

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інформаційних технологій та комп'ютерної інженерії
(повне найменування інституту, назва факультету (відділення))


Кафедра обчислювальної техніки
(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

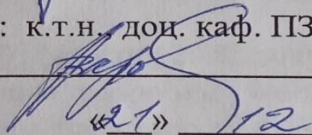
на тему:

«Комплекс спеціалізованих програмних засобів для здобуття професійних знань в галузі програмного забезпечення»


Виконав: студент 2-го курсу групи 2КІ-21м
спеціальності 123 «Комп'ютерна інженерія»


Лабчук О. М.
(прізвище та ініціали)

Керівник: к.т.н., доц. каф. ПЗ


Азаров О. Д.
(прізвище та ініціали)

Опонент: д.т.н., голова секції уб. каф. МБІС


Карпінець В. В.
(прізвище та ініціали)

«22» 12 2022 р.

Допущено до захисту

завідувач кафедри КІ

д.т.н. проф. Азаров О. Д.
(прізвище та ініціали)

«23» 12 2022 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра програмного забезпечення
Рівень вищої освіти II-й (магістерський)
Галузь знань – 12 Інформаційні системи
Спеціальність 123 – «Комп'ютерна інженерія»
Освітньо-професійна програма – Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ

Азаров О.Д.

“15” вересня 2022 року

ЗАВДАННЯ НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Лабчуку Олександрю Михайловичу

1. Тема роботи: «Комплекс спеціалізованих засобів для здобуття професійних знань в галузі програмного забезпечення»

Керівник роботи: Азаров Олексій Дмитрович, к.т.н., доцент кафедри ПЗ, затверджені наказом вищого навчального закладу від “15” вересня 2022 року № 205-А.

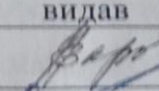
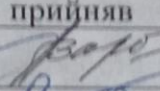
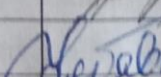
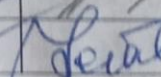
2. Строк подання студентом роботи: “01” грудня 2022 року.

3. Вихідні дані до роботи: середовища розробки WebStorm 2021, мови розробки Java Script та HTML, Css, операційна система – Windows 11, базові алгоритми для створення та проходження навчальних тестів, алгоритм для аналізу результатів та підбору додаткової інформації.

4 Зміст розрахунково-пояснювальної записки: вступ; аналіз та постановка задачі; аналіз даних; розробка структури системи для навчання студентів; розробка архітектури та алгоритмів програмного додатка; розробка програмних модулів для створення тестів, виведення результатів тестування та авторизації, розробка програмного додатку; тестування додатку; економічна частина; висновки; перелік посилань; додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): мета, об'єкт та предмет дослідження; завдання дослідження; аналіз стану питання; порівняння з аналогами; використані технології при розробці системи; тестування системи; економічне обґрунтування; наукова новизна одержаних результатів; практична цінність одержаних результатів, апробація та публікації.

6. Консультанти розділів роботи

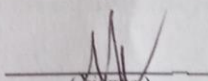
Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	виконання прийняв
1-4	Азаров О.Д. д.т.н., зав. каф.ОТ		
5	Небава М.І., д.е.н., професор ЕПВМ		

7. Дата видачі завдання: "14" вересня 2022 року.

КАЛЕНДАРНИЙ ПЛАН

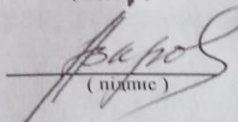
№ з/п	Назва етапів магістерської кваліфікаційної Роботи	Строк виконання етапів роботи	Примітка
1	Аналіз стану питання та постановка задачі дослідження	15.09.22 – 25.09.22	Вик.
2	Розробка методу та моделі системи тестування та системи навчального процесу.	26.09.22 – 28.09.22	Вик.
3	Розробка програмних засобів реалізації веб-сайту	29.09.22 – 12.10.22	Вик.
4	Тестування роботи програмного продукту	13.10.22 – 28.10.22	Вик.
5	Економічна частина	29.11.22 – 20.11.22	Вик.
6	Оформлення матеріалів до захисту МКР	21.11.22 – 25.11.22	Вик.

Студент


 (підпис)

Лабчук О.М.
 (прізвище та ініціали)

Керівник магістерської кваліфікаційної роботи


 (підпис)

Азаров О.Д.
 (прізвище та ініціали)

АНОТАЦІЯ

УДК 004.422.8

Лабчук О. М. Комплекс спеціалізованих засобів для здобуття професійних знань в галузі програмного забезпечення. Магістерська кваліфікаційна робота зі спеціальності 123 – комп'ютерна інженерія, освітня програма – комп'ютерна інженерія. Вінниця: ВНТУ, 2022.

На укр. мові. Бібліогр.: 36 назв; рис.: 29; табл.: 12.

Магістерська кваліфікаційна робота спрямована на розробку і створення веб-застосунку для здобуття професійних знань в галузі програмного забезпечення. Подальшого розвитку отримав метод перевірки набутих знань учнів, який, на відміну від існуючих, забезпечує використання нових задач та отримання оцінки у відсотковій шкалі, що дозволяє підлаштувати даний метод під будь-які вимоги до шкали оцінювання та підвищує універсальність його використання. Подальшого розвитку дістала модель тестування системи, яка, на відміну від існуючих, використовує стандартизовані підходи до підбору запитань та шкали оцінювання і забезпечує дистанційний доступ користувачів за допомогою використання веб-середовища, що розширює коло потенційних користувачів

Для розробки використано мову програмування Java Script, середовище розробки WebStorm, СКБД PostgreSQL та інструментарій GraphQL.

Ключові слова: Навчальна система, айті курси, тестування, перевірка знань, генератор запитань, Java Script, PostgreSQL.

ABSTRACT

UDC 004.422.8

Labchuk O. M. A complex of specialized tools for acquiring professional knowledge in the field of software. Master's qualification thesis on specialty 123 - computer engineering, educational program - computer engineering. Vinnytsia: VNTU, 2022.

In Ukrainian language. Bibliogr .: 36 titles; fig .: 29; tab .: 12.

The master's qualification work is aimed at the development and creation of a web application for the acquisition of professional knowledge in the field of software. The method of checking students' acquired knowledge received further development, which, unlike the existing ones, ensures the use of new tasks and obtaining an assessment on a percentage scale, which allows you to adjust this method to any requirements for the assessment scale and increases the universality of its use. The system testing model was further developed, which, unlike the existing ones, uses standardized approaches to the selection of questions and rating scales and provides remote user access through the use of the web environment, which expands the circle of potential users

Java Script programming language, WebStorm development environment, PostgreSQL database and GraphQL toolkit were used for development.

Keywords: Educational system, IT courses, testing, knowledge test, question generator, Java Script, PostgreSQL

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧ ДОСЛІДЖЕННЯ	11
1.1 Аналіз предметної області.....	11
1.2 Порівняльний аналіз аналогів.....	12
1.3 Аналіз перспектив розвитку програми.....	16
1.4 Аналіз методів реалізації веб-сайту.....	17
1.5 Постановка задач роботи.....	19
2 РОЗРОБКА МЕТОДУ ТА МОДЕЛІ СИСТЕМИ ТЕСТУВАННЯ	20
2.1 Аналіз інформаційного забезпечення.....	19
2.2 Вибір типу інтерфейсу.....	21
2.3 Розробка методу та моделі системи тестування.....	23
2.4 Аналіз вимог до дизайну сайту.....	28
2.5 Вибір колірної гами.....	30
3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ВЕБ-САЙТУ	35
3.1 Варіантний аналіз засобів реалізації сайту.....	35
3.2 Аналіз середовища розробки.....	36
3.3 Розробка систем керування базами даних.....	38
3.4 Програмна реалізація тестової системи.....	40
4 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ	49
4.1 Огляд рішень для тестування сайтів.....	49
4.2 Тестування роботи програмного продукту.....	51
5 ЕКОНОМІЧНА ЧАСТИНА	56
5.1 Комерційний та технологічний аудит науково-технічної розробки.....	56
5.2 Прогнозування витрат на виконання науково-дослідної роботи.....	59

					08-23.МКР.024.00.000 ПЗ							
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>				<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>		
<i>Розробив</i>		Лабчук О. М.			КОМПЛЕКС СПЕЦІАЛІЗОВАНИХ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗДОБУТТЯ ПРОФЕСІЙНИХ ЗНАЬ В ГАЛУЗІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ				6	104		
<i>Перевірів</i>		Азаров О. Д.										
<i>Рецензент</i>		Яремчук Ю.Є.						ВНТУ, гр. 2КІ-21м				
<i>Н.контр.</i>		Швець С. І.										
<i>Затвердж.</i>		Азаров О.Д.										

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором.....	64
ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	72
ДОДАТОК А Технічне завдання.....	74
ДОДАТОК Б Лістинг програми.....	78
ДОДАТОК В Ілюстративний матеріал до захисту бакалаврської дипломної роботи.....	95
ДОДАТОК Г Протокол перевірки роботи.....	104

					08-23.МКР.024.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми дослідження полягає в тому, що технології сьогодні не стоять на місці. Кількість інструментів, які використовуються для створення веб-сайтів, зростає з кожним роком, деякі з них не підтримуються або не використовуються розробниками. Створені ресурси орієнтовані на студентів, які хочуть покращити свої знання в певній галузі або просто хочуть перевірити свої знання. Додаток також корисний для вчителів, які можуть легко створювати тести для перевірки своїх учнів. Системи тестування набувають сьогодні все більшої популярності, оскільки за їх допомогою можна швидко й об'єктивно оцінити рівень знань учня. Тести використовуються скрізь, від невеликих гуртків до вищих навчальних закладів. Тим паче, що сьогодні всі навчальні заклади працюють дистанційно, системи тестування для них обов'язкові.

Такий вид контролю дає вчителям більш точну та об'єктивну інформацію про рівень знань своїх учнів. Тестування за стандартною процедурою може створити однакові умови тестування для всіх учнів. Це дозволяє проводити індивідуальне та диференційоване навчання. Для діагностики успішності дослідження можна використовувати різні форми тестування, але бажано, щоб обрана форма дозволяла швидко отримувати результати перевірки знань, умінь і компетентностей [1].

З усього сказаного можна зробити висновок, що тест - це не просто якийсь набір запитань, після якого можна дізнатися на скільки засвоєні знання, а важливий елемент в навчанні студента, який має містити в собі наступні складові:

- систему завдань;
- систему пред'явлення завдань;
- систему перевірки та обробки результатів;
- систему аналізу результату.

Існуючі системи перевірки знань студентів не забезпечують комплексних перевірок знань, а деякі використовують лише закриті запитання для поверхневої оцінки знань студентів. Тому існує нагальна потреба у розробці унікальних інструментів, спрямованих на об'єктивну, масштабну перевірку набутих студентами знань.

Зв'язок роботи з науковими програмами, планами, темами полягає в тому, що робота виконувалася згідно плану виконання наукових досліджень на кафедрі комп'ютерної інженерії.

Метою магістерської кваліфікаційної роботи є покращення якості знань студентів за рахунок розробленої спеціальної веб-системи, що забезпечує віддалений доступ користувачам.

Основними задачами роботи є:

- модель і метод роботи навчальної системи;
- розробити алгоритм для генетації тестів;
- розробити алгоритм для визначення результатів навчання;
- розробити графічний інтерфейс користувача для зручної взаємодії;
- розробити веб додаток для навчання програмуванню студентів;
- провести тестування сайту.

Об'єкт дослідження – процес розробки веб-системи для навчання студентів.

Предмет дослідження – алгоритми, методи та засоби для здійснення навчання та тестування студентів.

У процесі досліджень використовувались **методи дослідження**:

- методи перевірки знань і навичок учнів для реалізації навчання в актуальній формі;
- методи оцінювання знань учнів для отримання результатів знань учнів у актуальній формі;
- методи теорії алгоритмів для розробки алгоритмів тестування та підбору актуальної інформації для покращення знань учня;
- методи обробки інформації для генерації тестових питань й визначених правильних відповідей;
- методи розробки додатків для створення програмного забезпечення навчальної системи.

Наукова новизна отриманих результатів.

Подальшого розвитку отримав метод перевірки набутих знань учнів, який, на відміну від існуючих, забезпечує використання нових задач та отримання оцінки

у відсотковій шкалі, що дозволяє підлаштувати даний метод під будь-які вимоги до шкали оцінювання та підвищує універсальність його використання.

Подальшого розвитку дістала модель тестування системи, яка, навідміну від існуючих, використовує стандартизовані підходи до підбору запитань та шкали оцінювання і забезпечує дистанційний доступ користувачів за допомогою використання веб-середовища, що розширює коло потенційних користувачів.

Практична цінність отриманих результатів полягає в можливості проведення контрольних перевірок знань в середовищі веб-системи в онлайн-режимі.

Особистий внесок здобувача. Усі наукові результати, викладені у магістерській кваліфікаційній роботі, отримані автором особисто. У науковій роботі, опублікованій у співавторстві [3], автору належать такі результати: алгоритм створення навчальних програм, алгоритм розрахунку результатів навчання, алгоритм підбору інформації для покращення знань та алгоритм авторизації користувачів для отримання доступу до системи.

Апробація матеріалів магістерської кваліфікаційної роботи здійснена в доповіді на конференції ВНТУ електронні нацкові видання молодих вчених та студентів «Молодь в науці: дослідження, проблеми, перспективи».

Основні результати дослідженні опубліковані в науковій роботі – в тезах доповіді на конференції «Молодь в науці: дослідження, проблеми, перспективи» [3].

