. Підходить для складних та великих компонентів. 2. Забезпечує ефективну організацію логіки та стану компонентів.

Вибір між Options API, Class API та Composition API визначається в більшості випадків наявністю складною логіки в компонентах. І саме Composition API найбільше підходить для таких ситуацій і дозволяє значно краще впорядковувати код та створювати складну логіку компонентів, де необхідна висока модульність. Options API залишається зручним варіантом для більш простої логіки, а також для тих хто лише перейшов з Vue.js 2.6 на версію Vue.js 3 і ще не до кінця розуміє нюанси написання коду з Composition API. Необхідно зазначити, що Composition Api працює лише з новою версією Vue 3. Class API може використовуватися для тих, хто віддає перевагу об'єктно-орієнтованому підходу. Враховуючи найновітніші тенденції, а також для покращення структури коду, в дипломному проєкті буде використовуватися саме Composition API.

2.4 Вибір середовища розробки, сервісу збереження коду, хостингу

Visual Studio Code – редактор вихідного коду, розроблений Microsoft для Windows, Linux і macOS. Позиціонується як «легкий» редактор коду для кросплатформеної розробки веб і хмарних застосунків.

VS Code дозволяє розробляти як консольні застосунки, так і застосунки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб сайти, веб застосунки, веб служби як в рідному, так і в керованому кодах для всіх платформ.

У редакторі присутні вбудований відладчик, інструменти для роботи з Git і засоби рефакторинга, навігації по коду, автодоповнення типових конструкцій і контекстної підказки.

Продукт підтримує розробку для платформ ASP.NET і Node.js, і вважається легким рішення, яке дозволяє обійтися без повного інтегрованого середовища розробки. Великим плюсом редактора є підтримка великої кількості мов, таких як C ++, C #, Python, PHP, JavaScript та інших.

Перевагами даного продукту є:

– розширювана бібліотека доповнень і готових рішень;

– здатність працювати з великою кількістю мов програмування за рахунок бібліотеки доповнень;

– простота і гнучкість.

Саме тому для проєкту було обрано саме середовище розробки VS Code.

Подальшим кроком є визначення середовища розміщення коду, яке має бути максимально захищеним від стороннього втручання в код і при цьому бути зрозумілим та легкодоступним для розробника.

Кожен проєкт складається з безлічі файлів, до яких постійно вносяться зміни. Система контролю версій відстежує ці зміни. Контроль версій – це механізм, який вносить зміни в один файл або набір файлів протягом певного часу, щоб можна було повернутися до певної попередньої версії. Для управління версіями коду зазвичай використовується система контролю версій Git. Для цієї системи існують 3 основних середовища розміщення коду:

– GitHub;

– Gitlab;

– Bitbucket;

GitHub є одним з провідних сервісів для спільної розробки програмного забезпечення, використовуючи Git як основу для системи керування версіями. Вона пропонує як безкоштовні, так і преміальні план абонемента, доступні для її користувачів [33].

GitLab - це сервіс для управління життєвим циклом програмного забезпечення, який також базується на системі контролю версій Git. Він об'єднує у собі функціонал керування кодом, CI/CD (Continuous Integration/Continuous Deployment), моніторинг та інші можливості для спрощення процесу розробки. GitLab пропонує як відкритий вихідний код (open-source), так і приватні репозиторії, підтримуючи співпрацю команд на різних етапах розробки.

Bitbucket, від Atlassian, є ще одним популярним сервісом для розміщення коду, зосередженим на професійних командах та корпоративних клієнтах. Він підтримує як Git, так і Mercurial як системи контролю версій та інтегрується з різними інструментами Atlassian, такими як JIRA і Trello. Bitbucket відрізняється своєю зосередженістю на корпоративних потребах, пропонуючи функції для управління командою, звітності та планування проєктів. Однією з ключових особливостей Bitbucket, що вигідно відрізняє його від інших подібних сервісів – це потужна інтеграція з Jira або Trello, яка дозволяє одразу віднести певні зміни в гілці коду до певної задачі в Jira або Trello та відобразити там посилання на Bitbucket, де буде показано, які саме зміни внесені в межах цієї задачі.

Для даного проєкту було обрано GitHub завдяки наявності в цьому сервісі підтримки контролю версій Git та відсутності витрат для збереження коду.

Подальшим кроком є визначення фізичного місця розташування сайту. Фізичне місце розташування сайту називається хостингом. Для цього використовуються сервери – це потужні комп'ютери з спеціалізованим програмним забезпеченням. Вони перебувають під постійним технічним контролем і працюють безперервно, забезпечуючи неперервний доступ до веб ресурсу. Для забезпечення постійної роботи сайту необхідно розмістити його на одному з веб хостингів [34].

Вибір відповідного хостинг-провайдера є стратегічним рішенням для успішності будь-якого веб-додатку. Задля зручності та оптимізації вибору, було вирішено провести порівняльний аналіз п’яти провідних хостинг-провайдерів, серед яких 1 хостинг-провайдер створений в Україні:

- AWS (Amazon Web Services);
- Google Cloud;
- DigitalOcean;
- Render.com;
- Hostiq.

AWS (Amazon Web Services), незаперечний лідер у світі хмарних послуг, пропонує вражаючий рівень функціоналу. Однак його складність та високі тарифи можуть виявитися викликом для новачків та менших проєктів.

Google Cloud, зі своїм широким набором послуг, пропонує конкурентоспроможні ціни та можливості. Він може виявитися оптимальним варіантом для тих, хто шукає ефективний хмарний хостинг.

DigitalOcean здобув славу своєю простотою та оптимальними цінами. Це відмінний вибір для невеликих та середніх проєктів, де важливість легкості використання та доступність.

Render.com – це єдина хмара для створення та запуску програм і веб-сайтів із безкоштовним SSL, глобальним CDN, приватними мережами та автоматичним розгортанням від Git. Розгорнути автоматично проєкт можливо як з GitHub, так і з GitLab. Також можливо вказати для розміщення сайту власний домен.

Hostiq, завдяки якісній підтримці, конкурентоспроможним тарифам та можливості використання хмарного VPS, є беззаперечним варіантом для тих, хто шукає надійний та гнучкий хостинг українського походження.

Таким чином, для реалізації прототипу навчального веб середовища для генерування завдань було прийнято рішення розмістити веб середовище на сервері ресурсу Render.com.

2.5 Вибір інструменту для розробки дизайну

Одним з важливих пунктів для створення будь-якого веб застосунку – є дизайн оболонки майбутнього ресурсу, причому від вибору способу створення буде залежати як успішна співпраця з іншими дизайнерами над проєктом, так і переваги якими в кінцевому підсумку буде володіти веб застосунок.

Всього існує два напрямки для створення графічного оформлення: онлайн конструктори та спеціальне програмне забезпечення для настільних комп’ютерів. Найбільш часто використовуваними вважаються конструктори,

оскільки вони не вимагають особливих знань і умінь у сфері веб дизайну, будь-який бажаючий може створити собі сторінку в лічені години за допомогою цих ресурсів, так як в них закладена велика кількість готових шаблонів і колірних варіацій. В той ж час, конструктори сайтів – це повноцінні онлайн системи, які дозволяють створювати цілі ресурси для заробітку і просування свого бізнесу. Проєкти подібного плану, знамениті своїм високим рівнем безпеки у вигляді захисту від вірусів, різного роду атак на сервіс, і навіть наявністю функції фільтрації від спаму.

Якщо ж мова заходить про унікальну розробку дизайну, то в цьому випадку замовники звертаються безпосередньо до веб дизайнерів, які за допомогою настільного програмного забезпечення створюють проєкти, додаючи в них досить багато ефектів, яких немає в тих же самих онлайн-конструкторах. При виборі інструмента для створення якісного дизайну, важливим є швидкість виконання і «просунутість» самої програми. Також варто пам'ятати про популярність додатка в професійному середовищі. Адже від вибору більш поширеного ПО буде простіше працювати з іншими дизайнерами і верстальниками готового макету. При використанні однакового ПО інші партнери по проєкту без проблем зможуть відкрити файл у себе і внести необхідні правки [35].

На 2024 рік найпопулярнішими і в той же час функціональними є програми Figma (може також працювати у режимі онлайн конструктора), Sketch, та низка додатків з пакету Adobe Creative Cloud – які інтуїтивно зрозумілі і навіть чимось схожі між собою. Інтерфейс зазначених програм містить багато в чому подібну навігацію та інструменти, тому якщо дизайнер мав справу хоч з одним додатком від фірми Adobe Systems, то розібратися в інших йому не складе особливих труднощів. У список цих програм входять: Adobe Photoshop; Adobe Illustrator; Adobe XD [36].

Adobe Photoshop можна сміливо назвати професійною і найпопулярнішою програмою серед дизайнерів. Загальні переваги Adobe Photoshop: інтуїтивно зрозумілий інтерфейс, великий вибір інструментів для

роботи, малювання різних фігур і обрисів; можливість роботи з 3D-графікою; велика популярність програми серед професійних дизайнерів і відповідно можливість легко обмінюватися файлами. Програма спочатку була створена для ретуші і обробки фотографій, тож дозволяє підвищити якість використовуваних зображень і графіки [35]. Варто зазначити, що Photoshop коректно працює на більш продуктивних складових, і на відміну від тих же Sketch, та Figma створює занадто великі файли в порівнянні з перерахованими вище інструментами.

Adobe Illustrator – є одним з кращих рішень для роботи з векторною графікою. За допомогою цієї програми створюють різного роду ілюстрації, іконки, логотипи, рекламні листівки, банери, і макети веб сайтів. Вона має багато цікавих функцій, серед яких: Touch Type, Free Transform і Puppet Warp [36]. На відміну від растрової графіки, векторне зображення можна збільшувати або зменшувати без втрати якості. Тому в Adobe Illustrator можна легко створити графічну оболонку сайту, яка буде зберігати при використанні високу якість зображення в будь-якому розмірі.

Adobe XD – це комп'ютерна програма з пакету Adobe Creative Cloud, створена спеціально для веб дизайнерів. Основні можливості програми:

- створення своїх заготовок для компонентів;
- зручна система збереження змін для різних версій;
- файли XD можна відкривати і в photoshop для редагування;
- якісно працює з векторною графікою;
- можливість імпорту файлів з інших програм для веб дизайну;
- можна створювати анімацію, відео, і багато іншого;
- є доступ до спільного проєкту для інших дизайнерів.

В Adobe XD можна малювати макети з нуля для найрізноманітніших сайтів, адаптивних сторінок і веб застосунків. Програма відмінно підходить для UI дизайну, створення вайрфреймів та прототипування макетів для майбутніх ресурсів [37].

Програму Sketch використовують переважно для створення інтерфейсів сторінок, так як в ній для цього є все необхідне: прототипування, направляючі, сітки, символи, векторне редагування, бібліотеки і навіть експорт коду. Плюси роботи в Sketch: зручна робота з текстовими блоками; наявність символів і динамічних кнопок; використання логіки CSS і т.д. До недоліків програми можна віднести те, що програма працює тільки на Mac і не сумісна з продуктами Adobe.

Figma – це відносно нова програма для дизайну сайтів і за сумісництвом – найголовніший конкурент Sketch. Figma володіє тими ж функціями і можливостями, що і Sketch. Додаток працює практично на будь-якій операційній системі – і в версії для ПК, і прямо в браузері. Інтерфейс програми нагадує по навігації Sketch і Adobe XD одночасно, що полегшує засвоєння програми. До особливостей цієї програми слід віднести:

- режим редагування доступний для безлічі користувачів одночасно;
- має свій хмарний сервіс зберігання даних;
- дозволяє додавати різні компоненти;
- наявність сіток та напрямних [38].

Таким чином Figma – це відмінний варіант, для початківців дизайнерів і програмістів, які хотіли б спробувати себе в веб індустрії, програма має безліч різноманітних функцій і елементів, які можуть зацікавити будь-кого, хто хоч трохи пов'язаний з веб дизайном.

Незважаючи на величезну кількість засобів для створення графічної оболонки сайту веб дизайнер повинен сам вирішувати в якому додатку чи програмі йому створювати дизайн для майбутнього ресурсу. Вибір конкретної програми залежить від того, яким функціоналом сайт повинен володіти, особистих потреб і бажань замовника тощо. Так само, не варто забувати, що в проєкті по створенню сайту задіяні як мінімум ще кілька людей, від верстальника, до програміста, яким теж необхідно створити умови для продуктивної роботи над спільним проєктом. Тому вибір необхідних програм для веб дизайну є важливим етапом для правильного оформлення роботу і

задоволення потреб клієнта в поставлених завданнях, створення комфортних умов для роботи в команді. Таким чином, дизайн навчального веб середовища для генерування завдань буде розроблятися у середовищі Figma.

2.6 Висновки до другого розділу

У даному розділі було проведено аналіз ключових аспектів програмного та технічного забезпечення, необхідного для створення прототипу навчального веб середовища для генерування завдань.

