

Вінницький національний технічний університет

Факультет інформаційних технологій та комп'ютерної інженерії

Кафедра програмного забезпечення

Бакалаврська дипломна робота

на тему: «Розробка соціальної мережі для онлайн-навчання користувачів»

Виконав: студент IV курсу
групи 2ПІ-18б
спеціальності

121 – Інженерія програмного забезпечення

(шифр і назва напрямку підготовки, спеціальності)

Маренко Дана Вадимівна

(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ Бабюк Н.П.

(прізвище та ініціали)

Рецензент: к.т.н., доц. каф. КН Крилик Л.В.

(прізвище та ініціали)

Допущено до захисту

Зав. кафедри _____

« ____ » _____ 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти перший бакалаврський
Галузь знань 12 – Інформаційні технології
Спеціальність 121 – Інженерія програмного забезпечення
Освітньо-професійна програма – Інженерія програмного забезпечення

УЗГОДЖЕНО

Директор ТОВ «ЮК КРИСТАЛ»

Довбеус І. Я.

«__» _____ 2022 року

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н.

«25» березня 2022 р.

З А В Д А Н Н Я НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Маренко Дані Вадимівні

1. Тема роботи – «Розробка соціальної мережі для онлайн-навчання користувачів».

Керівник роботи: Бабюк Наталя Петрівна, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від 24 березня 2022 року №66.

2. Строк подання студентом роботи 13 червня 2022 року.

3. Вихідні дані до роботи: середовища розробки – Visual Studio Code, IntelliJ IDEA; система управління базами даних – MySQL Workbench; мова запитів – SQL; мови розробки – html, css, js, java; операційна система – Windows 7/8/8.1/10.

4. Зміст розрахунково-пояснювальної записки: вступ; обґрунтування вибору методу розробки та постановка задач дослідження; розробка алгоритмів та структури веб-додатка; розробка модулів програми; тестування програми, висновки; список використаних джерел, додатки, графічна частина.

5. Перелік графічного матеріалу: титульний слайд – автор, науковий керівник бакалаврської роботи, тема; мета, предмет і об'єкт дослідження; актуальність розробки; аналіз аналогів; новизна; розробка дизайну макету веб-сайту; використані технології; ER-модель бази даних соціальної мережі; блок-схема алгоритму реєстрації облікового запису користувача; блок-схема алгоритму фільтрації тестів за тегами;

блок-схема алгоритму підрахунку результату тестування; тестування веб-додатка; апробація та публікації результатів роботи; впровадження; результати розробки, фінальний слайд.

6. Консультанти розділів бакалаврської дипломної роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1–4	к.т.н Бабюк Н. П., доц. кафедри ПЗ		

7. Дата видачі завдання 25 березня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування вибору методу розробки та постановка задач дослідження	26.03.2022 – 02.04.2022	Виконано
2	Розробка структури та алгоритмів програмного продукту	03.04.2022 – 17.04.2022	Виконано
3	Розробка модулів програмного продукту	18.04.2022 – 20.05.2022	Виконано
4	Тестування програми	21.05.2022 – 25.05.2022	Виконано
5	Оформлення матеріалів до захисту БДР	26.06.2022 – 10.06.2022	Виконано

Студент

(підпис)

Маренко Д. В.

(прізвище та ініціали)

Керівник бакалаврської дипломної роботи

(підпис)

Бабюк Н. П.

(прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота складається з 60 сторінок формату А4, на яких є 37 рисунків, 3 таблиці, список використаних джерел містить 16 найменувань.

У бакалаврській дипломній роботі було проведено детальний аналіз сфери онлайн-навчання. Встановлено об'єкт, предмет, завдання та методи дослідження. Сформульовано мету дослідження – створення веб-додатку для доступного навчання у будь-якій сфері для різних рівнів поглиблення. Проведено аналіз аналогів та доведено актуальність розробки. Система буде давати змогу користувачам отримувати та укріплювати знання завдяки спілкуванню між собою, поширенню корисних матеріалів та перевірці знань – онлайн-тестуванню, результати якого будуть закріплені у профілі кожного користувача.

Було розроблено алгоритми та блок-схеми для критично важливих операцій, а також розроблено структуру соціальної мережі.

Веб-додаток було написано на мовах програмування JavaScript, а також мові розмітки HTML, та каскадних стилях CSS. У якості середовища розробки було обрано Visual Studio Code. База даних була створена завдяки технологіям MySQL Workbench. Кінцевий програмний продукт відповідає завданням та меті. У підсумку було отримано веб-додаток, який представляє собою соціальну мережу для онлайн-навчання користувачів.

Отримані в бакалаврській дипломній роботі результати можна використати для освітнього процесу у форматі онлайн-навчання.

Ключові слова: соціальна мережа, онлайн-навчання, веб-додаток.

ANNOTATION

The bachelor's thesis consists of 60 A4 pages, which have 37 figures, 3 tables, the list of sources used contains 16 items.

In the bachelor's thesis, a detailed analysis was performed of the field of online-learning. The object, subject, tasks and methods of research are established. The aim of the study is creating a web application for accessible online-learning in any field for different levels of deepening. The analysis of analogues is carried out and the urgency of development is proved. The system will enable users to acquire and strengthen knowledge through communication with each other, dissemination of useful materials and checking of knowledge – online-testing, the results of which will be fixed in the profile of each user.

Algorithms, block diagrams, have been developed for critical operations, as well as a social network structure.

The web application was written in JavaScript, HTML markup languages, and cascading CSS styles. The development environment was chosen Visual Studio Code. The database was created by using MySQL Workbench technologies. The final software product meets the objectives and goals. As a result, we received a web application, which is a social network for online user learning.

Results obtained in the bachelor's thesis can be used for the educational process in the online learning format.

Keywords: social network, online learning, web application.

ЗМІСТ

ВСТУП.....	8
1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	11
1.1 Аналіз стану галузі онлайн-навчання	11
1.2 Порівняльний аналіз аналогів розроблюваного додатку	12
1.3 Аналіз методів розв’язання поставленої задачі	16
1.4 Постановка задач дослідження	18
1.5 Висновки	19
2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ	20
2.1 Аналіз принципів створення тестувань	20
2.2 Розробка структури інтерфейсу соціальної мережі	21
2.3 Розробка дизайну інтерфейсу соціальної мережі	24
2.4 Розробка алгоритму роботи додатку	28
2.5 Висновки	33
3 РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ПРОДУКТУ	34
3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу	34
3.2 Розробка бази даних соціальної мережі для онлайн-навчання	35
3.3 Розробка клієнтської частини соціальної мережі для онлайн-навчання.....	37
3.4 Розробка серверної частини соціальної мережі для онлайн-навчання.....	44
3.5 Висновки	46
4 ТЕСТУВАННЯ ПРОГРАМИ	47
4.1 Аналіз методів тестування програмного забезпечення.....	47
4.2 Тестування розробленого програмного продукту	48
4.3 Розробка інструкції користувача	51
4.4 Висновки	59

ВИСНОВКИ.....	60
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	61
ДОДАТКИ.....	63
Додаток А – Технічне завдання	64
Додаток Б – Протокол перевірки на плагіат....	Ошибка! Закладка не определена.
Додаток В – Акт впровадження	68
Додаток Г – Лістинг програми.....	69
Додаток Д – Графічна частина.....	97

ВСТУП

Обґрунтування вибору теми дослідження. На сьогоднішній день існує багато шляхів здобуття знань. Одним із найпопулярніших є інтернет-матеріали. Традиційно це онлайн-курси, навчальні та наукові статті. Зрозуміло, що курси – найпростіший спосіб отримати знання та застосовувати їх на практиці, тому що там уже підготовлений фахівцями матеріал, з яким необхідно лише ознайомитись та засвоїти, але є одне але – вони практично всі платні. Не кожен має можливість дозволити собі звичайні зібрання інформації, яку можна відшукати у вільному доступі, з тестами та сертифікатом за кілька тисяч гривень.

Звідси постає питання: «А як тоді бути?», – відповідь на це питання є і доволі проста: розробити мережу, у якій користувачі зможуть ділитися знаннями між собою, створювати та проходити тестування на різноманітні теми зі шкільної програми та інших освітніх закладів.

У наші часи стає більш загостреним питання українізації кожної сфери діяльності людини, звідси впливає ще одна проблема – чимало курсів, тестувань та занять подаються англійською, а частіше російською мовою. Аби сприяти збереженню української мовної культури з'являється потреба у суто українській платформі, яка потенційно зможе використовуватись під час навчального процесу.

Потреба у соціальній мережі обумовлена тим, що, по-перше, зберігається якісний зворотній зв'язок між користувачами: кожен може запитати, отримати відповідь та подискутувати на тему, яка цікавить найбільше, знайти людей, які добре розуміються у певних темах, та отримати знання не тільки завдяки тестуванням, а й завдяки один одному. По-друге, сучасну молодь більше приваблює, коли є змога до самовираження, а саме можливість обирати оформлення своєї сторінки та доповнювати її тим контентом, який цікавить найбільше.

Розробка соціальної мережі для онлайн-навчання користувачів є актуальною

на даний час, особливо через довгострокове дистанційне навчання, яке чимало розвинулося. Програмне забезпечення має особливості, яких не спостерігається на аналогічних веб-сайтах та які виводять продукт на новий рівень – балансоване поєднання навчання та домашнього дозвілля. Один з найголовніших та критичних факторів є повна підтримка української мови, тобто усі матеріали та тестування будуть українською мовою, що дозволяє використовувати соціальну мережу не тільки для саморозвитку, а й у навчальних підрозділах різної акредитації.

Зв'язок роботи з науковими програмами, планами, темами. Робота виконувалася згідно плану виконання наукових досліджень на кафедрі програмного забезпечення.

Мета і завдання дослідження. Метою дослідження є створення веб-додатку з елементами соціальної мережі для підвищення ефективності та доступності навчання у будь-якій сфері для різних рівнів поглиблення знань.

Задачі дослідження:

- аналіз стану галузі онлайн-навчання;
- порівняльний аналіз аналогів;
- аналіз методів розв'язання задачі;
- постановка задач для веб-додатку для онлайн-навчання;
- розробка структури і алгоритмів програмного продукту;
- розробка дизайну інтерфейсу соціальної мережі;
- варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу;
- розробка бази даних соціальної мережі;
- розробка соціальної мережі для онлайн-навчання користувачів;
- тестування програми.

Об'єкт дослідження – процес створення алгоритмів для соціальної мережі для онлайн-навчання.

Предмет дослідження – методи та засоби для реалізації соціальної мережі для онлайн-навчання користувачів.

Методи дослідження. Протягом дослідження було використано методи розробки клієнтського інтерфейсу; методи онлайн-навчання; методи створення бази даних; методи семантичної розмітки; методи завдання стилів за допомогою бібліотеки; методи оптимізації клієнтського інтерфейсу та веб-додатка в цілому.

Новизна отриманих результатів. В веб-додатку модифіковано ведення рейтингових списків користувачів і тестів, що відрізняється від досліджуваних аналогів відображенням прогресу в навчанні для покращення мотивації слухача курсів у формі соціальної мережі та освітнього порталу українською мовою.

Практична цінність одержаних результатів. Практична цінність одержаних результатів полягає в тому, що на основі отриманих в бакалаврській дипломній роботі теоретичних положень розроблено алгоритми та програмні засоби соціальної мережі для онлайн-навчання користувачів.

Впровадження. Впровадження результатів дослідження підтверджуються відповідним актом та використовуються на ТОВ «ЮК Кристал» для комунікації між структурними підрозділами та підвищення кваліфікації працівників.

Особистий внесок. Усі наукові результати отримано здобувачем самостійно. У друкованих працях, опублікованих у співавторстві, автору належать такі результати: порівняння засобів для онлайн-навчання користувачів [5]; розробка алгоритмів соціальної мережі для онлайн-навчання користувачів [9].

Апробація роботи матеріали бакалаврської кваліфікаційної роботи доповідались на всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)» (2022 р., м. Вінниця).

Публікації. За тематикою дослідження опубліковано 2 наукові праці у збірниках матеріалів конференції.

Технічне завдання наведено в додатку А.

1 ОБҐРУНТУВАННЯ ВИБОРУ МЕТОДУ РОЗРОБКИ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ

1.1 Аналіз стану галузі онлайн-навчання

Онлайн-навчання – це метод дистанційного навчання, яке реалізовується завдяки підключенню до мережі Інтернет через комп'ютер, смартфон або планшет. На сьогоднішній день чимало компаній, навчальних закладів та репетиторів частково впровадили онлайн-навчання у свої програми і це не тільки через ситуацію у світі. Формат онлайн-занять часто є більш комфортним та продуктивним для студентів та викладачів через свої можливості: динамічне оновлення усіх матеріалів, показ презентацій, інтерактивні методи навчання, легке демонстрування своїх наробітків і це не кінець списку.

Незважаючи на чинники, які впливають на розвиток онлайн-навчання, менш уживаним та потрібним воно точно не стане. Дана галузь тільки розвивається та разом із нею з'являється все більше методів подачі та отримання інформації.

Після блокування в 2017 році популярні на той час соціальні мережі, якими користувалася переважна більшість користувачів, постала проблема очікування кращого аналогу. Було чимало проєктів та веб-додатків, які мали на меті створити український аналог заблокованих ресурсів, але ні один не зміг вийти на ринок та поширитись серед користувачів. Головною причиною неуспішного виходу на ринок таких веб-додатків є недосконалість функціоналу, невдалий підбір дизайну сторінки та погану рекламну кампанію.

Допоки немає настільки популярного та відомого аналогу соціальних мереж в Україні, люди використовують такі месенджери для спілкування та обміном інформацією: Telegram, Viber, Whatsup. Під час дистанційного навчання чимало викладачів проводило консультацію, спілкування та поширення інформації серед школярів та студентів саме через вищевказані месенджери, але вони самі по собі не можуть задовольнити усі потреби онлайн-навчання. Інша частина викладачів

проводила зворотній зв'язок з учнями через електронні пошти та спеціально створені освітні портали, наприклад, JetIq. Але і там є свої недоліки: студент не завжди може завчасно отримати повідомлення про надходження будь-якої інформації. Сповіщення про лист від викладача на десктоп версії взагалі не прийде, що уповільнює та дезорієнтує процес навчання.

Також доцільним буде згадати про офіційно створені онлайн-школи, які мають свої програми навчання та приймають нових учнів до себе. Одними із таких є онлайн школи «Альтернатива», «ОПТИМА», «Alterra School» та інші. Недоліком таких онлайн-шкіл є їх грошова недоступність. У наші часи далеко не у всіх є можливість сплачувати за навчання, тому безкоштовність сервісу є одним з критерієм пошуку та аналізу аналогів.

Виходячи з усього вищесказаного можна зробити висновок, що на ринку досі немає сервісу для онлайн-навчання, який би представляв собою балансування дозвілля та плідного навчання, що дозволило б не перенавантажувати учнів та допомогло б створити у користувачів позитивну асоціацію з навчанням, аби була мотивація та бажання отримувати та перевіряти знання.

Отже, було вирішено розробляти соціальну мережу для онлайн-навчання, яка сприятиме розвитку однойменної галузі та допоможе багатьом учням та викладачам із користю проводити як освітні процеси, так і дозвілля.

1.2 Порівняльний аналіз аналогів розроблюваного додатку

Для визначення конкурентоспроможності та актуальності розробки потрібно провести порівняльний аналіз аналогів. Підібрано критерії оцінювання згідно до мети розробки.

Потужний та стрімкий розвиток онлайн-навчання почався лише недавно, тому існує не дуже багато аналогів, які вміщували у собі функції соціальної мережі та онлайн-навчання, тому для аналізу підібрано веб-додатки, які у різних аспектах задовольняють тему розробки.

Stepik – російська освітня платформа та конструктор безкоштовних та платних відкритих онлайн-курсів та уроків [1]. Здебільшого на платформі такі дисципліни: програмування, математика, біоінформатика та біологія, економіка; головна мова курсів – російська, є курси англійською мовою. Цільові аудиторії – підлітки (більшість курсів з підготовки до вступних іспитів), студенти, молоді спеціалісти (див. рис. 1.1).

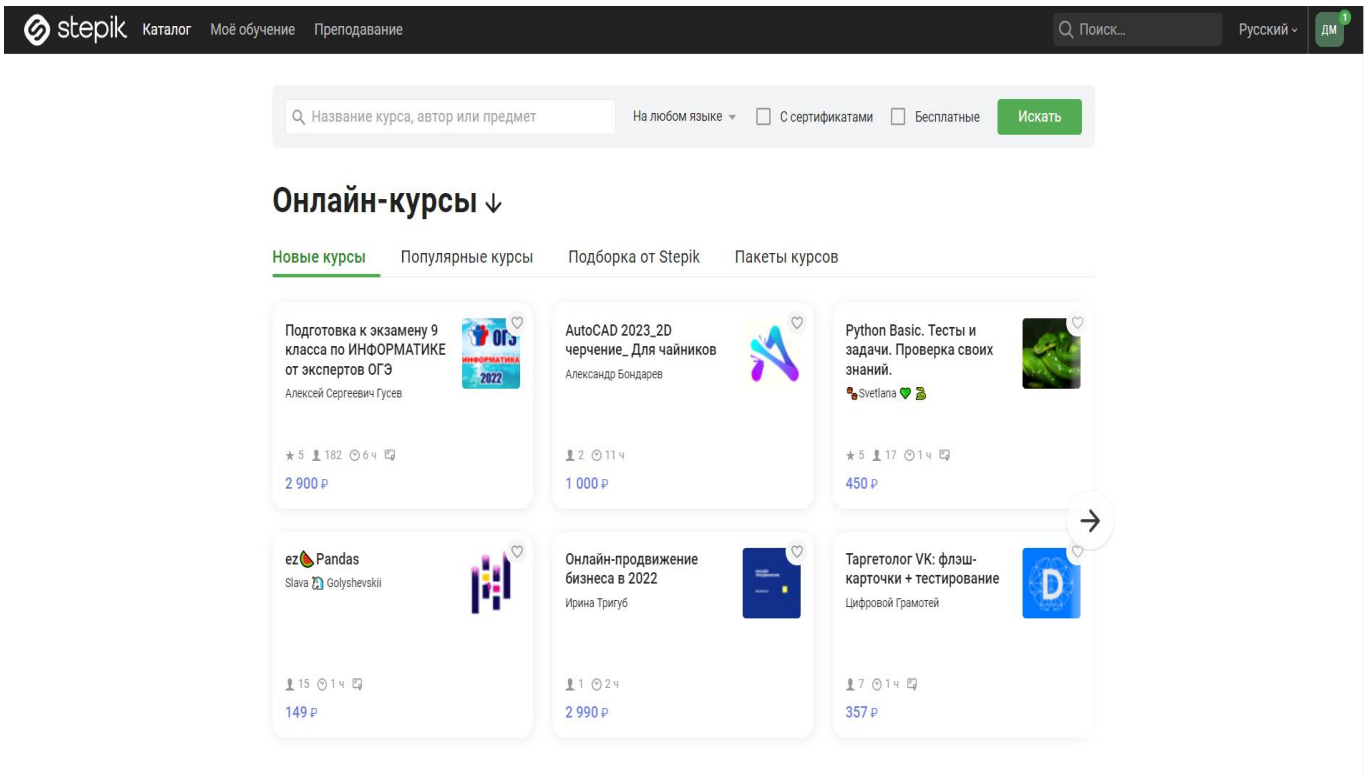


Рисунок 1.1 – Веб-сторінка освітньої платформи Stepik

Coursera – проект, який співпрацює з університетами, що публікують та ведуть у системі курси з різних галузей знань. Слухачі проходять курси, спілкуються з однокурсниками, складають тести та іспити безпосередньо на сайті Coursera [2]. Переважна кількість курсів англійською мовою, однак це скарбниця будь-якого навчального матеріалу, оскільки інтернет-сайт популярний на цілий світ та співпрацює з багатьма провідними компаніями. (див. рис. 1.2).

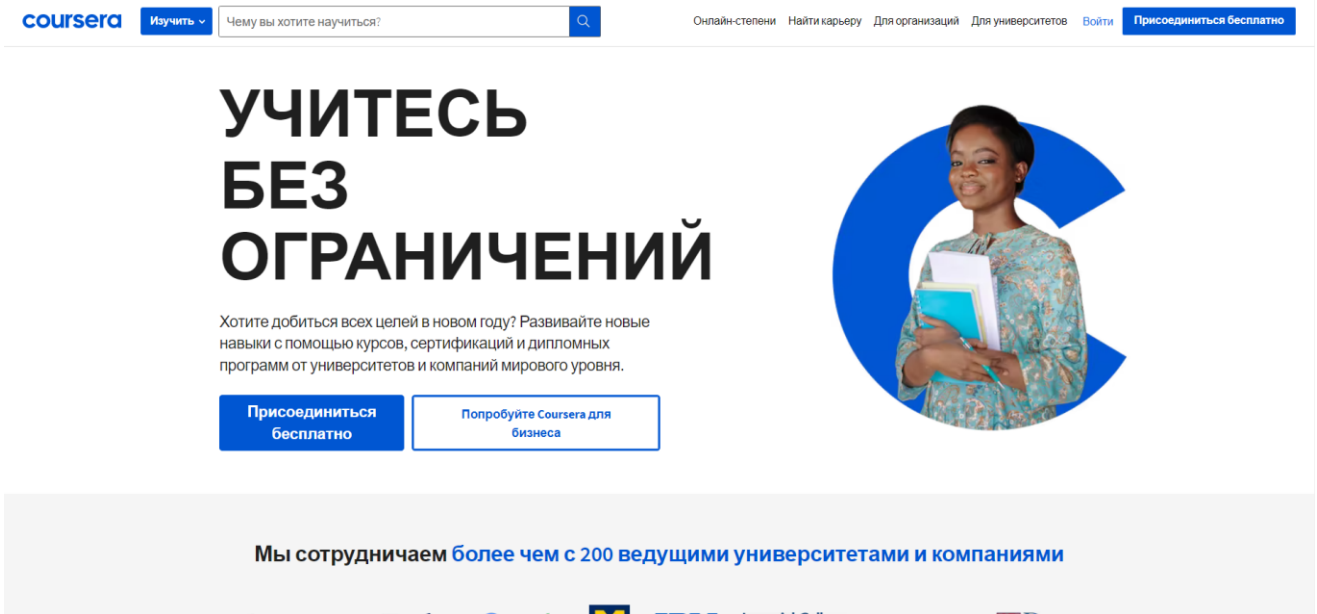


Рисунок 1.2 – Веб-сторінка освітнього проекту Coursera

Udemy – платформа з можливістю проходити курси та тестування [3]. Udemy пропонує пропозицію сотні онлайн-курсів рівно як для розваги, так і з метою розвитку висококласних професійних навичок, які допоможуть налагоджувати кар'єру. Недоліком вважається якість певних курсів, яка залишає бажати кращого, тому що будь-хто може викласти матеріал. (див. рис. 1.3).