1 АНАЛІЗ СТАНУ ПИТАННЯ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз навчальних систем для отримання професійних знань.

Сьогодні кожен навчальний заклад має систему іспитів і освітню систему для отримання спеціалізацій, а також державні навчальні заклади та приватні чи державні академії. Тестова система є невід'ємною частиною модульного управління знаннями студентів, особливо в онлайн-навчанні. Тестове управління знаннями, уміннями та навичками студентів дає змогу ефективно та об'єктивно оцінювати успішність навчання. Система оцінювання – тестовий контроль оцінювання знань, компетентностей і вмінь студента – дає можливість підвищити об'єктивність, ритмічність і всебічність контролю з урахуванням індивідуальних особливостей студента. Це супроводжується змінами психології учнів. Студенти повинні працювати самостійно, ініціативно та систематично протягом семестру. Тестові системи для оцінювання знань студентів в основному забезпечують такі методи психодіагностики для вимірювання та оцінки розвитку досягнутих навичок і особистих знань. Оцінювання рівня знань і вмінь. Визначає рівень інтелектуального розвитку. Оцінка рівня соціальної та особистісної цінності набутих знань. Тестовий контроль стає все більш популярним. Він поступово витісняє інші традиційні форми управління знаннями, особливо усні та письмові опитування. Принципова відмінність тестів для перевірки знань студентів від звичайних завдань полягає в тому, що тест є одночасно науково обґрунтованим методом і перевіркою характеру, компетентності та інших питань, що вимагаються від наукових організацій, це інструмент для вивчення багатьох наукових напрямків навчального процесу.

Незважаючи на значний прогрес технологій та активне використання тестових систем у різних навчальних закладах, проблеми незручних інтерфейсів та обмеженої функціональності системи так і не вирішені. В існуючих системах відсутні спеціальні аудиторії для студентів із системами підрахунку балів і алгоритмами, які виявляють прогалини в знаннях студентів і рекомендують тести для здачі відповідних спеціалізацій.

1.2 Порівняльний аналіз аналогів

Сьогодні існує безліч тестових систем, призначених для реалізації конкретних завдань або проблем. Кожна система має переваги та недоліки.

Основне завдання цих систем — перевірити знання користувача і за результатами рекомендувати наступні тести в аналогічних областях знань. Розглянемо найпоширеніші тест-системи та ті, що мають відношення до розв’язування задач сучасності.

Testorium [2] (рисунк 1.1) — безкоштовна система створення та тестування тестів, призначена для викладачів і учнів усіх навчальних закладів. Основна мета створення цієї системи – полегшити роботу вчителів та дати можливість учням самостійно перевірити свої знання. Недоліком цієї системи є відносно застарілі тести з різних предметної області.

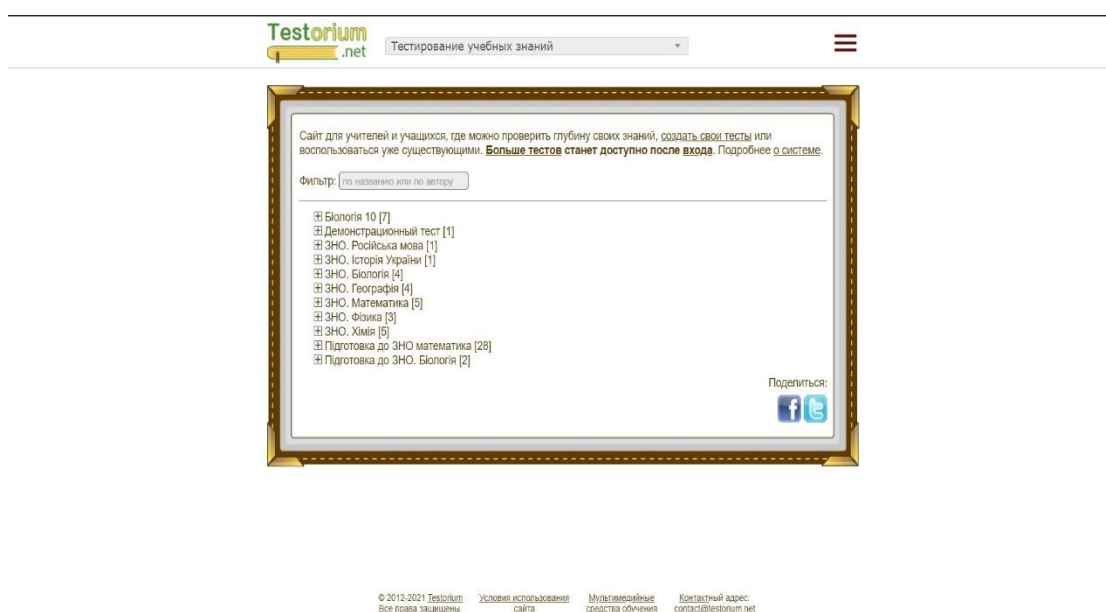


Рисунок 1.1 – Вікно веб додатку Testorium

Udemy[4] (рисунк 1.2) – Udemy — це платформа, яка дозволяє вчителям створювати онлайн-курси на обрані ними теми. Інструменти розробки курсів Udemy дозволяють створювати курси, завантажуючи відео, презентації, PDF-файли, аудіофайли, ZIP-файли та живі курси. Педагоги також можуть залучати та взаємодіяти з користувачами через онлайн-форуми.

Курси пропонуються в різних категоріях, включаючи бізнес і підприємництво, академічні науки, мистецтво, здоров'я та фітнес, мови, музику та технології. Udeyу також пропонує Udeyу для бізнесу, де компанії можуть отримати доступ до колекції з понад 7000 спеціалізованих курсів про все, від тактики цифрового маркетингу до офісної продуктивності, дизайну, менеджменту, програмування тощо. Організації також можуть використовувати Udeyу for Business для створення власного навчального порталу для корпоративного навчання.

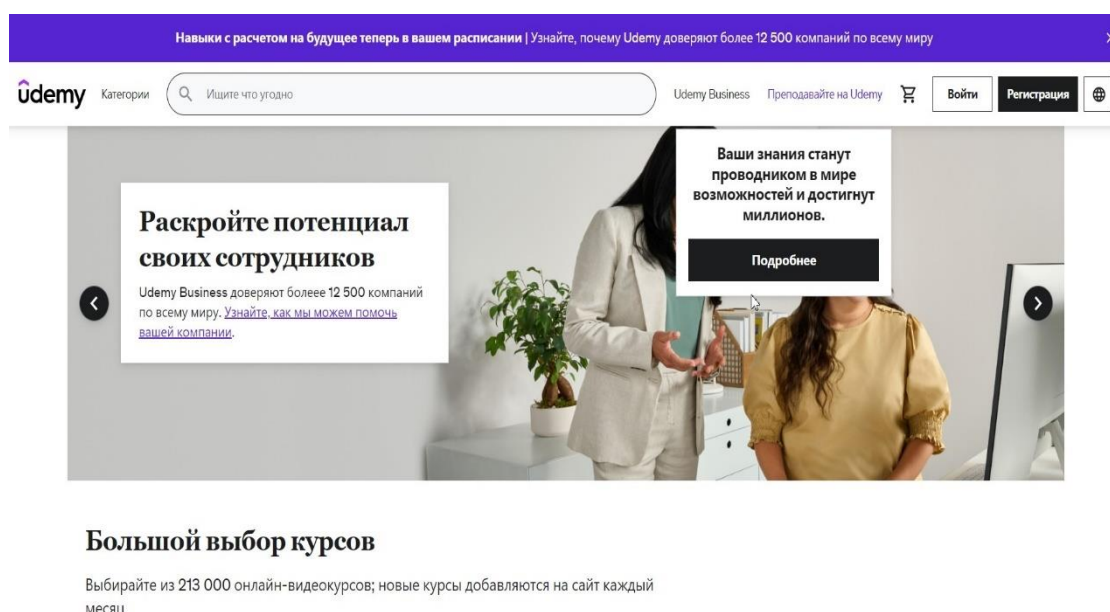


Рисунок 1.2 – Вікно веб додатку Udeyу

Крок [5] (рисунок 1.3) – навчальна система, що спеціалізується на ІТ-освіті. Комп'ютерна Академія має кілька основних форматів навчання. 1. Професійна підготовка (від 15 до 55 років): розробка програмного забезпечення, мережеві технології та системне адміністрування; Комп'ютерна графіка та дизайн. 2. Мала комп'ютерна академія (9-14 років): базова комп'ютерна освіта, практичні навички та теоретичні знання, літні табори 3. Спеціалізовані курси: Розробка та оновлення веб-проектів.

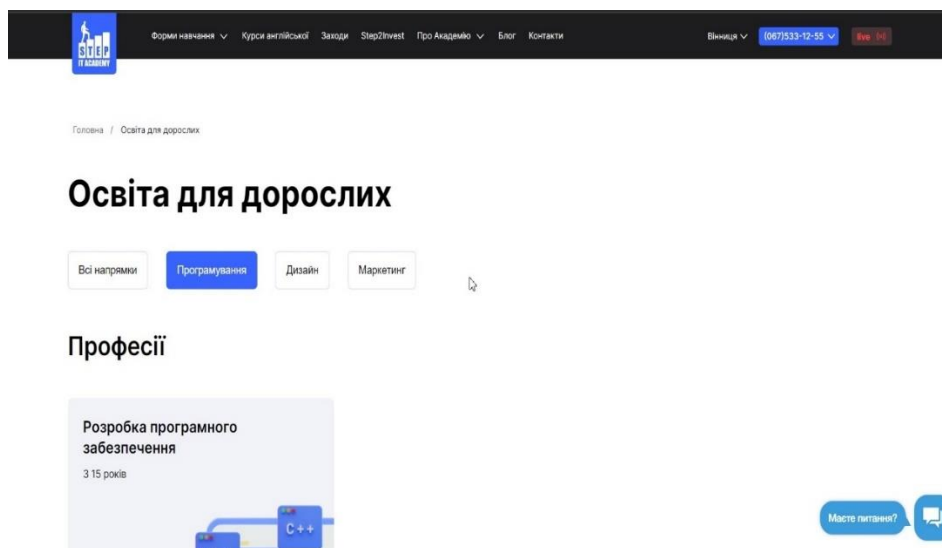


Рисунок 1.3 – Вікно веб додатку Крок

Тестування.укр [6] (рисунок 1.4) – Це система віддаленого тестування для тих, хто хоче отримати нові знання або вдосконалити та розширити свої наявні знання. Система має досить зрозумілий і зручний інтерфейс і корисну накладку у верхній частині сторінки. Недоліком цієї системи є вузькоспеціалізований тест з невеликою кількістю питань. Також не можна вибрати тестовий режим, вибрати час проходження тесту тощо.

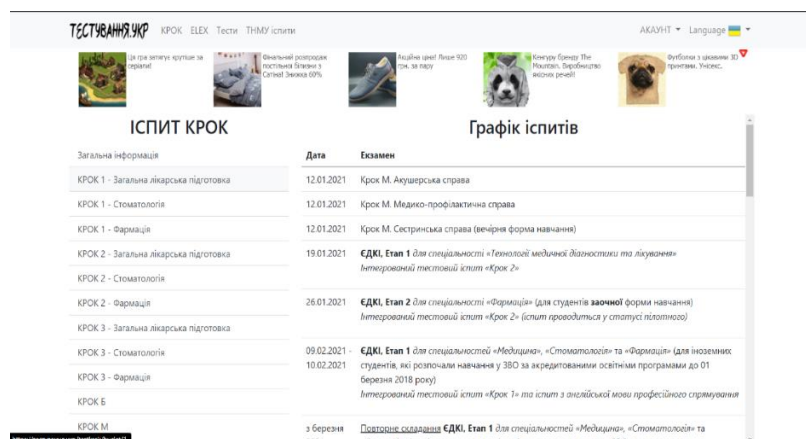


Рисунок 1.4 – Вікно веб додатку Тестування.укр

Після аналізу всіх аналогів були виявлені недоліки та сильні сторони систем, які були враховані при створенні власної системи під назвою «MyEasyQuiz» (табл. 1.1).

Таблиця 1.1 – Порівняльні характеристики програмних продуктів

Критерій	Testorium	Udemy	Крок	Тестування.укр	MyEasyQuiz
Кросбраузерність	+	+	+	+	+
Перехід по сторінках веб додатку без перезавантаження сторінок	-	+	-	-	+
PWA	-	-	-	-	+
Користування веб додатком без інтернету	-	-	-	-	+
Велика швидкість завантаження додатку перший раз	-	+	-	-	+
Зручний інтерфейс	-	+	-	-	+
Зручна навігація додатку	+	+	-	-	+

Таблиця порівняльних ознак показала, що розробка програмних продуктів є цільовою. У результаті ми маємо продукт, який усуває недоліки існуючих рішень і дозволяє користувачам зручно використовувати веб-додатки у своїх браузерах, а також звичайні додатки на своїх смартфонах. Ця веб-програма також дозволяє користувачам запускати тести та переглядати результати без підключення до Інтернету.

1.3 Аналіз перспектив розвитку розробки

Використання мережевих технологій значно зросло в усіх сферах життя людини, особливо в освіті. Однією з головних причин, через яку вчителі все більше турбуються про впровадження мережевих технологій, є зручність і простота

використання існуючих інструментів для пошуку, створення та використання мережевих ресурсів (NR). За допомогою ВР можна значно підвищити ефективність навчального процесу, активізувати навчальну та пізнавальну діяльність студентів, а також їх самостійну діяльність. Питанням впровадження мережевих технологій у навчальний процес та налагодження вчителів роботи з мережевими технологіями присвячено багато праць українських та зарубіжних авторів. Ці питання є предметом серйозної дискусії в науковій практиці, на науково-методичних конференціях, на сторінках журналів. За допомогою візуалізації різних форм навчальної інформації можна вирішити багато навчальних завдань.