Було обрано HTML, CSS разом з SASS, а також TypeScript, що перетворюється на JavaScript після збірки проєкту, адже браузер підтримує лише JavaScript, як мову для відображення інтерактивних подій на сайті. Було обрано фреймворк Vue.js враховуючи його переваги перед іншими фреймворками разом з новітньою бібліотекою керування станів Pinia та найновітнішим інструментом збірки Vite. Середовище розробки Microsoft VS Code було обрано з урахуванням зручності, відстутності додаткових витрат на його придбання та підтримки вибраних мов програмування. Середовище збереження коду GitHub було обрано завдяки наявності підтримки контролю версій Git та відсутності витрат для збереження коду. Хостинг Render.com було обрано, оскільки він дозволяє максимально швидко розгорнути клієнтську частину веб додатку написану за допомогою будь-якого фреймворку, в тому числі за допомогою Vue.js та є можливість розмістити веб-додаток безкоштовно на власному домені.

РОЗДІЛ 3 СТВОРЕННЯ НАВЧАЛЬНОГО ВЕБ СЕРЕДОВИЩА ДЛЯ ГЕНЕРУВАННЯ ЗАВДАНЬ

3.1 Розробка загальної архітектури проєкту

Навчальне веб середовище для генерування завдань, призначене для організації та проведення тестів у навчальному процесі, включає двох ключових категорій користувачів – викладачів та студентів. Було створено діаграму прецедентів, представлену на рис. 3.1, що відображає різні типи користувачів та доступні їм в межах системи навчального веб середовища для генерування завдань.

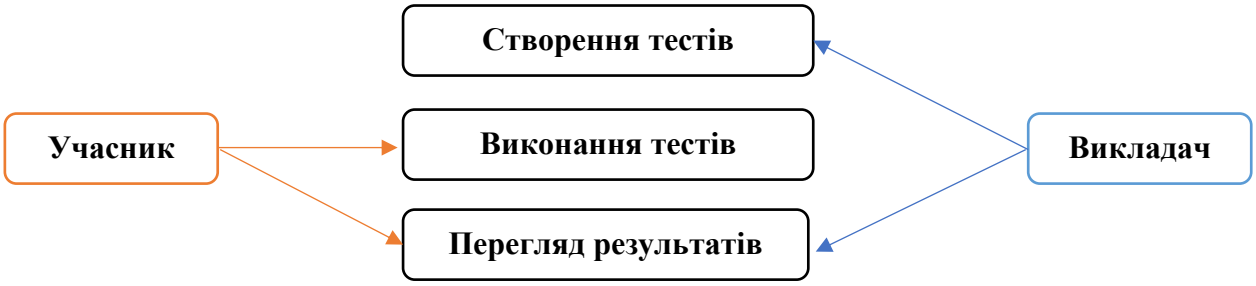


Рисунок 3.1 – Діаграма прецедентів системи навчального веб середовища для генерування завдань

Після перегляду даної діаграми прецедентів стає очевидним, що викладач відповідає за створення тестів, а учасник виконує їх. При цьому обидва типи користувачів мають можливість переглядати результати тестів.

Процес створення структури веб застосунку можна розділити на два етапи:

– структурування інформації;

– наочне зображення структури.

Структурування інформації – це необхідність класифікувати та групувати різні матеріали, об’єднувати їх у категорії чи заголовки та давати їм зручні для користувача імена. Це повинно враховувати логіку більшості користувачів.

Організаційна система навчального веб середовища для генерування завдань структурована у вигляді кількох блоків, кожен з яких містить різну кількість функціональних підблоків. Така організаційна структура спростить користування системою завдяки тому, що блоки будуть розташовані в послідовності, яку зазвичай використовують для розв'язання подібних задач.

На рис. 3.2 наведена схема структури навчального веб середовища для генерування завдань, на якій розташовані програмні модулі.

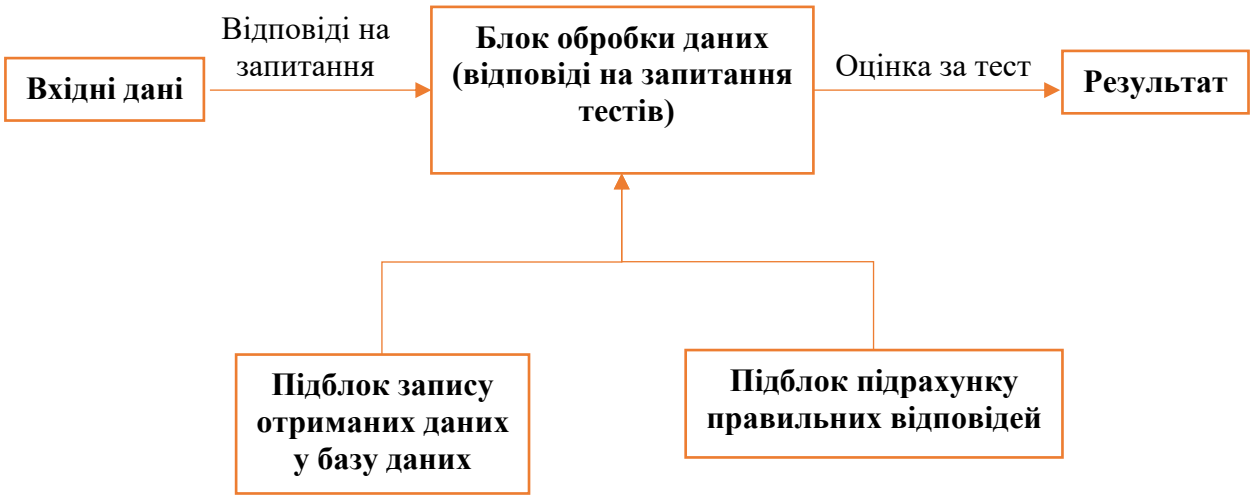


Рисунок 3.2 — Схема структури навчального веб середовища для генерування завдань

У систему вводяться вхідні дані – відповіді на питання, які піддаються обробці. Після цього система проводить розрахунки і виводить результат – оцінку за тест. Для відтворення даної структури, необхідне використання бази даних.

База даних – упорядкований набір логічно пов’язаних даних, які є спільними та призначеними для задоволення інформаційних потреб користувачів. Основне завдання бази даних – гарантувати зберігання великих обсягів інформації та забезпечити доступ до них для користувача або прикладної програми.

Отже, база даних складається з двох частин: інформації, що зберігається, та системи управління нею. Весь зміст системи а також її основні дані, тобто

дані про курси і тести зберігається в базі даних. Враховуючи поставлене завдання стосовно розробки саме клієнтської частини веб-додатку – ключовий акцент в цьому розділі буде зроблений саме клієнтській частині до якої не входить база даних.

Було вирішено, що архітектура прототипу навчального веб-середовища для генерування завдань буде складатись з декількох основних сторінок, які відображені в таблиці 3.1.

Таблиця 3.1

Сторінки прототипу навчального веб-середовища для генерування завдань з правами в межах сторінки відповідних категорій користувачів

№	Назва сторінки	Права учасника	Права викладача
1	Сторінка авторизації користувача	повні	повні
2	Сторінка реєстрації нового користувача	повні	повні
3	Сторінка курсу з переліком тестів	обмежені, без права редагування	повні
4	Сторінка перегляду тесту	відсутні	повні
5	Сторінка проходження тесту	повні	повні
6	Сторінка перегляду результатів проходження тесту	обмежені, без права редагування	повні
7	Сторінка редагування тесту	відсутні	повні
9	Сторінка перегляду статистики учасників тесту	відсутні	повні
10	Сторінка перегляду статистики учасників курсу	відсутні	повні

Таким чином, в процесі розробки архітектури проєкту та виділення ключових найнеобхідніших функцій було виділено 10 основних сторінок. Слід відзначити, що у викладачів є право на проходження тестів без збереження в результатів в базу даних для перевірки чи дійсно все правильно відображається в тесті для студента. Сторінки з різними правами для різних користувачів будуть відображатись по різному, в залежності від рівня прав

користувача. Також слід відзначити, що в залежності від ручної або автоматичної системи виставлення оцінок за проходження тесту сторінки для перегляду статистики успішності учасників тесту та курсу можуть відрізнятись та мати різні поля та можливості.

3.2 Розробка дизайну проєкту

Дизайн навчального веб середовища для генерування завдань розроблявся у середовищі Figma. Figma дозволяє розробляти інтерфейси в онлайн-додатку. У Figma дві ключові особливості: доступ до макету прямо з вікна браузера, можливість спільної роботи над макетами проєктів та можливість експорту прямо в pdf або зображення png, jpeg тощо.

У Figma було детально продумано та розроблено дизайн для кожної сторінки навчального веб середовища з генерування завдань. Кожен елемент інтерфейсу був чітко врахований з метою забезпечення оптимального користувацького досвіду та зручності використання.

Сторінка авторизації користувача відображена на рис. 3.3

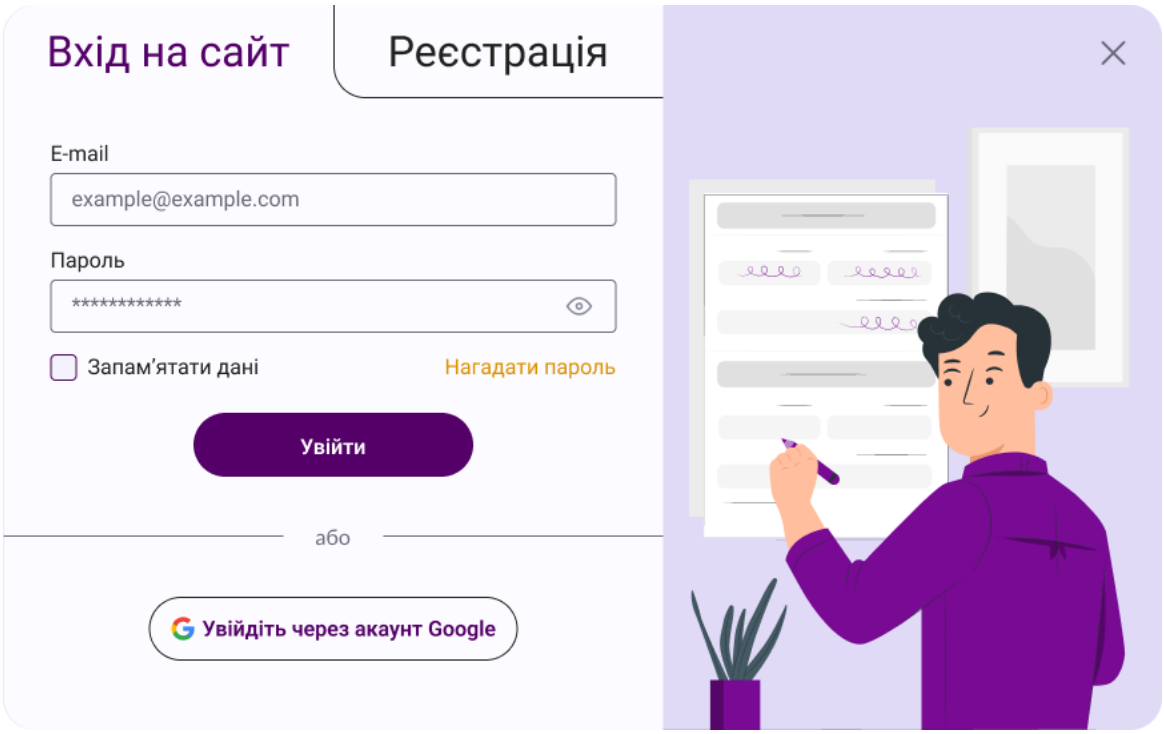


Рисунок 3.3 – Сторінка авторизації користувача

З рис. 3.3 стає очевидним, що на сторінці авторизації користувача мають бути присутні наступні поля:

– поле для вводу логіну користувача. Для даного веб-додатку логіном буде електронна пошта користувача, що має бути унікальною в системі;

– поле для вводу паролю користувача. Є функція відобразити пароль у вигляді відповідної іконки;

Також на сторінці авторизації присутні наступні кнопки:

– кнопка для увімкнення функції, щодо запам’ятовування даних користувача, щоб при наступному переході на сайт користувач міг автоматично авторизуватись, без повторного заповнення полів логіну та паролю;

– кнопка для нагадування паролю, після її натискання користувач має у спливаючому вікні вказати до якої саме пошти він забув пароль. Якщо така пошта вже зареєстрована в системі, то на неї прийде відповідний лист з унікальним посиланням для скидання паролю;

– кнопка «Увійти» після натискання якої проходить процес валідації поля email і у випадку, якщо email не дійсний – не відправляти запит на сервер, щоб не навантажувати його зайвий раз, а одразу показати помилку біля поля з email користувача. Якщо валідація поля email пройшла успішно – відправляти запит на сервер і перевіряти чи існує вказана пошта в системі і чи правильно був вказаний пароль для цієї пошти. Якщо такої пошти не існує в системі або пароль невірний, має відобразитись текст помилки над полем email про те що пошта або пароль вказані невірно. Слід звернути увагу, що не потрібно вказувати, що саме неправильно – пошта чи пароль, оскільки це може призвести до небажаних наслідків та намагань підібрати пароль до певної пошти.