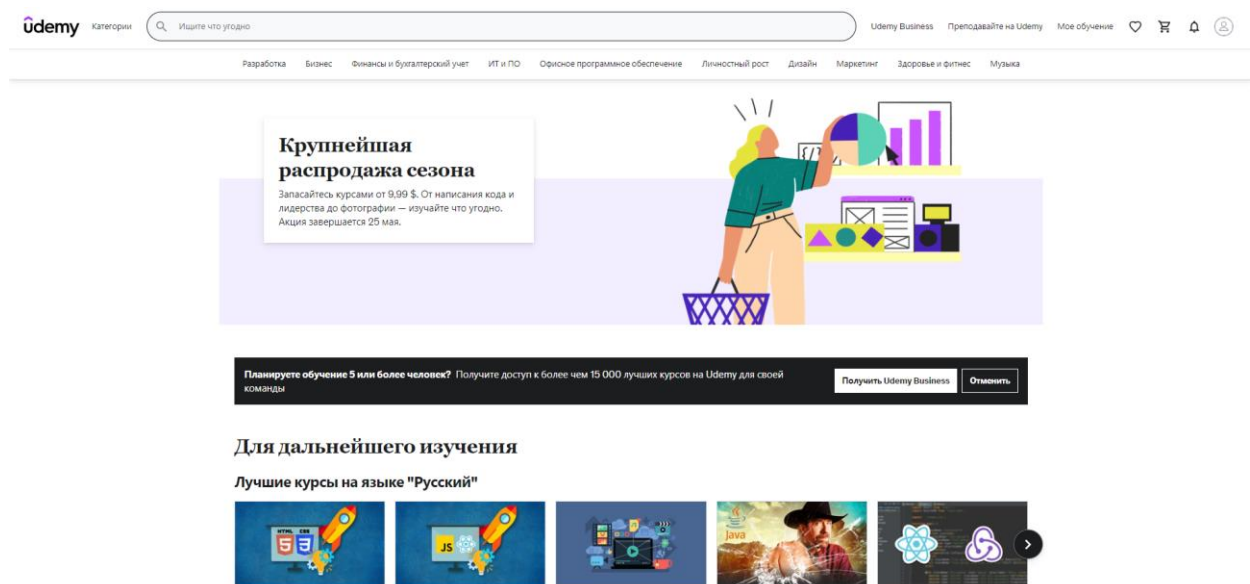


Рисунок 1.3 – Веб-сторінка освітньої платформи Udemy

Jamm School – дистанційна школа в онлайн-форматі, яка підготовлює учнів 5-11 класів та має свою освітню програму [4]. Основним недоліком є те, що вона підійде далеко не всім, а лише учням середньої та старшої шкіл (див. рис. 1.4).

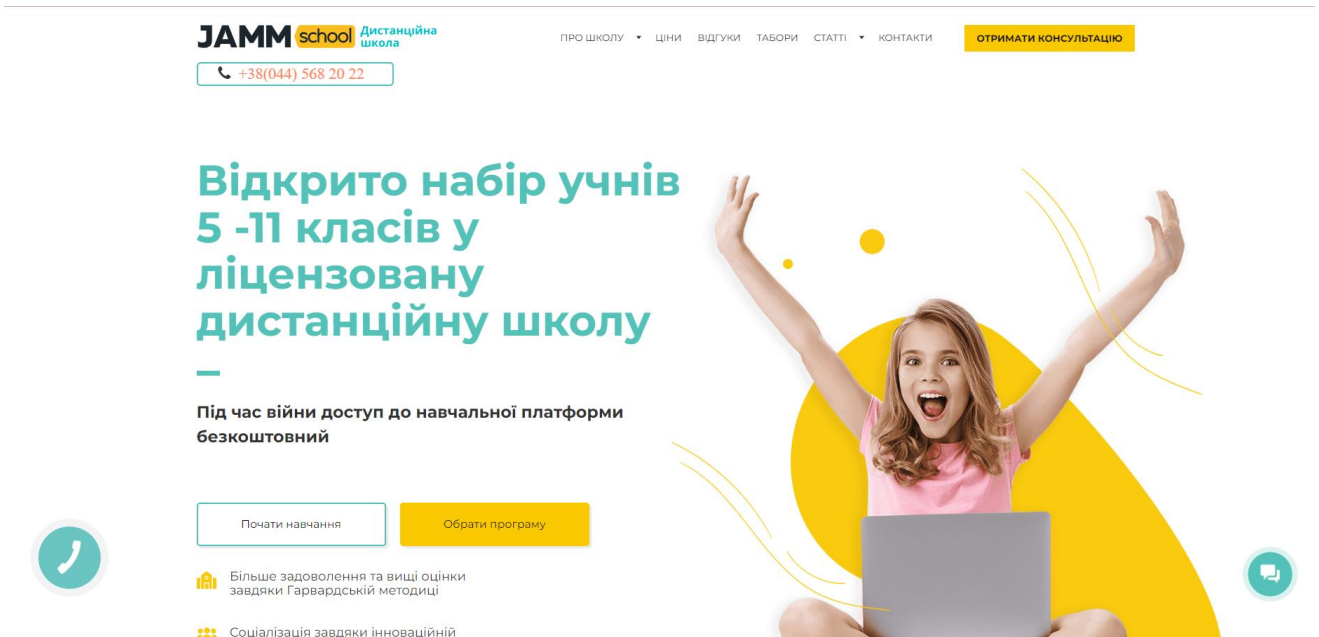


Рисунок 1.4 – Веб-сторінка онлайн-школи Jamm School

Результати аналізу наведено у таблиці 1.1.

Таблиця 1.1 – Порівняння аналогів

Критерії оцінювання	Stepik	Coursera	Udemy	Jamm School	UAlearning
Можливість створювати тестування	0	0	1	0	1
Можливість спілкуватися з іншими користувачами	0	0	0	1	1
Підтримується українська мова	0	0	0	1	1
Безкоштовно	0.5	0.5	0	0.5	1
Власний акаунт	1	1	1	0	1

Продовження таблиці 1.1

Рейтинг користувачів	1	0	0	1	1
Підійде для будь-якої аудиторії	1	1	1	0	1
Підсумковий результат	3.5	2.5	3	3.5	7

Виходячи з результатів аналізу аналогів, можна зробити висновок, що соціальна мережа для онлайн-навчання користувачів є актуальною на сьогоднішній день та має переваги, які роблять веб-сайт конкурентоспроможним. Також завдяки підтримці української мови, можливість спілкуватися та створювати тестування соціальну мережу потенційно можна використовувати під час навчального процесу, оскільки є можливість підтримувати зв'язок з іншими користувачами [5].

Також треба враховувати, що соціальна мережа повинна задовольняти потреби широкої аудиторії, тобто будь-хто зможе нею користуватися.

Отже, розробка соціальної мережі для онлайн-навчання користувачів є актуальною на даний час, особливо через довгострокове дистанційне навчання, яке чимало розвинулося за таких коротких проміжків часу. Програмне забезпечення має особливості, яких не спостерігається на аналогічних веб-сайтах та які виводять продукт на новий рівень – балансоване поєднання навчання та домашнього дозвілля. Один з найголовніших та критичних факторів є повна підтримка української мови, тобто усі тестування будуть українською мовою, що дозволяє використовувати соціальну мережу не тільки для саморозвитку, а й у навчальних підрозділах різної акредитації [5].

1.3 Аналіз методів розв'язання поставленої задачі

Аналіз методів розв'язання задачі є невід'ємною частиною розробки,

оскільки потрібно підібрати такий метод розробки, який підійде у конкретному випадку. Обрання методу розробки впливає на всю подальшу роботу, тому потрібно найточніше та найуважніше поставитися до цієї задачі.

Існують такі методи створення веб-сайту [6]:

- конструктори;
- фреймворки;
- використання CMS;
- програмування.

Створення веб-додатків за допомогою конструкторів полягає в доборі шаблону, порекомендованого конструктором, і підстроюванні його під власні потреби. Цей спосіб створення веб-сайту є найлегшим, оскільки він не вимагає від розробників знань у сфері діяльності веб-програмування. Таким чином, він бездоганно підходить для початківців, дозволяючи зберегти час та сили. Але цей спосіб зовсім не підходить для масштабування та підтримки сайту в майбутньому.

Наступний спосіб формування веб-сайту є розробкою за допомогою фреймворків. Вони здебільшого безкоштовні, і в той же момент у них є великі потенціали, які дають можливість створювати на їх базі будь-яких сторінок з неповторними можливостями та дизайном. З мінусів, крім знань мов веб-програмування та програм для створення сайтів, для успішної роботи з фреймворком буде необхідно витратити багато часу на оволодіння його структурою, а також дослідження функціоналу та доповнень до нього..

Третій спосіб являє собою розробку сайту за допомогою CMS – це зразковий інструментарій для тих, хто бажає розробити сайт своїми руками і не має при цьому великих знань веб-програмування. Будь-який CMS вважається програмним тісно спеціалізованим майданчиком для створення веб-проектів. За функціональними можливостями він краще будь-якого конструктора, але на відміну від інших способів є найбільш дорогим.

Четвертий і завершальний метод полягає у програмуванні «з нуля» і дає

можливість юзеру, по-перше, не збитися в написаному коді, і, по-друге, самостійно робити вибір, яким буде створюваний сайт. Не потрібно нічого під себе перетворювати – можна одразу писати власний незалежний додаток. Завдяки методу програмування значно легше підтримувати та масштабувати веб-додаток, немає залежності від сторонніх ресурсів і нічого не зіпсує успіх розробки.

Отже, розглянувши та проаналізувавши методи створення веб-додатків, було зроблено висновок, що для розв'язання поставленої задачі потрібно використати саме метод програмування «з нуля», який дозволяє взяти розробку у свої руки та не залежить від сторонніх ресурсів.

1.4 Постановка задач дослідження

Проаналізувавши стан галузі онлайн-навчання та дослідивши методи розв'язання теми «Розробка соціальної мережі для онлайн-навчання користувачів», було виділено задачі, які забезпечать якість та працездатність веб-модулів:

- визначити найбільш ефективний підхід до перевірки знань в онлайн просторі;
- розробити графічний інтерфейс користувача для взаємодії із різними модулями у соціальній мережі;
- розробити макет дизайну сторінок соціальної мережі;
- розробити базу даних, яка буде містити інформацію про користувачів та тести;
- розробити модуль авторизації та реєстрації користувача, а також отримання результатів тестування;
- здійснити тестування веб-додатку.

Отже, було поставлено задачі дослідження на тему «Розробка соціальної мережі для онлайн-навчання користувачів».

1.5 Висновки

У першому розділі було проаналізовано сучасний стан галузі онлайн-навчання.

Провівши порівняльний аналіз аналогів, було зроблено висновок, що дана розробка є актуальною та у повній мірі задовольнить потребу як у навчанні, так і у дозвіллі.

Обрано метод самостійної розробки «з нуля» веб-додатку.

Було визначено задачі дослідження, які потрібно виконати для успішної розробки на тему «Розробка соціальної мережі для онлайн-навчання користувачів».

2 РОЗРОБКА СТРУКТУРИ ТА АЛГОРИТМІВ ПРОГРАМНОГО ПРОДУКТУ

2.1 Аналіз принципів створення тестувань

Основний спосіб перевірки та закріплення знань у соціальних мережах – тестування різноманітних тем і виставлення їм балів. Звичайні користувачі також можуть створювати нові тести, щоб розширити каталог і дозволити їм внести свій внесок у процес навчання інших користувачів.

Найважливішим для тесту є правильна назва, оскільки вона відображає основну логіку тесту та розкриває його тему. Оскільки книга оцінюється за обкладинкою, тест буде оцінюватися за її назвою. Коротка, лаконічна назва — початок майбутнього успіху тесту. Якщо це не спрацює, то потрібно більше подумати над темою тестування, чи не надто вона складна? Крім того, найкращий спосіб зацікавити користувачів складанням тесту – це поставити запитання в заголовку, щоб вони хотіли знати відповідь і були мотивовані пройти тест.

Необхідно визначити, для чого створено той чи інший тест: щоб донести якісь знання іншим користувачам або досягти якоїсь мети. Все залежить від деталей тесту та його логіки.

Далі потрібно визначити, що найважливіше в тесті – результати чи запитання. Оскільки соціальна мережа створена для онлайн-навчання, то, звичайно, спочатку потрібно задавати питання і приділяти їм більше уваги, щоб отримана інформація могла зберігатися в свідомості користувача. Звичайно, це не означає, що неможливо створити тести, орієнтовані на результати, і кожен може сам вирішити, що в його тесті є найважливішим.

Ще одне негласне правило створення тестів – робити якомога менше запитань, підбирати їх так, аби висвітлити логіку та проблематику заданої теми, але також не навантажувати користувача, який під кінець тестування може задуматися, а про що взагалі цей тест? Ідеальна кількість запитань для детальної перевірки знань з певної теми – 10-15, але, звісно, якщо є змога укоротити

тестування, потрібно це робити. Потрібно намагатися дотримуватися балансу: якщо питань багато, то нехай і питання, і варіанти відповідей будуть максимально короткими.

Під кінець головне, щоб результати відповідали назві тесту. У гіршому випадку, пройшовши тест, користувач лише розчарується та більше не захоче проходити тестування. Звичайно, це також може послужили додатковою мотивацією для створення власного тесту, оскільки інші не задовольняють користувача, але це лише часткові випадки.

Також потрібно максимально толерантно відноситись до будь-яких результатів при їх описі та розуміти, що у кожного свої переживання та різний ступінь емоційності. Результати повинні мотивувати користувача пройти тестування ще краще, а не закинути та ніколи більше не зайти на соціальну мережу.

Отже, було проаналізовано основні принципи створення тестів для соціальної мережі для онлайн-навчання користувачів.

2.2 Розробка структури інтерфейсу соціальної мережі

Розробка структури інтерфейсу додатку є одним з найважливіших етапів розробки інтерфейсу користувача. Саме на цьому етапі потрібно зрозуміти, що потрібно користувачеві, грамотно та лаконічно розташувати ті або інші елементи веб-сторінки.

Основна мета етапу логічного проектування полягає в тому, щоб упорядкувати розділи сайту, привести їх до зручного й зрозумілого вигляду. Це включає організацію інформації на сайті, побудову його структури та навігації по розділах таким чином, щоб користувачі сайту могли легко та швидко знайти необхідну інформацію [7].

Для грамотного проектування структури інтерфейсу соціальної мережі потрібно знати стандарти юзабіліті.

Одним з основних правил дизайну інтерфейсу є аспект зручності використання. Перше правило полягає в тому, що інтерфейси завжди створюються для певної групи людей, що характеризуються їхніми характеристиками, знаннями, навичками, очікуваннями, сильними та слабкими сторонами. Система ніколи не буде створена для пересічного користувача, оскільки майже жоден користувач не зможе використовувати сайт, крім як задовольнити більшість користувачів.

Не можна забувати, що рішення, яке ідеально підходить для користувача в одній ситуації, може бути абсолютно неприйнятним в іншій, і немає цілком загальної «функції», яка працює скрізь. Наприклад, голосове керування може бути чудовим рішенням для операторів ізольованих кімнат, але працювати з кількома операторами в одній кімнаті буде складно.

У свою чергу, якщо сайт не може допомогти досягти юзеру своїх цілей, отже, рішення щодо проектування програмного забезпечення були безуспішними. Для того, щоб програма могла допомогти їх досягти, потрібно з'ясувати, з чого складаються ці цілі, провести дослідження можливої аудиторії та спроектувати сайт заново.

Звісно, універсальних порад щодо розробки структури сторінки немає, але є декілька загальноприйнятих нюансів:

- існує декілька констант у проектуванні. Наприклад, те що логотип розміщується зліва зверху, а стрічка пошуку справа зверху;

- елементи для взаємодії з користувачем, кнопки, повинні виділятися, щоб одразу було зрозуміло, що робити на сторінці;

- найголовніші елементи, які представляють ідею та ціль веб-сайту, повинні бути на «першому плані», щоб користувач, зайшовши на сторінку, одразу знайшов те, за чим прийшов.

Розроблено структуру головної сторінки соціальної мережі для онлайн-навчання користувачів (див. рис. 2.1).

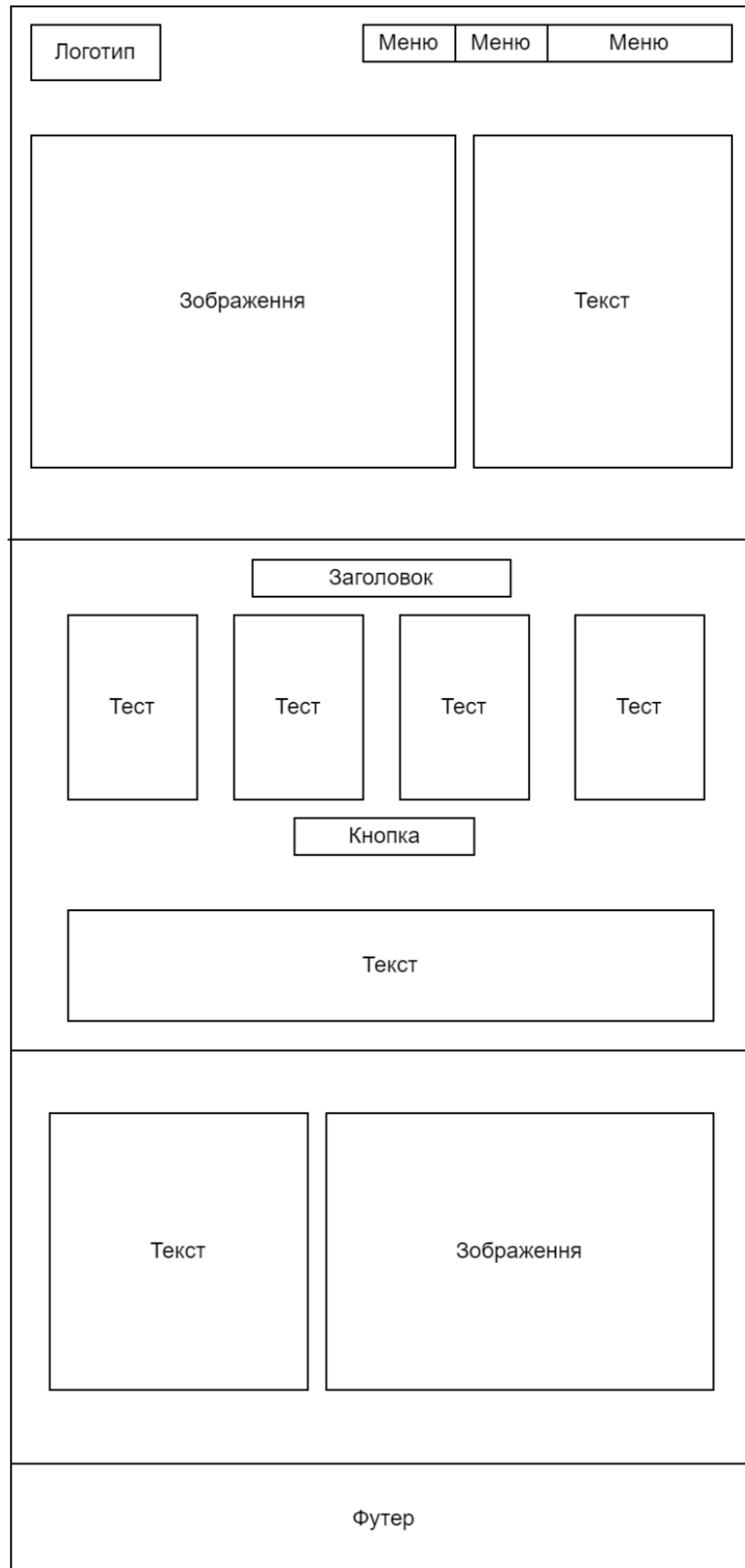


Рисунок 2.1 – Структура інтерфейсу головної сторінки

Дана структура соціальної мережі дозволяє користувачеві одразу зрозуміти, про що цей веб-сайт, швидко авторизуватися та почати користуватися послугами сторінки – тестуваннями.

Отже, було розроблено структуру інтерфейсу головної сторінки соціальної мережі для онлайн-навчання користувачів.

2.3 Розробка дизайну інтерфейсу соціальної мережі

Практичний і привабливий інтерфейс – висока ознака якості інтернет-сайту. Елементи інтерфейсу повинні бути закономірно структуровані та взаємопов'язані. Проектування інтерфейсу базується на законах візуального дизайну, іконографіки та друкарні. Використовувати інноваційні закономірності інформаційної архітектури та основні аксіоми поведінкової психології людей.

Візуальне оформлення обличчя сайту, його сприйняття, графічний інтерфейс – це перше, з чим стикається користувач. Ще до оцінки зручності сайту користувачі бачать, як він виглядає. Розробка візуального оформлення будь-якого сайту – це творчий процес. На цій стадії створюються всі елементи дизайну сайту відповідно до стилю подачі інформації і загальної концепції, сформульованої у креативному брифі проекту. Після цього дизайн сайту затверджується, доводиться до досконалості та тестується [7].

Головне завдання веб-інтерфейсів – зробити максимально зручним процес серфінгу по сайту та максимально спростити життя користувачеві. При розробці інтерфейсів бажано дотримуватися кількох правил [8]:

- у структурі не видумувати надто оригінальних ідей;
- розуміти цілі та поведінкові звички користувачів;
- не вирішувати інтерфейсом сторонніх завдань;
- не ускладнювати надмірно логіку;
- інтерфейс має бути виконаний в єдиному стилі.

Для розробки інтерфейсів є чимало сервісів та програм, одними з яких є Sketch та Figma.

Figma – онлайн-сервіс для розробки інтерфейсів та прообразів з перспективою організації колективної роботи в режимі реального часу. Його можна запустити як у веб-браузері, так і як додаток на робочому столі.

Плюси:

- усі розробки зберігаються у хмарі;
- кросплатформленість;
- можливість одночасно працювати над проектом декількома людьми;
- історія змін в реальному часі та резервні копії.

Недоліки:

- є платна підписка;
- необхідне підключення до Інтернет-мережі.

Sketch – векторний графічний редактор. Використовується для проектування інтерфейсів мобільних програм та веб-сайтів.

Плюси:

- можливість експортувати графічні зображення;
- є інструмент «піпетка».

Недоліки:

- лише для MacOS;
- недосконалі векторні інструменти.

Усі макети дизайну інтерфейсу було вирішено розробляти у сервісі для розробки інтерфейсів та прототипування Figma.

Було прийнято рішення використати у якості кольорової гами відтінки чорного та жовтого кольорів, які у поєднанні виглядають незвичайно та гармонічно.

Розроблено макет дизайну інтерфейсу головної сторінки (див. рис. 2.2).