Забезпечення посиленого навчання, посиленої навчально-пізнавальної діяльності, виховання та розвитку критичного та образного мислення, візуального пізнання, грамотності та образотворчої культури. Сьогодні важко уявити компанію без власного сайту. Цей сайт є фасадом для будь-якої організації в Інтернеті. Веб-сайт – це інтерфейс між компанією та її оточенням (партнери, постачальники, клієнти). Тому створення веб-сайту є одним із основних завдань будь-якої компанії, пов'язаної з Інтернетом. Сьогодні все більше і більше підприємств використовують Інтернет для обміну даними та оптимізації обслуговування клієнтів. Найбільш поширеною формою онлайн-презентації є створення веб-сайту (веб-представництва). Структурована інформація на сайті дає партнерам та клієнтам повне уявлення про компанію та її діяльність. Споживачі отримують багато корисних URL-адрес з рекламних банерів веб-сайтів, комерційних та комерційних публікацій, з яких вони можуть дізнатися ціни та замовити продукти [7].

1.4 Аналіз методу реалізації веб сайту

Не всі системи перевірки знань можна поділити на групи. Кожна система унікальна по-своєму, але функціонал у всіх однаковий. Одна система перевірки знань учнів дає більше повноважень, інша менше. Але результат роботи таких систем практично однаковий, лише кожна система відповідає певним вимогам.

Вимоги встановлюються самим навчальним закладом. Наприклад, як обробляються результати, хто може переглядати результати, коли можна пройти тест тощо. буд. Деякі тестові системи включають відкриті питання, і навіть з кількома варіантами відповідей. Це означає, що користувачі повинні відкрито відповідати на такі запитання та висловлювати свою думку щодо конкретних питань. Такі питання продуктивніші, оскільки користувачі не можуть вгадати правильну відповідь. У цьому випадку результати тесту будуть точнішими. Але не багато систем містять такі питання. Тому що його реалізація потребує більше часу та ресурсів. Для реалізації системи, що містить відкриті тестові питання, необхідно використовувати нейронні мережі, які перевіряють такі питання, використовуючи синоніми слів та різні перефразування речень [8].

Системи з питаннями закритого типу не надто складні у реалізації. Кількість відповідей питання тесту зазвичай варіюється від 2 до 8, а кількість правильних відповідей залежить від числа, зазначеного вчителем, який створив тест. Також для перевірки можна використовувати картинки, прикріплені як до питання, так і відповіді. Але найбільш поширеними в наші дні є аудіо-або відеозапитання. Такі питання використовуються повсюдно, починаючи з систем, які допомагають вивчати іноземні мови та тестувати іноземні мови для іспитів з іноземної мови. Користувач спочатку чує питання своєю рідною чи іншою мовою, а потім відповідає на запитання у стандартному режимі.

За допомогою такої системи кожен учитель має можливість самостійно створювати тести з різних предметів різного типу складності. Студенти можуть перевірити свої знання з цих тестів індивідуально або за допомогою викладача в режимі реального часу.

Завдяки системі вчителю не потрібно перевіряти, чи пройдено тест. Це з тим, що сама система обробляє це відразу після завершення тесту. Вчителі також можуть переглядати результати тестів, відкривати пройдені запитання та бачити, які відповіді надали учні на конкретні питання. Це скорочує час, що витрачається на перевірку різних завдань, та автоматизує взаємодію між вчителями та учнями.

Існує також система, що дозволяє обмінюватись тестами. Це означає, що

вчителі можуть ділитися своїми тестами коїться з іншими вчителями. Так ви покращите свої тести і внесете деякі зміни до своїх та своїх колег. Це також забезпечує покращені методи тестування, і знання учнів перевіряються більш критично.

Обробка результатів є також важливим етапом перевірки знань учнів. Результати можуть відображатися у різних форматах і можуть бути переглянуті певними користувачами у певний час. Найбільш поширеною формою результату є нормальний результат, одержаний після завершення тесту. Вони включають кількість набраних балів, загальну кількість питань, кількість питань, на які були надані правильні відповіді, і результати у процентній шкалі.

Не менш важливою особливістю системи є можливість завантажувати PDF-файли, що містять більш складні результати. Цей файл містить усю інформацію, доступну в старому вихідному форматі. Однак список усіх заданих користувачем запитань для цього тесту все ще доступний. Це дає користувачеві можливість не тільки дізнатися результати, але й побачити, що вони неправильно представили, а потім повернутися до тестування знову з покращеними знаннями.

Функціональність таких систем не обмежена. Завжди є що вдосконалити систему, зробити роботу продуктивнішою, полегшити роботу вчителя, а учням надати зворотний зв'язок, який допоможе покращити знання з різних спеціальностей.

1.5 Постановка задач роботи

Після аналізу розробки навчальної системи, визначаються наступні задачі, які потрібно виконати для створення програмних модулів:

- зробити аналіз предметної області;
- визначити трендові тенденції розробки дизайну сайтів;
- розробити метод та модель системи тестування;

- обрати засоби для розробки додатків;
- розробити алгоритми роботи програми;
- розробити дизайн веб-сайту;
- розробити веб-сайт;
- протестувати веб-сайт.

2. РОЗРОБКА МЕТОДУ ТА МОДЕЛІ СИСТЕМИ ТЕСТУВАННЯ

2.1 Аналіз інформаційного забезпечення

Інформаційне наповнення сайту – важливе завдання, обсяг, складність та різноманітність якої часто недооцінюють. Потрібно писати та публікувати новини, готувати матеріали та проводити презентації.

Якісний контент – це матеріал, який поєднує в собі кілька якостей, таких як оригінальність, корисність, релевантність, декоративність та інформація. Якісний вміст сприяє підвищенню рейтингу вашого сайту в пошукових системах і залучає більше відвідувачів на ваш сайт.

Алгоритми пошукових систем швидко змінюються, і вже недостатньо заповнити веб-сторінку великою кількістю посилань. Також потрібно наповнити сайт хорошим контентом. Пошукові системи враховують унікальність контенту, тобто середній час, який користувач витрачає на читання статей на сайті. Також враховується дата додавання документа та дата останнього оновлення. Чим свіжіший і актуальніший контент, тим швидше сайт підніметься на вершину.

Корисний контент приваблює користувачів на сайт. Якщо користувач натрапить на цікаву статтю, він обов'язково поділиться нею, і кількість відвідувачів збільшиться, сайт стане відомим і створиться хороша репутація. Все це допоможе досягти головної мети - сайт стане популярним[9].

Вміст веб-сайту можна розділити на такі типи:

- текст;
- відео-контент;
- інфографіка;
- графічні зображення.

Якісним контентом відрізняються наступні ключові принципи його наповнення:

— Унікальність, унікальний зміст - унікальний зміст. Пошукові системи надають вагу захищеному авторським правом тексту та зображенням, які не відповідають матеріалам на інших веб-сайтах. Інформативність. Матеріал повинен бути корисним для користувачів.

— Актуальність. Актуальні події - це не просто підсумок останніх подій, що відбулися останнім часом. Актуальність тепер є тим, що читачі можуть сприйняти та втілити.

— Грамотність. Текст необхідно перевірити на наявність орфографічних і граматичних помилок.

— Структурування та ієрархія тексту. Заголовок, підзаголовок, список. Читачам зручніше використовувати цей тип тексту. Пошукові боти віддають перевагу чітко структурованому вмісту.

— Наявність правильних посилань. Допомагають читачам орієнтуватися у матеріалі та без зайвих дій забезпечують прямий доступ до важливих частин.

Веб-застосунки зазвичай використовуються не тільки для відображення певного контенту, але також можуть замінити настільні програми, які використовуються тільки в браузері. Однією з переваг такого підходу є те, що веб-додаток є кросплатформовим сервісом, оскільки клієнт не залежить від будь-якої конкретної операційної системи. Ця універсальність та відносна простота розробки зробили веб-застосунки повсюдними.

Веб-програми використовуються для вирішення проблем користувачів. Наприклад, інтернет-магазин з функцією закупівлі та різними системами для проходження тестів з різних спеціальностей.

2.2 Вибір типу інтерфейсу

Веб-інтерфейс – це набір інструментів, які користувачі використовують для взаємодії з веб-сайтом або веб-програмою через браузер. Веб-інтерфейси стали популярними завдяки зростаючій популярності Всесвітньої павутини та широкому використанню веб-браузерів.

Інтерфейс веб-сайту впливає на те, наскільки швидко та легко відвідувачі зможуть знайти потрібну їм інформацію та чи захочуть вони повернутися на ваш веб-сайт, тому важливо розробити якісний інтерфейс та вибрати тему для свого веб-сайту. Інтерфейс має відповідати функціоналу сайту. Для кожного типу веб-сайтів існує безліч відповідних інтерфейсів, які можна реалізувати. Крім того, інтерфейс має бути сучасним і відповідати останнім тенденціям з різними графічними елементами та анімацією.

Візуальний дизайн, широко відомий як графічний дизайн, дизайн інтерфейсу користувача, комунікаційний дизайн і візуальна комунікація, описує естетичний вигляд об'єктів користувацького інтерфейсу. Графічна обробка елементів

інтерфейсу вважається формою візуального дизайну. Мета візуального дизайну – передати інформацію аудиторії за допомогою візуальних елементів, таких як кольори, зображення та символи. Базові знання гештальт-психології та бачення забезпечують когнітивну перспективу для створення ефективної візуальної комунікації.

Інформаційна архітектура — це мистецтво та наука створення та організації інформації про продукти та послуги для забезпечення зручності використання та пошуку.

Структурування зводить інформацію до основних будівельних блоків, та був об'єднує їх. Організації групують ці одиниці відповідно до їх унікальних семантичних властивостей. Маркування означає використання відповідного формулювання для полегшення навігації та пошуку.

Можливість пошуку – найважливіший чинник успішної інформаційної архітектури. Пошук в інформаційній архітектурі зазнає невдачі, коли користувачі не можуть знайти потрібну їм інформацію без перегляду, пошуку та запитів. Навігація має бути зручно побудована для легкого пошуку контенту.

Є багато важливих факторів для розуміння дизайну взаємодії та того, як він забезпечує чудовий досвід потенційних клієнтів. Як ви знаєте, створення відмінного інтерфейсу користувача вимагає, щоб дизайн взаємодії грав провідну роль і допомагав визначити, що має найбільший сенс для користувача. Високий попит на покращену взаємодію з користувачем і зосередженість на кінцевому користувачеві робить важливим для дизайнерів розробляти концепції, які відповідають очікуванням користувачів, а також компоненти інтерфейсу користувача та останні стандарти [10]. При роботі необхідно враховувати кілька факторів. Дехто з них:

- визначення режиму взаємодії, який найкраще відповідає контексту;
- включення потреб користувачів, які зібрані протягом досліджень користувачького досвіду;
- поведінка інтерфейсу, така як «drag-drop», вибір і дію миші;
- ефективна взаємодія кращих сторін системи;

- створення інтерфейсу з найкращими можливостями для користувачів;
- підтримка узгодженості у всій системі.

Зручність використання — це ступінь, до якого користувач використовує продукт у певному середовищі використання для ефективного й задовільного досягнення конкретних цілей. Зручність використання доповнює всі використовувані інструменти та поширюється на цифрові та нецифрові пристрої. Таким чином, це частина користувацького досвіду, але не повна. Частина зручності використання, яка перетинається з дизайном взаємодії з користувачем і стосується здатності людей використовувати систему або програму. Зручність використання важлива для позитивного сприйняття користувачами, але сама по собі це не гарантує.

Доступність системи описує легкість доступу, використання та розуміння. Що стосується дизайну взаємодії з користувачем, це також може стосуватися загальної ясності інформації та функціональності. Це скорочує криву навчання, задіяну в системі. У багатьох випадках доступність пов'язана з доступністю та зручністю.

2.3 Розробка методів та моделей тестування системи

Авторизація в систему. Користувач отримує доступ до форми входу або форми реєстрації. Є два способи входу в систему. Перший спосіб – за допомогою електронної пошти та пароля. Другий спосіб використовує авторизацію Google. Обидва методи реалізуються за допомогою автентифікації Firebase. Ця технологія дозволяє швидко та ефективно додавати користувачів до окремої бази даних, де ви можете легко ними маніпулювати (рис. 2.1). Після реєстрації кожному користувачеві присвоюється унікальний токен, який використовується для зберігання всіх даних користувача в системі.

Identifier	Providers	Created	Signed In	User UID ↑
qwer@gmail.com	✉	May 20, 2021		Snnal7Ky1tbI7ZPLrFowqHMYvWq2
labchuk.s66@gmail.com	✉	May 20, 2021		F8cWvRp0UaXNWitb4LQJv6lcgJl1
kvladfcvd@rgfref.frgv	✉	Jan 6, 2021	Jan 6, 2021	aHuVR1xitZTXRZH86htKBPh2Qm...
a@gmail.com	✉	Mar 26, 2021	Mar 26, 2021	lI0MCOdXXGhBFrN91GkW0mH4yl...

Рисунок 2.1 – Збереження користувачів

Режими тестування. Всі тести системи можна розділити на три основні групи. Тести можна здати в будь-який час. Ця форма підходить для студентів, які хочуть потренуватися або перевірити свої знання самостійно. Наступні контрольні роботи – це контрольні роботи, заплановані на певний час, тобто викладач може перевірити знання студента в необхідний час і проконтролювати студента. І остання група - це тести, яким можна призначити тривалість або тривалість для кожного питання. Ця форма тестування більше підходить для закритих запитань і питань, які не вимагають багато роздумів.