– кнопка «Увійдіть через аккаунт Google» має перенаправляти користувача безпосередньо на сторінку авторизації від Google, після успішного проходження якої користувач стає авторизованим та потрапляє на головну сторінку сайту зі списком курсів, як авторизований користувач.

Сторінка реєстрації користувача відображена на рис. 3.4

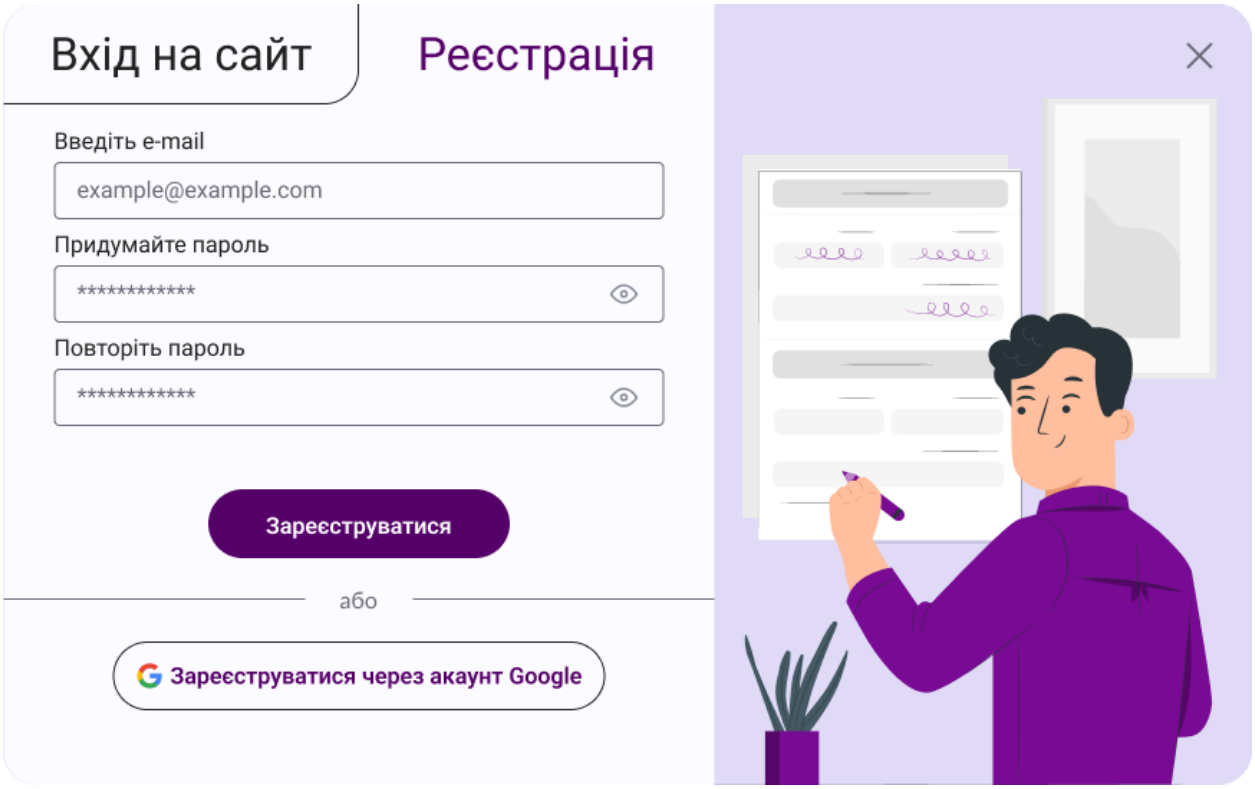


Рисунок 3.4 – Сторінка реєстрації користувача

Згідно рис. 3.4 на сторінці реєстрації користувача мають бути також присутні декілька полів. Зокрема поле для вводу email, яке перевіряється на валідність разом з полями паролю після натискання кнопки «Зареєструватися». Окремо слід зазначити, що на цій сторінці валідується також правильність паролю, щоб він включав букви у різних регістрах, цифри, спец символи та був не менше 10 символів. Також присутня можливість зареєструватись через акаунт Google аналогічно, як і випадку з авторизацією користувача.

На рис. 3.5 відображено сторінку з курсами для викладача, яка включає головний заголовок проєкту, що буде спільним для багатьох сторінок та складається з логотипу проєкту, кнопки «Підримати проєкт», пошти поточного користувача та кнопки для виходу з поточного акаунту. Зліва відображене головне меню, в якому є кнопка «Додати курс» та список курсів

користувача. Праворуч від головного меню відображено назву курсу, ПІБ викладача, кількість учасників та список тестів відповідного курсу.

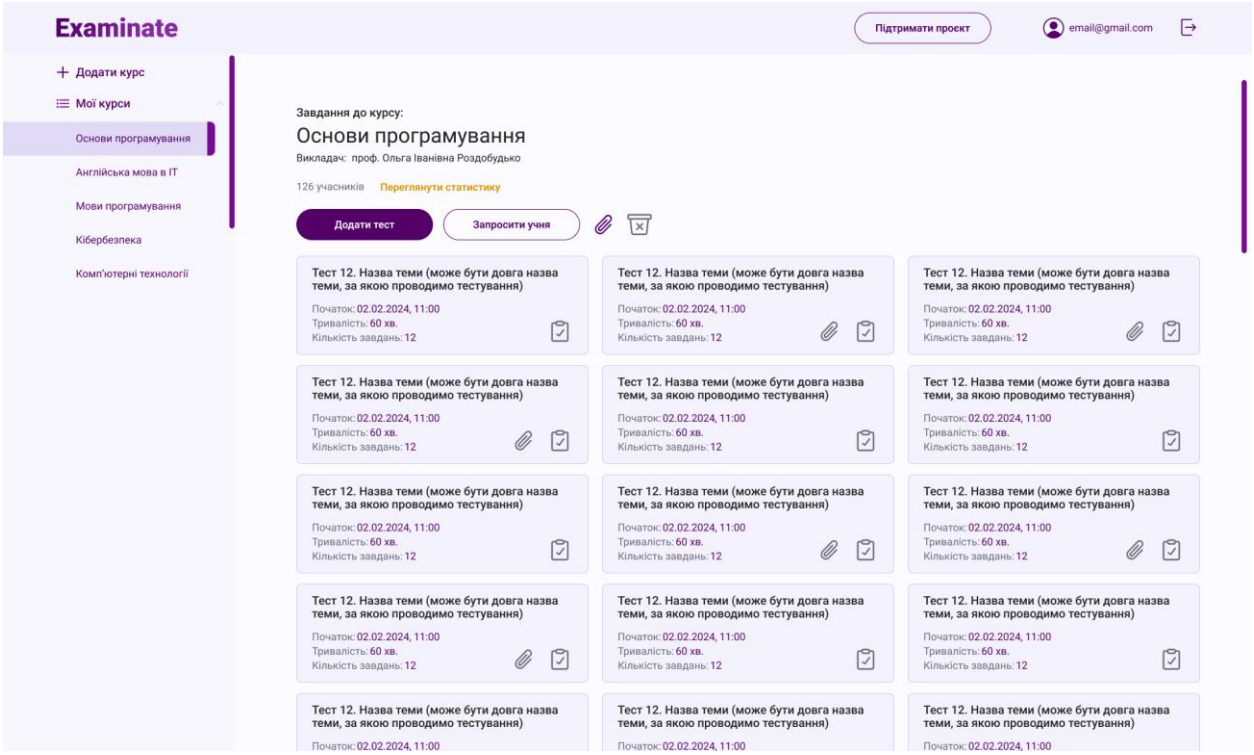


Рисунок 3.5 – Сторінка курсів викладача

Серед активних функцій для викладача на даній сторінці присутні:

– перегляд статистики курсу;
– додавання нового тесту курсу;
– створення запрошення для учасника;
– додати вкладення до курсу;
– видалення курсу;

На рис. 3.6 відображено сторінку курсів, але вже для учасника курсу. На ньому відображено, що для студента це має бути майже така сама сторінка, як і у викладача, але без активних функцій. Також на цій сторінці помітно, що в тестах також можуть вкладення, що стосуються всього тесту в цілому. Наявність таких вкладень позначена відповідною іконкою.

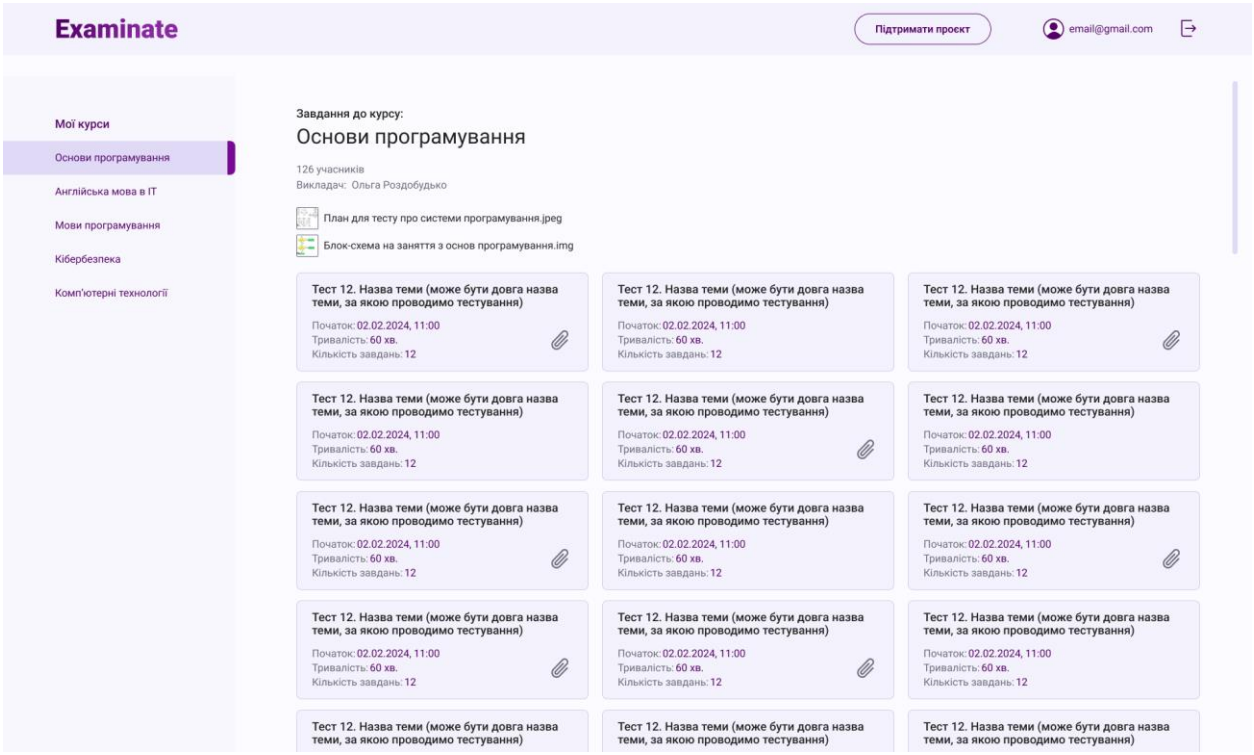


Рисунок 3.6 – Сторінка курсів учасника

На рис. 3.7 відображено сторінку перегляду тесту для викладача на якій викладачу доступні наступні функції:

– перевірка тестів, що були виконані учасниками та виставлення оцінок;

– можливість редагування тесту, тобто загальної інформації стосовно тесту з можливістю вказання максимальної кількості балів за тест;

– додавання нових завдань до тесту або редагування вже існуючих з виставленням максимальних балів за кожен тест;

– додавання вкладень, що відносяться до тесту в цілому;

– додавання вкладень, що відносяться до конкретного завдання. Вони можуть містити таблиці, формули, малюнки, схеми тощо;

– можливість видалення тесту (на рис. 3.7 ще не відображена відповідна іконка);

– можливість здійснити додаткові просунуті налаштування тесту. Зокрема чи відображати всі запитання по черзі для студента чи лише одне запитання на екран і тільки після відповіді студента здійснювати перехід на наступне запитання, генерувати випадкові варіанти тестів для кожного

студента з обмеженням максимальної кількості тестів на 1 варіант, ставити 0 балів за неповну відповідь, коли в тесті кілька правильних відповідей або розраховувати оцінку за тест у %, за кожну правильну відповідь та чи показувати одразу результат проходження тесту студенту у випадку, якщо тест не містить відкритих питань;

– можливість вибору типу завдання: тест з однією правильною відповіддю, тест з кількома правильними відповідями, відкрите питання;

Окремо слід відзначити, що на даній сторінці також планується розробка експериментальної функції, якою можна буде скористатись за допомогою додаткової кнопки «Додати список завдань», яка буде переадресовувати на нову сторінку з текстовим полем в яке можна буде вставити текст усіх завдань одразу, головне щоб кожне завдання мало свій порядковий номер у форматі «1.» та підпункти у форматі «1)» або «a)» з позначенням правильної відповіді в кінці підпункту знаком +. Якщо такого позначення не буде, то на етапі перегляду тесту можна буде позначити правильну відповідь. Після збереження текстового поля у викладача буде вибір або додати всі ці завдання до вже існуючих у тесті або замінити усі попередні тести – новими. Вкладення за необхідності можна буде додати на етапі перегляду тесту. Це дозволить значно скоротити час на створення тестів.