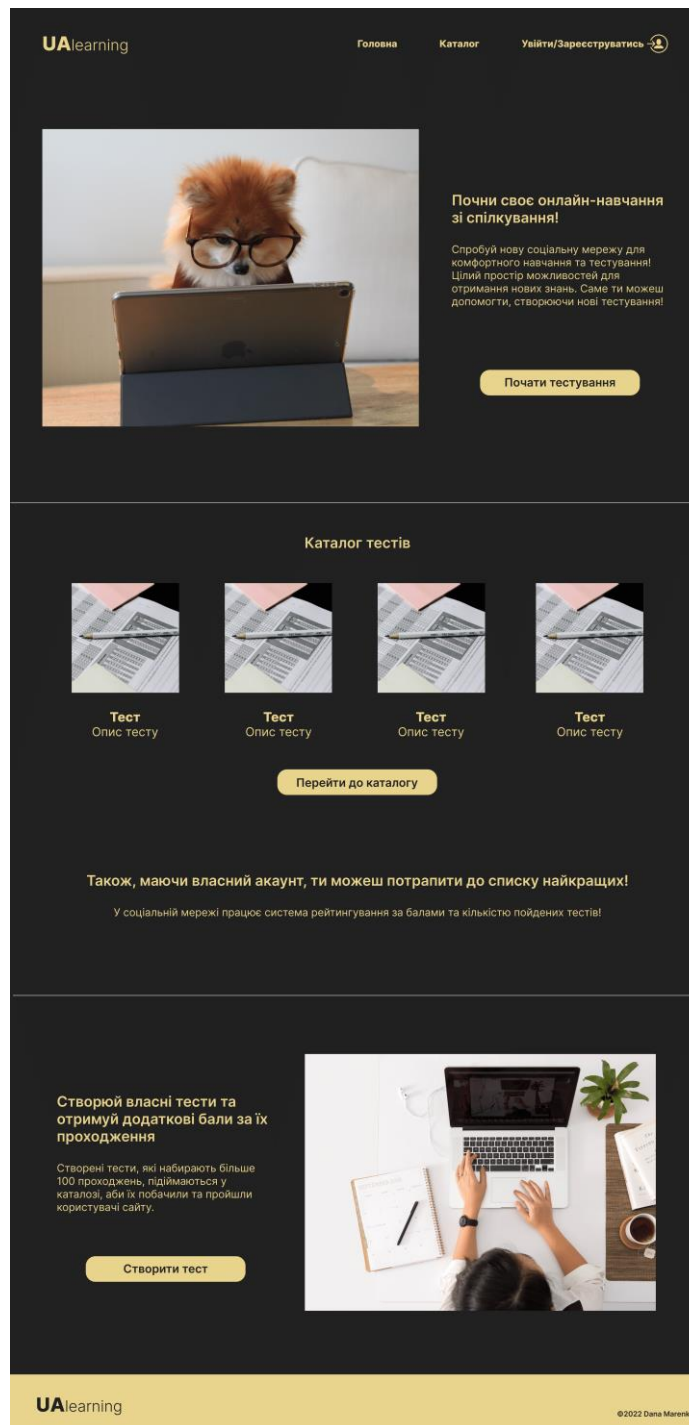


Рисунок 2.2 – Макет дизайну інтерфейсу головної сторінки

Можна побачити, що усі потрібні для користувача можливості коротко та лаконічно розписані на головній сторінці із можливістю приступити до користування веб-сайтом.

Макет дизайну інтерфейсу сторінки профілю показано на рисунку 2.3.

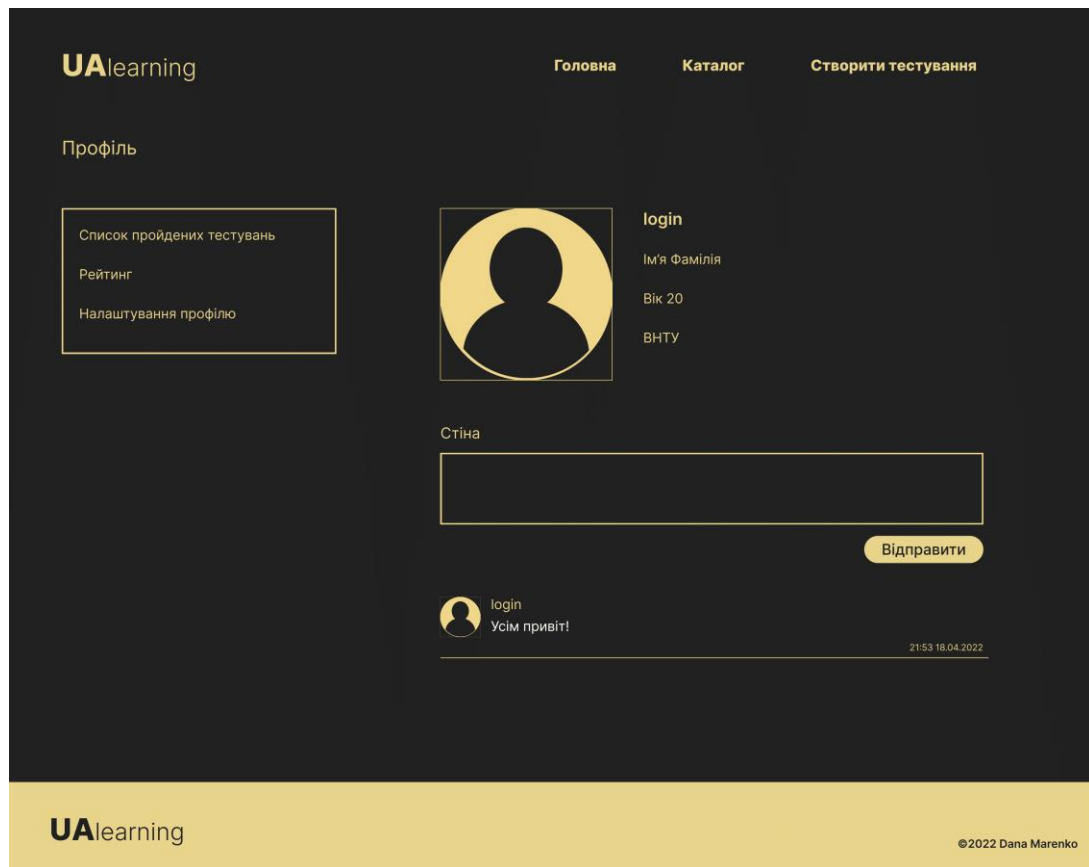


Рисунок 2.3 – Макет дизайну інтерфейсу сторінки профілю

Розроблений макет дизайну інтерфейсу сторінки профілю представляє собою одну сторінку із блоком панелі інструментів зліва, з якої можна перейти до таких сторінок: список пройдених тестувань, рейтинг, налаштування профілю.

Також після авторизації користувачу доступне меню «Створити тестування». Натискаючи цю кнопку, користувач переходить до сторінки створення нового тесту, де, заповнивши всі обов'язкові поля, можна розробити новий тест, який зможе пройти кожен користувач, який матиме бажання.

Згадавши поля для введення, переходимо до розробки макету сторінки реєстрації нового користувача соціальної мережі.

Розроблений макет дизайну інтерфейсу сторінки реєстрації представлено на рисунку 2.4.

UAlearning

Логін

Ім'я
Ім'я П'ростянка

Вік

e-mail

Пароль

Повторіть пароль

Створити обліковий запис

UAlearning

©2022 Dana Marenko

Рисунок 2.4 – Макет дизайну інтерфейсу сторінки реєстрації

У формі реєстрації присутні лише критично важливі поля для заповнення анкети користувача.

Отже, було розроблено макет дизайну інтерфейсу сторінок соціальної мережі для онлайн-навчання користувачів.

2.4 Розробка алгоритму роботи додатку

Було розроблено декілька алгоритмів. Розглянемо основні із них.

Алгоритм реєстрації нового користувача представляє собою перевірку валідності введених даних перед створенням нового акаунту. Перевіряються такі

поля: поле логіну перевіряється на наявність таких же записів у базі даних; поле паролю перевіряється, використовуючи регулярні вирази; поле віку перевіряється на те, чи є користувачу більше 12 років, щоб користуватися соціальною мережею [9].

Блок-схема алгоритму реєстрації нового користувача представлено на рисунку 2.5.

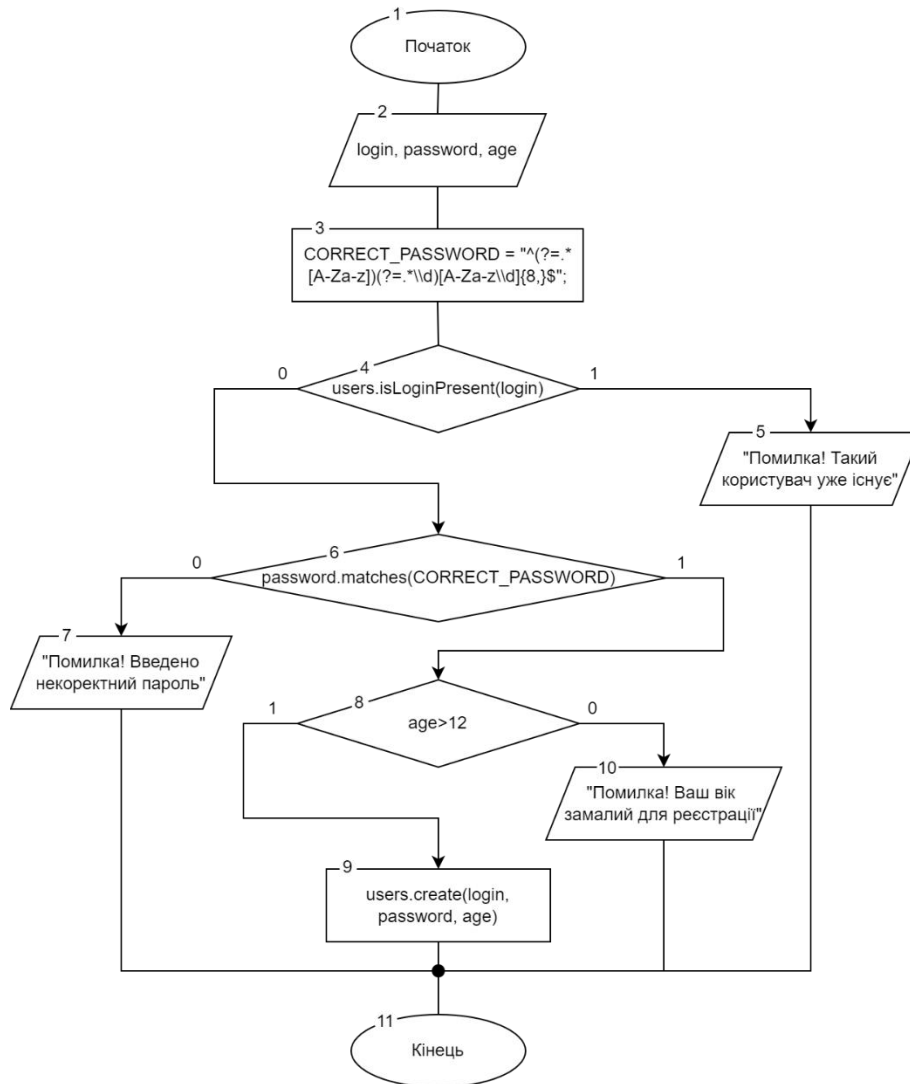


Рисунок 2.5 – Блок-схема алгоритму реєстрації нового користувача

Реєстрація користувачів представляє собою один із багатьох модулів соціальної мережі, що розробляється [9].

Розглянемо наступний алгоритм – створення результату проходження тесту. Зберігає результат та оновлює рейтинг користувача, базуючись на оцінці, яка була отримана в результаті проходження тестування.

Блок-схема алгоритму створення результату проходження тесту показана на рисунку 2.6.

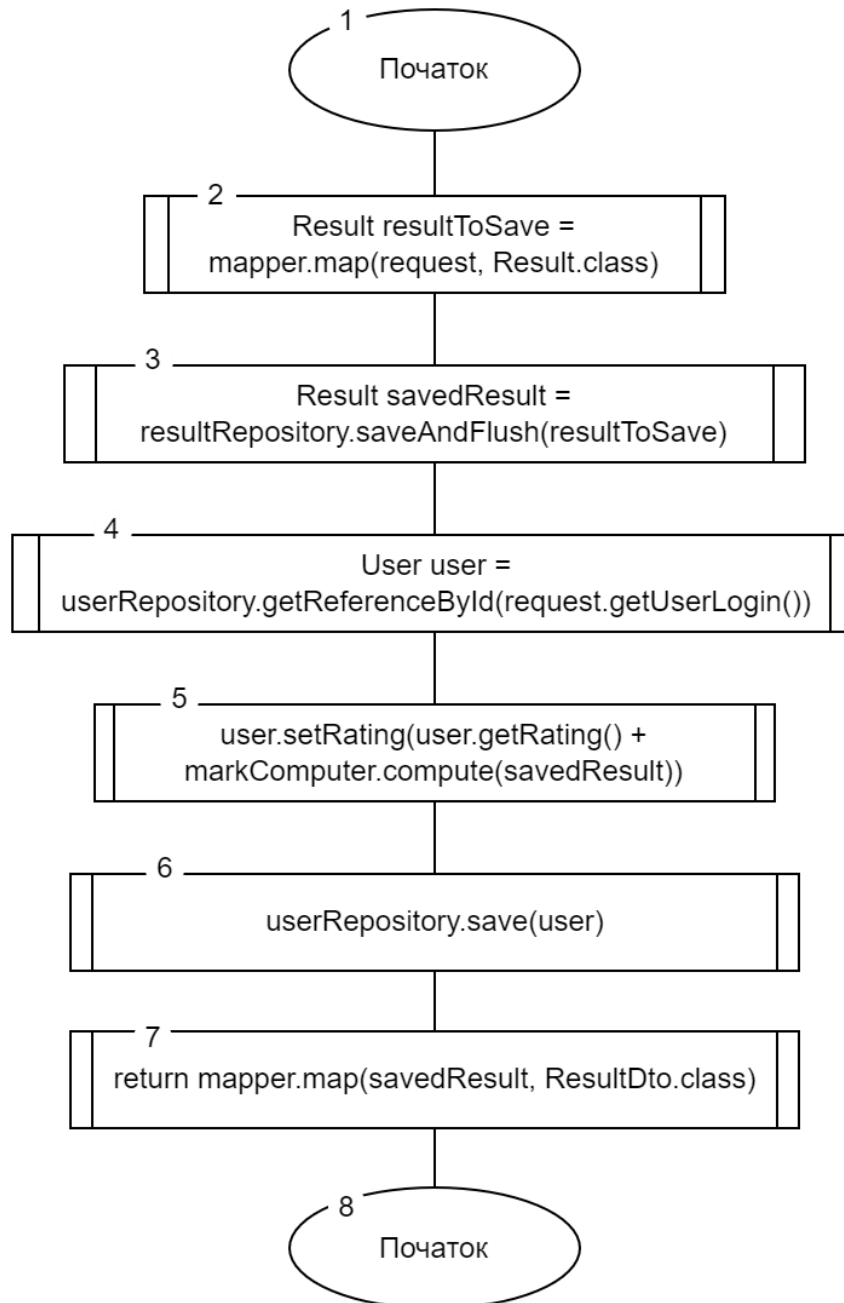


Рисунок 2.6 – Блок-схема алгоритму створення результату проходження тесту

Розглянемо наступний алгоритм – обрахування результату проходження тестування користувачем. Він представляє собою обробку результатів за таким принципом: оцінка дорівнює поділенню кількості правильних відповідей користувача, помноженого на сто, на загальну кількість правильних відповідей. Спочатку відбувається пошук відповідей на запитання тесту, далі – загальної кількості правильних відповідей, а потім іде пошук кількості правильних відповідей користувача. У кінці алгоритму проводиться виведення отриманих результатів.

Блок-схема алгоритму підрахунку результату проходження тестування користувачем показана на рисунку 2.7.

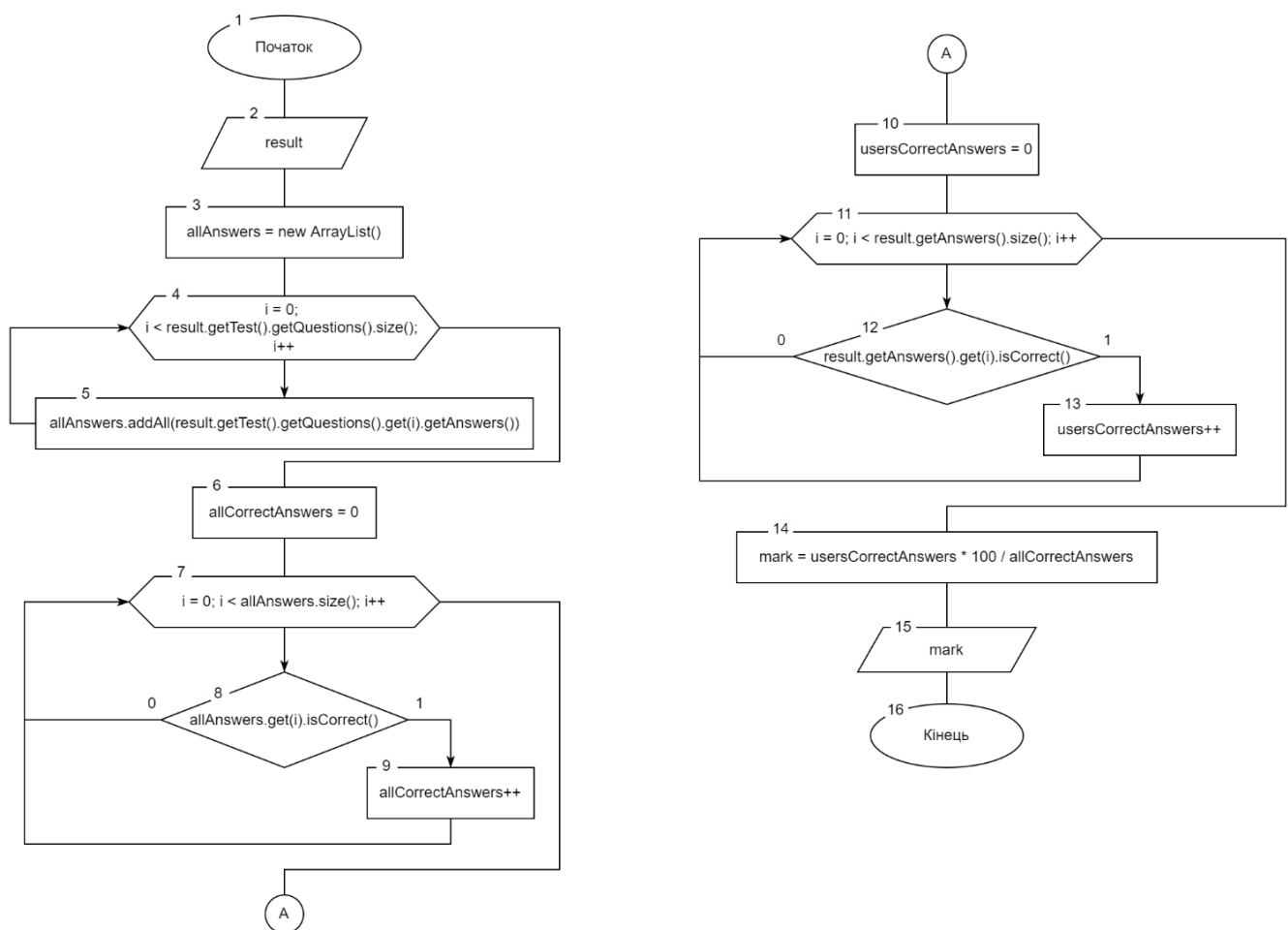


Рисунок 2.7 – Блок-схема алгоритму підрахунку результату тестування

Алгоритм фільтрації тестів за тегами є наступним алгоритмом на розглядання. На вхід отримуємо тег, за яким буде відбуватися фільтрація. Отримуємо тести, створюємо масив тестів, які будуть мати потрібний тег та повернені користувачеві. У циклі перевіряємо, чи мають тести отриманий тег, якщо так – додаємо у масив тестів на повернення, якщо ні – нічого не робим. Після закінчення циклу повертаємо масив тестів із заданим тегом.

Блок-схема алгоритму фільтрації тестів за тегами показана на рисунку 2.8.

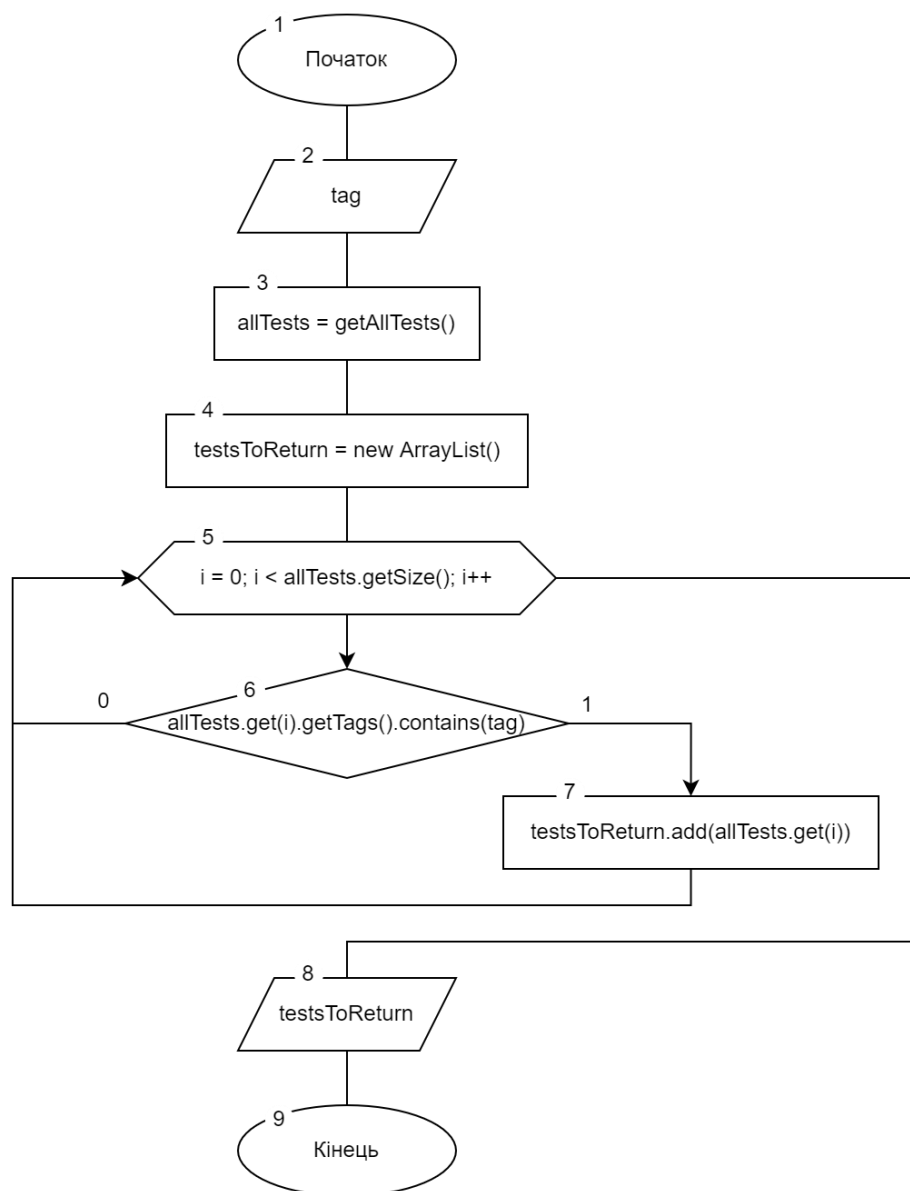


Рисунок 2.8 – Блок-схема алгоритму фільтрації тестів за тегами

Отже, було розроблено алгоритми роботи модулів соціальної мережі для онлайн-навчання користувачів.