Крім того, для кожного тесту можна додати налагоджувальну роботу, тобто після проходження тесту користувач може одразу побачити правильну відповідь для кожного тесту та порівняти її з відповіддю своїми словами. Також до кожного тесту можна додати роботу з виправлення помилок, тобто після проходження тесту користувач може одразу побачити правильну відповідь кожного тесту та порівняти її з вашою.

Режими тесту можна вибрати на етапі створення тесту, але їх також можна вибрати, навіть якщо тест створюється без режиму. У налаштуваннях тесту вчитель може призначити студенту певний тест і вибрати необхідні режими проходження для цього студента.

На рисунку 2.2 зображено модель розробки тесту. Цей шаблон відображає послідовність кроків створення тесту, тоді як підкроки не потрібні для завершення створення тесту.

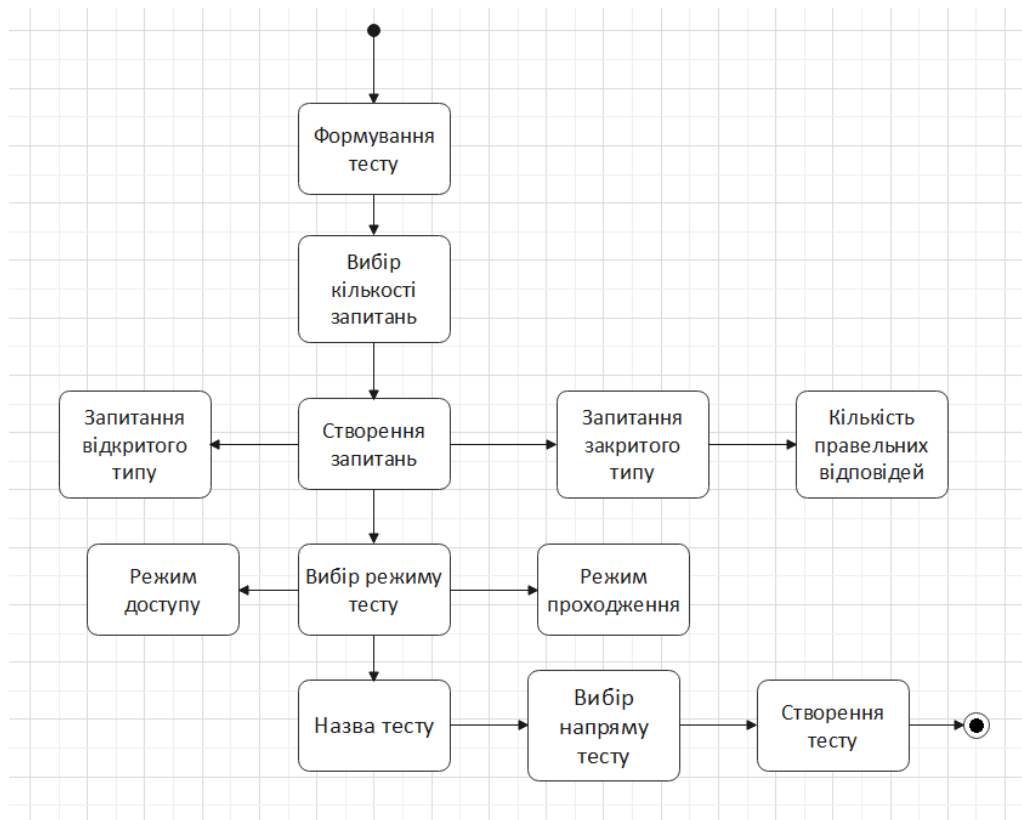


Рисунок 2.2 – Модель формування тесту

Метод проведення тестування:

- зареєструватися в системі;
- обрати дисципліну та тему для проходження тестування;
- реалізувати метод підбору різнотипових питань для формування тестових завдань;
- проходження етапу тестування з ідентифікованим рівнем складності самого тесту та підбору відповідних питань;
- формування оцінки по завершенню проходження тесту;
- збереження оцінки для можливості проведення статистичного дослідження, щодо визначення прогресу користувача;
- адаптація підбору питань до поточного рівня успішності відповідного користувача.

Етап проходження тесту з вибором визначає різні типи питань. Для перевірки ефективності використовується метод вибору запитань, які відповідають рівню знань учнів. Оцінка знань студента розраховується на основі тестів, які він склав раніше. Цей метод працює, якщо користувач пройшов хоча б один тест. Тому що питання можуть бути як з однією правильною відповіддю, так і з декількома або взагалі на відповідність, тобто кожне завдання має різний тип складності. У цьому випадку система просить користувача відповісти на питання з найнижчим рівнем складності, потім система видає більш складні питання. Тобто тести проходять від найлегшого запитання до найскладнішого. Якщо загальний бал студента в системі добрий, то контрольна робота відбудеться в порядку, визначеному викладачем при створенні тесту[11].

Відгуки користувачів в системі. Бали підраховуються після складання студентом тесту. Викладач може виставити найменший можливий бал для проходження тесту. Крім того, викладачі можуть вибрати бажану шкалу оцінювання студентів. Результати розраховуються кількома методами. Перший спосіб полягає в сумуванні всіх балів за звичайну правильну відповідь, тобто користувач отримує набраний бал і максимально можливий бал у цьому тесті. Другий спосіб оцінки більш складний. Він дозволяє конвертувати зафіксовані бали у відсотках і для кожного періоду часу вказувати певну оцінку в іншій шкалі. Наприклад, результати від 60% до 74% перетворюються на оцінку 3 (незадовільно), результати від 74% до 85% перетворюються на оцінку 4 (добре) і так далі.

Також результати проходження тесту можна отримати в окремому PDF файлі. Учень може натиснути кнопку “Скачати результати”, після він отримає файл з всією інформацією про проходження даного тесту.

Результат пройденного тесту

Назва тесту: Основи Java Script
 ПІБ студента: Лабчук Олександр Михайлович
 Група студента: 1ПІ-17Б
 Дата проходження: 15:17:54 12-04-2021

4

Оцінка:

Набрано балів: 38
 Максимальна кількість балів: 50
 Зроблено спроб: 1
 Кількість правильних відповідей: 22
 Питань всього: 28

Шкала оцінок:

"5" : 90%
"4" : 75% - 89%
"3" : 60% - 74%
"2" : 55% - 59%
"1" : 20% - 54%
"0" : 19%

[Пройти тест знову](#)

Рисунок 2.3 – Результати тестування в PDF файлі

Крім оцінки за тест, система також має загальний рейтинг користувача. Цей бал розраховується на основі аналізу всіх зданих користувачем тестів, а також враховується бал, виставлений викладачем. Але найвищий бал можна отримати, лише якщо учні створять власні індивідуальні тести, які допоможуть іншим учням підвищити рівень своїх знань.

Крім того, в процесі проектування будується модель «з'єднання сутностей» – модель ER. Модель ER — це модель даних, яка дозволяє описувати концептуальні схеми за допомогою загальних структур блоків.

Модель ER — це спосіб опису шаблонів даних. Існує багато моделей представлення знань, але одним із найбільш практичних інструментів для уніфікованого представлення даних незалежно від розгорнутого програмного забезпечення є модель «відношення сутностей». Важливо, що всі існуючі моделі даних (ієрархічні, мережеві, реляційні, об'єктні) можна імітувати з моделі «сутність-зв'язок», тому вона є найбільш загальною.

Модель ER розроблена для веб-додатку тестової системи (рис. 2.5). У результаті ми отримуємо процес, який описує та визначає, як працює веб-додаток. Дані представлені у вигляді компонентів (сутностей), пов'язаних один з одним певними зв'язками, що демонструє залежності та вимоги між ними. Сутності можуть мати різні атрибути, що їх описують. Діаграми, створені для графічного зображення цих сутностей, атрибутів і зв'язків, називаються діаграмами «сутність-зв'язок» (рис. 2.6).

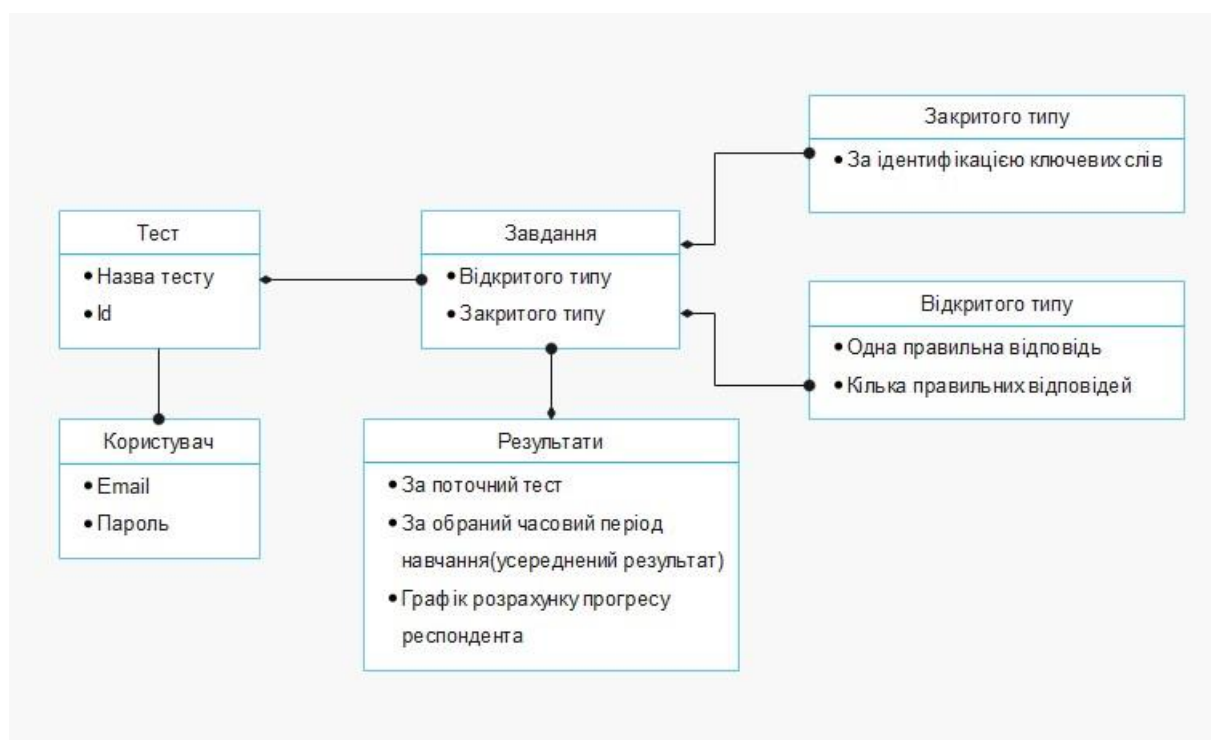


Рисунок 2.6 – ER-модель

Сутність - це деяка абстракція існуючого об'єкта, процесу чи явища або уявлення про об'єкт. Найчастіше використовується термін сутність, тому важливо розрізнити поняття типу сутності та екземпляру. У випадку тестової системи об'єктами є дані користувача, дані тесту та дані результатів тесту.

2.4 Аналіз вимог до дизайну сайту

Дизайн сайту - дуже важлива частина впливу на Інтернет-аудиторію. При створенні та розробці веб-дизайну слід звернути увагу на комфорт і зручність інформаційного наповнення сайту. Текст не повинен бути пов'язаний із фоном

сторінки, інакше його буде нечитабельно. Це негативно позначиться на популярності мережевих ресурсів, що негативно позначиться на доході, який приносить сайт своїм власникам. Ви повинні знати, як вимірювати при використанні графіки та анімації. Їх найкраще використовувати для фокусування та підкреслення важливого тексту. Так, ілюстрації полегшать монотонне нарощування рядків у великих текстових таблицях. Крім того, для зручності читання великі тексти слід ділити на глави та підзаголовки.

Веб-дизайн – це візуальна частина сайту. Для веб-сайтів він відіграє таку ж роль, як дизайн друку та макет паперу. Взагалі «веб-дизайн» називається веб-розробкою, тобто створенням загального веб-сайту, дизайном структури сайту, навігації, а іноді й системи управління [12].

Основним завданням дизайну є:

- ергономіка - простота у використанні об'єкту дизайну;
- інтерфейс користувача - психологічні характеристики оператора, що працює з об'єктом дизайну;
- новітні тенденції - використання нових технологій в дизайні щоб спростити інші завдання.

Функції, що забезпечує дизайн веб-сайту:

- використання кольорових рішень, дизайн шрифтів, графічні елементи, анімацію, щоб привернути увагу відвідувачів веб-сайту;
- створення унікального вигляду веб-сторінок, для виділення власного продукту з поміж конкурентів;
- дає можливість розміщувати актуальне інформаційне наповнення, яке буде доступним і читабельним.

Схема інтерфейсу головної сторінки сайту представлена на рисунку 2.7. Головна сторінка тестового сайту невелика, складається з кількох блоків і слайдера, що містить ключову інформацію про систему, щоб користувачі могли ознайомитися з можливостями сайту.