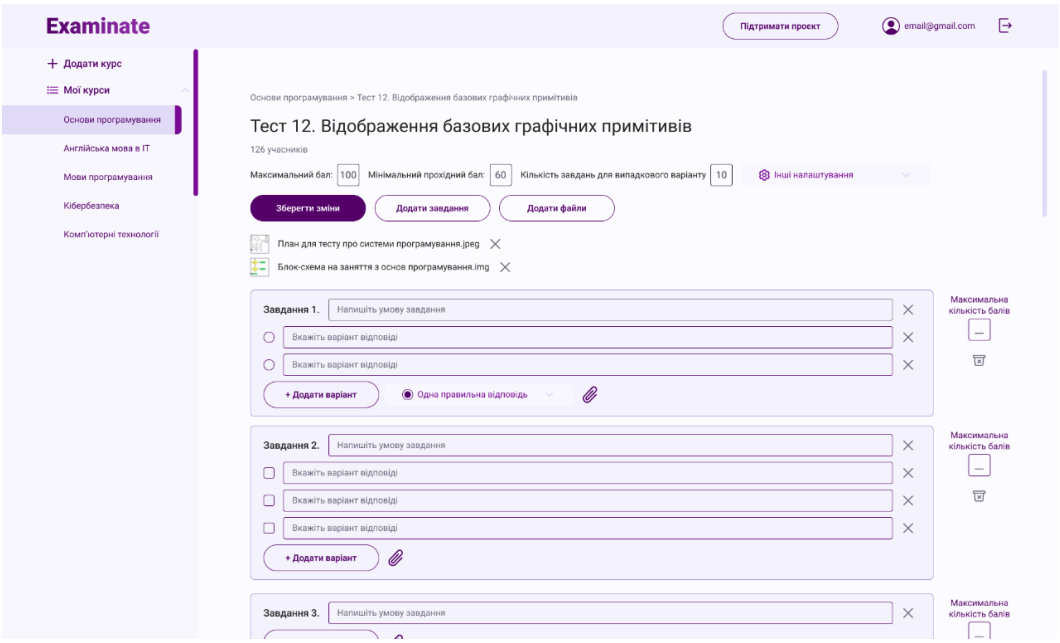


Рисунок 3.7 – Сторінка перегляду та редагування тестів викладачем

На рис. 3.8-3.9 зображені сторінки статистики тесту з автоматичними оцінками та ручним виставленням балів викладачем в разі потреби або якщо тести містять відкриті питання.

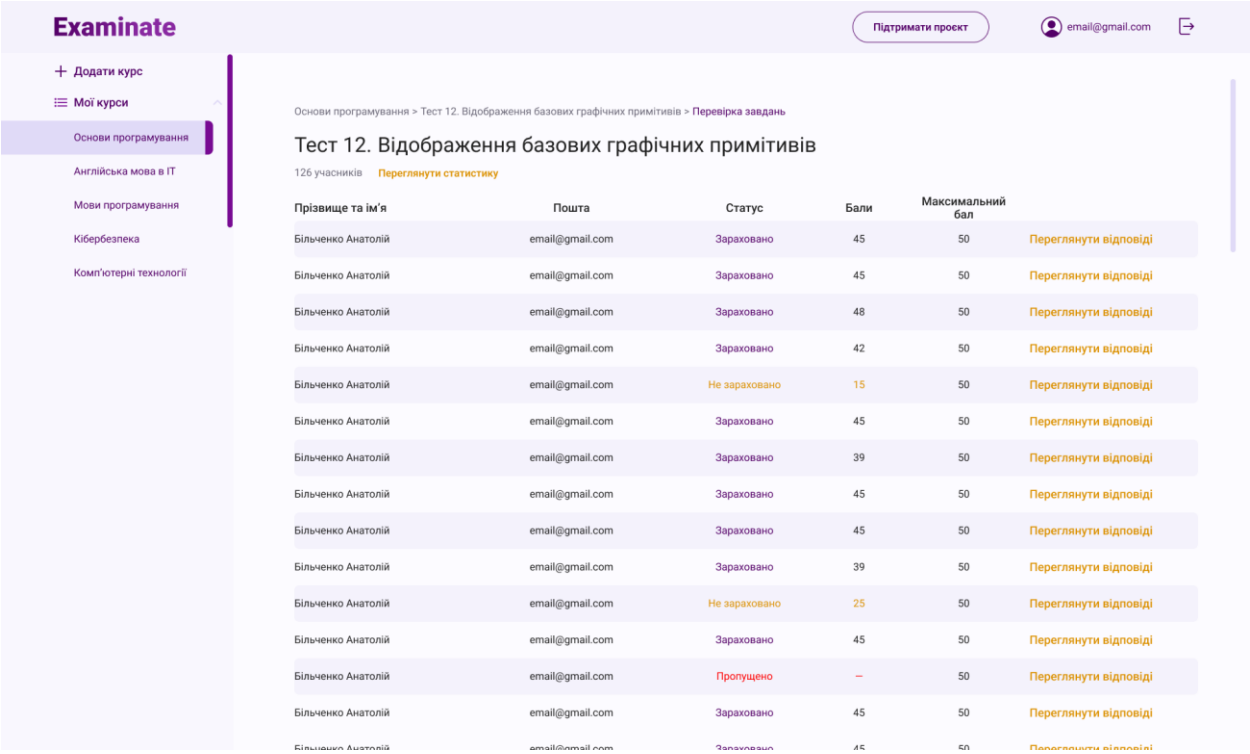


Рисунок 3.8 – Сторінка перегляду викладачем статистики тесту з автоматичними результатами тестів

Згідно вищенаведеного рис. 3.8 викладачу буде доступна повна інформація щодо успішності складання тесту кожним учасником курсу. Зокрема буде видно прізвище, ім’я, пошта, статус проходження тесту, набрані бали, максимальний бал кожного учасника курсу. Також у викладача буде можливість переглянути відповіді кожного учасника та побачити хто з учасників курсу взагалі не з’явився для проходження тесту у зазначені терміни.

Відповідно до рис. 3.9 викладачу буде доступна аналогічна статистика по кожному учаснику тесту, аналогічно можливості та додатково реалізована можливість виставляти бали за тест вручну. Такий функціонал буде

необхідний в разі наявності відкритих питань, які викладач має перевірити особисто.

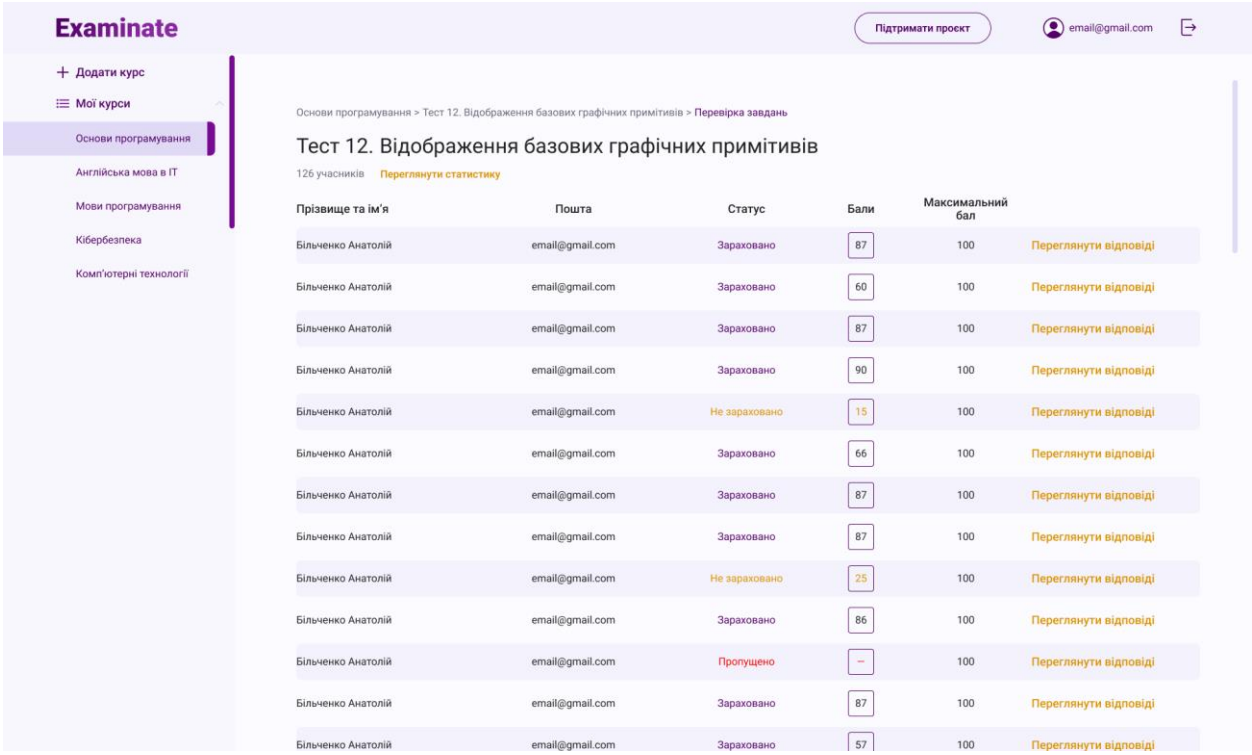


Рисунок 3.9 – Сторінка перегляду викладачем статистики тесту, що включають відкриті відповіді чи потребують ручного виставлення оцінок

На рис. 3.10 відображено статистику всього курсу по кожному учаснику. На цій сторінці викладач буде мати змогу:

– перегляд базової інформації по кожному учаснику курсу;

– перегляд успішності кожного учасника тесту, а саме скільки всього тестів здано та скільки тестів було успішно здано;

– можливість перегляду тестів кожного учасника після натискання на кнопку «переглянути роботу» на окремій сторінці, яка не була позначена як першочергова та не увійде до першої версії прототипу навчального веб-середовища;

Також будуть створені окремі сторінки для перегляду просунутої статистики по кожному курсу та кожному тесту в цілому з графіками та дашбордами, що будуть відображати середню успішність студентів, середній

час проходження тесту, з якими питаннями студенти найчастіше не справляються тощо. Ці сторінки були не були позначені, як першочергові, тож не увійдуть в першу версію прототипу навчального веб-середовища.

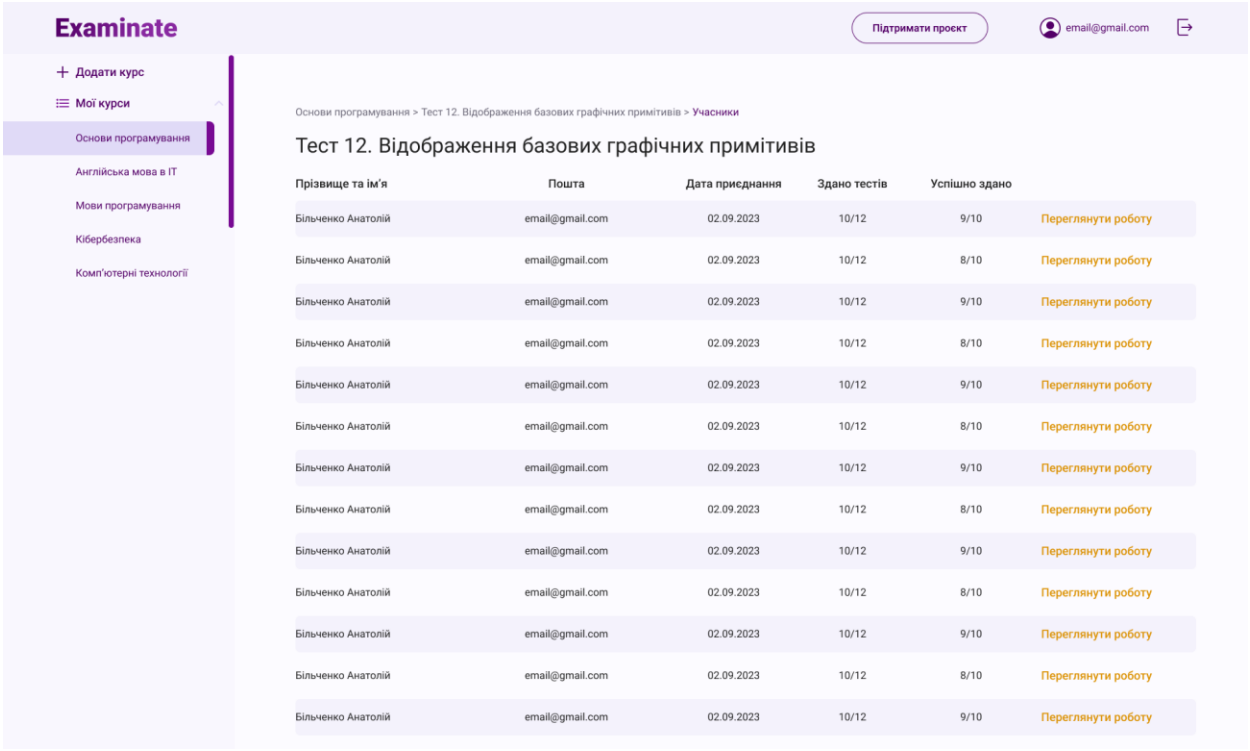


Рисунок 3.10 – Перегляд статистики курсу викладачем

На рис. 3.11-3.12 відображені сторінки проходження тесту в ролі учасника курсу, коли доступна лише одна відповідь на один екран. На цих сторінках відображені відмінності між тестовим типом завдання та завданням з відкритою відповіддю.