2.5 Висновки

У другому розділі було проаналізовано принципи створення тестів для соціальної мережі та зроблено висновок, що тестування повинно мати такі характеристики: зрозумілу назву, вичерпну кількість питань або стислість самих питань та відповідей, мотивуючі результати, незважаючи на їх якість. Розроблено структуру інтерфейсу головної сторінки соціальної мережі для онлайн-навчання користувачів. Розроблено макет дизайну інтерфейсу користувача сторінок соціальної мережі. Розроблено такі алгоритми роботи веб-додатку: реєстрації, створення результату проходження тесту, підрахунку результату проходження тесту та фільтрації тестів за тегами.

3 РОЗРОБКА МОДУЛІВ ПРОГРАМНОГО ПРОДУКТУ

3.1 Варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу

HTML – це мова гіпертекстової розмітки сторінки. Вона використовується для того, щоб дати браузеру зрозуміти, як потрібно відображати завантажений сайт. Мова складається з тегів – це своєрідні команди, які перетворюються на візуальні об'єкти в браузері користувача [10]. Оскільки HTML5 вважається останньою версією мови для структурування та зображення вмісту світової мережі, вирішено було застосовувати саме її для розробки соціальної мережі для навчання користувачів.

Завдяки п'ятій версії HTML можна розміщувати контент в семантичні блоки, що допомагають браузеру осягнути, який контент міститься в цьому компоненті. Тому сторінка може з більшою ймовірністю бути однією з перших при пошуку, що допоможе в просуванні соціальної мережі.

CSS – це мова розмітки, яка використовується для візуального оформлення веб-сайтів. Об'єкти на сторінці розміщуються за допомогою HTML. А ось CSS відповідає за те, як ці об'єкти виглядають. Їх розмір, колір, фонове зображення, рівень прозорості, розташування щодо інших елементів, поведінка при наведенні курсору, візуальна зміна кнопок при натисканні тощо [11]. Було прийнято рішення використовувати не що інше як CSS3, тому що це остання версія, тому дозволить скористатися великою кількістю можливостей для створення вдалого дизайну сайту.

JavaScript – це мова програмування, яка дозволяє створити динамічно оновлюваний контент, керує мультимедіа, анімує зображення, втім, робить все, що можливо. Дозволяє застосовувати складні речі на web сторінці [12]. JavaScript використовується для розробки клієнтської частини веб-сайту. Також при можливості за допомогою JavaScript можна розробити частину функціоналу.

Java – це мова високого рівня, тобто вона схожа на людську, на відміну від мов низького рівня, які нагадують машинний код [13]. Мову програмування Java було вирішено використовувати для розробки серверної частини соціальної мережі для онлайн-навчання користувачів. У якості інтегрованого середовища розробки було вирішено обрати IntelliJ IDEA.

Visual Studio Code – редактор вихідного коду, виконаний Microsoft для Windows, Linux та MacOS. Позиціонується як «простий» редактор коду для розробки веб- та хмарних проектів для різних платформ. Містить відладчик, прилади для роботи з Git, підсвічування синтаксису, IntelliSense та ресурси для рефакторингу [14]. Visual Studio Code дає можливість вільно і зручно писати код, особливо завдяки підсвічуванням синтаксису, що зменшує час розробки. У якості інтегрованого середовища розробки було вирішено використовувати Visual Studio Code.

Отже, проведено варіативний аналіз та обґрунтовано вибір засобів для реалізації програмного засобу.

3.2 Розробка бази даних соціальної мережі для онлайн-навчання

У таких веб-додатках, як соціальна мережа, багато інформації, яку потрібно десь зберігати. Для цього було вирішено використовувати базу даних.

База даних може бути визначена як структурна сукупність даних, що підтримуються в активному стані та відображає властивості об'єктів зовнішнього (реального) світу. В базі даних містяться не тільки дані, але й описи даних, і тому інформація про форму зберігання вже не схована в сполучення «файл-програма», вона явним чином декларується в базі [15].

Першочергове і основне, що дуже важливо взяти до відома – це практичність застосування системи управління базами даних (СУБД), що дається, і ступінь її масштабування, а також потенціал інтеграції з тими продуктами, з якими вона матиме контакт.

MySQL Workbench – програма для візуального проектування баз даних, що поєднує проектування, моделювання, створення та використання БД у цілісне безшовне оточення для системи баз даних MySQL. Вона дає можливість зобразити модель бази даних, а в свою чергу дає потенціал редагування даних у таблиці в графічному варіанті.

Зваживши усе вищесказане, було зроблено висновок, що у якості СУБД підходить використовувати саме MySQL Workbench, оскільки вона дозволяє візуально представити модель бази даних, а також є зручний редактор SQL запитів.

Перечислені суті та визначені ключі, які однозначно ідентифікують поля даних сутей:

- user (<login>, password, name, surname, age, institution, email, rating);
- test (<idtest>, name_test, description_test, completed_couter_test);
- question (<idquestion>, test_idtest, text_que);
- answer (<idanswer>, text, correct, question_idquestion);
- result (<idresult>, user_login, test_idtest, submitted);
- post (<idpost>, text, timestamp);
- tag (<idtag>, name).

Створену таблицю характеристики зв'язків бази даних соціальної мережі можна побачити у таблиці 3.1.

Таблиця 3.1 – Характеристики зв'язків бази даних соціальної мережі

Ім'я суті 1	Ім'я суті 2	Тип зв'язку	Клас належності
test	question	1:N	Обов., обов.
question	answer	1:N	Обов., обов.
answer	result	N:M	Обов., обов.
user	result	1:N	Обов., обов.
user	post	1:N	Обов., обов.
user	post	N:M	Обов., обов.
test	tag	N:M	Обов., обов.

Використовуючи таблицю 3.1, побудовано результуючу ER-модель соціальної мережі для онлайн-навчання користувачів, яка представлена на рисунку 3.1.

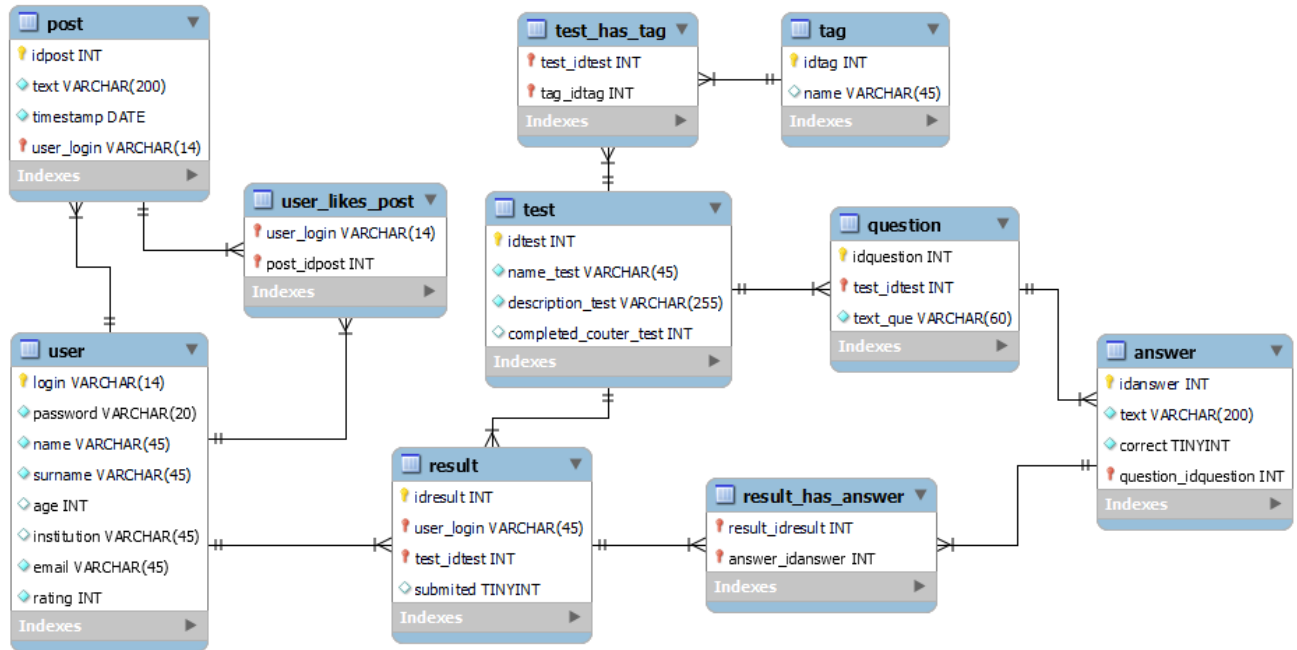


Рисунок 3.1 – ER-модель бази даних соціальної мережі

Таблиця «result_has_answer» є проміжною та утворена унаслідок зв'язку N:M між таблицями result та answer. Аналогічно створені таблиці «user_likes_post» та «test_has_tag» – вони також є проміжними та містять ключі сутностей, в результаті зв'язку яких утворилися.

Отже, було обґрунтовано вибір СУБД та розроблено ER-модель бази даних соціальної мережі для онлайн-навчання користувачів.

3.3 Розробка клієнтської частини соціальної мережі для онлайн-навчання

Розробку веб-додатку доцільно починати із HTML завдяки семантичним тегам, які покращать пошук веб-сторінки у Інтернет-просторі. Це буде так названий каркас програмного додатку.

Далі до роботи під'єднується розробка стилів за допомогою CSS. Це допоможе додати веб-сторінці приємного вигляду та відтворити макети дизайну у реальність. Поєднання HTML та CSS вже робить веб-додаток схожим на повноцінне програмне забезпечення, але це далеко не все.

Починається розробка мовою JavaScript, за допомогою якої можна створити динамічно оновлюваний контент, керувати мультимедіа, анімувати зображення та взагалі все, що можливо.

Втім, починаємо розробку зверху вниз. На рисунку 3.2 зображено шапку соціальної мережі. Вона містить логотип веб-сайту у лівому верхньому кутку та панель навігації.



Рисунок 3.2 – Розроблена шапка головної сторінки додатку

Верхній частині (шапці) веб-сторінки відповідає наведений нижче фрагмент коду файлу index.html (див. рис. 3.3).

```
<header class="b-header">
  <div>
    <div>
      
    </div>
  </div>
  <nav>
    <ul class="b-header_nav">
      <li class="b-header_nav-item"><a href="#" class="b-header_nav-item-link">Головна</a></li>
      <li class="b-header_nav-item"><a href="#" class="b-header_nav-item-link">Каталог</a></li>
      <li class="b-header_nav-item"><a href="#" class="b-header_nav-item-link">Увійти/Зареєструватись 
  </nav>
</header>
```

Рисунок 3.3 – Фрагмент коду головної сторінки

Далі розробляється наступний блок головної сторінки. Він містить медіа-контент у вигляді зображення песика в окулярах, який працює за планшетом,

заголовок тексту, сам текст, а також кнопку для переходу на іншу сторінку (для неавторизованих користувачів – на сторінку авторизації, для авторизованих – на сторінку каталогу тестів).

Його вигляд представлений на рисунку 3.4.

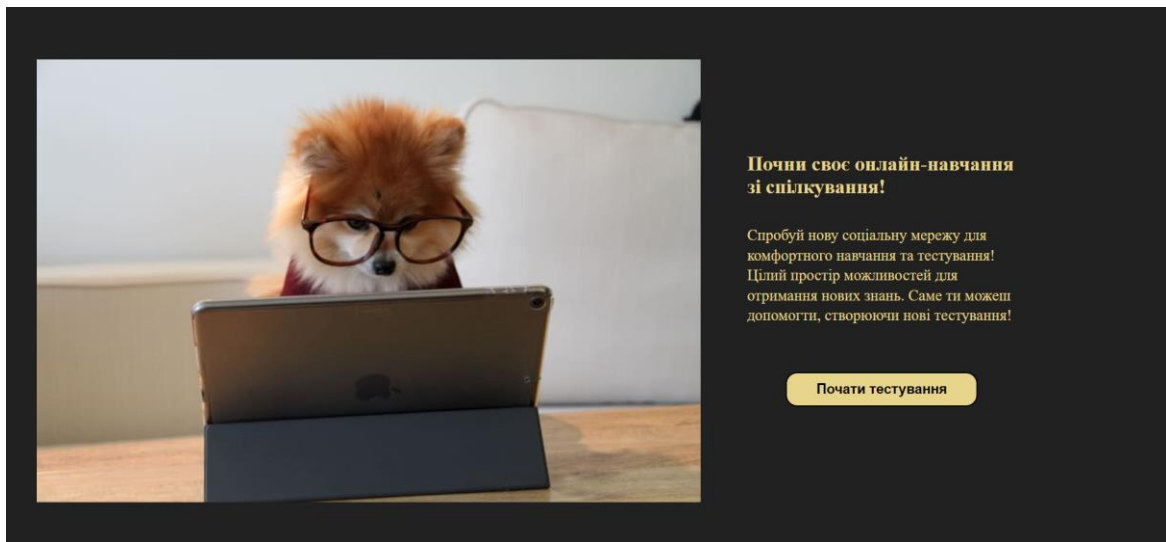


Рисунок 3.4 – Розроблений блок головної сторінки додатку

Йому відповідає наступний фрагмент коду (див. рис. 3.5).

```
<section class="b-start">
  <div>
    
  </div>
  <div class="b-start_desc">
    <h2 class="b-start_desc-header">
      Почни своє онлайн-навчання зі спілкування!
    </h2>
    <p class="b-start_desc-text">
      Спробуй нову соціальну мережу для комфортного навчання та тестування!
      Цілий простір можливостей для отримання нових знань. Саме ти можеш допомогти, створюючи нові тестування!
    </p>
    <button class="b-start_btn">Почати тестування</button>
  </div>
</section>
```

Рисунок 3.5 – Фрагмент коду головної сторінки додатку

За цією логікою розроблено наступні кілька блоків головної сторінки соціальної мережі. Одним із найважливіших елементів є футер, оскільки він є на

кожній сторінці веб-сайту. Його вигляд представлений на рисунку 3.6.



Рисунок 3.6 – Футер соціальної мережі

Його код можна побачити на рисунку 3.7.

```
<footer class="b-footer">
  <div>
    
  </div>
  <h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>
```

Рисунок 3.7 – Лістинг коду футера

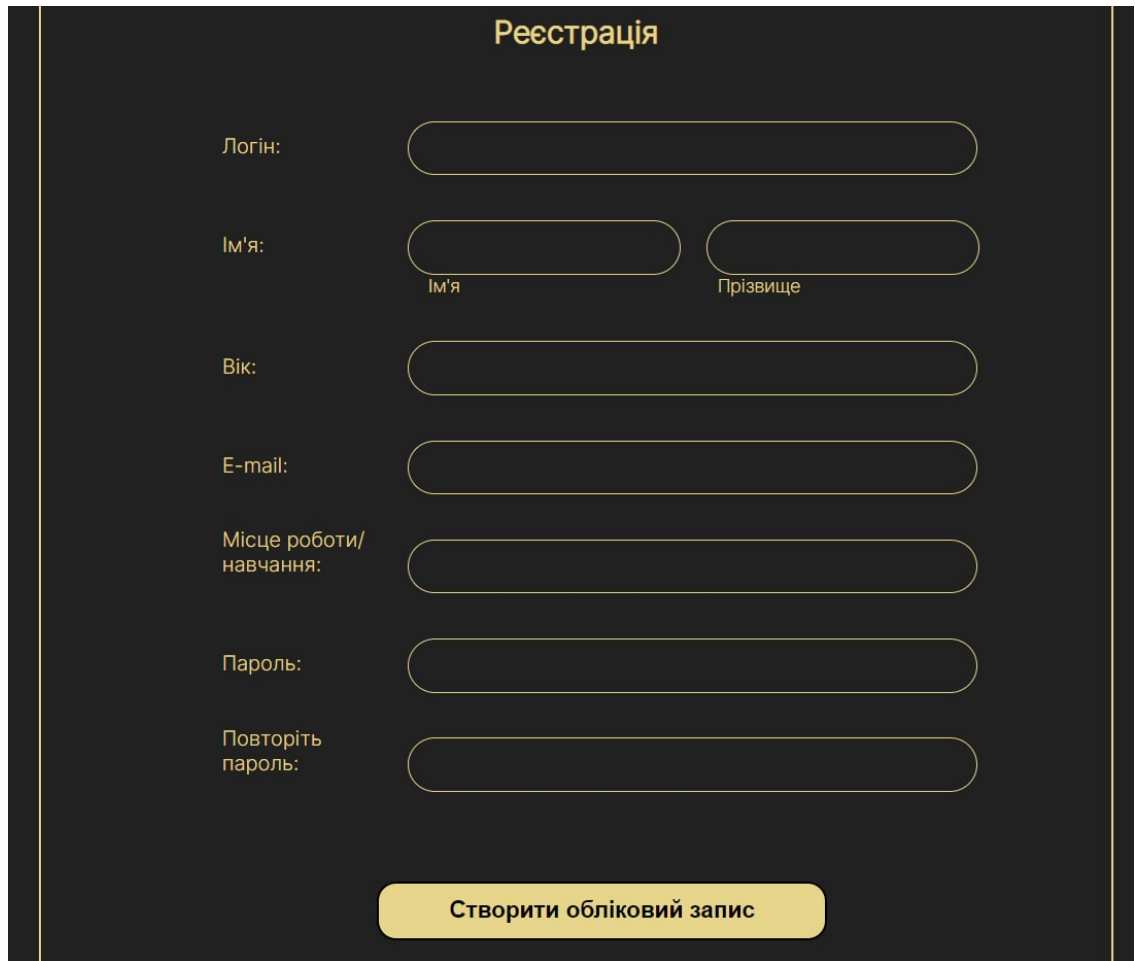
Футер виконано у мінімалістичному стилі – нічого зайвого.

Далі розглянемо розробку сторінки реєстрації нового користувача. Вона містить форму, яка складається з таких полей:

- логін;
- ім'я (ім'я та фамілія);
- вік
- поштову скриньку;
- місце роботи/навчання;
- пароль;
- підтвердження паролю.

Поля для паролю та його підтвердження приховують написану інформацію одразу після написання.

Вигляд форми реєстрації можна переглянути на рисунку 3.8.



Реєстрація

Логін:

Ім'я:

Вік:

E-mail:

Місце роботи/
навчання:

Пароль:

Повторіть
пароль:

Створити обліковий запис

Рисунок 3.8 – Форма реєстрації соціальної мережі

Фрагмент лістингу коду форми наведено на рисунку 3.9.

```

<label>
  <span class="b-form_fieldname">Місце роботи/навчання: </span>
  <input type="text" name="test" class="b-form_field">
</label>
<label>
  <span class="b-form_fieldname">Пароль: </span>
  <input type="password" name="test" class="b-form_field">
</label>
<label>
  <span class="b-form_fieldname">Повторіть пароль: </span>
  <input type="password" name="test" class="b-form_field">
</label>
  <button class="b-form_btn">Створити обліковий запис</button><br/>
</form>

```

Рисунок 3.9 – Фрагмент лістингу коду сторінки реєстрації

Розглянемо розробку сторінки профілю користувача. Він містить усю введену інформацію під час реєстрації про користувача. Також на сторінці профілю можна залишати повідомлення.

Профіль користувача можна побачити на рисунку 3.10.

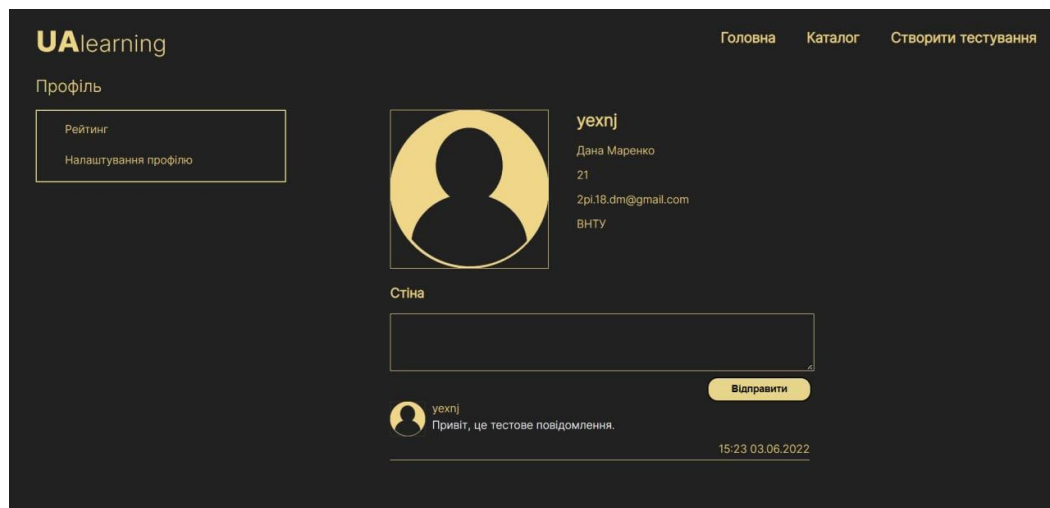


Рисунок 3.10 – Сторінка профілю користувача

Фрагмент лістингу коду сторінки облікового запису можна побачити на рисунку 3.11.

```

1  document.cookie = 'page=профіль';
2
3  fillUserInfo(getCookie('userLogin'));
4
5  function fillUserInfo(login) {
6      let userLogin = document.querySelector('.b-profile_info-user-login');
7      let userName = document.querySelector('.b-profile_info-user-name');
8      let userAge = document.querySelector('.b-profile_info-user-age');
9      let userEmail = document.querySelector('.b-profile_info-user-email');
10     let userWorkplace = document.querySelector('.b-profile_info-user-workplace');
11
12     fetch(`http://localhost:8080/user/${login}`)
13     .then(res => res.json())
14     .then(userData => {
15         userLogin.innerText = login;
16         userName.innerText = `${userData.name} ${userData.surname}`;
17         userAge.innerText = userData.age;
18         userEmail.innerText = userData.email;
19         userWorkplace.innerText = userData.institution;
20     })
21 }

```

Рисунок 3.11 – Фрагмент лістингу коду профілю

Розглянемо розробку сторінки каталогу тестувань. У каталозі є можливість шукати тести за назвою та описом. При введенні тексту у пошукову стрічку наповнення каталогу динамічно змінюється. Також є можливість фільтрувати тестування за тегом, після натиснення на потрібний будуть відображатися лише ті тести, які містять цей тег.