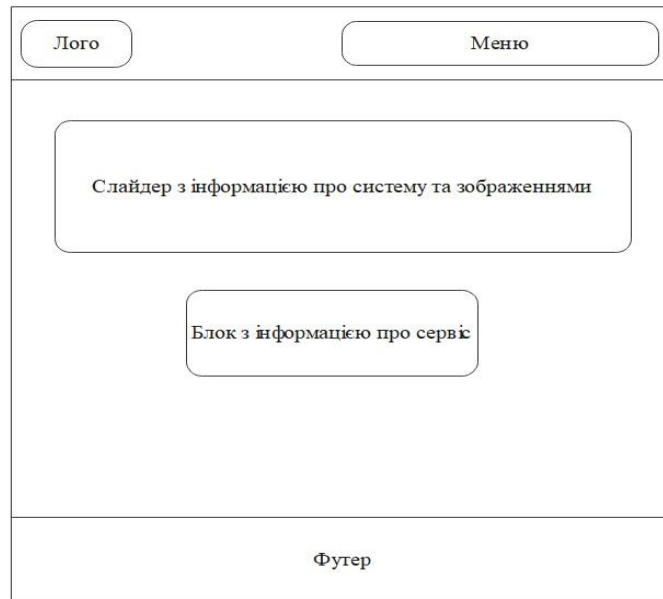


Рисунок 2.7 – Інтерфейс головної сторінки

Головною умовою успіху веб-сайту є взаємодія та вдале поєднання різних частин сторінки.

2.5 Вибір колірної гами

Важливу роль у створенні веб-сайту відіграє дизайн і вибір кольору. Правильний вибір дуже важливий, адже колірна гамма впливає на настрій людини і певною мірою на його поведінку. Колір, безумовно, може привернути або відштовхнути увагу. Щоб не втратити потенційних клієнтів, власникам мережевих ресурсів варто враховувати думку професійних дизайнерів. Але буде корисно мати уявлення про гармонійний вибір кольорів для ділянки. Як вибрати кольори, які сервіси допоможуть «переглянути» кольори сайту та їх поєднання та як працює палітра. Що ще врахувати при виборі колірної гами для веб-ресурсів.

Основним інструментом для поєднання відтінків є колірний круг. Завдяки цьому можна знайти ефектні поєднання, які гармонійно виглядають між собою. Але спочатку потрібно освоїти принципи роботи з нею.

Гармонійні комбінації народжуються з:

— Інтерактивні кольори для формування трикутника в колі. Класична трійка - це коли кожен елемент трійки розташований на однаковій відстані. Один колір відіграє домінуючу роль, а два інші – допоміжну.

— 3 кольорів, що створюють прямокутник. Складна схема, в якій перший колір є основним, другий другорядним, а другий акцентним. Не всі чотири кольори використовуються як основні. Це порушує баланс і, простіше кажучи, дизайн ріже око.

— Контрасти, тобто змішання двох кольорів, розташованих по колу навпроти.

Білий колір вибрано для основного фону сайту. Білий - найпростіший, але в той же час елегантний і спокійний колір, що асоціюється з чистотою, точністю, новизною, спокоєм, невинністю, простотою, акуратністю, взимку та ніжністю. Білий дизайн є символом елегантності, мінімалізму та стилю та використовується у багатьох блогах, онлайн-портфоліо, електронній комерції та інших веб-сайтах. Шрифти, фони, гіперпосилання та зображення виконані у білому кольорі, що часто привертає увагу читача, оскільки відкриває простір та забезпечує ясність [13].

Навігації по сайту в лівій частині сторінки теж надали білий колір, але з невеликою тінню. Посилання на інші сторінки відображаються сірим кольором. Це тому, що він привертає найбільшу увагу білому тлі (рис. 2.8).

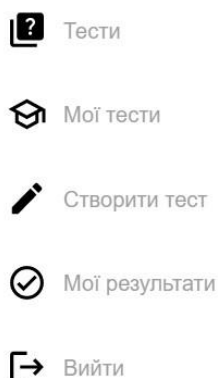


Рисунок 2.8 – Навігація сайту

І не менш важливим елементом, на який звертають увагу користувачі є логотип. Логотип зробили простим, лише текст, що відповідає основним завданням

системи, та жодних картинок. Колір логотипу такий самий сірий, як і посилань на інші сторінки, не такий яскравий, як чорний, але гармонійний з білим (рис. 2.9).

MY EASY QUIZ

Рисунок 2.9 – Логотип сайту

Головна сторінка, на яку вперше заходять користувачі, складається з двох блоків та футера. Назва системи та невеликий коментар про її функцію розміщені в першому центральному блоці (рис. 2.10).

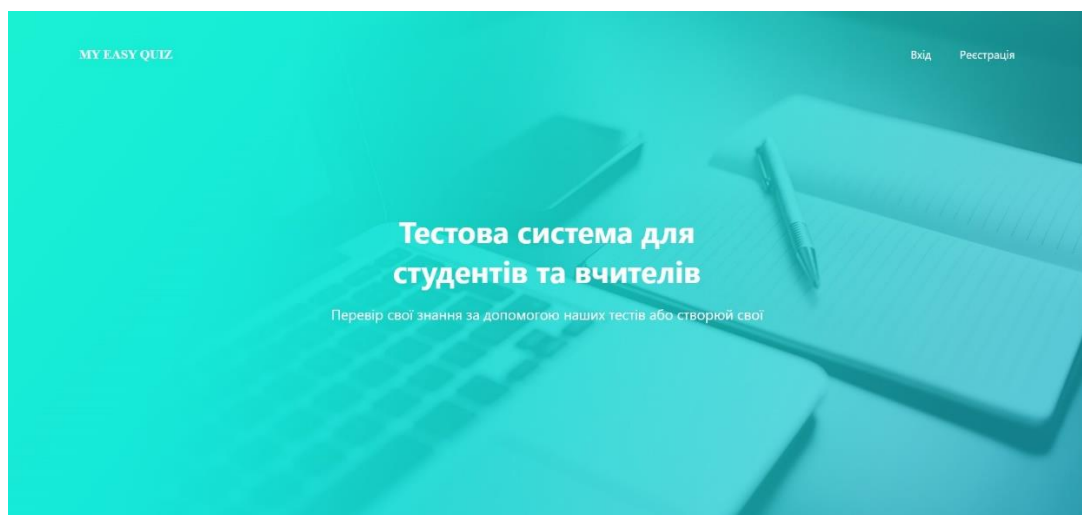


Рисунок 2.10 – Перший блок головної сторінки

У наступному блоці розміщувалася інформація про систему, але це не було масштабною інструкцією про систему як для студентів, так і для читачів. Колір тексту був чорним для гармонії з білим тлом і використовувався у поєднанні з навчальними картинками (рис. 2.11).



Рисунок 2.11 – Інформаційний блок головної сторінки

Сторінка реєстрації або входу також оформлена у світлих тонах, форма для заповнення даних біла, а для бічних фонів ми вибрали трохи темніші кола, щоб візуально сприймався обсяг форми. Кнопки «Реєстрація» та «Увійти» були реалізовані зеленим кольором, оскільки його було обрано другим кольором системи (рис. 2.12).

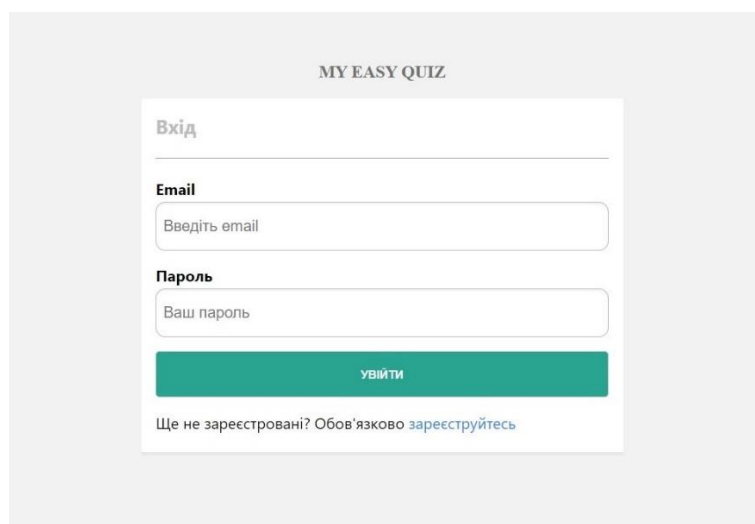


Рисунок 2.12 – Інформаційний блок головної сторінки

Після авторизації користувач потрапляє у саму систему. Усі сторінки системи були виконані у білому кольорі. На навігаційну панель була додана тінь, щоб надати їй більше об'єму і виділити панель серед інших блоків. Вміст розміщується

посередині. Вгорі кожної сторінки знаходиться чорний заголовок, який відповідає кожній сторінці (рис. 2.13).

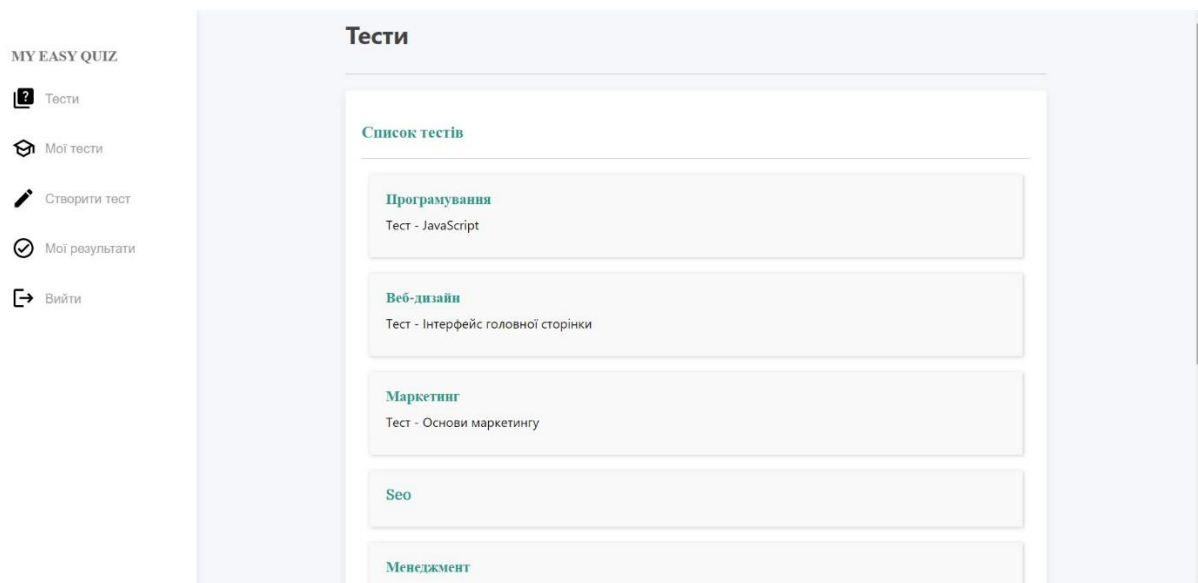


Рисунок 2.13 – Сторінка тестів системи.

3 РОЗРОБКА ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ВЕБ-САЙТУ

3.1. Варіантний аналіз засобів реалізації сайту

На сьогоднішній день існує безліч способів розробки веб-сайту. Основними з них є ручна розробка з використанням HTML, використання інструментів розробки програмного забезпечення, використання систем інструментів, таких як CMS, використання загальних фреймворків (SPA) і розробка на платформах SaaS в рамках CLOUD.

З появою стандарту HTML ручні методи реалізації стали найпоширенішими. Основним інструментом розробки був Блокнот. Однак, цей метод має очевидні недоліки. Цей метод вимагає багато часу. Більше того, створити нормальний сайт без таких мов програмування, як CSS та JavaScript досить складно.

Існує безліч готових рішень, що дозволяють прискорити та спростити розробку веб-сайтів. Вони надають можливість генерувати HTML-код, розробляти сайти у візуальному режимі та багато іншого. В основному включено: Adobe DreamWeaver, Microsoft FrontPage, HotDog, AceHTML [14].

Веб-додаток (SPA) — один із найпопулярніших та найпродуктивніших способів створення веб-сайту. Практично кожна компанія, організація чи установа мають свій сайт. Веб-сайт — чудовий спосіб для бізнесу залучити клієнтів, але ваша власна сторінка має бути простою у використанні та швидкою для навігації та навігації. SPA повністю відповідає вищезазначеним вимогам. SPA (односторінковий додаток) — це веб-додаток або веб-сайт, який розміщується на одній сторінці, щоб надати користувачам можливість роботи на робочому столі. В односторінковому додатку весь необхідний код (HTML, JavaScript, CSS) завантажується разом із сторінкою або завантажується динамічно за необхідності (зазвичай у відповідь дії користувача). Сторінка не оновлюється та не перенаправляє користувача на іншу сторінку під час роботи. Взаємодія з односторінковими програмами часто включає динамічний зв'язок з веб-сервером. Програми SPA повністю функціональні як на стаціонарних, так і на мобільних пристроях. «Великі» комп'ютери, планшети, смартфони та згодом прості телефони

(деякі) можуть без проблем працювати з сайтами, побудованими за принципами SPA. Крім того, багатий інтерфейс користувача, так званий інтерфейс користувача. Оскільки існує лише одна веб-сторінка, легко створювати багатофункціональні інтерфейси користувача. Це спрощує зберігання інформації про сеанс, керування станом перегляду та (у деяких випадках) керування анімацією. SPA значно знижує кількість так званих «блукаючих» або множинних завантажень того самого контенту. Якщо на сайті використовуються шаблони, відвідувачі сайту повинні завантажити розмітку шаблону разом з основним вмістом сторінки. Існує ряд базових бібліотек та фреймворків, які щодня розробляються, щоб полегшити розробникам створення SPA. Основні фреймворки для SPA: React.js, Vue.js, Angular.js, Svelte.js (табл. 3.1).