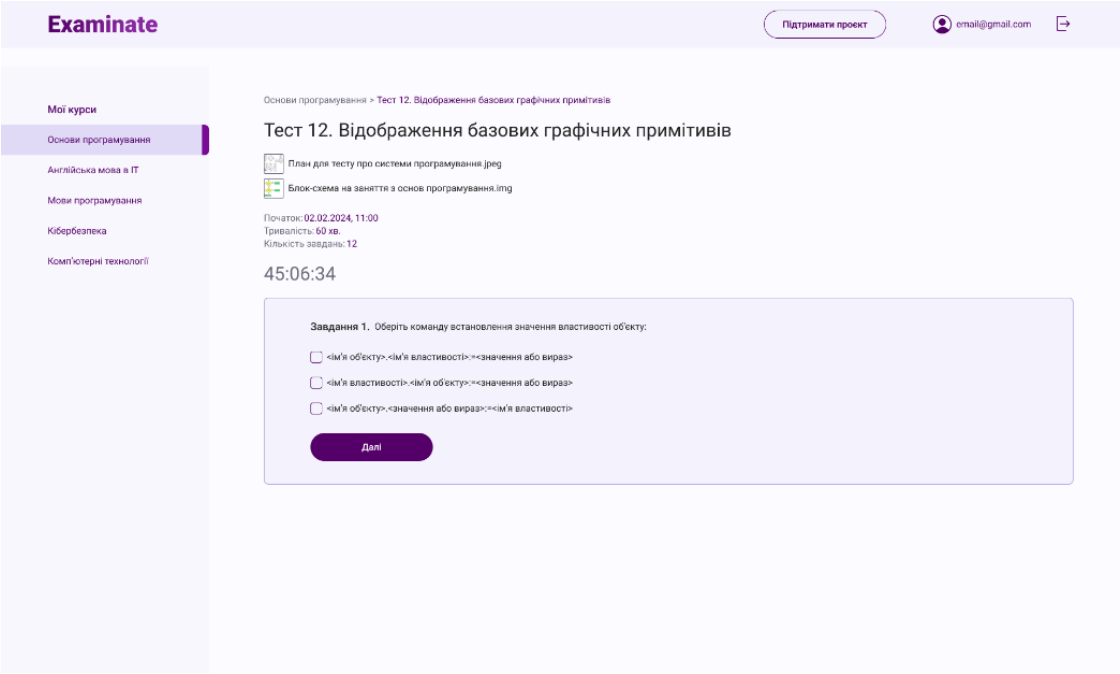


Рисунок 3.11 – Сторінка проходження тесту з тестовим типом завдання

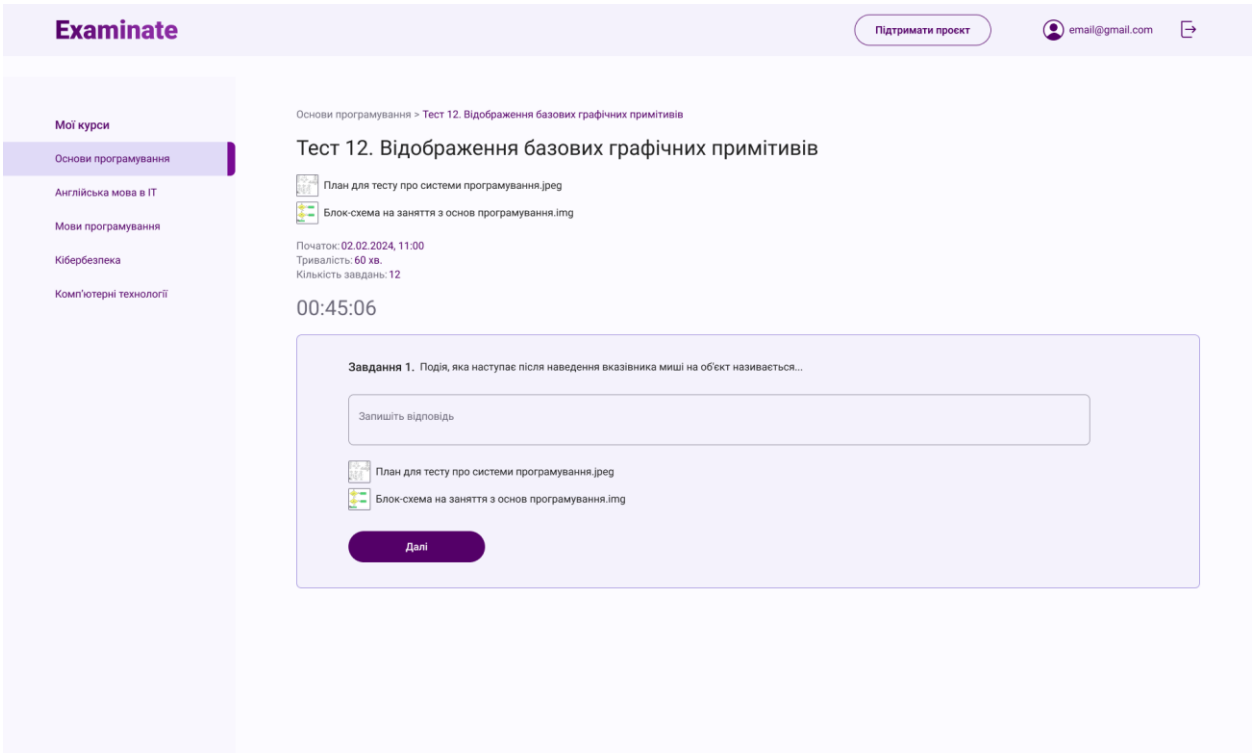


Рисунок 3.12 – Сторінка проходження тесту з тестовим типом завдання

На рис. 3.13 відображено сторінку завершення тесту учасником курсу. Якщо в цьому тесті немає відкритих питань та буде вказаний автоматичний розрахунок балів викладачем, то учасник курсу зможе одразу побачити власну оцінку. В іншому випадку кількість набраних балів не буде відображена.

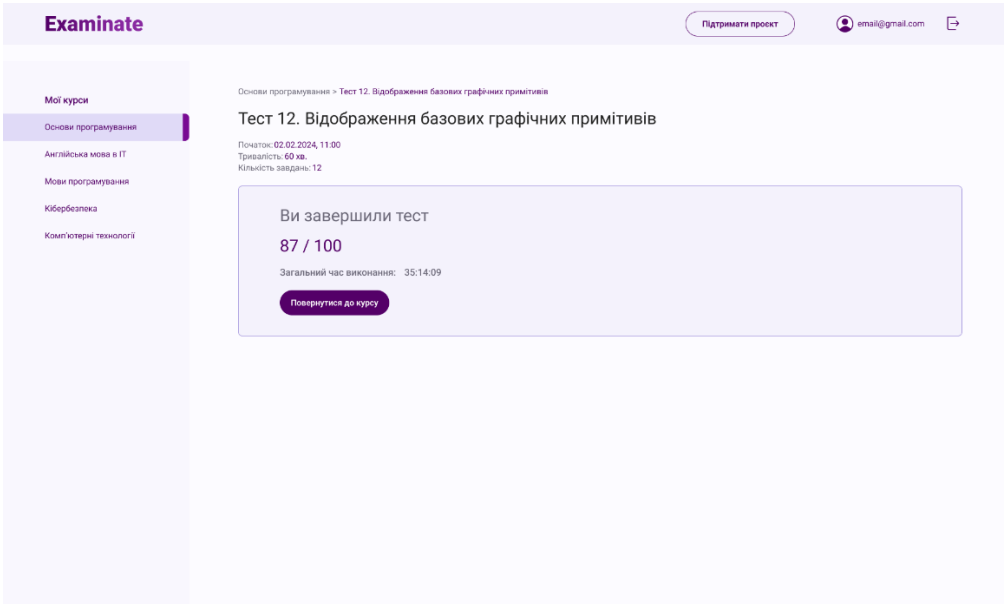


Рисунок 3.13 – Сторінка закінчення проходження тесту з тестовим типом завдання та автоматичним розрахунком балів

При розробці дизайну одним з головних завдань було створення та дотримання визначеної палітри кольорів. Для цього було розроблено декілька палітр кольорів та обрано палітру, що найбільш естетично відображає сторінки та не повторює відомі платформи. Також головним завданням було створення естетичного та інтуїтивно зрозумілого інтерфейсу веб-додатку для користувачів. Палітру кольорів, основні компоненти, які можна перевикористати на різних сторінках з основними станами сторінок було винесено в окремі сторінки дизайну.

При реалізації дизайну для прототипу навчального веб середовища для генерування завдань будуть використані визначені у другому розділі технології для розробки веб-додатку.

3.3 Програмна реалізація і результат виконаної роботи

У результаті виконання дипломного проєкту було реалізовано заплановану мету проєкту, тобто – створення прототипу частини навчального веб середовища для генерування завдань. Лістинг коду навчального веб

середовища для генерування завдань представлено в Додатку А. На рис. 3.14 детально зображено структуру проєкту створену інструментом збірки Vite.

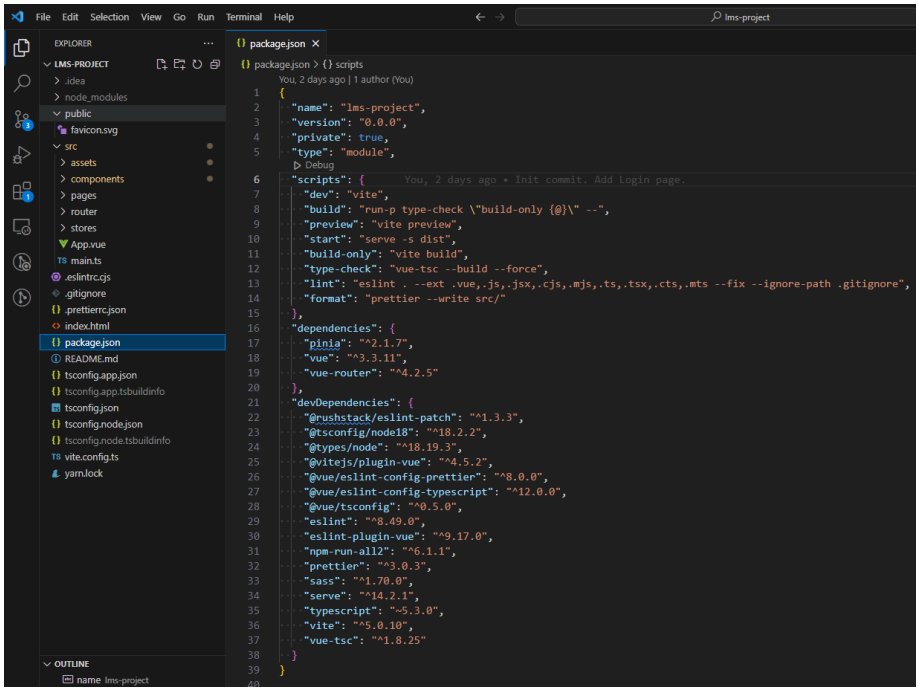


Рисунок 3.14 – Структура навчального веб середовища для генерування завдань

Більшу частину файлів в корені проєкту створено за допомогою команди «npm create vue@latest», що наведена в офіційній технічній документації фреймворку Vue.

В файлі package.json знаходятся всі основні залежності, що поділені на ті що необхідні лише при розробці додатку та ті що потрібні завжди навіть при розгортанні на сервері. Також в даному файлі відображено основну інформацію про репозиторій коду та скрипти, які можна виконати в терміналі.

Перед початком роботи необхідно обрати, який пакетний менеджер буде використовуватись для завантаження та зберігання залежностей, необхідних для коректної роботи проєкту. Після проведення ретельного порівняння наявних пакетних менеджерів npm, yarn та pnpm було обрано пакетний менеджер yarn, оскільки він швидше виконує команди та краще оптимізує місце для збереження залежностій при розгортанні на сервері. Pnpm

вважається ще швидше ніж yarn, проте на разі цей пакетний менеджер є відносно новим та не користується такою популярністю, як npm або yarn і немає такого широкого кола користувачів та достатньої підтримки, що значно збільшує ризики виникнення незапланованих помилок при розгортанні на сервері.