Вид сторінки каталогу представлено на рисунку 3.12.

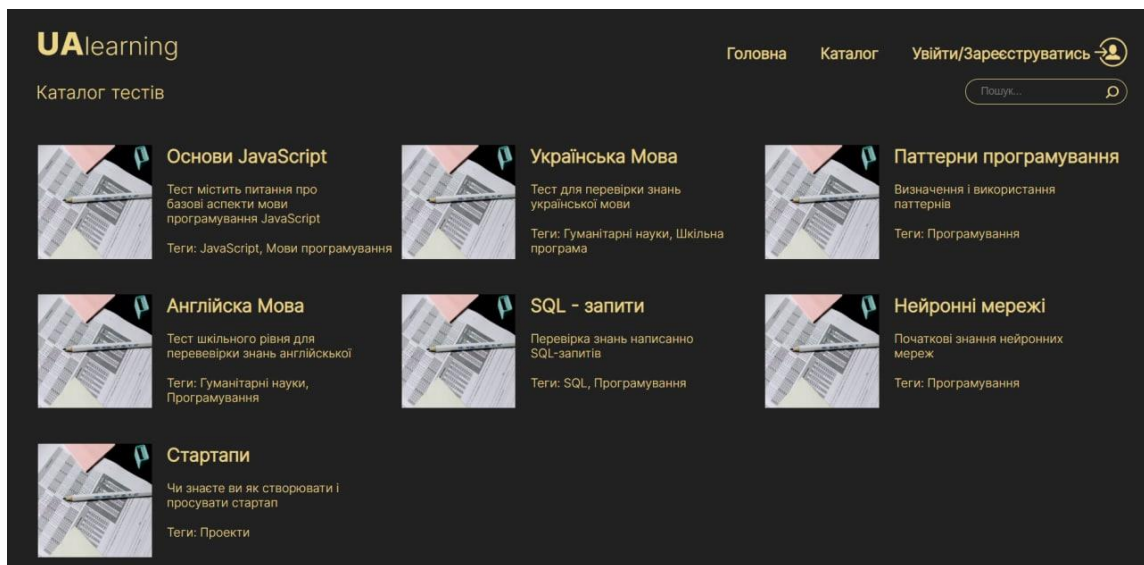


Рисунок 3.12 – Сторінка каталогу

Фрагмент лістингу коду наведено на рисунку 3.13.

```
function findTests(inputValue) {
  const catalogTests = document.querySelectorAll('.b-catalog_tests-test');

  for (const test of catalogTests) {
    const testName = test.querySelector('.b-catalog_tests-test-name');
    const testDesc = test.querySelector('.b-catalog_tests-test-desc');

    if (!testName.innerText.toLowerCase().includes(inputValue.toLowerCase()) &&
        !testDesc.innerText.toLowerCase().includes(inputValue.toLowerCase())) {
      test.style.display = 'none';
    } else {
      test.style.display = 'flex';
    }
  }
}
```

Рисунок 3.13 – Фрагмент лістингу пошуку

Лістинг коду представляє собою функцію, написану на JavaScript, яка відповідає за пошук по каталогу. Пошук відбувається у реальному часі та оновлює каталог за рахунок введеного тексту у пошукову стрічку. Пошук здійснюється за назвою тестування та його описом, аби розширити можливі варіанти тестів, які отримає користувач.

Отже, було описано розробку клієнтської частини соціальної мережі для онлайн-навчання користувачів. Лістинг коду наведений у додатку Г.

3.4 Розробка серверної частини соціальної мережі для онлайн-навчання

Оскільки у результаті варіативного аналізу для розробки серверної частини соціальної мережі було обрано мову програмування Java, а також була розроблена база даних, використовуються такі бібліотеки та фреймворки для полегшення розробки: Spring Framework та Hibernate.

Spring Framework – це платформа загального призначення з відкритим кодом для платформи Java. Spring надає рішення для багатьох завдань, з якими стикаються розробники Java. Через його широкий спектр функцій важко визначити найважливіші конструктивні елементи, включаючи елементи, які вони містять. Spring можна розглядати як набір менших фреймворків або фреймворків всередині фреймворків. Більшість із цих фреймворків можуть працювати незалежно одна від одної, але вони пропонують більше функціональних можливостей, коли вони використовуються спільно. Простіше кажучи, без фреймворка Spring багато чого не може працювати: служби, контролери, робота з базою даних тощо.

Hibernate – це бібліотека для мови програмування Java, призначена для вирішення проблем об'єктно-реляційного відображення, і це найпопулярніша реалізація специфікації JPA. Зменшить низькорівневе програмування при використанні реляційних баз даних. Може використовуватися як у процесі проектування системи класів і таблиць «з нуля», так роботи з вже існуючою базою.

Ніibernate автоматизує генерацію SQL-запитів і позбавляє розробника ручної обробки набору даних і перетворення об'єктів, що вийде у результаті, роблячи набагато легшим перетягування програми на будь-які бази даних SQL.

Почнемо з опису бази даних соціальної мережі. Він потрібен для взаємодії з базою даних. На рисунку 3.14 представлений опис таблиці «User».

```
package com.vntu.marenko.ualearning.server.model;
import lombok.Data;
import javax.persistence.Entity;
import javax.persistence.Id;

@Data
@Entity
public class User {

    @Id
    private String login;
    private String password;
    private String name;
    private String surname;
    private int age;
    private String institution;
    private String email;
    private long rating;
}
```

Рисунок 3.14 – Опис таблиці «User»

Відповідно до цього було описано такі таблиці: Post, Answer, Question, Result, Tag, Test.

Далі прописуємо логіку для серверної частини (див. рис. 3.15).

```
@Override
public User getUser(String id) { return userRepository.getReferenceById(id); }

@Override
public List<User> top10() {
    Pageable pageable = PageRequest.of( page: 0, size: 10, Sort.by(Sort.Direction.DESC, ...properties: "rating"));
    return userRepository.findAll(pageable).getContent();
}
```

Рисунок 3.15 – Фрагмент лістингу коду логіки користувача

Описано метод, який повертає інформацію про користувача за його айді. Також розроблено логіку рейтингування на сторінці рейтингу. Метод повертає відсортований за спаданням список користувачів. Використовується на сторінці рейтингу.

Далі створюються контролери для безпосередньої обробки запитів від клієнта та повернення результатів. На рисунку 3.16 показано фрагмент лістингу коду.

```
@GetMapping("/{id}")
public UserDto getUser(@PathVariable String id) { return mapper.map(userService.getUser(id), UserDto.class); }

@GetMapping("/rating")
public List<UserDto> rating() { return mapper.mapAsList(userService.top10(), UserDto.class); }
```

Рисунок 3.16 – Фрагмент лістингу коду контролера користувача

Даний контролер оброблює описану вище логіку користувача соціальної мережі для повернення інформації про користувача та рейтингу.

Отже, було описано розробку серверної частини соціальної мережі для онлайн-навчання користувачів на прикладі розробки юзера веб-сайту.

3.5 Висновки

Проведено варіативний аналіз обґрунтування вибору засобів для реалізації програмного засобу та зроблено висновок, що будуть використовуватися такі засоби розробки: HTML, CSS, JavaScript та Java. Розроблено базу даних соціальної мережі та представлено ER-модель. Розроблено соціальну мережу для онлайн-навчання користувачів.

4 ТЕСТУВАННЯ ПРОГРАМИ

4.1 Аналіз методів тестування програмного забезпечення

Для того, щоб будь-який продукт вийшов на ринок, обов'язково потрібно провести тестування програмного забезпечення. Необхідно серйозно поставитись до процесу створення веб-сайту.

Тестування – це відхилення фактичних результатів від очікуваних, іншими словами, це процес пошуку багів (помилки у коді). Основне правило тестування веб-сайту – показати користувачеві, наскільки зручним і логічним буде елемент, прості та можливі кроки для пошуку тієї чи іншої інформації [16].

Тестування чорного ящика – це процес отримання та відбору тестових випадків на основі аналізу специфікацій (функціональних чи нефункціональних), компонентів чи систем без посилання на їх внутрішню структуру [16].

Метод «Чорного ящика» можна використовувати для всіх рівнів тестування (від модульності до прийому), де є специфікація.

Переваги:

- можна починати писати тест-кейси, як тільки готова специфікація;
- тестувальнику немає необхідності знати мови програмування і заглиблюватися в особливості реалізації програми.

Недоліки: тестується тільки дуже обмежена кількість шляхів виконання програми.

На відміну від «чорного ящика» у підході «білого ящика», тестер проводить тести на основі знання обладнання та логіки частини бекенда, що тестується [16]. Тому під час тестування з використанням підходу «білого ящика» створюються сценарії для перевірки певної частини серверної частини, а не певної моделі поведінки користувача.

Переваги:

- тестування може проводитися на ранніх етапах: немає необхідності чекати

створення призначеного для користувача інтерфейсу;

– можна провести більш ретельне тестування, з покриттям великої кількості шляхів виконання програми.

Недоліки: для виконання тестування білого ящика необхідна велика кількість спеціальних знань.

Перевірка методом сірого ящика – спеціальний спосіб випробування працездатності програмного забезпечення, у якому вся функціональність продукту вивчається без аналізу вихідного коду. Цей метод вважається своєрідним поєднанням методів «чорного» та «білого» ящиків.

Переваги: є оптимальним варіантом, оскільки містить складові тестування чорного та білого ящиків.

Недоліки: неможливо перевірити всі ймовірні потоки введення і виведення, тому що на це йде надто багато часу.

Отже, провівши аналіз методик функціонального тестування, було вирішено обрати метод «чорного ящика», адже він дозволяє прискорити процес тестування, порівнюючи фактичний результат з потрібним.

4.2 Тестування розробленого програмного продукту

Test case – це тестовий артефакт, суть якого полягає у виконанні певної кількості дій і/або умов, необхідних для перевірки певної функціональності програмної системи, що розробляється [16].

Структура даного артефакту полягає в «тріїці»:

- що треба зробити;
- очікуваний результат;
- фактичний результат.

Тестові випадки можуть бути:

– позитивні – використовуються тільки очікувані правильні дані і перевіряється, чи відповідає робота додатку очікуванням;

– негативні – використовуються як правильні, так і неправильні дані і ставиться за мету перевірка виняткових ситуацій.

Для підтвердження результатів роботи соціальної мережі для онлайн-навчання користувачів здійснено тестування ряд таких функцій:

- відкриття веб-сторінки;
- авторизації;
- реєстрації;
- фільтрації тегів;
- пошук тестів на сторінці каталогу;
- редагування інформації про користувача;
- проходження тестування;
- створення тестування.

Детальний опис тестових випадків наведено у таблиці 4.1.

Таблиця 4.1 – Тестові випадки

Іденти-фікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-1	Перевірка відображення елементів головної сторінки	1. Відкрити соціальну мережу.	1. Відображається головна сторінка. 2. Відображаються всі три медіа-файли, які містяться на головній сторінці.	Виконано
ТВ-2	Перевірка роботи авторизації	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Ввести дані та натиснути «Увійти».	1. Відображається сторінка входу в обліковий запис. 2. Відбувається вхід в існуючий обліковий запис. 3. Відображається сторінка профілю.	Виконано
ТВ-3	Перевірка роботи реєстрації	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Натиснути на надпис	1. Відображається сторінка входу в обліковий запис. 2. Відображається сторінка реєстрації.	Виконано

Продовження таблиці 4.1

Іденти-фікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
		«Ресстрація». 4. Ввести дані та натиснути «Створити обліковий запис».	3. Відбувається вхід у створений обліковий запис. 4. Відображається сторінка профілю.	
ТВ-4	Перевірка відображення сторінки «Каталог»	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Каталог».	1. Відображається сторінка каталогу та усі можливі тестування.	Виконано
ТВ-5	Перевірка фільтрації тестів	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Каталог». 3. Натиснути на обраний тег в описі тестування.	1. Відображається сторінка каталогу та лише ті тестування, які містять обраний тег.	Виконано
ТВ-6	Перевірка пошуку тесту в каталозі	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Каталог». 3. У поле пошуку ввести запит.	1. Відображається сторінка каталогу. 2. Відображаються тестування, назва та/або опис яких містить введений запит у пошукову стрічку.	Виконано
ТВ-7	Перевірка редагування інформації про користувача	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Ввести дані та натиснути «Увійти». 4. Натиснути «Налаштування профілю». 5. Ввести потрібні дані та натиснути «Зберегти».	1. Відображається профіль зі зміненою інформацією про користувача.	Виконано
ТВ-8	Перевірка відображення сторінки «Рейтинг»	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Ввести дані та натиснути «Увійти». 4. Натиснути «Рейтинг» на сторінці профілю.	1. Відображається сторінка рейтингу з двома таблицями – рейтингом серед користувачів та тестувань.	Виконано

Продовження таблиці 4.1

Ідентифікатор	Назва	Методика проведення тестування	Очікуваний результат	Результат
ТВ-9	Перевірка проходження тестування	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Ввести дані та натиснути «Увійти». 4. Натиснути на елемент навігаційної панелі «Каталог». 5. Натиснути на обраний тест. 6. Пройти тест та натиснути на «Завершити тестування».	1. Відображається сторінка результатів проходження тестування: бал та доля правильних відповідей по відношенню до їх кількості.	Виконано
ТВ-10	Перевірка створення тестування	1. Відкрити соціальну мережу. 2. Натиснути на елемент навігаційної панелі «Увійти/Зареєструватись». 3. Ввести дані та натиснути «Увійти». 4. Натиснути на «Створити тестування» на сторінці профілю. 5. Заповнити поля та натиснути «Створити тестування».	1. Відображається сторінка зі створеним тестом, який можна одразу пройти.	Виконано

Отже, створено десять тестових випадків, за допомогою яких було проведено тестування графічного інтерфейсу та функціоналу програмного продукту, а також підтверджено його працездатність та відповідність виконання дій очікуваному результату.

4.3 Розробка інструкції користувача

Інструкція користувача являє собою документ, який містить покрокову довідку для роботи з програмним продуктом. Вона повинна відповідати на всі

базові питання, які можуть виникнути у людей під час користування веб-додатком.

Після запуску програми видно головну сторінку веб-додатку (див. рис. 4.1).

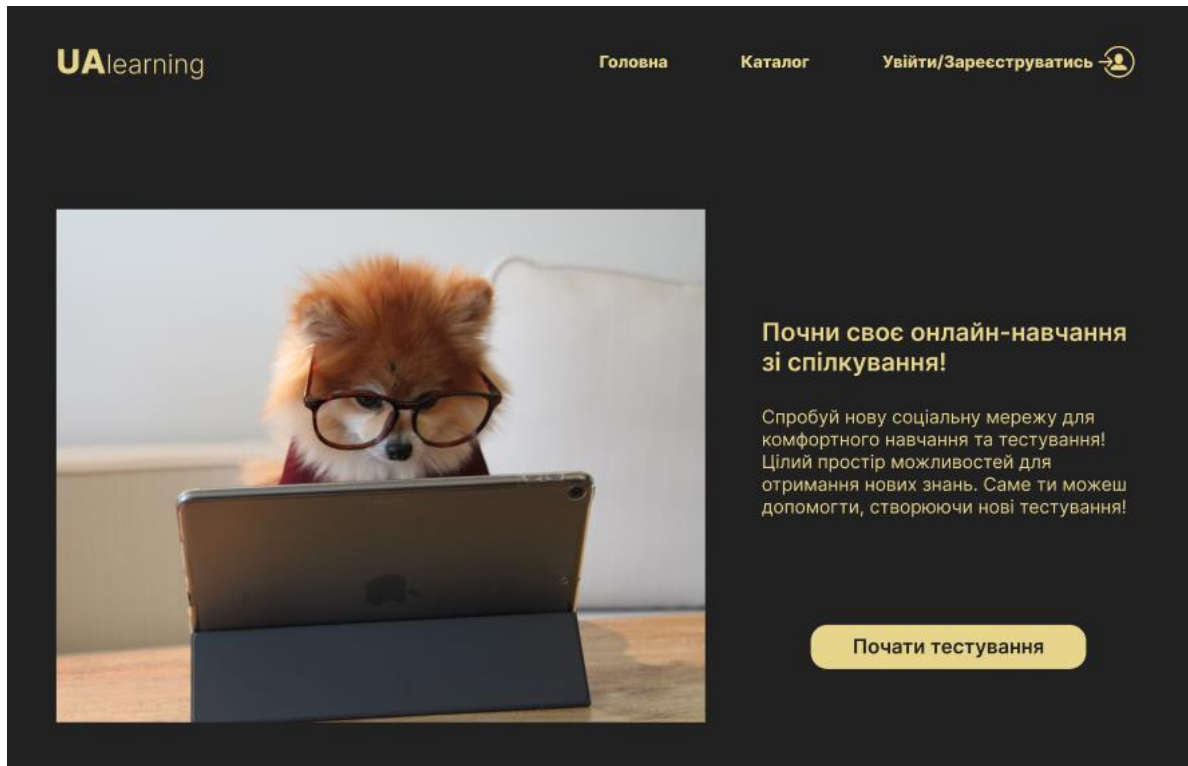


Рисунок 4.1 – Головна сторінка соціальної мережі

Головна сторінка має декілька блоків, з яких також можна перейти на такі сторінки: каталог, авторизація, створення тестування. На ці сторінки можна перейти, використовуючи панель навігації, але вона змінюється залежно від того, яка сторінка відкрита. Наприклад, на сторінці профілю з'явиться елемент навігаційної панелі «Створити тестування», коли на інших сторінках замість нього буде «Профіль».

Для авторизації потрібно натиснути на елемент навігаційної панелі «Увійти/Зареєструватися». Після цього відкриється сторінка авторизації, на якій у випадку наявності акаунту потрібно ввести логін та пароль та натиснути на кнопку «Увійти» (див. рис. 4.2).

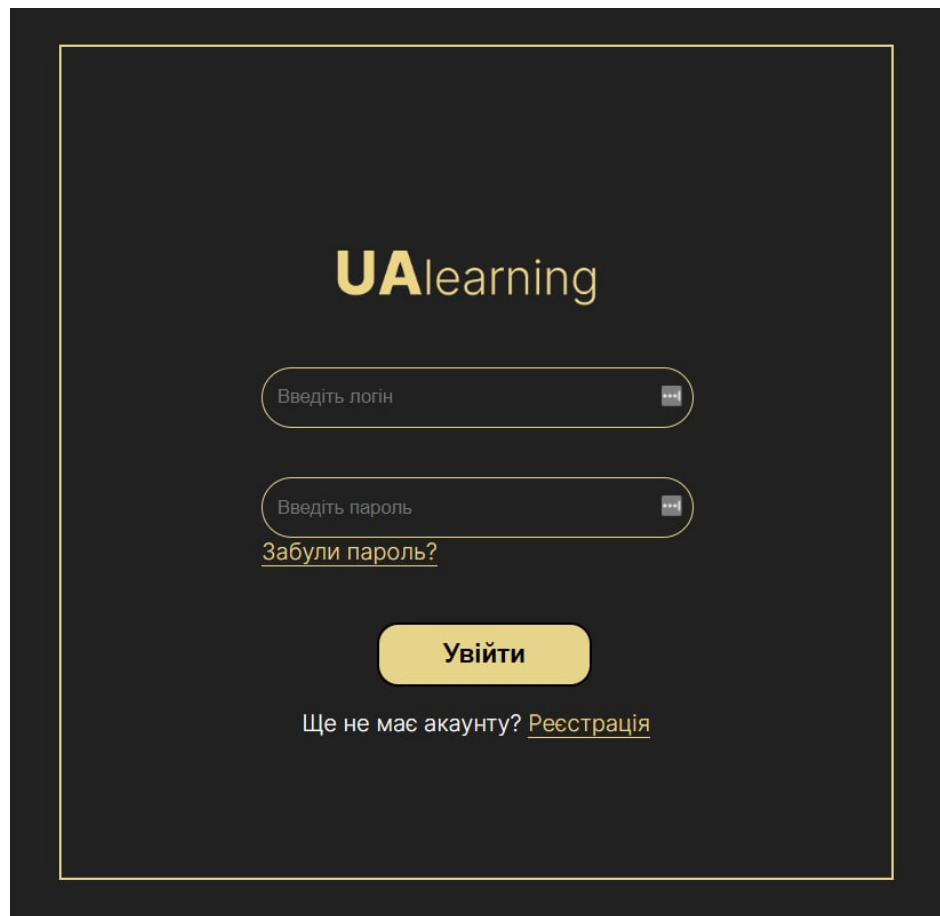


Рисунок 4.2 – Сторінка авторизації

У випадку, якщо користувач не має акаунту у соціальній мережі, на сторінці авторизації йому потрібно натиснути на надпис «Реєстрація» під кнопкою «Увійти». Після цього відкриється сторінка реєстрації, на якій потрібно заповнити усі поля форми та натиснути на кнопку «Створити обліковий запис». Потрібно зазначити, що логін, вказаний при реєстрації, змінити неможливо.

Поле «Місце роботи/навчання» не є обов'язковим для заповнення при реєстрації, але його бажано заповнити потім при редагуванні даних облікового запису, оскільки так буде набагато простіше знайти колег, однокласників та однокурсників.

Поле «Пароль» приховує усю введену інформацію для дотримання конфіденційності та безпеки даних користувача (див. рис. 4.3).

Реєстрація

Логін:

Ім'я:
Ім'я Прізвище

Вік:

E-mail:

Місце роботи/
навчання:

Пароль:

Повторіть
пароль:

Рисунок 4.3 – Сторінка реєстрації

У випадку, якщо спроба реєстрації не є успішною, потрібно впевнитись, що введені логін та пошта не повторюється та повторити спробу.

Після авторизації користувач автоматично переходить на сторінку профілю соціальної мережі. Для того, щоб змінити облікові дані, потрібно натиснути на надпис «Налаштування профілю» у виділеному блоці зліва вверху сторінки. Після чого відкриється сторінка редагування даних користувача (див. рис. 4.4).

Деяку введеної інформації під час реєстрації змінити неможливо – логін залишиться такий, як був вказаний при створенні облікового запису. Оскільки веб-сторінка є соціальною мережею, було запроваджено вікові обмеження на реєстрацію на сайті: зареєструватися може будь-хто старше 12 років. Пароль змінювати можна за бажанням – сторінка не буде вимагати заповнення цих полей.

Редагування профілю

Логін:

Ім'я:
Ім'я Прізвище

Вік:

E-mail:

Місце роботи/
навчання:

Пароль:

Повторіть
пароль:

Рисунок 4.4 – Сторінка редагування даних користувача

Для того, щоб відкрити рейтинг користувачів та тестів, потрібно натиснути на надпис «Рейтинг» у виділеному блоці зліва вверху сторінки. Після чого відкривається сторінка рейтингу (див. рис. 4.5).