Таблиця 3.1 – Порівняння сучасних технологій розробки веб-застосунків

	Python Frameworks	JS Frameworks	.NET
Багатолатформеність	+	+	-
Простота використання	+/-	+	+/-
Продуктивність	+/-	+/-	+
Швидкість	+	+	+
Розподіл дизайну та логіки	+/-	+/-	+

Проаналізувавши порівняльні результати новітніх технологій розробки веб-додатків, ми дійшли висновку, що найшвидші та найпродуктивніші сайти засновані на JS-фреймворках. Але головне перевагу фреймворку JS, тобто. React, - це простота використання та мультилатформеність [15].

3.2 Аналіз середовища розробки

Для розробки веб-додатків були вибрані технології HTML5, CSS3, JavaScript та React.js. Для роботи з цими технологіями було обрано середовище розробки WebStorm. JetBrains WebStorm - це інтегроване середовище розробки для різних програмних інструментів з використанням JavaScript, HTML та CSS від JetBrains,

засноване на платформі IntelliJ IDEA. WebStorm – це покращена версія PhpStorm, що пропонує лише частину функцій цієї програми. [14] WebStorm поставляється з прийнятними модулями JavaScript (наприклад, для Node.js), які також доступні безкоштовно в PhpStorm [16].

WebStorm підтримує JavaScript, CoffeeScript, TypeScript та Dart. WebStorm допомагає вам кодувати на JavaScript, мовами, скомпільованих у JavaScript, HTML та CSS, а також за допомогою Node.js. Автодоповнення коду, проста навігація, миттєве виявлення помилок та рефакторинг доступні всіма цими мовами.

WebStorm допомагає розробляти програми за допомогою Angular, React, Vue.js та Meteor. Якщо ви використовуєте React Native, PhoneGap, Cordova або Ionic для розробки мобільних програм або Node.js, WebStorm також може допомогти вам підвищити продуктивність. І все це в одній IDE.

На основі аналізу коду вашого проекту середовище IDE пропонує найкращі варіанти автозаповнення для всіх мов, що підтримуються. Сотні вбудованих перевірок миттєво вказують на можливі помилки та пропонують варіанти виправлення. Це середовище допомагає швидко орієнтуватися в коді та економити час під час роботи над великими проектами. Одним клацанням миші ви можете перейти до визначення будь-якого методу, функції чи змінної або знайти його застосування у своєму проекті.

Він також має вбудовані інструменти для налагодження, відстеження та тестування клієнтських програм та програм Node.js. Продумані інтеграції роблять вирішення цих завдань простішим і менш складним. WebStorm має вбудовані інструменти для налагодження коду клієнта (працюючого в браузері Chrome) і програм Node.js. Встановлюйте точки зупинки, виконуйте код покроково та переглядайте змінні безпосередньо у середовищі IDE.

IDE інтегрується з Karma, Mocha, Jest та Protractor, що спрощує тестування додатків за допомогою WebStorm. Запускайте та налагоджуйте тести прямо в IDE, переглядайте результати у зручному форматі та легко переходьте до тестованого коду.

3.3 Робота систем керування базами даних

Програмування веб-сайту виконується для створення корисних функцій. Використання бази даних є одним із найважливіших компонентів динамічного веб-програмування. Незалежно від того, створюєте сторінки «на льоту» чи реагуєте на дії відвідувачів сайту, завжди існує взаємодія з базою даних.

База даних веб-сайту (БД) використовується для зберігання різноманітної інформації та є просто набором пов'язаних таблиць. Розмір таблиць бази даних є змінним, а їх кількість довільною. Інформація, необхідна для роботи веб-сайту, зберігається в базі даних на сервері, наприклад інформація про клієнтів, категорії продуктів, статистичні дані тощо.

Програмування динамічного сайту використовує для виконання різні сценарії. Зазвичай він ділиться на серверну та клієнтську частини. Сценарії на стороні сервера можна використовувати для програмування веб-сайтів для обробки даних, введених у веб-форми відвідувачами веб-сайту, для створення динамічних сторінок, а також для надсилання та отримання файлів cookie. Сценарії на стороні сервера звертаються до бази даних для отримання інформації, необхідної для виконання таких операцій. Сценарії доступу до бази даних називаються запитами, а SQL (мова структурованих запитів) широко використовується для створення запитів до бази даних. Ви можете використовувати SQL для додавання, видалення та редагування записів у таблицях бази даних, вилучення даних за різними критеріями та сортування даних.

Програмування сайту управління базою даних виконується за допомогою клієнт-серверної системи управління базами даних (СУБД), такої як Oracle, MS SQL Server, PostgreSQL або MySQL. Бази даних клієнт-сервер активно використовують запити. Його переваги включають забезпечення високої надійності, доступності та розширеної безпеки бази даних.

СУБД MySQL – це система управління базами даних, яка зазвичай використовується при програмуванні веб-сайтів. СУБД MySQL підтримує безліч існуючих типів таблиць (InnoDB, MyISAM та ін), а завдяки відкритій архітектурі

та ліцензії GPL в СУБД MySQL постійно з'являються нові типи таблиць. Управління базами даних за допомогою MySQL дуже зручне, тому ця система була затребувана та популярна.

Система керування реляційними базами даних Microsoft SQL Server надається Microsoft на комерційній основі (за винятком безкоштовної версії Express Edition). Ця СУБД використовує мову запитів Transact-SQL та підтримується в настільних/серверних операційних системах Windows. До складу СУБД Microsoft SQL Server входить графічне ПЗ для побудови та оптимізації запитів (SQL Management Studio та Studio Express)[16].

Oracle Database, система управління об'єктно-реляційними базами даних Oracle, працює у Windows, Unix, Linux та MacOS. Наприклад, база даних Oracle має ширший діапазон, ніж MySQL. СУБД Oracle відрізняється високою продуктивністю, широким набором можливостей та унікальними технологіями (RAC, RAT та ін.). Рідко використовується при програмуванні сайтів для малого бізнесу через дорожнечу. Більше того, дуже складно знайти хостинг, який підтримує базу даних.

База даних Firebase Realtime – це веб-сайт бази даних Firebase NoSQL. Дані зберігаються у форматі JSON та синхронізуються в режимі реального часу з кожним підключеним клієнтом. Пов'язані функції включають інтеграцію з додатковими операційними системами Android і iOS, реалізації API для JavaScript, Java, Objective-C і Node.js, стилі REST з численних фреймворків JavaScript, таких як AngularJS, React Включає функціональність, яка використовує базові дані , Подивитися. js, Ember.js та Backbone.js.

Надається API для шифрування даних. При створенні кросплатформових програм з використанням пакетів SDK для iOS, Android та JavaScript всі клієнти спільно використовують один екземпляр бази даних у режимі реального часу, яка автоматично оновлюється останніми даними.

Firebase Realtime Database дозволяє створювати багатofункціональні спільні програми, що забезпечують безпечний доступ до бази даних безпосередньо з клієнтського коду. Дані зберігаються локально, а події в реальному часі

продовжують працювати навіть в автономному режимі, забезпечуючи повне реагування на запити користувачів. Коли пристрій повторно підключається, база даних у реальному часі синхронізує локальні зміни даних з віддаленими оновленнями, які відбулися, коли клієнт був в автономному режимі, автоматично поєднуючи всі зміни. База даних реального часу надає гнучкий мову правил на основі виразів під назвою Firebase Real-time Database Security Rules для визначення структури ваших даних і того, коли їх можна читати або записувати. Завдяки інтеграції з Firebase Authentication розробники можуть контролювати, хто та як може отримувати доступ до даних.

Тому для забезпечення функціональності нашого тестового веб-додатку було обрано базу даних Firebase Realtime. Ми рекомендуємо використовувати саме цю базу даних, оскільки вона швидка та проста у використанні, оскільки дозволяє розділити всю інформацію на декілька груп на нашому тестовому сайті.

3.4 Програмна реалізація

Збільшуючи функціональність про програмних модулів, вони сприяють підвищенню ефективності та взаємодії користувачів з мережевими додатками. Додаючи різні модулі на свій сайт, ви можете додавати різні функції і перетворювати свій сайт на невелику програму тільки для браузера. Ви також можете використовувати програмні модулі для відображення даних у вашій базі даних у різних блоках. Не менш важливою функцією, якщо користувач ще не зареєстрований, є створення форми для входу до свого облікового запису або реєстрації.

Для тестування веб-додатків розроблено різноманітні програмні модулі, які використовуються як у дизайні сайту, так і у його функціоналі.

Ось кілька основних прикладів: слайдер з інформацією та зображеннями на головній сторінці, система входу в особистий кабінет або реєстрація, система підрахунку результатів після тестування, анімація на сторінці, система взаємодії.

Для розробки модуля використовувалася мова програмування JavaScript та його фреймворк React. Створюйте власні тести на сторінці Мої тести як приклад використання програмних модулів.

Лістинг-код функцій реалізації:

```
function createOptionControl(index) {
  return createControl(
    {
      label: `Відповідь ${index}`,
      errorMessage: 'Поле не може бути пустим',
      id: index,
    },
    {
      required: true,
    }
  )
}
```

За домогою цієї функцію реалізовано валідацію полей введення, що дозволяє системі відправляти на сервер тільки коректну інформацію.

```
function createFormControl() {
  return {
    question: createControl(
      {
        label: 'Введіть питання',
        errorMessage: 'Поле не може бути пустим',
      },
      { required: true }
    ),
    option1: createOptionControl(1),
    option2: createOptionControl(2),
    option3: createOptionControl(3),
    option4: createOptionControl(4),
  }}
}
```

Функція `createFormControl` повертає об'єкт за допомогою якого створюються поля введення для створення тестів.

```
class QuizCreator extends Component {
  state = {
    isValid: false,
    rightAnswerId: 1,
    formControls: createFormControl(),
    nameTest: '',
    createTest: false
  }
  submitHandler = (event) => {
    event.preventDefault()
  }
  addQuestionHandler = (event) => {
    event.preventDefault()
    const {
      question,
      option1,
      option2,
      option3,
      option4,
    } = this.state.formControls
    const questionItem = {
      question: question.value,
      id: this.props.quiz.length + 1,
      rightAnswerId: this.state.rightAnswerId,
      answers: [
        { text: option1.value, id: option1.id },
        { text: option2.value, id: option2.id },
        { text: option3.value, id: option3.id },
        { text: option4.value, id: option4.id },
      ],
    }
  }
}
```

QuizCreator — це компонент класу сторінки створення тесту, який містить форму створення тесту та посилання на інші сторінки.

```

this.props.createQuizQuestion(questionItem)
this.setState({
  isFormValid: false,
  rightAnswerId: 1,
  formControls: createFormControl(),
})
}
createQuizHandler = (event) => {
  event.preventDefault()
  console.log('fef')
  this.setState({
    isFormValid: false,
    rightAnswerId: 1,
    formControls: createFormControl(),
  })
  this.props.createQuizName(this.state.nameTes)
  this.props.finishCreateQuiz()
  console.log(this.nameTes)
}
changeHandler = (value, controlName) => {
  console.log(value)
  const formControls = { ...this.state.formControls }
  const control = { ...formControls[controlName] }
  control.value = value
  control.touched = true
  control.valid = validate(control.value, control.validation)
  formControls[controlName] = control
  this.setState({
    formControls,
    isFormValid: validateForm(formControls),
  })
}

```

```
}
```

Функції `ChangeHandler` та `CreateQuizHandler` зчитують дані від користувача, записують їх у локальний стан та надсилають запити на сервер для запису до бази даних.

```
createTestVisible = () => {
  this.setState({
    createTest: !this.state.createTest
  })
}
changeHandlerName = (value) => {
  // console.log(value)
  let oldNameTes = this.state.nameTest
  oldNameTes = value
  this.setState({
    nameTes: oldNameTes,
  })
  console.log(this.state.nameTes)
}
renderControl() {
  return Object.keys(this.state.formControls).map((controlName, index) =>
  {
    const control = this.state.formControls[controlName]
    return (
      <Fragment key={controlName + index}>
        <Input
          value={control.value}
          valid={control.valid}
          touched={control.touched}
          label={control.label}
          errorMessage={control.errorMessage}
          shouldValidate={!control.validation}
          onChange={(event) =>
```

```

        this.changeHandler(event.target.value, controlName)
      }
    />
    {index === 0 ? <hr /> : null}
  </Fragment>
)
})
}

```

Функція `RenderControl` проходить циклом по запитаннях тесту та відображає лише те запитання на якому зараз знаходиться користувач.

```

nameTest() {
  return (
    <InputName
      onChangeName={(event) =>
this.changeHandlerName(event.target.value)}
    />
  )
}
selectChangeHandler = (event) => {
  this.setState({
    rightAnswerId: +event.target.value,
  })
}
render() {
  const cls = [classes.Wrapper]
  if (this.state.createTest) {
    cls.push(classes.Show)
  }
  const select = (
    <Select
      label="Виберіть правильну відповідь"
      value={this.state.rightAnswerId}
      onChange={this.selectChangeHandler}

```

```

    options={[
      { text: 1, value: 1 },
      { text: 2, value: 2 },
      { text: 3, value: 3 },
      { text: 4, value: 4 },
    ]}
  />
)

```

Рішення Firebase Authenticate також використовувалося під час розробки системного модуля входу та реєстрації. Для цього використовується окрема невелика база даних, де зберігаються лише дані входу користувача. Перевірка форми також використовувалася для правильного входу до системи.

```

validateControl(value, validation) {
  if (!validation) {
    return true
  }
  let isValid = true
  if (validation.required) {
    const emptyCount = value.split(' ').length - 1
    isValid = value.trim() !== '' && !emptyCount && isValid
  }
  if (validation.email) {
    isValid = is.email(value) && isValid
  }
  if (validation.minLength) {
    isValid = value.length >= validation.minLength && isValid
  }

  return isValid
}

onChangeHandler = (event, controlName) => {
  const formControls = { ...this.state.formControls }
  const control = { ...formControls[controlName] }

```

```

control.value = event.target.value
control.touched = true
control.valid = this.validateControl(control.value, control.validation)
formControls[controlName] = control
let isValid = true
Object.keys(formControls).forEach((name) => {
  isValid = formControls[name].valid && isValid
})
this.setState({
  formControls,
  isValid,
})
}

```

Функція, що наведено зверху обробляє дані в формі входу та перевіряє на їх коректність, якщо дані перевірено успішно то користувач попадає в систему.

```

renderInputs() {
  return Object.keys(this.state.formControls).map((controlName, index) =>
  {
    const control = this.state.formControls[controlName]
    return (
      <Input
        key={controlName + index}
        type={control.type}
        placeholder = {control.placeholder}
        value={control.value}
        valid={control.valid}
        touched={control.touched}
        label={control.label}
        errorMessage={control.errorMessage}
        shouldValidate={!control.validation}
        onChange={(event) => this.onChangeHandler(event, controlName)}
      />
    )
  }
)
}

```

```
)  
})  
}
```

Вищевказані функції є основними функціями веб-додатку та вирішують завдання, поставлені перед використанням тестового сайту.