Для додавання усіх необхідних бібліотек залежностей безпосередньо в репозиторій коду необхідно скористатись командою yarn install. Перед виконанням команди, слід впевнитись, що в операційній системі встановлені глобально Node.js, yarn. Деякі бібліотеки залежностей не підтримують застарілі версії Node.js раніше 12 версії, тому слід пересвідчитись, що в операційній системі встановлена сучасна версія Node.js [39]. Далі для додавання нових залежностей, що не відображені в package.json необхідно скористатись командою « yarn add <назва бібліотеки залежності> ».

Після успішного встановлення всіх необхідних бібліотек залежностей локально, необхідно виконати команду «yarn dev». Результат виконання даної команди в терміналі відображено на рис. 3.15

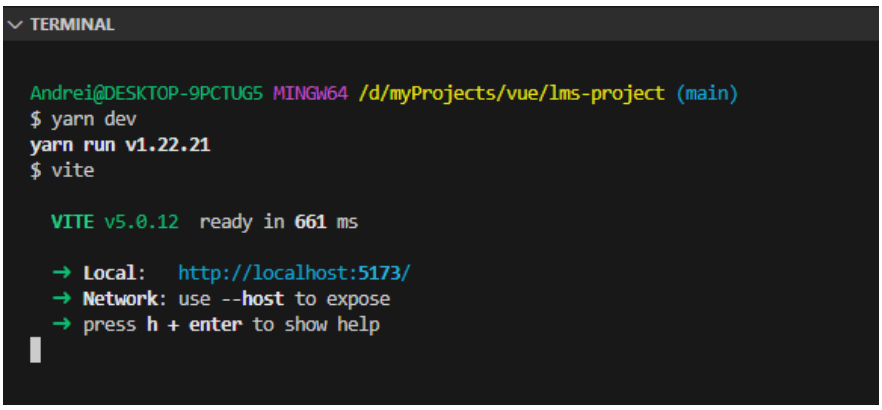


Рисунок 3.15 – Результат виконання команди «yarn dev» в терміналі VS Code

В результаті виконання даної команди в браузері за адресою http://localhost:5173/ буде розгорнуто сайт, що доступний лише для локального комп’ютеру розробника. Якщо в браузері попередньо не було здійснено входу у веб-додаток або не збережено дані для входу, то з усіх сторінок користувача буде примусово перенаправлено на сторінку авторизації, де користувач зможе

авторизуватись чи зареєструватись. Відображення цієї сторінки в браузері наведено на рис. 3.16.

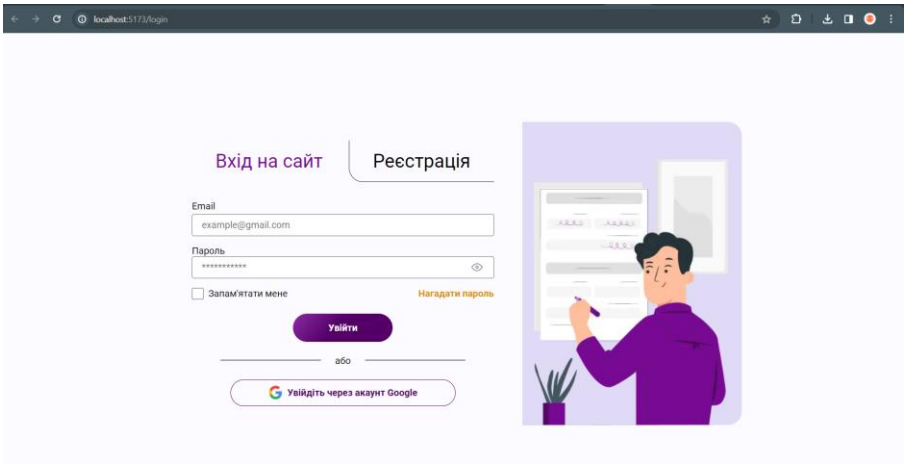


Рисунок 3.16 – Відображення сторінки авторизації сайту в браузері

Після внесення змін в код відповідно до розробленого дизайну, можливо здійснити збереження останньої версії коду за допомогою системи версій Git та відправити код до віддаленого репозиторію, що розміщений в хмарі сервісу GitHub за допомогою серії стандартних команд системи контролю версій Git.

Для тестування та перегляду сайту іншими користувачами окрім розробника, необхідно розгорнути сайт на сервері, що буде доступний публічно для інших користувачів. Для цього було використано хостинг Render.com та було придбано власний домен examinate.online.

Основними етапами процесу розгортання сайту на Render.com на власному домені examinate.online були наступні:

– реєстрація на сервісі Render.com;

– обрання необхідного типу сервісу та підключення GitHub репозиторію до сервісу Render.com для можливості автоматичного деплою після внесення змін до певної «гілки» коду в системі контролю версій Git, яку буде обрано на сервісі для здійснення прив’язки (рис. 3.17);

– обрання піддомену Render.com для першого розміщення сайті в межах Render.com без використання власного домену за адресою examinate.onrender.com (рис. 3.18).

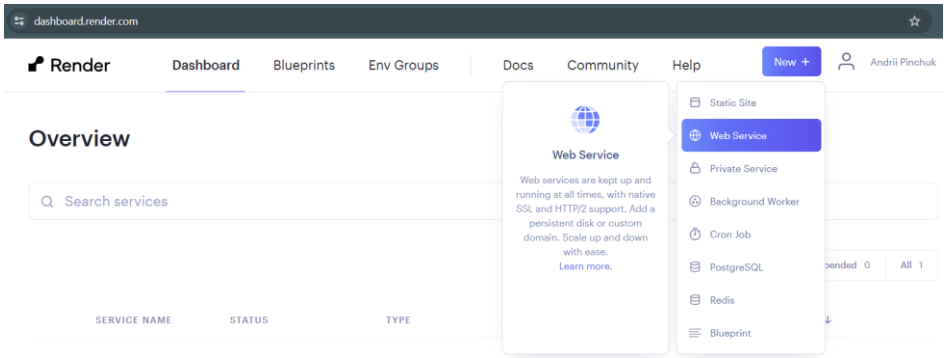


Рисунок 3.17 – Обрання необхідного типу сервісу на Render.com

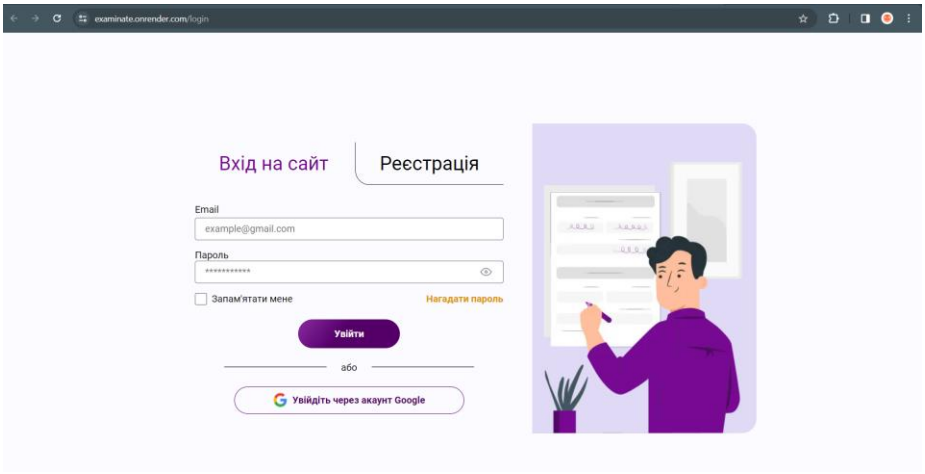


Рисунок 3.18 – Відображення сайту на піддомені сервісу Render.com

Далі в налаштуваннях існуючого сервісу необхідно обрати опцію «користувацький домен» та внести зміни в налаштування DNS у провайдера, в якого було придбано домен. Враховуючи той факт, що домен було придбано в українського провайдера Hostiq – зміни в DNS налаштування необхідно було внести в їхньому кабінеті (рис. 3.19)

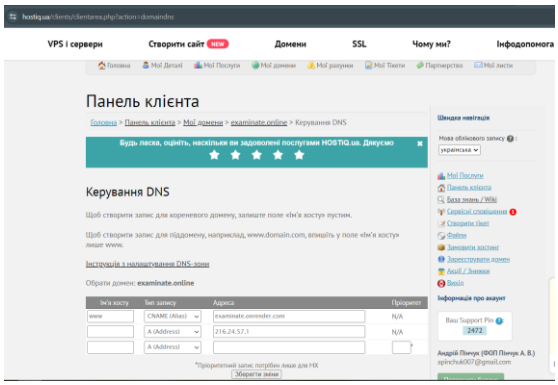


Рисунок 3.19 – Керування налаштуваннями DNS в панелі клієнта Hostiq

При спробі змінити налаштування виявилось, що Hostiq не дозволяє змінювати параметр CNAME (Alias) для кореневого домену examinate.online, однак сервіс Render.com був готовий до такого розвитку подій та одразу пропонує створювати піддомен з www.examinate.online (рис. 3.20) на який буде встановлений автоматичний редирект з examinate.online. Це показує, що Render.com вчасно здійснив аналіз ризиків, отримав оцінку потенційних ризиків та їх вплив на проєкт та прийняв відповідне рішення, що дозволило мені, як користувачу сервісу досягти поставленої мети. Оцінка ризиків повинна проводитись на кожному етапі розробки програмного забезпечення, щоб команда розробників мала можливість вчасно виявити проблеми та прийняти відповідні заходи [40]. Враховуючи вірогідність ризику непередбачуваних проблем через відсутність налаштувань для кореневого домену на Hostiq додатково був змінений параметр A (Address) на випадок непередбачуваних проблем з піддоменом www.examinate.online.

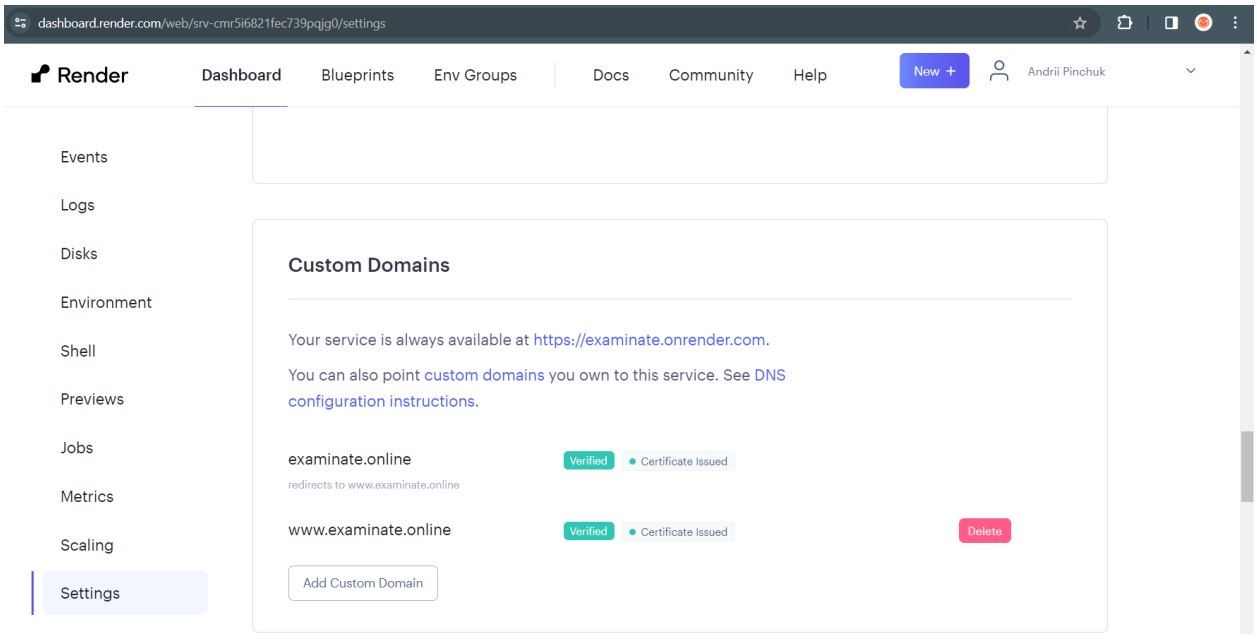


Рисунок 3.20 – Налаштування користувацього домену на сервісі Render.com

Після успішного налаштування DNS, сайт був успішно розгорнутий на власному домені (рис. 3.21). Окремо слід відзначити, що нові зміни в коді в прив’язаній «гілці» коду системи контролю версій Git одразу призводять до

автоматичного оновлення сайту не тільки на піддомені Render.com, а й на власному домені. Простота налаштувань, розгортання коду та підготовка до можливих несприятливих сценаріїв стосовно розміщення сайтів безумовно є великою перевагою Render.com з поміж інших подібних сервісів. Навіть якщо після змін в коді сталася помилка при спробі розгортання сайту – Render.com одразу повідомить власника за допомогою листа на електронну пошту. Також можна отримати детальну інформацію про помилку в логах серверу, що доступні для перегляду у власному кабінеті.