UAlearning Головна [Каталог](#) [Створити тестування](#)

Рейтинг користувачів

№	Логін	Бали
1	yehnj	1337
2	babiuk_NP	1168
3	lerrka	800
4	simchik	777
5	kahalniak	524
6	danke	357
7	ivan_KO	322
8	prikol	228

Рейтинг тестів

№	Назва тесту	К-сть проходжень
1	Українська Мова	69
2	Стартапи	46
3	Паттерни програмування	36
4	Англійська Мова	25
5	Основи JavaScript	13
6	SQL - запити	7
7	Нейронні мережі	3

Рисунок 4.5 – Сторінка рейтингу

Для того, щоб почати тестування, потрібно перейти на сторінку каталогу. Є два варіанти того, як це зробити:

- натиснувши елемент навігаційної панелі «Каталог»;
- на головній сторінці натиснути на кнопку «Перейти до каталогу».

Для пошуку у каталозі потрібно ввести потрібне значення у пошукову стрічку. Наповнення каталогу одразу зміниться, як буде введений текст (див. рис. 4.6).

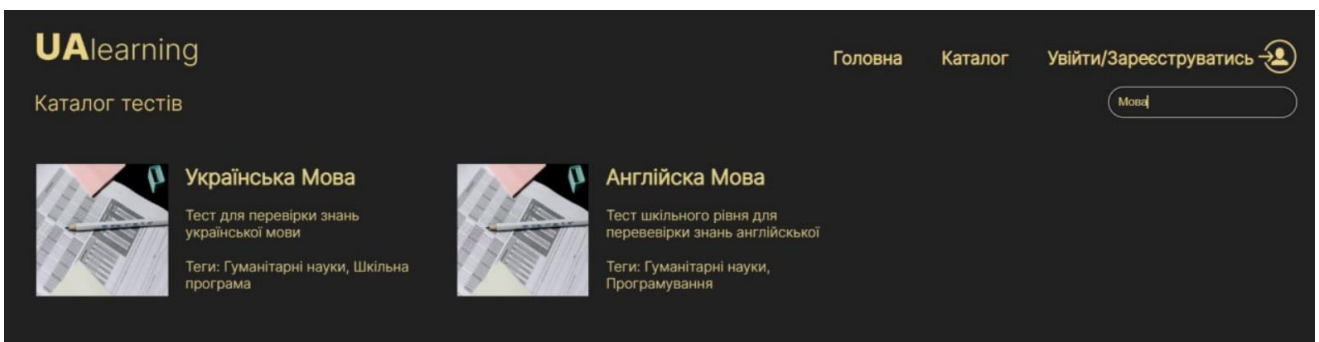


Рисунок 4.6 – Пошук у каталозі

Для фільтрації тестів за тегом потрібно натиснути на обраний тег, який відображається в інформації про тестування. Замість усіх тестів будуть лише ті, які мають обраний тег. Теги були впроваджені для спрощення пошуку тестів у каталозі.

Після того, як відкрилась сторінка каталогу, потрібно обрати тест, який хочете пройти. Для цього потрібно натиснути на назву обраного тестування. Після цього з'явиться сторінка з даними тесту та самим тестом. Потрібно обирати одну правильну відповідь. Зазвичай, якщо правильних відповідей декілька, то автор тестування вказує про це у самому запитанні, правильних відповідей на яке буде декілька. Після проходження тесту потрібно натиснути на кнопку «Завершити тестування» (див. рис. 4.7).

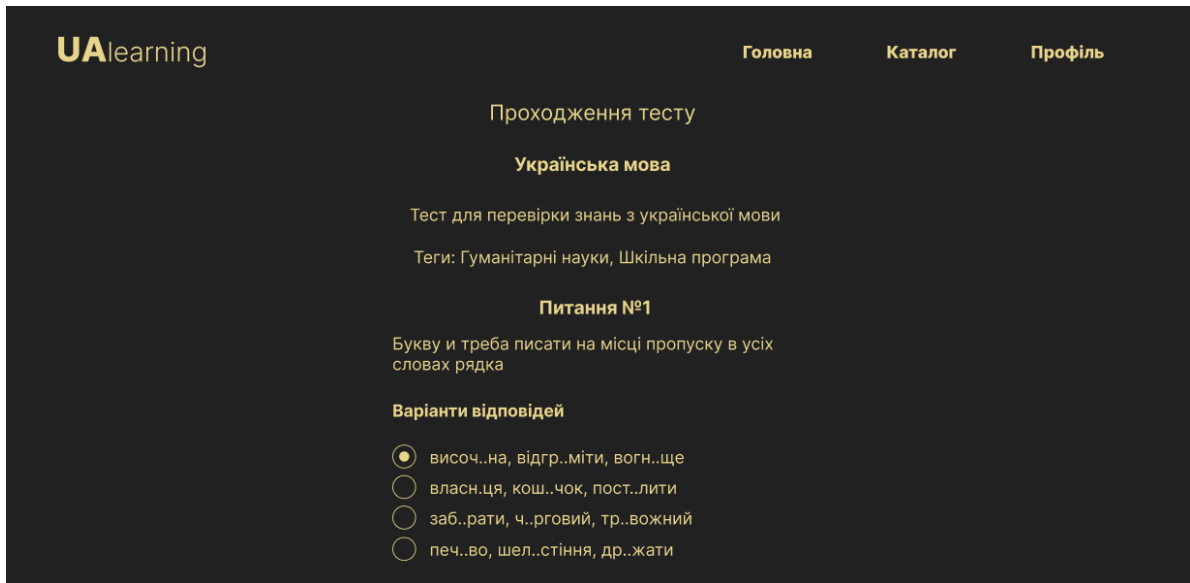


Рисунок 4.7 – Проходження тестування

Після завершення тестування з'являється сторінка з результатами тесту (див. рис. 4.8). Для виходу потрібно натиснути кнопку «Зберегти», що автоматично переадресує користувача на сторінку каталогу з тестуваннями.

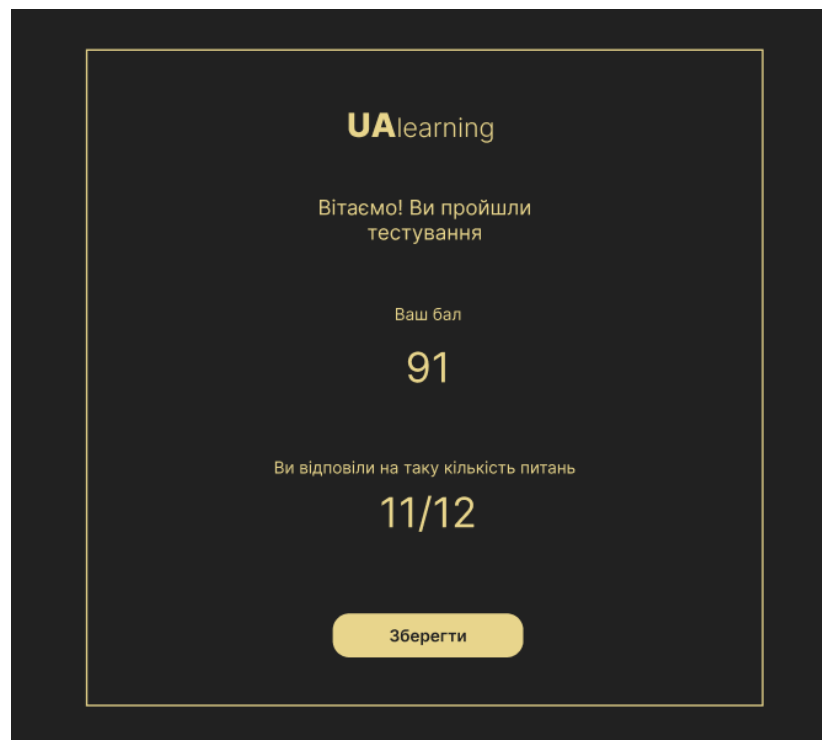


Рисунок 4.8 – Результат тестування

Є два шляхи створення власних тестувань (див. рис. 4.9):

– натиснувши елемент навігаційної панелі «Створити тестування» на сторінці профілю;

– на головній сторінці натиснути на кнопку «Створити тестування».

The screenshot shows the 'UAlearning' website interface for creating a test. At the top, there is a navigation bar with 'UAlearning' on the left and 'Головна', 'Каталог', and 'Профіль' on the right. The main heading is 'Створення тестування'. Below this, there are three input fields: 'Назва' (Name), 'Опис' (Description), and 'Теги' (Tags). Underneath is a section titled 'Питання' (Question) containing a 'Текст питання' (Question text) field, a 'Текст відповіді' (Answer text) field, and a 'Правильна' (Correct) radio button. A plus sign icon is located to the left of the answer field. At the bottom of the form are two buttons: 'Додати питання' (Add question) and 'Створити тестування' (Create test).

Рисунок 4.9 – Сторінка створення тестування

Перейшовши на сторінку створення тестування та заповнивши усі поля, треба натиснути кнопку «Створити тестування». Після чого відкриється сторінка створеного тесту, який одразу можна пройти.

Отже, було створено інструкцію користувача соціальної мережі для онлайн-навчання.

4.4 Висновки

Було проаналізовано методи тестування програмного забезпечення та вирішено проводити тестування саме за методом «Чорного ящика». Проведено тестування розробленої соціальної мережі за допомогою тестових випадків. Протестовано функціонал: відкриття веб-сторінки, авторизації, реєстрації, фільтрації тегів, пошук тестів на сторінці каталогу, редагування інформації про користувача, проходження тестування, створення тестування. Розроблено інструкцію користувача, де описано кожну сторінку веб-сайту та як її відкрити.

ВИСНОВКИ

У контексті дипломної роботи було створено соціальну мережу для підвищення ефективності та доступності навчання у будь-якій сфері для різних рівнів поглиблення знань.

У першому розділі було проаналізовано сучасний стан галузі онлайн-навчання. Провівши порівняльний аналіз аналогів, було зроблено висновок, що дана розробка є актуальною. Проведено аналіз методів розв'язання поставленої задачі та вирішено, що веб-додаток буде розроблятися «з нуля». Було визначено задачі, які потрібно виконати для успішної розробки соціальної мережі.

У другому розділі було проаналізовано принципи створення тестувань. Розроблено структури інтерфейсу соціальної мережі. Розроблено макет дизайну інтерфейсу користувача сторінок соціальної мережі та обрано кольорову гаму. Розроблено та описано одні з найголовніших алгоритмів соціальної мережі.

У третьому розділі проведено варіантний аналіз і обґрунтування вибору засобів для реалізації програмного засобу і визначено, що веб-додаток буде розроблятися за допомогою таких засобів: HTML, CSS, JavaScript – для клієнтської частини, та Java – для серверної частини веб-сайту. Розроблено базу даних для зберігання інформації про користувачів соціальної мережі та тести. Розроблено соціальну мережу для онлайн-навчання користувачів.

У четвертому розділі проведено аналіз методів тестування програмного забезпечення і зроблено висновок, що тестування соціальної мережі для онлайн-навчання користувачів буде проводитися за допомогою методу «чорного ящика». Проведено тестування програмного веб-додатку за допомогою тестових випадків. Розроблено інструкцію користувача із детальним описом користування соціальною мережею.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Stepik. [Електронний ресурс]. – Режим доступу: <https://welcome.stepik.org/ru>
2. Coursera. [Електронний ресурс]. – Режим доступу: <https://www.coursera.org/>
3. Udemy. [Електронний ресурс]. – Режим доступу: <https://www.udemy.com/>
4. Jamm School. [Електронний ресурс]. – Режим доступу: <https://jammschool.com.ua/>
5. Маренко Д. В. Бабюк Н.П. Порівняння засобів для онлайн-навчання користувачів. / Молодь в науці: дослідження, проблеми, перспективи (МН-2022), Вінниця: ВНТУ, 2022. 2 с.
6. Способы создания сайта. [Електронний ресурс]. – Режим доступу: <http://cccp-blog.com/sozдание-saytov/sposoby-sozdaniya-sajta>
7. Романюк О. Н. Веб-дизайн і комп'ютерна графіка / О. Н. Романюк, Д. І. Кательніков, О. П. Косовець. – Вінниця, 2007. – 142 с
8. Дизайн и проектирование пользовательского интерфейса. [Електронний ресурс]. – Режим доступу: <https://dlogic.com.ua/service/proektirovanie-interfeisov/>
9. Маренко Д. В. Бабюк Н.П. Розробка алгоритмів соціальної мережі для онлайн-навчання користувачів. / Молодь в науці: дослідження, проблеми, перспективи (МН-2022), Вінниця: ВНТУ, 2022. 2 с.
10. Что такое HTML. [Електронний ресурс]. – Режим доступу: https://skillbox.ru/media/code/chto_takoe_html/
11. Основы CSS. [Електронний ресурс]. – Режим доступу: <https://timeweb.com/ru/community/articles/chto-takoe-css>
12. Что такое JavaScript? [Електронний ресурс]. – Режим доступу: https://developer.mozilla.org/ru/docs/Learn/JavaScript/First_steps/What_is_JavaScript

13. Шилдт Г. Java 8. Керівництво для початківців: пер. з англ. М. Вільямс / Г. Шилдт – Санкт-Петербург: Вільямс, 2015. – 712 с.
14. Visual Studio Code. [Електронний ресурс]. – Режим доступу: <https://code.visualstudio.com/>
15. Романюк О. Н. Організація баз даних і знань. / О.Н. Романюк, Т.О. Савчук // Навчальний посібник. – Вінниця: УНІВЕРСУМ – Вінниця. – 2003. – 123 с.
16. Савин Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. – М.: Дело, 2007. – 312 с.

ДОДАТКИ

Додаток А – Технічне завдання
Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

УЗГОДЖЕНО

Директор ТОВ «ЮК КРИСТАЛ»

Довбеус І. Я.

«__» _____ 2022 року

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Романюк О. Н.

«25» березня 2022 р.

Технічне завдання
на бакалаврську дипломну роботу
«Розробка соціальної мережі для онлайн-навчання користувачів»
за спеціальністю 121 – Інженерія програмного забезпечення

Керівник бакалаврської дипломної роботи:

к.т.н., доцент кафедри ПЗ Бабюк Н. П

«____» _____ 2022 р.

Виконав:

студент гр. 2ПІ-186 Маренко Д. В.

«____» _____ 2022 р.

Вінниця – 2022 року

1 Найменування та галузь застосування.

Бакалаврська робота: «Розробка соціальної мережі для онлайн-навчання користувачів».

Сфера застосування – навчальна (інформаційні технології).

2 Підстава для розробки.

Підставою для розробки бакалаврської дипломної роботи є рішення засідання кафедри програмного забезпечення (протокол №12 від «07» 2022 року).

3 Мета та призначення розробки.

Мета виконання бакалаврської дипломної роботи – створення веб-додатку з елементами соціальної мережі для підвищення ефективності та доступності навчання у будь-якій сфері для різних рівнів поглиблення знань.

Призначення роботи – розробка програмного продукту для онлайн-навчання користувачів.

4 Вихідні дані для проведення НДР.

Вихідні дані для розробки є індивідуальне завдання на бакалаврську дипломну роботу на розробку соціальної мережі для онлайн-навчання користувачів.

5 Технічні вимоги.

Принципи створення тестів у соціальній мережі для онлайн навчання користувачів; формування каталогу тестів; створення облікового запису; розроблена база даних; розроблений веб-додаток.

6 Конструктивні вимоги.

Веб-додаток має відповідати ергономічним та технічним вимогам. текстова

та графічна документація повинна відповідати стандартам України.

7 Перелік технічної документації, що пред'являється по закінченню робіт:

- пояснювальна записка до бакалаврської дипломної роботи;
- технічне завдання;
- лістинг програми.

8 Вимоги до рівня уніфікації та стандартизації.

При розробці програмних засобів слід дотримуватися уніфікації і ДСТУ.

9 Стадії та етапи розробки.

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1	Обґрунтування вибору методу розробки та постановка задач дослідження	26.03.2022 – 02.04.2022	Виконано
2	Розробка структури та алгоритмів програмного продукту	03.04.2022 – 17.04.2022	Виконано
3	Розробка модулів програмного продукту	18.04.2022 – 20.05.2022	Виконано
4	Тестування програми	21.05.2022 – 25.05.2022	Виконано
5	Оформлення матеріалів до захисту БДР	26.06.2022 – 10.06.2022	Виконано

10 Порядок контролю та прийняття.

Всі етапи бакалаврської роботи контролюються науковим керівником згідно плану по виконанню роботи. Прийняття бакалаврської кваліфікаційної роботи здійснюється ДЕК, затвердженою зав. кафедрою згідно з графіком захисту. Дозволяється корегування бакалаврської дипломної роботи

Додаток Б
ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Розробка соціальної мережі для онлайн-навчання користувачів

Тип роботи: БДР

Підрозділ : кафедра програмного забезпечення, ФІТКІ

Науковий керівник: к.т.н., доцент Бабюк Наталя Петрівна

Оригінальність	94,3%
Схожість	5,7%

Аналіз звіту подібності

■ **Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.**

□ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

□ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Черноволик Г. О.

Ознайомлені з повним звітом подібності, який був згенерований системою Unichек

Автор роботи _____

Маренко Д.В.

Керівник роботи _____

Бабюк Н.П.

Додаток В – Акт впровадження

Додаток Г – Лістинг програми

Лістинг файлу index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <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:wght@300&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="style.css">
  <title>BDR</title>
</head>
<body>
<header class="b-header">
  <div>
    <div>
      
    </div>
  </div>
  <nav>
    <ul class="b-header_nav">
      <li class="b-header_nav-item"><a href="index.html" class="b-header_nav-
item-link">Головна</a></li>
      <li class="b-header_nav-item"><a href="pages/catalog.html" class="b-
header_nav-item-link">Каталог</a></li>
      <li class="b-header_nav-item"><a href="pages/authorization.html"
class="b-header_nav-item-link js-auth-link">Увійти/Зареєструватись </a></li>
    </ul>
  </nav>
</header>

<section class="b-start">
  <div>
    
  </div>
  <div class="b-start_desc">
    <h2 class="b-start_desc-header">
      Почни своє онлайн-навчання зі спілкування!
    </h2>
    <p class="b-start_desc-text">

```

Спробуй нову соціальну мережу для комфортного навчання та тестування!
Цілий простір можливостей для отримання нових знань. Саме ти можеш допомогти, створюючи нові тестування!

```

    </p>
    <button class="b-start_btn">Почати тестування</button>
  </div>
</section>

<section class="b-catalog">
  <h2 class="b-catalog_heading">
    Каталог тестів
  </h2>
  <div class="b-catalog_tests">
    <div class="b-catalog_test">
      <div>
        
      </div>
      <h3 class="b-catalog_test-name">Тест</h3>
      <p class="b-catalog_test-desc">Опис тесту</p>
    </div>
    <div class="b-catalog_test">
      <div>
        
      </div>
      <h3 class="b-catalog_test-name">Тест</h3>
      <p class="b-catalog_test-desc">Опис тесту</p>
    </div>
    <div class="b-catalog_test">
      <div>
        
      </div>
      <h3 class="b-catalog_test-name">Тест</h3>
      <p class="b-catalog_test-desc">Опис тесту</p>
    </div>
    <div class="b-catalog_test">
      <div>
        
      </div>
      <h3 class="b-catalog_test-name">Тест</h3>
      <p class="b-catalog_test-desc">Опис тесту</p>
    </div>
  </div>
  <button class="b-catalog_btn">Перейти до каталогу</button>

  <h2 class="b-catalog_info-heading">Також, маючи власний акаунт, ти можеш потрапити до списку найкращих!</h2>

```

```

    <p class="b-catalog_info-text">У соціальній мережі працює система рейтингування
за балами та кількістю пройдених тестів!</p>
</section>

```

```

<section class="b-create">
  <div class="b-create_desc">
    <h2 class="b-create_desc-heading">Створи власні тести та отримуй додаткові
бали за їх проходження</h2>
    <p class="b-create_desc-text">Створені тести, які набирають більше 100
проходжень, підіймаються у каталозі, аби їх побачили та пройшли користувачі
сайту.</p>
    <button class="b-create_desc-btn">Створити тест</button>
  </div>
  <div>
    
  </div>
</section>

```

```

<footer class="b-footer">
  <div>
    
  </div>
<h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>

```

```

<script src="js/homepage.js"></script>
<script src="js/script.js"></script>
</body>
</html>

```

Лістинг файлу profile.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <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:wght@300&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="../style.css">
  <title>BDR</title>
</head>
<body>

```

```

<header class="b-header">
  <div>
    <div>
      
    </div>
  </div>
  <nav>
    <ul class="b-header_nav">
      <li class="b-header_nav-item"><a href="../index.html" class="b-
header_nav-item-link">Головна</a></li>
      <li class="b-header_nav-item"><a href="../pages/catalog.html" class="b-
header_nav-item-link">Каталог</a></li>
      <li class="b-header_nav-item"><a href="../pages/createtest.html"
class="b-header_nav-item-link">Створити тестування</a></li>
    </ul>
  </nav>
</header>

<h2 class="b-profile_heading">Профіль</h2>
<div class="b-profile">
  <div class="b-profile_nav">
    <ul class="b-profile_nav-list">
      <li class="b-profile_nav-list-item"><a href="rating.html" class="b-
profile_nav-list-link">Рейтинг</a></li>
      <li class="b-profile_nav-list-item"><a href="changeuserinfo.html"
class="b-profile_nav-list-link">Налаштування профілю</a></li>
    </ul>
  </div>

  <div class="b-profile_info">
    <div class="b-profile_info-user">
      <div>
        
      </div>
      <div class="b-profile_info-user-data">
        <h2 class="b-profile_info-user-login">login</h2>
        <p class="b-profile_info-user-name">name surname</p>
        <p class="b-profile_info-user-age">20</p>
        <p class="b-profile_info-user-email">e-mail</p>
        <p class="b-profile_info-user-workplace">workplace</p>
      </div>
    </div>
    <h3 class="b-profile_info-feed-heading">Стіна</h3>
    <div class="b-profile_info-feed">
      <textarea class="b-profile_info-feed-input" name="" id="" cols="80"
rows="5"></textarea><br/>

```



```

        <button class="b-profile_info-feed-btn">Відправити</button>
    </div>

    <div class="b-feed">
        <div class="b-feed_item">
            <div class="b-feed_item-info">
                <div>
                    
                </div>
                <div>
                    <p class="b-feed_item-login">уехнј</p>
                    <p class="b-feed_item-text">Привіт, це тестове
повідомлення.</p>
                </div>
            </div>
            <p class="b-feed_item-date">15:23 03.06.2022</p>
        </div>
    </div>
</div>

<footer class="b-footer">
    <div>
        
    </div>
    <h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>

<script src="../../js/profile.js"></script>
</body>
</html>

```

Лістинг файлу registration.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <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:wght@300&display=swap"
rel="stylesheet">
    <link rel="stylesheet" href="../../style.css">

```

```

    <title>BDR</title>
</head>
<body>
<header class="b-header">
    <div>
        <div>
            