4 ТЕСТУВАННЯ РОБОТИ ПРОГРАМНОГО ПРОДУКТУ

4.1 Огляд рішень для тестування сайтів

Перш ніж публікувати розроблений веб-сайт у «Free Float», слід зробити кілька кроків, щоб виявити та виправити помилки у веб-дизайні, верстці та взаємодії з користувачем. Без усього цього неможливо досягти добрих результатів.

Це означає, що вашому проекту буде складніше вийти у топ пошукових систем.

Тестування веб-додатків дозволяє перевірити роботу всіх функцій щодо обсягу дозволів, швидкості та стабільності, читабельності інформації, зручності навігації по сайту, запобігання мережевим атакам і т.д. Ваші послуги та методи тестування можуть відрізнитися, але ви повинні з самого початку визначитися зі стратегією та дотримуватись її. Порядок операцій має бути ретельно продуманий і послідовно виконуватись [17].

Основні види або роботи при тестуванні:

- співвідношення фактичного дизайну та відповідного дизайну в ТЗ;
- перевірка валідності верстки;
- функціональне тестування;
- навантажене тестування;
- тестування безпеки сайту;
- тестування сумісності.

Не всяка розробка веб-сайтів обходиться без недоліків дизайну. Точне порівняння макетів та готових дизайнів непросто, тому що потрібно бачити, як виглядає кожен елемент та сторінка. Крім того, версії для мобільних пристроїв і планшетів вимагають більшої уваги та створюють модель, яка застосовується до кожного гаджета та браузера.

Перевірка є невід'ємною частиною тестування, щоб розроблені веб-ресурси виглядали однаково у різних браузерах, завантажувалися швидше та дозволяли пошуковим системам краще розуміти контент.

Існує стандарт, консорціум World Wide Web (W3C), який визначає правила та положення для створення сайтів. На його основі було розроблено W3C Markup Validation Service.

Один із найбільш трудомістких видів випробувань, які потребують точних функціональних розрахунків. Наприклад, розглянемо інтернет-магазин, який має безліч статусів для придбання певної кількості товарів, а не лише акції товарів. Усі комбінації мають бути перевірені.

Функціонал залежить від типу ресурсу що перевіряється, але крім того є базові елементи, на які звертають увагу:

- тестування форм введення даних;
- перевірка колектності пошуку та релевантності результатів видачі;
- тестування навігації – перевірка посилань;
- перевірки завантаження даних в базу даних;
- перевірка реєстраційної форми та входу.

Навантажувальне тестування важливе для запобігання несподіваному відключенню вашого сайту в години пік. Тому існують деякі сервіси, які можуть імітувати поступове збільшення кількості користувачів вашого сайту. Якщо час завантаження вашого сайту такий же, як і за низького рівня трафіку, то все гаразд. Для інтернет-магазинів ви можете протестувати виконання паралельних транзакцій, щоб отримати час відгуку для всіх важливих бізнес-функцій.

Тестування безпеки сайту залежить від списку вразливостей сайту. Важливо вчасно усунути їх, щоб хакери не змогли «зламати» канал для залучення нових клієнтів через Інтернет. Системи безпеки перевіряють методи виявлення вторгнень, рейтинги безпеки сайтів та аналіз ризиків доступу зловмисників до конфіденційних даних.

Основні етапи перевірки на вразливість:

- контроль доступу;
- діагностика аутентифікації;
- валідація даних;
- криптографія;
- механізм обробки помилок;
- інтеграція зі сторонніми сервісами;
- перевірка стійкості сайту до Dos/DDos атак;
- конфігурація сервера.

4.2 Тестування роботи програмного продукту

Під час розробки важливо перевірити функціональність вашого сайту, перш ніж розміщувати його на хостингу. Тести запускаються для перевірки поведінки веб-програми.

Спочатку проводиться тестування інтерфейсу користувача, потім функціональне тестування виконується на кожній сторінці. Функціональне тестування дуже важливе у роботі тестувальника, оскільки головна вимога — вміти користуватися сайтом. Виконання конфігураційних дій без порушень, конкретні результати, інтеграція з іншими програмами та безпека – тільки такі можливості веб-додатків можна вважати добрими. На цьому етапі ви можете стежити за технічними завданнями та перевіряти функціональну відповідність вашого сайту. З огляду на всі вимоги клієнта ми перевіряємо всі функції, які необхідно було розробити для конкретного проекту (сайту). Додатково перевіряється робота сайту та його функцій на наявність помилок коду [18].

Не треба плутати тестування інтерфейсу із тестуванням зручності використання. Це два різні етапи тестування загалом. UI – Цей тест перевіряє відповідність графічного інтерфейсу вашого сайту.

Тестування інтерфейсу:

- перевірка на відповідність для всіх стандартів графічних інтерфейсів;
- тестування на різних діагоналях;
- перевірка сумісності зі всіма браузерами(кросбраузерність);
- тестування інтерфейсу на смартфонах, кпп, планшетах;
- локалізоване тестування: точність перекладу, перевірка довжини назв та інше.

Все більше людей виходять в Інтернет через свої смартфони. Тому тестування мобільної версії вашого сайту є важливим етапом розробки проекту.

SQL-ін'єкція, шкідливий код у запитах до бази даних, є найнебезпечнішим типом атаки. Він може виконувати довільний код та атакувати комп'ютер користувача після перегляду зараженої сторінки [19].

Робота на цьому етапі дуже важлива, оскільки допомагає виявити всі помилки та дефекти, що виникають у процесі розробки програмного продукту.

Цей програмний продукт протестовано за допомогою валідатора HTML. Переконайтеся, що у кодї дисплея немає помилок. На рис. 4.1 показано результат перевірки сторінки index.html.

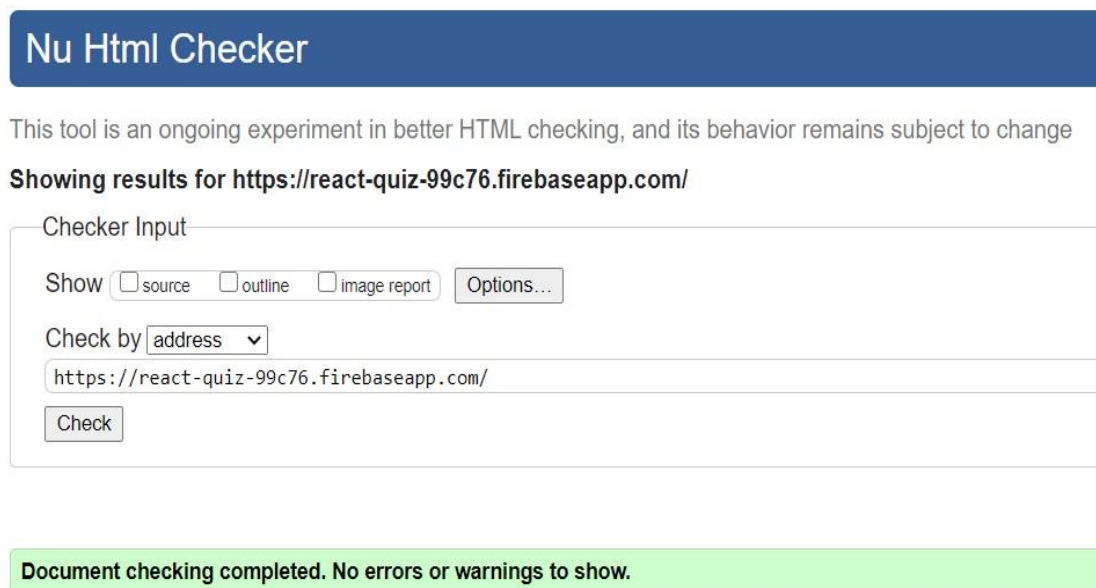


Рисунок 4.1 – HTML тестування сайту

Наступна перевірка програмного продукту за допомогою CSS валідатора. На рисунку 4.2 зображено результат тестування веб-сайту.

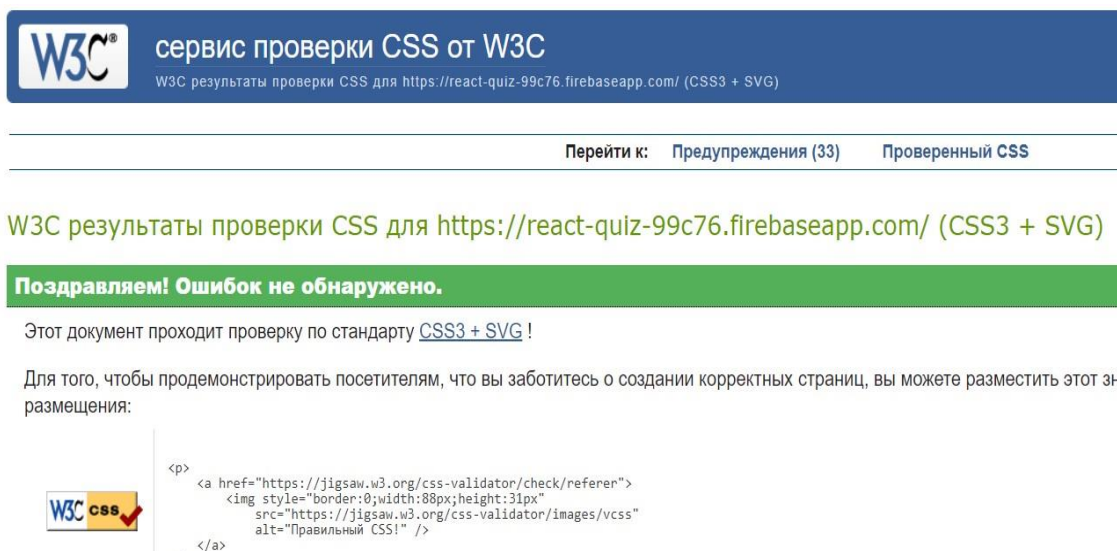
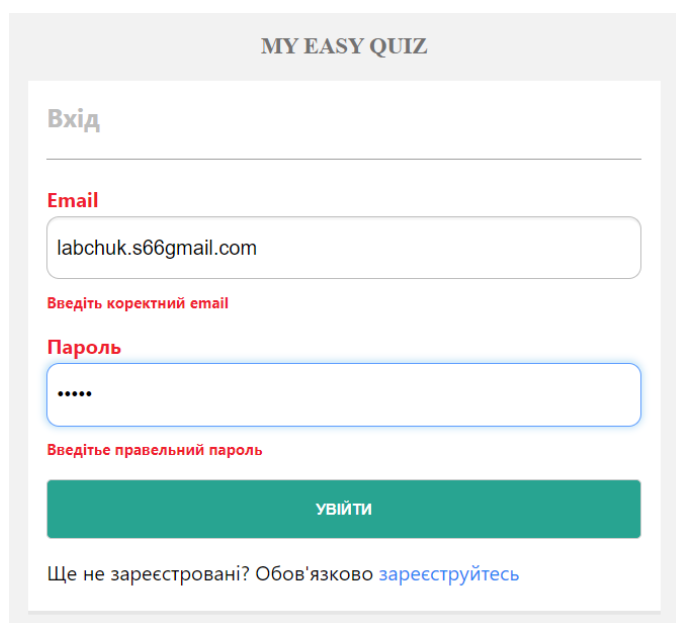


Рисунок 4.2 – Тестування CSS

Після тестування на різних комп'ютерах проблеми нам також необхідно переконатися, що алгоритми та методи системи працюють правильно, без помилок. Форми входу та реєстрації протестовані. Дані коректно вводяться та відображаються у полі. Якщо користувач вводить невірні дані, перевірка даних працюватиме, і користувач не зможе продовжити вхід або реєстрацію (рис. 4.3). Ваша адреса електронної пошти повинна відповідати цим критеріям, а ваш пароль повинен складатися не менше ніж з шести символів і не містити пробілів.



The image shows a login form titled "MY EASY QUIZ". It has a "Вхід" (Login) section. The "Email" field contains "labchuk.s66gmail.com" and has a red error message below it: "Введіть коректний email". The "Пароль" (Password) field contains six dots and has a red error message below it: "Введіть правильний пароль". There is a green "УВІЙТИ" (Login) button. At the bottom, there is a link: "Ще не зареєстровані? Обов'язково [зареєструйтесь](#)".