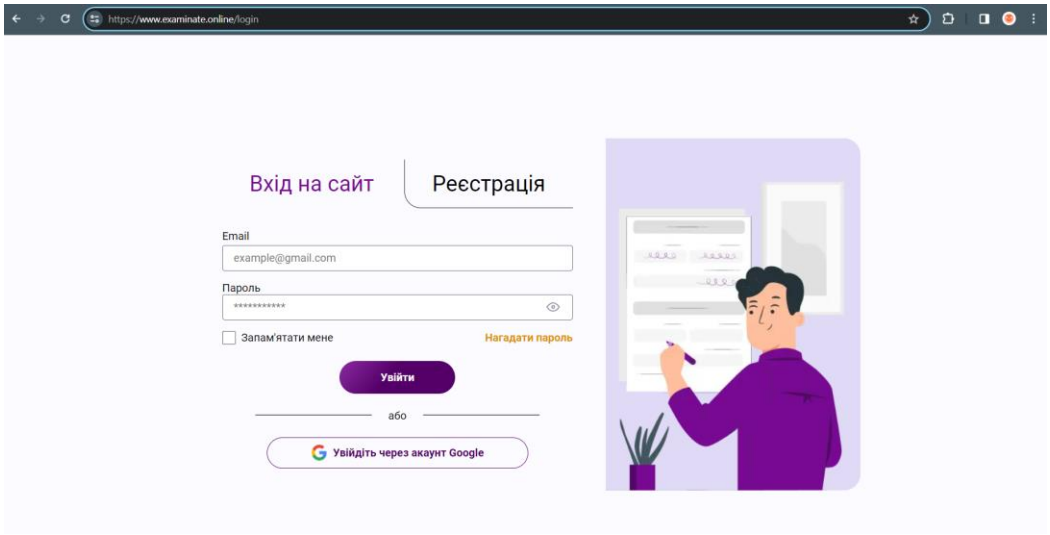


Рисунок 3.21 – Відображення сайту на власному домені

Окремо слід зауважити, що для успішного розгортання веб-додатку, що побудований на TypeScript краще перед збереженням останньої версії коду здійснити перевірку типів на відповідність стандартам TypeScript в терміналі VS Code за допомогою скрипту «yarn type-check», що виконає команду «vue-tsc --build --force» та у випадку наявності помилок з типами даних – одразу покаже їх в терміналі. Однією з найпоширеніших помилок є відсутність вказання типу для параметру функції, що призведе до помилки «Typescript: TS7006: Parameter 'a' implicitly has an 'any' type». Якщо не виправити таку помилку до збереження та відправки коду в хмарний репозиторій – розгортання сайту з новими змінами не буде виконано і в логах буде відображено цю помилку і повторне успішне розгортання сайту буде

можливе лише після виправлення цією помилки та оновлення коду в хмарному репозиторії, що підключений до Render.com.

3.4 Висновки до третього розділу

У даному розділі було розглянуто процес створення прототипу частини навчального веб середовища для генерування завдань. що почався з етапу визначення базової архітектури проєкту. Під час проходження цього етапу було визначено дві необхідні категорії користувачів проєкту, а саме учасник та викладач та які функції вони будуть здійснювати в межах навчального веб-середовища з генерації завдань. Відповідно до цього було побудовано діаграму прецедентів та виявлено, що студент може проходити тести в межах курсів та переглядати свої результати тестів, а викладач може створювати тести та переглядати результати всіх учасників власного курсу.

На першому етапі визначення базової архітектури проєкту також було визначено необхідні сторінки для першої версії прототипу частини навчального веб середовища для генерування завдань та визначено права кожної категорії користувачів в межах кожної необхідної сторінки.

Наступним етапом стало створення дизайну сторінок з урахуванням всіх необхідних функцій для користувача, дотримання палітри кольорів, яка в результаті тестування різноманітних макетів дизайну, була визнана найкращою. Дизайн сторінок створювався з урахуванням сучасних тенденцій до створення подібних платформ, мінімізації кількості елементів на сторінці та забезпечення зручності в користуванні та інтуїтивно зрозумілого дизайну, як для викладачів, так і для студентів.

Програмна реалізація проєкту та отримані результати виконаної роботи підтверджують відповідність поставленим завданням. Розглядаючи деталі виконання та демонстрацію функціональності навчального веб середовища, було відзначено його потенціал для ефективного використання у навчальних цілях.

ВИСНОВКИ

Дослідження наявних проблем у сучасній дистанційні освіті в Україні, зокрема у наявних освітніх платформах та навчальних середовищах в умовах постійного зростання вимог до ефективності та якості навчального процесу показало, що в начальних середовищах, які є невід'ємними компонентами освітніх платформ існує ряд невирішених проблем, як технічних так і нетехнічних. Серед технічних було виділено:

– обмеженість функціоналу освітніх платформ, що присутні в системі дистанційної освіти в Україні. На сьогодні в межах цих платформ не можливо створити тестові завдання, що можуть включати формули, таблиці та вкладення до кожного завдання та неможливо генерувати випадкові варіанти для студентів, що зобов'язує викладачів для кожного окремого варіанту створювати окремі тестові завдання, що веде до значних витрат у часі;

– відсутність універсальних навчальних веб середовищ поза межами освітніх платформ, які б дозволили будувати тестові завдання з урахуванням вищезазначених наявних проблем в освітніх платформах. В Україні вже існують просунуті веб середовища, які мають дійсно вражаючий набір функцій, проте на сьогодні вони існують лише в межах комерційних освітніх платформ та прив'язані до контексту курсів, що вже існують в межах цих платформ та в них немає можливості створення тестових завдань користувачами.

Таким чином, в результаті вищенаведеного дослідження проблем сучасної системи дистанційної освіти, зокрема освітніх платформ та навчальних веб середовищ, аналізу предметної області та огляду існуючих розв'язків вищенаведених проблем в системі дистанційної освіти України було сформовано завдання для практичного дослідження, яке полягає у розробці прототипу нового універсального навчального веб-середовища, оскільки доопрацювання вже існуючих освітніх платформ та навчальних веб середовищ вимагає обов'язкової згоди їх власників і у випадку їх згоди це

може зайняти значно більше часу, ніж розробка нового навчального веб-середовища для генерування завдань.

В результаті дослідження існуючих методів та технологій розробки веб-додатків в цілому та зокрема освітніх платформ, в тому числі навчальних веб середовищ було проаналізовано широкий спектр існуючих інструментів для розробки та виявлено, що на сьогодні найкращим вибором для розробки не лише освітніх платформ, навчальних веб середовищ, а й в цілому веб-додатків буде поєднання наступних інструментів:

– HTML5, CSS3, SASS як веб технології;

– JavaScript та TypeScript, як мови програмування;

– фреймворк Vue (3 версія), як фреймворк для реалізації інтерфейсу з використанням підходу Composition APi, бібліотеки загального стану додатку Pinija, а також інструменту збірки Vite.

– Visual Studio Code, як середовище розробки;

– Figma, як інструмент для розробки дизайну та прототипування;

– GitHub, як сервіс збереження коду в хмарному репозиторії;

– Render.com, як провайдер хостингу;

– Hostiq, як провайдер для придбання домену;

Завдяки використанню цих інструментів вдалось розробити прототип навчального веб середовища для генерування завдань та розмістити його в мережі Інтернет, яке відповідає всім вимогам, що були сформульовані на етапі постановки завдання та вирішує вищенаведені проблеми в системі дистанційної освіти України завдяки усуненню наявних технічних обмежень освітніх платформ, зокрема в межах навчальних веб середовищ для генерування завдань.

Основним напрямком подальшого дослідження є пошук інших наявних обмежень освітніх платформ та навчальних веб середовищ для генерування завдань, які можливо вирішити за допомогою сучасних інструментів розробки та дослідження нових інструментів розробки, що можуть більш ефективно усувати наявні обмеження та покращувати рівень розвитку освітніх платформ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Цеслів О.В. Основи програмування та веб-дизайн: навчальний посібник. Київ: КПІ, 2020. 149 с.

2. Гунченко Ю. О. Методи аналізу та синтезу розробки web–додатків. Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. 2017. №. 57. С. 96–104.

3. Спірін О. М., Колос К. Р. Технологія організації масового дистанційного навчання учнів в умовах карантину на базі платформи Moodle. Інформаційні технології і засоби навчання. 2020. №5. С. 29–58.

4. Видайчук Т. Л. Модернізація філологічної освіти та науки України: глобалізаційні виклики, аналіз дієвості інноваційних освітніх веб платформ. Академічні візії. 2023. №16. URL: https://elibrary.kubg.edu.ua/id/eprint/43716/1/T_Vydaichuk_AV_2023_FUFKM.pdf.

5. Шерман, М. І., Степаненко, Н. В., Фельбуш, А. В. Педагогічні засади створення навчального веб-ресурсу з дисципліни «Інформатика і системологія» для майбутніх екологів. Інформаційні технології в освіті. 2017. № 3. С. 21–39.

6. Тамаркіна О. Л. Самостійна робота студентів ВЗО в умовах дистанційного навчання. Державний педагогічний університет імені Івана Франка. 2020. №34. С. 228–232.

7. Биков В. Ю. Цифрова трансформація суспільства і розвиток комп'ютерно-технологічної платформи освіти і науки України. Інформаційно-цифровий освітній простір України: трансформаційні процеси і перспективи розвитку: Матеріали методологічного семінару НАПН України, м. Київ, 4 квіт. 2019 р. Київ, 2019. С. 20–26.

8. Острога М., Шамоня В., Шершень О. Цифрові освітні веб платформи як інструмент реалізації неформальної освіти. Освіта. Інноватика. Практика. 2022. №4. С. 27–36.

9. Пінчук А. В., Ульяновська Ю. В. Функціональні й нефункціональні вимоги до розробки програмного забезпечення та їх роль в управлінні якістю. Економіко-правові та управлінсько-технологічні виміри сьогодення: молодіжний погляд: матеріали міжнар. наук.-практ. конф., м. Дніпро, 4 лист. 2022 р. Дніпро, 2022. С. 411–412.

10. Левус Є. В., Марусенкова Т. А., Нитребич О. О. Життєвий цикл програмного забезпечення: навч. посіб. Львів: Львівська політехніка, 2017. 208 с.

11. Антоненко В. М., Мамченко С. Д., Рогушина Ю. В. Сучасні інформаційні системи і технології управління знаннями: навч. посіб. Ірпінь: Нац. університет ДПС України, 2016. 212 с.

12. Пінчук А. В., Критенко О. О. Цифровізація митної справи: сучасний стан та перспективи розвитку. Цифрове суспільство: управління, фінанси та соціум: матеріали міжнар. наук.-практ. конф., м. Дніпро, 28 квіт. 2023 р. Дніпро, 2023. С. 219–221.

13. Amirault R. J. Distance learning in the 21st century university: Key issues for leaders and faculty. Quarterly Review of Distance Education. 2012. Т. 3, №. 4. С. 253–254.

14. Гасинець Я. С., Вакерич М. М., Куртяк Ф. Ф. Цифрова трансформація освіти майбутнього: стандарти, норми та правила. Академічні візії. 2023. №16. URL: https://academy-vision.org/index.php/av/article/view/143.

15. Морзе Н.В. Інформаційні системи: навч. посіб. Івано-Франківськ: ЛілеяНВ, 2015. 384 с.

16. Leslie A., Beverley E., Sian M. P. Enhancing the online learning experience using virtual interactive classrooms. Australian Journal of Advanced Nursing. 2015. Т. 32, № 32/4. С. 22–31.

17. Самойчук К. О., Паляничка Н. О., Верхоланцева В. О. Особливості організації самостійної роботи для студентів технічних спеціальностей в умовах дистанційного навчання. Таврійський державний

агротехнологічний університет імені Дмитра Моторного. 2022. №16. С. 206–210. URL: http://elar.tsatu.edu.ua/handle/123456789/16650.

18. Сіняєва О. Особливості використання інформаційних технологій в освіті. Освіта. Інноватика. Практика. 2023. Т. 11, № 7. С. 98–104.

19. Мельник Р. А. Програмування веб-застосувань (фронт-енд та бек-енд): навч. посіб. Львів: Львівська політехніка, 2018. 248 с.

20. Keith J., Sambells J. DOM scripting: Web design with Javascript and the document object model. Springer, 2010, 352 с.

21. Williams G. Laying the CSS3 Foundations. Learn HTML5 and JavaScript for Android. Berkeley, CA: Apress, 2012. С. 157–173.

22. Grant. K. J. CSS in Depth. Manning, 2018. 472 с.

23. Shukla A. Modern JavaScript Frameworks and JavaScript's Future as a Full-Stack Programming Language. Journal of Artificial Intelligence & Cloud Computing. 2023. № 144. С. 2–5.

24. Bierman G., Abadi M., Torgersen M. Understanding typescript. ECOOP 2014 Object-Oriented Programming: 28th European Conference, Uppsala, Sweden, July 28–August 1, 2014. Uppsala, 2014. С. 257–281.