        </div>
    </div>
    <nav>
        <ul class="b-header_nav">
            <li class="b-header_nav-item"><a href="../../../index.html" class="b-
header_nav-item-link">Головна</a></li>
            <li class="b-header_nav-item"><a href="../../../pages/catalog.html" class="b-
header_nav-item-link">Каталог</a></li>
            <li class="b-header_nav-item"><a href="../../../pages/authorization.html"
class="b-header_nav-item-link">Увійти/Зареєструватись </a></li>
        </ul>
    </nav>
</header>

<div class="b-form_container b-form_reg">
    <div>
        
    </div>
    <h2 class="b-form_heading">
        Реєстрація
    </h2>
    <form class="b-form_register">
        <label>
            <span class="b-form_fieldname">Логін: </span>
            <input type="text" name="test" class="b-form_field">
        </label>
        <label>
            <span class="b-form_fieldname">Ім'я:</span>
            <input type="text" name="test" class="b-form_field b-form_name">
            <input type="text" name="test" class="b-form_field b-form_name">
            <div class="b-form_name-names">
                <div class="b-form_name-names-name">Ім'я</div>
                <div class="b-form_name-names-surname">Прізвище</div>
            </div>
        </label>
        <label>
            <span class="b-form_fieldname">Вік: </span>
            <input type="text" name="test" class="b-form_field">
        </label>
    </form>

```

```

<label>
  <span class="b-form_fieldname">E-mail: </span>
  <input type="text" name="test" class="b-form_field">
</label>
<label>
  <span class="b-form_fieldname">Місце роботи/навчання: </span>
  <input type="text" name="test" class="b-form_field">
</label>
<label>
  <span class="b-form_fieldname">Пароль: </span>
  <input type="password" name="test" class="b-form_field">
</label>
<label>
  <span class="b-form_fieldname">Повторіть пароль: </span>
  <input type="password" name="test" class="b-form_field">
</label>
  <button class="b-form_btn">Створити обліковий запис</button><br/>
</form>
</div>

<footer class="b-footer">
  <div>
    
  </div>
  <h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>

<script src="../js/register.js"></script>
</body>
</html>

```

Лістинг файлу catalog.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <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:wght@300&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="../style.css">
  <title>BDR</title>
</head>

```

```

<body>
<header class="b-header">
  <div>
    <div>
      
    </div>
  </div>
  <nav>
    <ul class="b-header_nav">
      <li class="b-header_nav-item"><a href="../../../index.html" class="b-
header_nav-item-link">Головна</a></li>
      <li class="b-header_nav-item"><a href="../../../pages/catalog.html" class="b-
header_nav-item-link">Каталог</a></li>
      <li class="b-header_nav-item"><a href="../../../pages/authorization.html"
class="b-header_nav-item-link js-auth-link">Увійти/Зареєструватись </a></li>
    </ul>
  </nav>
</header>
<section class="b-catalog b-maincatalog b-catalog_fix">
  <div class="b-catalog_header">
    <h2 class="b-catalog_header-heading">Каталог тестів</h2>
    <label><input type="text" class="b-catalog_header-search"
placeholder="Пошук..."><span class="b-catalog_header-search-
icon">&#8981;</span></label>
  </div>

  <div class="b-catalog_tests">

  </div>
</section>

<footer class="b-footer">
  <div>
    
  </div>
  <h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>

<script src="../../../js/catalog.js"></script>
</body>
</html>

```

ЛІСТИНГ файлу rating.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <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:wght@300&display=swap"
rel="stylesheet">
  <link rel="stylesheet" href="../style.css">
  <title>BDR</title>
</head>
<body>
<header class="b-header">
  <div>
    <div>
      
    </div>
  </div>
  <nav>
    <ul class="b-header_nav">
      <li class="b-header_nav-item"><a href="../index.html" class="b-
header_nav-item-link">Головна</a></li>
      <li class="b-header_nav-item"><a href="catalog.html" class="b-header_nav-
item-link">Каталог</a></li>
      <li class="b-header_nav-item"><a href="../pages/createtest.html"
class="b-header_nav-item-link">Створити тестування</a></li>
    </ul>
  </nav>
</header>

<section class="b-rating">
  <div class="b-rating_users">
    <h2 class="b-rating_users-heading">Рейтинг користувачів</h2>
    <table class="b-rating_users">

      </table>
  </div>

  <div class="b-rating_tests">
    <h2 class="b-rating_tests-heading">Рейтинг тестів</h2>
    <table class="b-rating_tests">
      </table>
  </div>
</section>

```

```

<footer class="b-footer">
  <div>
    
  </div>
  <h4 class="b-footer_author">&copy; 2022 Dana Marenko</h4>
</footer>

<script src="../../js/rating.js"></script>

</body>
</html>

```

ЛІСТИНГ файлу style.css

```

body {
  margin: 0;
  background-color: #212121;
  color: #E8D58C;
  font-family: 'Inter', sans-serif; }

.b-header {
  display: flex;
  justify-content: space-between;
  padding: 30px 40px 0; }
.b-header_img {
  width: 187px;
  height: 44px; }
.b-header_nav {
  list-style: none;
  margin: 0; }
.b-header_nav-item {
  display: inline;
  margin-left: 40px;
  font-weight: 800;
  font-size: 20px;
  line-height: 24px; }
.b-header_nav-item-link {
  text-decoration: none;
  color: #E8D58C; }
.b-header_nav-item-link:hover {
  text-decoration: underline; }

.b-login_icon {
  width: 43px;
  height: 38px;
  position: relative;

```

```
top: 10px; }

.b-start {
margin: 100px 40px 100px;
display: grid;
grid-template-columns: 60% 40%; }
.b-start_image {
width: 100%; }
.b-start_desc {
width: 350px;
margin: 100px 0 0 60px; }
.b-start_desc-header {
font-size: 26px; }
.b-start_desc-text {
margin-top: 35px;
font-size: 20px;
line-height: 26px; }
.b-start_btn {
background-color: #E8D58C;
margin-top: 40px;
width: 250px;
height: 45px;
font-weight: bold;
font-size: 18px;
border-radius: 15px;
position: relative;
left: 50px; }

.b-catalog_tests {
display: flex;
justify-content: space-around; }

.b-catalog_heading {
text-align: center;
margin-bottom: 50px; }

.b-catalog_test {
text-align: center; }
.b-catalog_test-image {
width: 200px;
height: 200px;
margin-bottom: 25px; }
.b-catalog_test-name {
margin: 0;
font-size: 24px; }
.b-catalog_test-desc {
margin: 0; }
```

```
    font-size: 20px; }

.b-catalog_btn {
  background-color: #E8D58C;
  margin-top: 40px;
  width: 250px;
  height: 45px;
  font-weight: bold;
  font-size: 18px;
  border-radius: 15px;
  position: relative;
  left: 42%; }

.b-catalog_info-heading {
  margin-top: 100px;
  text-align: center;
  font-size: 32px; }

.b-catalog_info-text {
  margin-top: 30px;
  text-align: center;
  font-size: 24px; }

.b-create {
  margin: 100px 150px 75px;
  display: grid;
  grid-template-columns: 40% 60%; }
  .b-create_desc {
    width: 350px;
    margin-top: 60px; }
    .b-create_desc-heading {
      font-size: 26px; }
    .b-create_desc-text {
      font-size: 20px; }
    .b-create_desc-btn {
      background-color: #E8D58C;
      margin-top: 40px;
      width: 250px;
      height: 45px;
      font-weight: bold;
      font-size: 18px;
      border-radius: 15px;
      position: relative;
      left: 40px; }
  .b-create_img {
    width: 100%; }
```



```

.b-footer {
  background-color: #E8D58C;
  display: flex;
  justify-content: space-between;
  color: #212121;
  padding-top: 20px;
  height: 100px;
  width: 100%; }
  .b-footer_container {
    padding-right: 250px; }
  .b-footer_logo {
    width: 187px;
    height: 43px;
    margin: 20px 20px 0; }
  .b-footer_author {
    margin: 60px 15px 0 0; }

.b-form_heading {
  text-align: center; }

.b-form_logo {
  width: 185px;
  height: 45px;
  margin-left: 50px; }

.b-form_container {
  margin: 100px auto;
  padding: 140px;
  width: 300px;
  height: 300px;
  border: 2px #E8D58C solid; }

.b-form_reg, .b-form_change {
  width: 550px;
  height: 750px;
  padding-top: 80px; }
  .b-form_reg .b-form_register, .b-form_reg .b-form_changeinfo, .b-form_change .b-
form_register, .b-form_change .b-form_changeinfo {
    width: 600px; }
  .b-form_reg .b-form_logo, .b-form_change .b-form_logo {
    margin: 30px 0 50px 180px; }
  .b-form_reg .b-form_name, .b-form_change .b-form_name {
    width: 190px !important; }
  .b-form_reg .b-form_name:not(:first-of-type), .b-form_change .b-
form_name:not(:first-of-type) {
    margin-left: 15px; }
  .b-form_reg .b-form_name-names, .b-form_change .b-form_name-names {

```

```

    display: flex;
    padding-left: 160px;
    font-size: 14px; }
    .b-form_reg .b-form_name-names-surname, .b-form_change .b-form_name-names-
surname {
    margin-left: 200px; }
    .b-form_reg .b-form_date, .b-form_change .b-form_date {
    width: 112px !important; }
    .b-form_reg .b-form_date:not(:first-of-type), .b-form_change .b-
form_date:not(:first-of-type) {
    margin-left: 15px; }
    .b-form_reg .b-form_date-names, .b-form_change .b-form_date-names {
    display: flex;
    padding-left: 160px;
    font-size: 14px; }
    .b-form_reg .b-form_date-names-month, .b-form_change .b-form_date-names-month {
    margin-left: 125px; }
    .b-form_reg .b-form_date-names-year, .b-form_change .b-form_date-names-year {
    margin-left: 110px; }
    .b-form_reg .b-form_field, .b-form_change .b-form_field {
    width: 420px; }
    .b-form_reg .b-form_fieldname, .b-form_change .b-form_fieldname {
    display: inline-block;
    width: 140px; }
    .b-form_reg .b-form_btn, .b-form_change .b-form_btn {
    width: 350px;
    margin-top: 70px;
    left: 20%; }

.b-form_field {
    border: 1px #E8D58C solid;
    border-radius: 25px;
    color: #E8D58C;
    padding: 10px;
    width: 280px;
    height: 20px;
    margin-top: 35px;
    background-color: #212121; }

.b-form_link {
    color: #E8D58C;
    text-decoration: none;
    border-bottom: 1px #E8D58C solid; }

.b-form_noacc {
    color: #fff;
    text-align: center; }

```

```
.b-form_btn {
  background-color: #E8D58C;
  margin-top: 40px;
  width: 250px;
  height: 45px;
  font-weight: bold;
  font-size: 18px;
  border-radius: 15px;
  width: 150px;
  position: relative;
  left: 27%; }

.b-profile {
  display: grid;
  padding: 0 40px;
  grid-template-columns: 25% 75%; }
  .b-profile_heading {
    padding: 0 40px;
    font-weight: normal; }
  .b-profile_nav {
    border: 2px #E8D58C solid;
    height: 100px; }
    .b-profile_nav-list {
      list-style: none; }
      .b-profile_nav-list-item:not(:first-of-type) {
        margin-top: 25px; }
      .b-profile_nav-list-link {
        color: #E8D58C;
        text-decoration: none; }
  .b-profile_info {
    margin-left: 150px; }
    .b-profile_info-user {
      display: flex; }
      .b-profile_info-user-data {
        margin-left: 40px; }
      .b-profile_info-user-login {
        margin-top: 0; }
      .b-profile_info-user-pfp {
        width: 228px;
        height: 228px; }
    .b-profile_info-feed-input {
      border: 1px solid #E8D58C;
      color: #E8D58C;
      background-color: #212121; }
    .b-profile_info-feed-btn {
      background-color: #E8D58C;
```

```

    margin-top: 40px;
    width: 250px;
    height: 45px;
    font-weight: bold;
    font-size: 18px;
    border-radius: 15px;
    width: 150px;
    height: 35px;
    font-size: 14px;
    margin-top: 5px;
    position: relative;
    left: 49%; }

.b-feed {
margin-bottom: 100px; }
.b-feed_item {
border-bottom: 1px solid #E8D58C;
width: 603px; }
.b-feed_item-date {
text-align: right;
margin: 5px 0; }
.b-feed_item-pfp {
width: 50px;
height: 50px;
margin-right: 10px; }
.b-feed_item-info {
display: flex; }
.b-feed_item-login {
margin: 0; }
.b-feed_item-text {
margin: 5px 0;
color: #fff; }

.b-catalog_fix {
padding: 0 40px;
height: 100vh; }

.b-maincatalog .b-catalog_header {
display: flex;
justify-content: space-between; }
.b-maincatalog .b-catalog_header-heading {
font-weight: normal; }
.b-maincatalog .b-catalog_header-search {
width: 200px;
height: 30px;
padding-left: 10px;
background-color: #212121;

```

```

color: #E8D58C;
border: 1px solid #E8D58C;
border-radius: 25px;
margin-top: 16px; }
.b-maincatalog .b-catalog_header-search::-webkit-input-placeholder {
padding-left: 10px;
font-size: 14px; }
.b-maincatalog .b-catalog_header-search-icon {
font-size: 32px;
position: absolute;
right: 50px;
top: 88px; }

.b-maincatalog .b-catalog_tests {
display: grid;
grid-template-columns: 33% 33% 33%; }
.b-maincatalog .b-catalog_tests-test {
display: flex;
margin-top: 35px; }
.b-maincatalog .b-catalog_tests-test-name {
margin-top: 0; }
.b-maincatalog .b-catalog_tests-test-desc {
font-size: 16px;
margin-top: 20px;
width: 250px; }
.b-maincatalog .b-catalog_tests-test-tags {
font-size: 16px;
margin-top: 20px; }
.b-maincatalog .b-catalog_tests-test-img {
width: 150px;
height: 150px;
margin-right: 20px; }

.b-rating {
margin: 0 40px 550px;
display: flex;
justify-content: space-around;
font-size: 18px; }
.b-rating_users, .b-rating_tests {
border-collapse: collapse; }
.b-rating_head, .b-rating_body {
border: 1px solid #E8D58C; }
.b-rating_head-item, .b-rating_body-item {
border-right: 1px #E8D58C solid;
padding: 5px 10px; }
.b-rating_head-number, .b-rating_body-number {
width: 40px; }

```

```
.b-rating_head-login, .b-rating_body-login {
  width: 240px; }
.b-rating_head-rating, .b-rating_body-rating {
  width: 90px; }

.b-createtest {
  width: 50%;
  margin: 0 auto; }
.b-createtest_heading {
  text-align: center; }
.b-createtest_form {
  width: 65%;
  margin: 0 auto; }
.b-createtest_input {
  border: 1px #E8D58C solid;
  border-radius: 25px;
  color: #E8D58C;
  padding: 10px;
  width: 280px;
  height: 20px;
  margin-top: 35px;
  background-color: #212121; }
  .b-createtest_input-name {
    display: inline-block;
    width: 150px; }
  .b-createtest_input-desc-name {
    vertical-align: top;
    margin-top: 50px; }
.b-createtest_desc {
  height: auto; }
.b-createtest_questions {
  margin: 0 auto; }
  .b-createtest_questions-heading {
    text-align: center; }
.b-createtest_question-add-img {
  width: 35px;
  height: 35px;
  position: relative;
  top: 10px; }
.b-createtest_question-input {
  border: 1px #E8D58C solid;
  border-radius: 25px;
  color: #E8D58C;
  padding: 10px;
  width: 100%;
  height: 20px;
  margin-top: 10px; }
```

```

    background-color: #212121; }
.b-createtest_question-btn {
    background-color: #E8D58C;
    margin-top: 40px;
    width: 250px;
    height: 45px;
    font-weight: bold;
    font-size: 18px;
    border-radius: 15px;
    width: 250px;
    height: 35px;
    position: relative;
    left: 110px; }
.b-createtest_question-answers {
    width: 380px;
    margin: 0 auto; }
    .b-createtest_question-answers-item {
        border: 1px #E8D58C solid;
        border-radius: 25px;
        color: #E8D58C;
        padding: 10px;
        width: 300px;
        height: 20px;
        background-color: #212121; }
    .b-createtest_question-answers-radio {
        position: relative;
        accent-color: #E8D58C;
        width: 30px;
        height: 30px;
        top: 10px; }
    .b-createtest_question-answers-btn {
        background-color: #E8D58C;
        margin-top: 40px;
        width: 250px;
        height: 45px;
        font-weight: bold;
        font-size: 18px;
        border-radius: 15px;
        margin-top: 30px;
        font-size: 16px;
        width: 200px;
        height: 30px;
        position: relative;
        left: 135px; }
.b-createtest_labelname {
    margin: 35px 0 0; }
.b-createtest_labels {

```

```
    display: flex;
    justify-content: space-between; }

.b-test {
width: 30%;
margin: 0 auto; }
.b-test_header {
    text-align: center; }
.b-test_question-number {
    text-align: center; }
.b-test_question-answers {
padding: 0;
list-style: none; }
.b-test_question-answers-item {
    font-size: 18px; }
.b-test_question-answers-correct {
    width: 15px;
    height: 15px;
    accent-color: #E8D58C; }
.b-test_submit {
background-color: #E8D58C;
margin-top: 40px;
width: 250px;
height: 45px;
font-weight: bold;
font-size: 18px;
border-radius: 15px;
position: relative;
left: 40%;
margin-bottom: 100px; }

.b-result {
border: 2px #E8D58C solid;
text-align: center;
width: 40%;
height: 500px;
padding-top: 100px;
margin: 50px auto 75px;
font-size: 18px; }
.b-result_logo {
    width: 135px;
    height: 35px; }
.b-result_grade, .b-result_right-answers-count {
    font-size: 32px;
    font-weight: bold; }
.b-result_save {
background-color: #E8D58C;
```



```
margin-top: 40px;
width: 250px;
height: 45px;
font-weight: bold;
font-size: 18px;
border-radius: 15px; }
```

Лістинг файлу changeinfo.js

```
document.cookie = 'page=Зміна профілю';

function getUserFormData() {
  return {
    userLogin: document.querySelector('.js-form_login'),
    userName: document.querySelector('.js-form_name'),
    userSurname: document.querySelector('.js-form_surname'),
    userAge: document.querySelector('.js-form_age'),
    userEmail: document.querySelector('.js-form_email'),
    userInstitution: document.querySelector('.js-form_institution'),
    userPassword: document.querySelector('.js-form_password'),
    userPasswordAgain: document.querySelector('.js-form_passwordAgain')
  }
}

function fillUserInfo() {
  const userFormData = getUserFormData();
  const login = getCookie('userLogin');

  fetch(`http://localhost:8080/user/${login}`)
  .then(res => res.json())
  .then(data => {
    userFormData.userLogin.value = login;
    userFormData.userName.value = data.name;
    userFormData.userSurname.value = data.surname;
    userFormData.userAge.value = data.age;
    userFormData.userEmail.value = data.email;
    userFormData.userInstitution.value = data.institution;
  })
}

function getCookie(name) {
  const value = `; ${document.cookie}`;
  const parts = value.split(`; ${name}=`);
  if (parts.length === 2) return parts.pop().split(';').shift();
}
```

```
fillUserInfo();
```

Лістинг файлу PostServiceImpl.java

```
package com.vntu.marenko.ualearning.server.service.impl;

import com.vntu.marenko.ualearning.server.model.Post;
import com.vntu.marenko.ualearning.server.repository.PostRepository;
import com.vntu.marenko.ualearning.server.service.PostService;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PostServiceImpl implements PostService {

    private final PostRepository postRepository;

    public PostServiceImpl(PostRepository postRepository) {
        this.postRepository = postRepository;
    }

    @Override
    public Post savePost(Post post) {
        return postRepository.save(post);
    }

    @Override
    public Post getPost(int id) {
        return postRepository.getReferenceById(id);
    }

    @Override
    public List<Post> getPostsForUser(String userId) {
        return postRepository.findByUser(userId);
    }
}
```

Лістинг файлу TestServiceImpl.java

```
package com.vntu.marenko.ualearning.server.service.impl;

import com.vntu.marenko.ualearning.server.dto.CreateTestRequest;
import com.vntu.marenko.ualearning.server.model.Test;
import com.vntu.marenko.ualearning.server.repository.TagRepository;
import com.vntu.marenko.ualearning.server.repository.TestRepository;
```

```
import com.vntu.marenko.ualearning.server.service.TestService;
import ma.glasnost.orika.MapperFacade;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class TestServiceImpl implements TestService {

    private final TestRepository testRepository;

    private final TagRepository tagRepository;

    private final MapperFacade mapper;

    public TestServiceImpl(TestRepository testRepository,
                          TagRepository tagRepository,
                          MapperFacade mapper) {
        this.testRepository = testRepository;
        this.tagRepository = tagRepository;
        this.mapper = mapper;
    }

    @Override
    public List<Test> getAllTests() {
        return testRepository.findAll();
    }

    @Override
    public List<Test> getTestsWithTag(int tagId) {
        return testRepository.findAllByTags(tagRepository.getReferenceById(tagId));
    }

    @Override
    public Test getTest(int id) {
        return testRepository.getReferenceById(id);
    }

    @Override
    public Test createTest(CreateTestRequest testRequest) {
        Test testToSave = mapper.map(testRequest, Test.class);
        testToSave.setTags(tagRepository.findAllById(testRequest.getTags()));
        return testRepository.save(testToSave);
    }
}
```

Лістинг файлу UserServiceImpl.java

```

package com.vntu.marenko.ualearning.server.service.impl;

import com.vntu.marenko.ualearning.server.exception.UnauthorisedException;
import com.vntu.marenko.ualearning.server.exception.ValidationException;
import com.vntu.marenko.ualearning.server.model.User;
import com.vntu.marenko.ualearning.server.repository.UserRepository;
import com.vntu.marenko.ualearning.server.service.UserService;
import org.springframework.data.domain.PageRequest;
import org.springframework.data.domain.Pageable;
import org.springframework.data.domain.Sort;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Optional;

@Service
public class UserServiceImpl implements UserService {

    private static final String CORRECT_PASSWORD = "^(?=.*[A-Za-z])(?=.*\\d)[A-Za-z\\d]{8,}$";

    private final UserRepository userRepository;

    public UserServiceImpl(UserRepository userRepository) {
        this.userRepository = userRepository;
    }

    @Override
    public String login(String login, String password) {
        Optional<User> user = userRepository.findById(login);
        if (user.isPresent() &&
            user.get().getLogin().equals(login) &&
            user.get().getPassword().equals(password)) {
            return login.concat("_").concat(password);
        }
        throw new UnauthorisedException();
    }

    @Override
    public String register(User user) {
        if (userRepository.existsById(user.getLogin())) {
            throw new ValidationException("Помилка! Такий користувач уже існує");
        }
        if (user.getPassword().matches(CORRECT_PASSWORD)){
            if (user.getAge() > 12) {

```

```

        userRepository.save(user);
        return login(user.getLogin(), user.getPassword());
    } else {
        throw new ValidationException("Помилка! Ваш вік замалий для
реєстрації");
    }
}
else{
    throw new ValidationException("Помилка! Введено некоректний пароль");
}
}

@Override
public User getUser(String id) {
    return userRepository.getReferenceById(id);
}

@Override
public List<User> top10() {
    Pageable pageable = PageRequest.of(0, 10, Sort.by(Sort.Direction.DESC,
"rating"));
    return userRepository.findAll(pageable).getContent();
}

@Override
public User updateUser(String login, User userToSave) {
    User fromDb = userRepository.getReferenceById(login);
    if (userToSave.getLogin() != null) {
        fromDb.setLogin(userToSave.getLogin());
    }
    if (userToSave.getPassword() != null) {
        fromDb.setPassword(userToSave.getPassword());
    }
    if (userToSave.getName() != null) {
        fromDb.setName(userToSave.getName());
    }
    if (userToSave.getSurname() != null) {
        fromDb.setSurname(userToSave.getSurname());
    }
    if (userToSave.getAge() != 0) {
        if (userToSave.getAge() > 12) {
            fromDb.setAge(userToSave.getAge());
        } else {
            throw new ValidationException();
        }
    }
    if (userToSave.getInstitution() != null) {

```

```

        fromDb.setInstitution(userToSave.getInstitution());
    }
    if (userToSave.getEmail() != null) {
        fromDb.setEmail(userToSave.getEmail());
    }
    return userRepository.save(fromDb);
}
}

```

Лістинг файлу AuthorisationController.java

```

package com.vntu.marenko.ualearning.server.controller;

import com.vntu.marenko.ualearning.server.dto.LoginRequest;
import com.vntu.marenko.ualearning.server.dto.UserRequest;
import com.vntu.marenko.ualearning.server.model.User;
import com.vntu.marenko.ualearning.server.service.UserService;
import ma.glasnost.orika.MapperFacade;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class AuthorisationController {

    private final UserService userService;

    private final MapperFacade mapper;

    public AuthorisationController(UserService userService, MapperFacade mapper) {
        this.userService = userService;
        this.mapper = mapper;
    }

    @PostMapping("/login")
    public String login(@RequestBody LoginRequest request) {
        return userService.login(request.getLogin(), request.getPassword());
    }

    @PostMapping("/register")
    public String register(@RequestBody UserRequest request) {
        return userService.register(mapper.map(request, User.class));
    }
}

```

Лістинг файлу PostController.java

```

package com.vntu.marenko.ualearning.server.controller;

import com.vntu.marenko.ualearning.server.dto.CreatePostRequest;
import com.vntu.marenko.ualearning.server.dto.PostDto;
import com.vntu.marenko.ualearning.server.model.Post;
import com.vntu.marenko.ualearning.server.service.PostService;
import ma.glasnost.orika.MapperFacade;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("post")
public class PostController {

    private final MapperFacade mapper;

    private final PostService postService;

    public PostController(MapperFacade mapper, PostService postService) {
        this.mapper = mapper;
        this.postService = postService;
    }

    @PostMapping
    public PostDto createPost(@RequestBody CreatePostRequest request) {
        Post toSave = mapper.map(request, Post.class);
        Post saved = postService.savePost(toSave);
        return mapper.map(saved, PostDto.class);
    }

    @GetMapping("/{id}")
    public PostDto getPost(@PathVariable int id) {
        return mapper.map(postService.getPost(id), PostDto.class);
    }

    @GetMapping("/user/{userId}")
    public List<PostDto> getPostsForUser(@PathVariable String userId) {
        List<Post> posts = postService.getPostsForUser(userId);
        posts.sort((p1, p2) -> p2.getTimestamp().compareTo(p1.getTimestamp()));
        return mapper.mapAsList(posts, PostDto.class);
    }
}

```

Лістинг файлу UserController.java

```

package com.vntu.marenko.ualearning.server.controller;

import com.vntu.marenko.ualearning.server.dto.UserDto;
import com.vntu.marenko.ualearning.server.dto.UserRequest;
import com.vntu.marenko.ualearning.server.model.User;
import com.vntu.marenko.ualearning.server.service.UserService;
import ma.glasnost.orika.MapperFacade;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@RequestMapping("user")
public class UserController {

    private final UserService userService;

    private final MapperFacade mapper;

    public UserController(UserService userService, MapperFacade mapper) {
        this.userService = userService;
        this.mapper = mapper;
    }

    @GetMapping("/{id}")
    public UserDto getUser(@PathVariable String id) {
        return mapper.map(userService.getUser(id), UserDto.class);
    }

    @GetMapping("/rating")
    public List<UserDto> rating() {
        return mapper.mapAsList(userService.top10(), UserDto.class);
    }

    @PutMapping("/{userId}")
    public UserDto updateUser(@PathVariable String userId, @RequestBody UserRequest
request) {
        return mapper.map(userService.updateUser(userId,
                                                    mapper.map(request,
                                                                    User.class)),
                           UserDto.class);
    }
}

```


Додаток Д – Графічна частина

ГРАФІЧНА ЧАСТИНА
РОЗРОБКА СОЦІАЛЬНОЇ МЕРЕЖІ ДЛЯ ОНЛАЙН-НАВЧАННЯ
КОРИСТУВАЧІВ

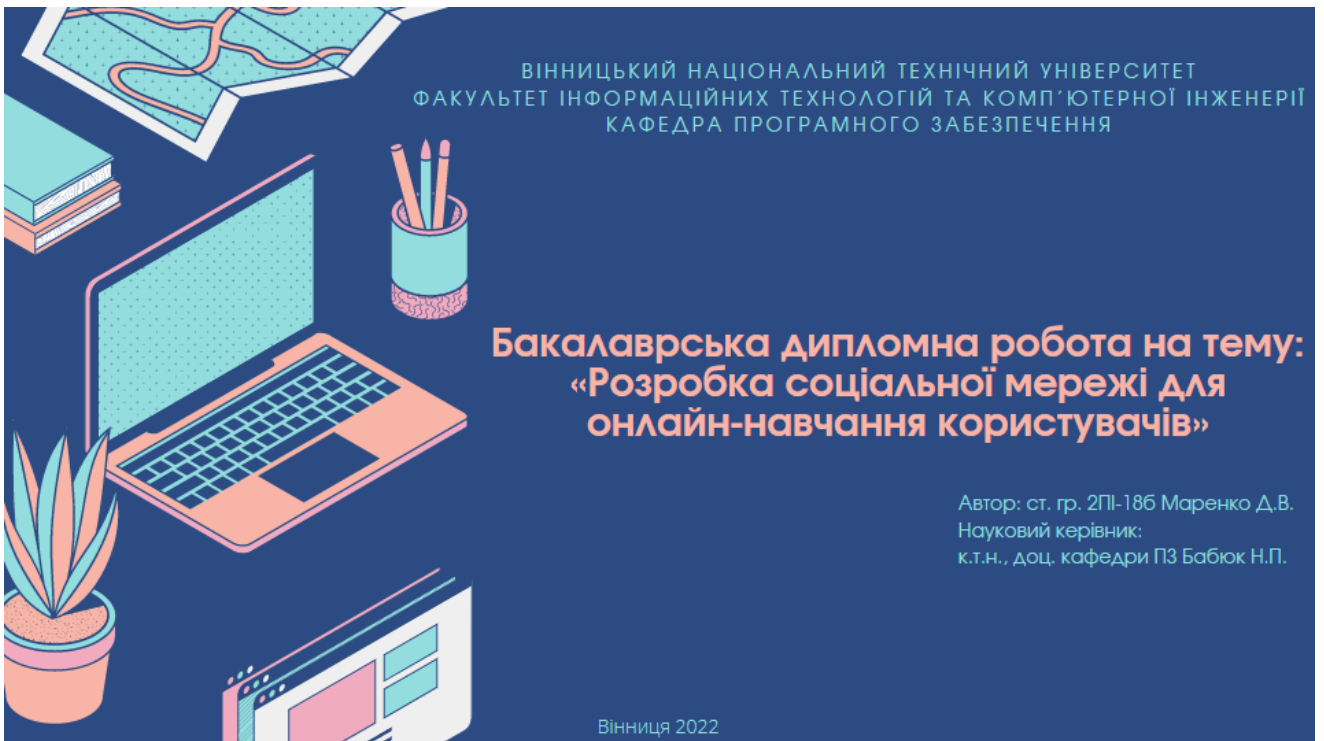


Рисунок Д.1 – Титульний слайд



Рисунок Д.2 – Мета, предмет та об'єкт дослідження

Актуальність розробки

- ЧИМАЛИЙ РОЗВИТОК ДИСТАНЦІЙНОГО НАВЧАННЯ;
- ВІДСУТНІСТЬ ПОПУЛЯРНОЇ УКРАЇНСЬКОЇ СОЦІАЛЬНОЇ МЕРЕЖІ;
- МАЛА КІЛЬКІСТЬ ПОШИРЕНИХ ОНЛАЙН-ТЕСТУВАНЬ УКРАЇНСЬКОЮ МОВОЮ;
- ПОЄДНАННЯ НАВЧАННЯ ТА ДОЗВІЛЛЯ.

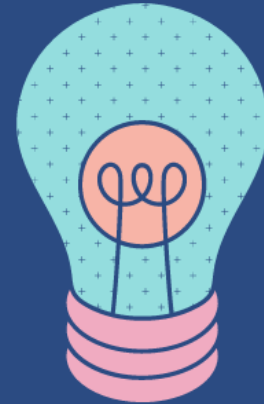


Рисунок Д.3 – Актуальність розробки

Аналіз аналогів

Критерії оцінювання	Stepik	Coursera	Udemy	Jamm School	UAlearning
Можливість створювати тестування	-	-	+	-	+
Можливість спілкуватися з іншими користувачами	-	-	-	+	+
Підтримується українська мова	-	-	-	+	+
Безкоштовно	+/-	+/-	-	-	+
Рейтинг користувачів	+	-	-	+	+
Власний акаунт	+	+	+	-	+

Рисунок Д.4 – Аналіз аналогів



Рисунок Г.5 – Новизна

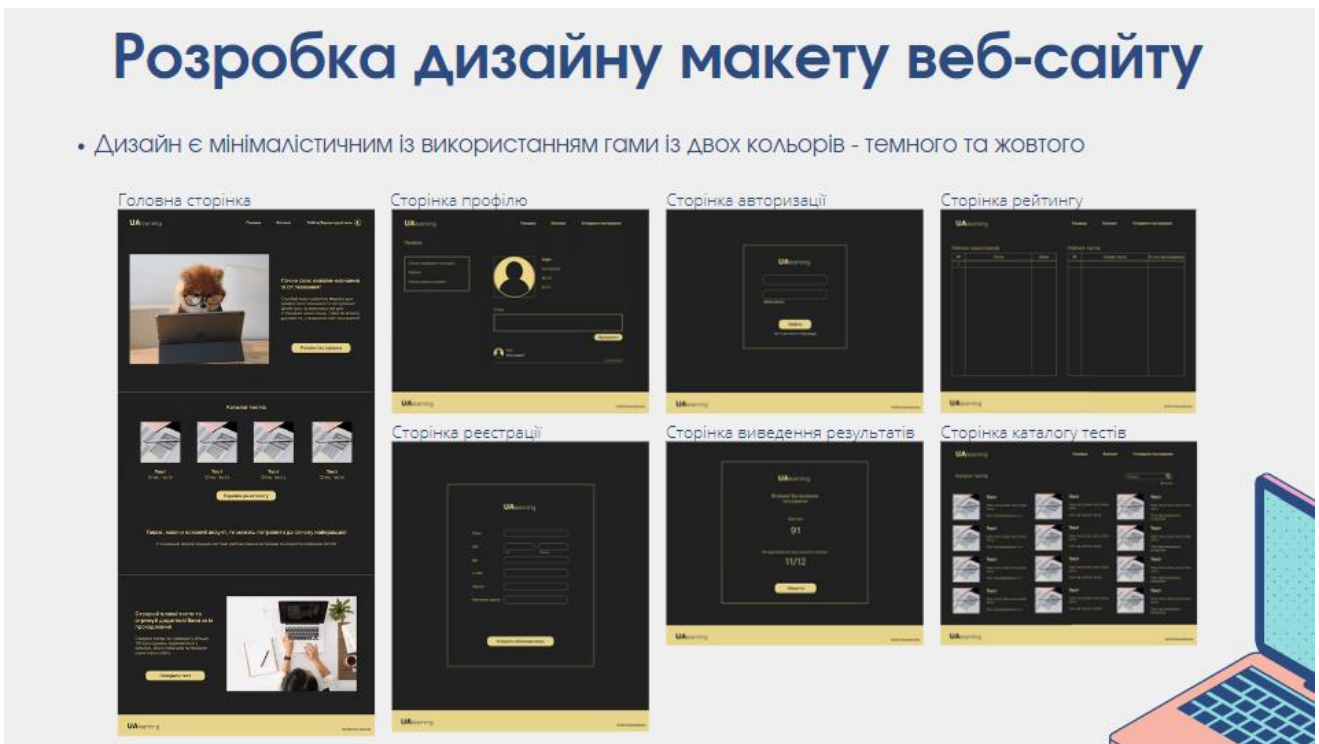


Рисунок Д.6 – Розробка дизайну макету веб-сайту



Рисунок Д.7 – Використані технології

ER-модель бази даних соціальної мережі

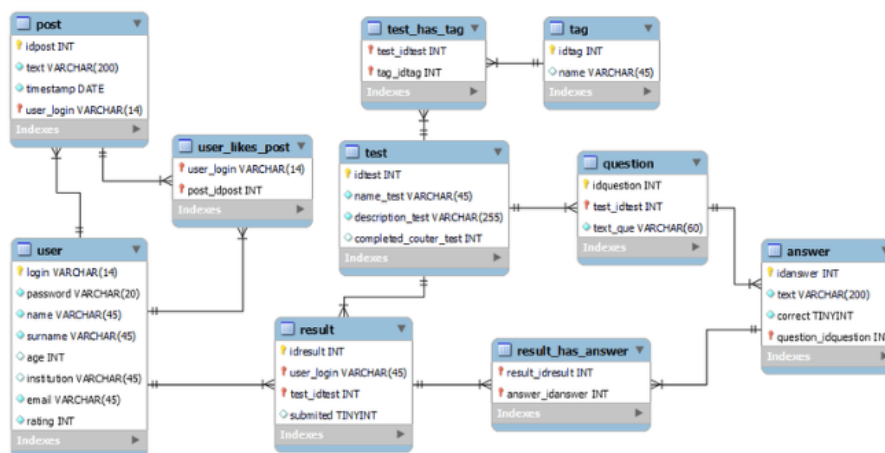


Рисунок Д.8 – ER-модель бази даних соціальної мережі

Блок-схема алгоритму реєстрації облікового запису користувача

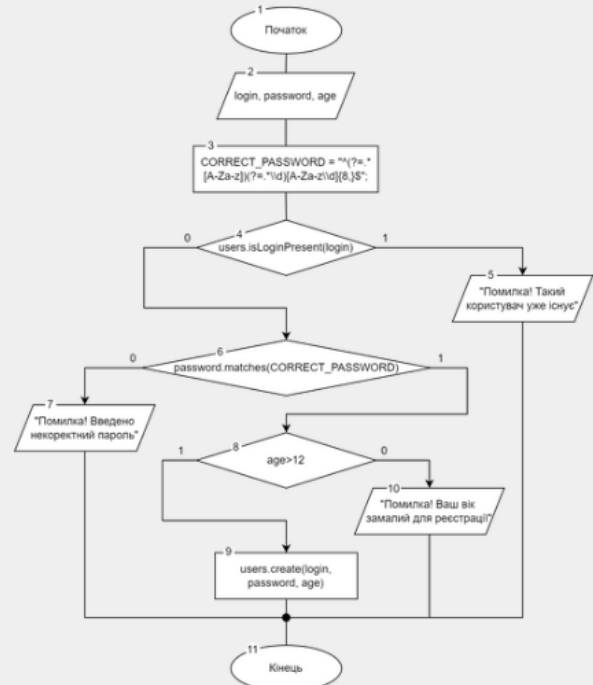


Рисунок Д.9 – Блок-схема алгоритму реєстрації облікового запису користувача

Блок-схема алгоритму фільтрації тестів за тегами

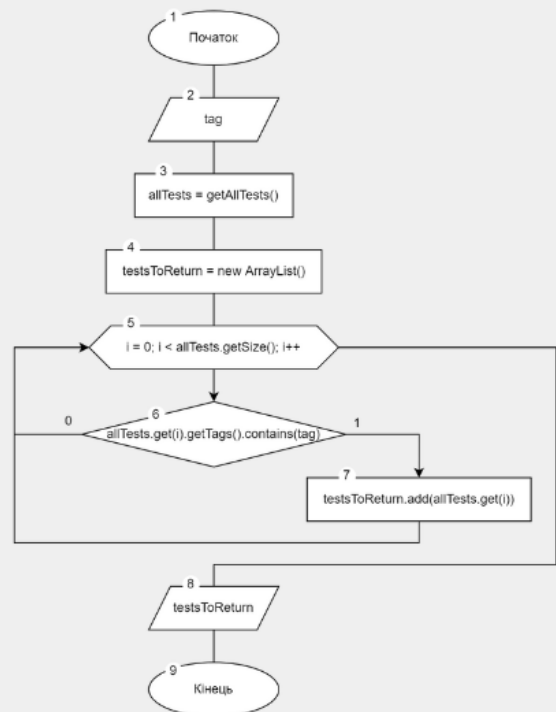


Рисунок Д.10 – Блок-схема алгоритму фільтрації тестів за тегами



Рисунок Д.11 – Блок-схема алгоритму підрахунку результату тестування

Тестування веб-додатку

Головна сторінка

Авторизація

Редагування профілю

Рейтинг

Рейтинг користувачів

№	Логін	Бали
1	user1	1337
2	babuk_MP	1108
3	leptka	900
4	sitchik	777
5	kahatniak	524
6	danke	357
7	ivan_ID	322
8	prkol	228

Рейтинг тестів

№	Назва тесту	К-сть проходжень
1	Українська мова	69
2	Стартали	48
3	Патерни програмування	36
4	Англійська мова	25
5	Основи JavaScript	13
6	SQL - запити	7
7	Нейронні мережі	3

Рисунок Д.12 – Тестування веб-додатку

Апробація та публікації результатів роботи

Матеріали бакалаврської кваліфікаційної роботи доповідалися на:

всеукраїнській науково-практичній інтернет-конференції «Молодь в науці: дослідження, проблеми, перспективи (МН-2022)» (2022 р., м. Вінниця)

За тематикою дослідження опубліковано 2 наукові праці у збірниках матеріалів конференції

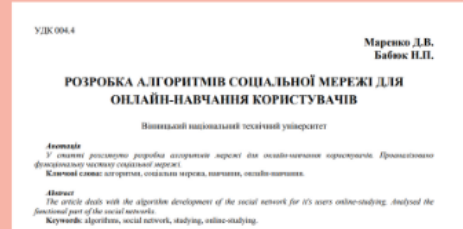
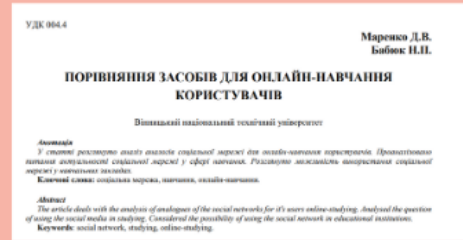


Рисунок Д.13 – Апробація та публікації результатів роботи

Впровадження

Впровадження результатів дослідження підтверджуються відповідним актом та використовуються на ТОВ «ЮК Кристал» для

- комунікації між структурними підрозділами;
- підвищення кваліфікації працівників.

Рисунок Д.14 – Впровадження



Рисунок Д.15 – Результати роботи

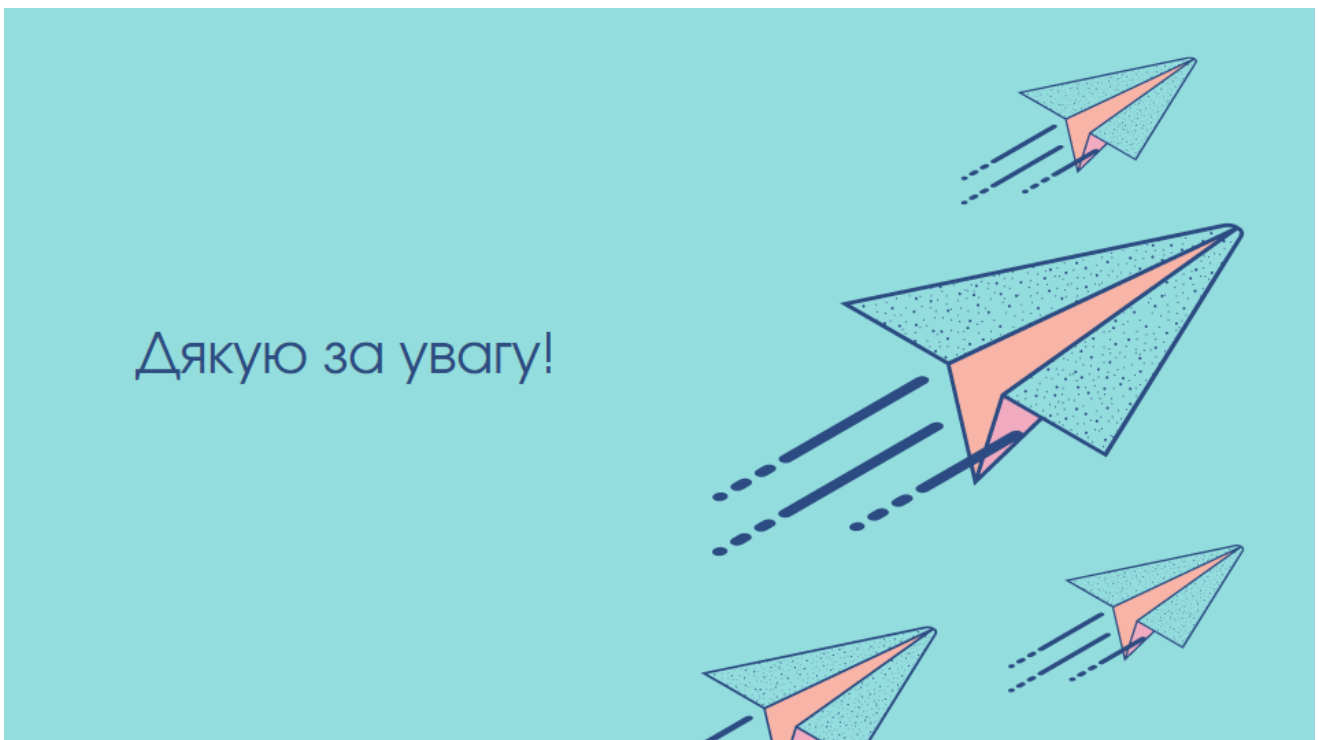


Рисунок Д.16 – Фінальний слайд