25. O'Hanlon P. Advanced TypeScript Programming Projects: Build 9 Different Apps with TypeScript 3 and JavaScript Frameworks Such as Angular, React, and Vue. Packt Publishing, 2019. 415 с.

26. Macrae C. Vue. js: up and running: building accessible and performant web apps. O'Reilly Media, 2018. 171 с.

27. Hanchett E., Listwon B. Vue. js in Action. Manning, 2018. 375 с.

28. Танянський О., Руденко Д. Порівняльний аналіз популярних JavaScript-фреймворків та бібліотек для front-end розробки. Інформаційні системи та технології: матеріали міжнар. наук.-техн. конф., м. Харків, 10–15 вер. 2018 р. Харків, С. 347–349.

29. Проценко М. М. Аналіз фреймворків як засобів розробки web-додатків. Міжнародний науковий журнал "Інтернаука". 2016. Т. 1, № 6. С. 86–89.

30. Похваленна О. Д. Порівняльний аналіз технічних характеристик найпопулярніших фреймворків та бібліотек JS. Радіоелектроніка та молодь у XXI столітті : матеріали 27-го Міжнар. молодіж. форуму, 10–12 травня 2023 р. – Харків : ХНУРЕ, 2023. – Т. 6., № 1. – С. 313–314.

31. Bielak K., Borek B., Plechawska-Wójcik M. Web application performance analysis using Angular, React and Vue. js frameworks. Journal of Computer Sciences Institute. 2022. Т. 23, С. 77–83.

32. Kyriakidis A., Maniatis K., You E. The majesty of Vue. js. Packt Publishing, 2016. 240 с.

33. Kalliamvakou E. et al. The promises and perils of mining github. Proceedings of the 11th working conference on mining software repositories. – 2014. С. 92-101.

34. Pikkuhookana V., Soini J. Software development using cloud services. Oulu University of Applied Sciences, 2023. 60 с.

35. Kokate U., Shinohara K., Tigwell G. W. Exploring Accessibility Features and Plug-ins for Digital Prototyping Tools. The 24th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '22): Proceedings of the scient. conf., c. Athens, oct. 2022 y. Athens, 2022 С. 2–5.

36. Hoppe J. Adobe Illustrator. A complete course and compendium of features. Rocky Nook, 2020. 388 с.

37. Schwarz D. Jump Start Adobe XD. – SitePoint, 2017. 170 с.

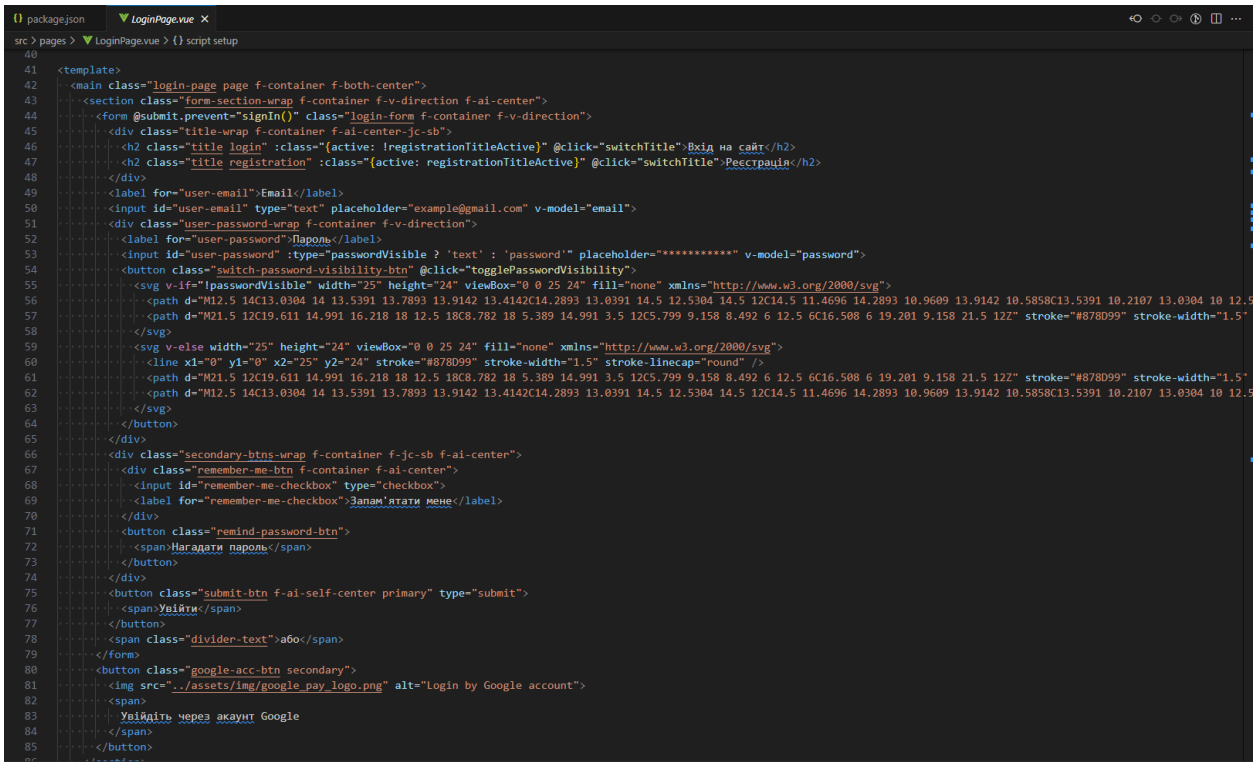
38. Staiano F. Designing and Prototyping Interfaces with Figma: Learn essential UX/UI design principles by creating interactive prototypes for mobile, tablet, and desktop. Packt Publishing Ltd, 2022. 382 с.

39. Mardan A., Practical Node.js: Building Real-World Scalable Web Apps. Apress, 2014. 300 с.

40. Пінчук А. В., Рудянова Т. М. Шляхи управління ризиками у процесі розробки програмного забезпечення проєкту. Інноваційні технології, моделі управління кібербезпекою ІТМК-2023: матеріали міжнар. наук. конф., м. Дніпро, 18–20 квіт. 2023 р. Дніпро, 2023. С. 16–18.

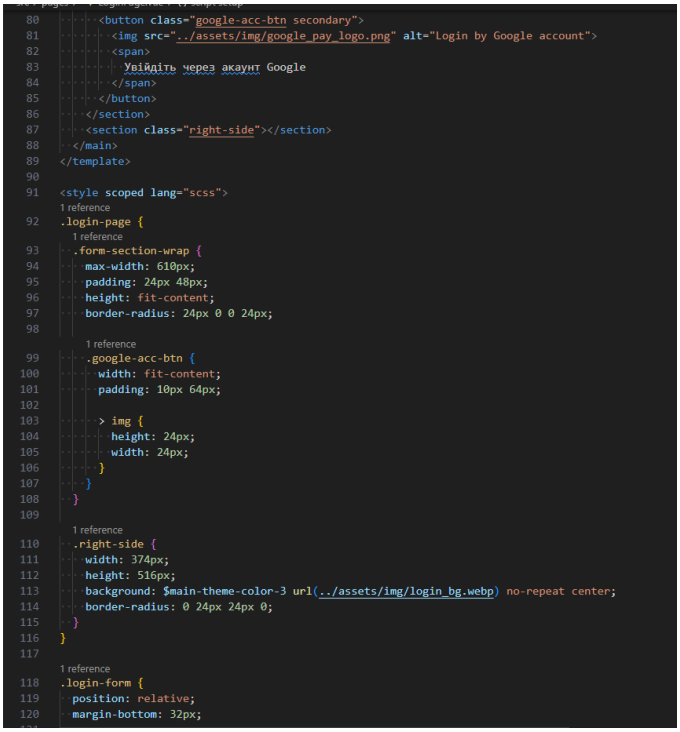
ДОДАТОК А

Лістинг коду прототипу навчального веб середовища для генерування завдань



```
<template>
  <main class="login-page page f-container f-both-center">
    <section class="form-section-wrap f-container f-v-direction f-ai-center">
      <form @submit.prevent="signIn()" class="login-form f-container f-v-direction">
        <div class="title-wrap f-container f-ai-center-jc-sb">
          <h2 class="title login" :class="{active: !registrationTitleActive}" @click="switchTitle">Вхід на сайт</h2>
          <h2 class="title registration" :class="{active: registrationTitleActive}" @click="switchTitle">Реєстрація</h2>
        </div>
        <label for="user-email">Email</label>
        <input id="user-email" type="text" placeholder="example@gmail.com" v-model="email">
        <div class="user-password-wrap f-container f-v-direction">
          <label for="user-password">Пароль</label>
          <input id="user-password" :type="passwordVisible ? 'text' : 'password'" placeholder="**********" v-model="password">
          <button class="switch-password-visibility-btn" @click="togglePasswordVisibility">
            <svg v-if="!passwordVisible" width="25" height="24" viewBox="0 0 25 24" fill="none" xmlns="http://www.w3.org/2000/svg">
              <path d="M12.5 14C13.0304 14 13.5391 13.7893 13.9142 13.4142C14.2893 13.0391 14.5 12.5304 14.5 12C14.5 11.4696 14.2893 10.9609 13.9142 10.5858C13.5391 10.2107 13.0304 10 12.5
              <path d="M21.5 12C19.611 14.991 16.218 18 12.5 18C8.782 18 5.389 14.991 3.5 12C5.799 9.158 8.492 6 12.5 6C16.508 6 19.201 9.158 21.5 12Z" stroke="#878D99" stroke-width="1.5"
            </svg>
            <svg v-else width="25" height="24" viewBox="0 0 25 24" fill="none" xmlns="http://www.w3.org/2000/svg">
              <line x1="0" y1="0" x2="25" y2="24" stroke="#878D99" stroke-width="1.5" stroke-linecap="round" />
              <path d="M21.5 12C19.611 14.991 16.218 18 12.5 18C8.782 18 5.389 14.991 3.5 12C5.799 9.158 8.492 6 12.5 6C16.508 6 19.201 9.158 21.5 12Z" stroke="#878D99" stroke-width="1.5"
              <path d="M12.5 14C13.0304 14 13.5391 13.7893 13.9142 13.4142C14.2893 13.0391 14.5 12.5304 14.5 12C14.5 11.4696 14.2893 10.9609 13.9142 10.5858C13.5391 10.2107 13.0304 10 12.5
            </svg>
          </button>
        </div>
        <div class="secondary-btns-wrap f-container f-jc-sb f-ai-center">
          <div class="remember-me-btn f-container f-ai-center">
            <input id="remember-me-checkbox" type="checkbox">
            <label for="remember-me-checkbox">Запам'ятати мене</label>
          </div>
          <button class="remind-password-btn">
            <span>Нагадати пароль</span>
          </button>
        </div>
        <button class="submit-btn f-ai-self-center primary" type="submit">
          <span>Увійти</span>
        </button>
        <span class="divider-text">або</span>
      </form>
      <button class="google-acc-btn secondary">
        <img src="../assets/img/google_pay_logo.png" alt="Login by Google account">
        <span>
          Увійдіть через акаунт Google
        </span>
      </button>
```

Рисунок А.1 – Лістинг коду сторінки веб-додатку



```
      <button class="google-acc-btn secondary">
        <img src="../assets/img/google_pay_logo.png" alt="Login by Google account">
        <span>
          Увійдіть через акаунт Google
        </span>
      </button>
    </section>
    <section class="right-side"></section>
  </main>
</template>

<style scoped lang="scss">
1 reference
.login-page {
  1 reference
  .form-section-wrap {
    max-width: 610px;
    padding: 24px 48px;
    height: fit-content;
    border-radius: 24px 0 0 24px;

    1 reference
    .google-acc-btn {
      width: fit-content;
      padding: 10px 64px;

      > img {
        height: 24px;
        width: 24px;
      }
    }
  }

  1 reference
  .right-side {
    width: 374px;
    height: 516px;
    background: $main-theme-color-3 url(../assets/img/login_bg.webp) no-repeat center;
    border-radius: 0 24px 24px 0;
  }
}

1 reference
.login-form {
  position: relative;
  margin-bottom: 32px;
```

Рисунок А.2 – Лістинг коду сторінки веб-додатку

ДОДАТОК Б

ВИСНОВОК

оригінальності тексту

Кваліфікаційна робота (наукова робота монографія, підручник, навчальний посібник, стаття, тези тощо) «Методи та технології створення навчального веб середовища для генерування завдань», що підготовлена Пінчуком Андрієм Васильовичем було перевірено за допомогою ліцензійного програмного забезпечення перевірки наукової та навчально-методичної літератури, що є в Університеті митної справи та фінансів «Система виявлення збігів/ідентичності/схожості (Система «Unplag»).

Текстових запозичень, які б вплинули на належний рівень представленої Пінчуком Андрієм Васильовичем кваліфікаційної роботи «Методи та технології створення навчального веб середовища для генерування завдань» не виявлено.

Дата

Відповідальна особа, яка здійснила перевірку на плагіат

ДОДАТОК В

Копії публікацій здобувача