Рисунок 4.2 – Валідація форм

Також було проведено тест на правильність виведення результатів тестування. Після цього жодних багів та помилок не виявлено, все працює коректно, результати розраховуються коректно (рис. 4.3).

Java Script

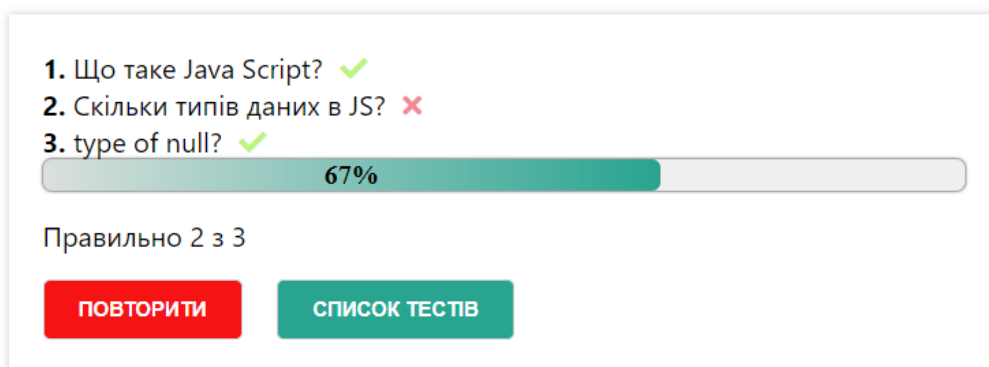


Рисунок 4.3 – Результат проходження тесту

Наступний тест створення навчального тесту. Було перевірено створення загального тесту по спеціальності з будь якими запитаннями та різними режимами, також заповнення базової інформації про тест, наприклад назву тесту та його спеціалізацію (рис. 4.4).

Новий тест

✓ ТЕСТ СТВОРЕНО ✗

Введіть питання

Відповідь 1

Відповідь 2

The image shows a form titled 'Новий тест' (New Test). At the top, there is a black alert box with a green checkmark and the text 'ТЕСТ СТВОРЕНО' (TEST CREATED) and a close button. Below the alert, there are three input fields: 'Введіть питання' (Enter question), 'Відповідь 1' (Answer 1), and 'Відповідь 2' (Answer 2).

Рисунок 4.3 – Алерт успішно створеного тесту

Веб-додаток був протестований з використанням усіх правил та можливих методів. Коли ви тестуєте свій сайт у різних браузерах, ваш сайт виглядатиме однаково у всіх браузерах та на мобільних пристроях. Також було перевірено юзабіліті сайту, встановлено, що сайт має зручну навігацію та чітку структуру. Цей сайт був протестований з використанням валідаторів HTML та CSS, і помилок виявлено не було.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку розроблених спеціалізованих програмних засобів для здобуття професійних знань в галузі програмного забезпечення, який розроблено для покращення якості освіти та для оптимізації віддаленого навчального процесу. Останнім часом все більше компаній та академій які починають працювати у віддаленому форматі.

Особливістю розробки є те, що даний застосунок має зручний інтерфейс, систему для зручної перевірки домашніх завдань, систему отримання результатів та систему проведення лекцій, що дозволяє оптимізувати процес навчання віддалено. Також застосунок не менш актуальний для студентів. Для них застосунок пропонує систему тестування після якого система надає матеріал для покращення знань.

Аналогом може бути WebStudy за ціною 110000 грн., або j StudeVerse за ціною 150000 грн.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п'ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 5.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність в реальних умовах

Продовження табл. 5.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 5.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
----	--	---	---	---	---

12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту
----	---	--	---	--	---

Усі дані по кожному параметру занесено в таблиці 5.2

Таблиця 5.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	3
Експлуатаційні витрати	3	4	3
Ринок збуту	4	3	4
Конкурентоспроможність	3	4	3
Фахівці з технічної і комерційної реалізації	4	3	4
Фінансування	4	4	3
Матеріально-технічна база	3	3	3
Термін реалізації ідеї	4	4	4
Супровідна документація	4	3	4
Сума	43	42	43
Середньоарифметична сума балів	$(43+42+43) / 3 = 42,67$		

За даними таблиці 5.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 5.3.

Таблиця 5.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький

11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок того, що даний застосунок має зручний інтерфейс, систему для зручної перевірки домашніх завдань, систему отримання результатів та систему проведення лекцій, що дозволяє оптимізувати процес навчання віддалено. Також застосунок не менш актуальний для студентів. Для них застосунок пропонує систему тестування після якого система надає матеріал для покращення знань.

5.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

5.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

T_p – число робочих днів за місяць, 20 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.4.

Таблиця 5.4 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	37000	1850,00	32	59200,000
Програміст	34500	1725,00	32	55200,000

Всього	114400,00
--------	-----------

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

5.2.2 Додаткова заробітна плата розробників, які брати участь в розробці обладнання/програмного продукту.

Додаткову заробітну плату прийнято розраховувати як 15 % від основної заробітної плати розробників та робітників:

$$Z_d = Z_o \cdot 15 \% / 100 \%, \quad (4.2)$$

$$Z_d = (114400,00 \cdot 15 \% / 100 \%) = 17160,00 \text{ (грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$H_3 = (Z_o + Z_d) \cdot 22 \% / 100\% \quad (4.3)$$

$$H_3 = (114400,00 + 17160,00) \cdot 22 \% / 100 \% = 28943,20 \text{ (грн.)}$$

5.2.4. Оскільки для розроблювального пристрою не потрібно витратити матеріали та комплектуючі, то витрати на матеріали і комплектуючі дорівнюють нулю.

5.2.5 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді розраховується за формулою:

$$A = \frac{Ц}{T_в} \cdot \frac{t_{вук}}{12} \text{ [Грн.]}. \quad (4.4)$$

де Ц – балансова вартість обладнання, грн.;

T – термін корисного використання обладнання згідно податкового законодавства, років

$t_{\text{вик}}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 47000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,60 міс.

$$A_{\text{обл}} = \frac{47000}{2} \times \frac{1,6}{12} = 3133,33 \text{ грн.}$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.5.

Так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив (Webstorm) не амортизується, а його вартість включається у вартість розробки повністю, $B_{\text{нем.ак.}} = 1040$ грн.

Таблиця 5.5 – Амортизаційні відрахування на матеріальні та нематеріальні ресурси для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн.
Комп'ютер та комп'ютерна периферія (Asus Vivobook X512FL, Iphone 11)	47000	2	1,60	3133,333
Офісне обладнання (меблі)	25000	4	1,60	833,333
Приміщення	985000	20	1,60	6566,667
Всього				10533,33

5.2.6 Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_n, \quad (4.5)$$

де V – вартість 1 кВт-години електроенергії для 1 класу підприємства, $V = 6,2$ грн./кВт;

P – встановлена потужність обладнання, кВт. $P = 0,4$ кВт;

Φ – фактична кількість годин роботи обладнання, годин.

K_n – коефіцієнт використання потужності, $K_n = 0,9$.

$$V_e = 0,9 \cdot 0,4 \cdot 8 \cdot 32 \cdot 6,2 = 571,392 \text{ (грн.)}$$

5.2.7 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників:

$$I_e = (Z_o + Z_p) \cdot \frac{H_{ib}}{100\%}, \quad (4.6)$$

де H_{ib} – норма нарахування за статтею «Інші витрати».

$$I_e = 114400,00 * 74\% / 100\% = 84656 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників:

$$H_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.7)$$

де $H_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$H_{нзв} = 114400,00 * 143 \% / 100 \% = 163592 \text{ (грн.)}$$

5.2.8 Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 114400,00 + 17160,00 + 28943,20 + 10533,33 + 1040 + 571,39 + 84656 + \\ + 163592 = 420895,93 \text{ грн.}$$

5.2.9 Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \quad (4.8)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то $\eta=0,1$; технічного проектування, то $\eta=0,2$; розробки конструкторської документації, то $\eta=0,3$; розробки технологій, то $\eta=0,4$; розробки дослідного зразка, то $\eta=0,5$; розробки промислового зразка, то $\eta=0,7$; впровадження, то $\eta=0,9$. Оберемо $\eta = 0,5$, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 420895,93 / 0,5 = 841792 \text{ грн.}$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів цієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

з) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором можна об'єднати, враховуючи визначені ситуації з відповідними умовами.

5.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta\Pi_i = (\pm\Delta\Pi_o \cdot N + \Pi_o \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\rho}{100}\right), \quad (4.9)$$

де $\pm\Delta\Pi_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

Π_o – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $\Pi_o = \Pi_o \pm \Delta\Pi_o$;

Π_b – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році $\vartheta = 18\%$.

Припустимо, що при прогнозованій ціні 8800 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на 1500 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 2000 шт., протягом другого року – на 1500 шт., протягом третього року на 1000 шт. До моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta\Pi_1 = (0*1500 + (8800 + 1500)*2000)*0,8333*0,26*(1 - 0,18) = 3126933,208 \text{ грн.}$$

$$\Delta\Pi_2 = (0*1500 + (8800 + 1500)*(2000+1500))*0,8333*0,26*(1 - 0,18) = 6404883,077 \text{ грн.}$$

$$\Delta\Pi_3 = (0*1500 + (8800 + 1500)*(2000+1500+1000))*0,8333*0,26*(1 - 0,18) = 8234849,671 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 17766665,96 грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків $ПП$, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1 + \tau)^i}, \quad (5.10)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

T – період часу, протягом якого виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо, починаючи з першого року:

$$\begin{aligned} \text{ПП} = & (3126933,208/(1+0,1)^1) + (6404883,077/(1+0,1)^2) + (8234849,671/ \\ & / (1+0,1)^3) = 2842666,55 + 5293291,799 + 6186964,441 = 14322922,79 \text{ грн.} \end{aligned}$$

Далі розраховують величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{inv} * ZB, \quad (4.11)$$

де k_{inv} – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{inv} = 2 \dots 5$, але може бути і більшим;

ZB – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 841792 = 1683583,70 \text{ грн.}$$

Тоді абсолютний економічний ефект E_{abc} або чистий приведений дохід (NPV , *Net Present Value*) для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{abc} = \text{ПП} - PV, \quad (4.12)$$

$$E_{abc} = 14322922,79 - 1683583,70 = 12639339,09 \text{ грн.}$$

Оскільки $E_{abc} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої норми дохідності (*IRR, Internal Rate of Return*) вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_e . Для цього використаємо формулу:

$$E_e = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.13)$$

$T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_e = \sqrt[3]{(1 + 12639339,09/1683583,70) - 1} = 1,041$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.14)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні $d = (0,09...0,14)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05...0,5)$.

$$\tau_{\min} = 0,14 + 0,05 = 0,19.$$

Так як $E_b > \tau_{\min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_g}, \quad (4.15)$$

$$T_{ок} = 1 / 1,041 = 0,96 \text{ р.}$$

Оскільки $T_{ок} < 3$ -х років, а саме термін окупності рівний 0,96 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 841792 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,96 роки.

ВИСНОВКИ

З кожним роком освіта людей все більше переходить до Інтернету, або онлайн-режиму. Кожен навчальний заклад має свій сайт чи програму з освітніми ресурсами. Тому було прийнято рішення розробити більш просунуте та зручне веб-додаток для тестування студентів.

Було проаналізовано завдання та обрано відповідні рішення з використанням сучасного HTML5, CSS 3, JavaScript та його фреймворку React.js. Проведено аналіз останніх розробок веб-технологій для обґрунтування вибору засобів реалізації програмного продукту. Проведено порівняльний аналіз існуючих реалізацій з урахуванням усіх недоліків та переваг.

Розроблено метод та моделі і алгоритми роботи тестової системи.

Розроблено мережеву структуру веб-додатку та розробили інформаційне наповнення зі зручним інтерфейсом. Інтерфейс простий, але вибраний таким чином, щоб його було легко використовувати та легко переміщатися сайтом. Основними сторінками веб-додатків є «Головна», «Тести», «Мої тести», «Вхід» та «Реєстрація».

Дизайн веб-сайту розроблено із загальноприйнятих правил з урахуванням особливостей сприйняття кольорів користувачами.

Було проаналізовано основні інструменти для реалізації веб-сайтів, обрано середовище розробки WebStorm, а також відібрано найкращі інструменти для розробки сайтів для студентського тестування. Порівняльний аналіз показав, що використання React.js виявилось найефективнішим. Розроблено різноманітні програмні модулі. Візуальний дизайн веб-сайту був створений за допомогою каскадних таблиць стилів з HTML 5 та CSS 3.

Було проведено комерційну оцінку розробки, щоб показати, що технологія має достатній рівень комерційної життєздатності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методи оцінювання студентів [Електронний ресурс] – Режим доступу до ресурсу: <https://osvita.ua/vnz/reports/pedagog/14679/>.
2. Тестова система «Testorium» [Електронний ресурс] – Режим доступу до ресурсу: <https://www.testorium.net/ua/>
3. Бевз С. В. Розробка веб-додатку для тестування знань студентів/ С. В. Бевз, С. М. Бурбело, О. В. Гаврилюк, О. М. Лабчук [Електронний ресурс] – Режим доступу до ресурсу: <https://conferences.vntu.edu.ua/index.php/mn/mn2021/paper/view/13237>.
4. Тестова система «На урок» [Електронний ресурс] – Режим доступу до ресурсу: <https://naurok.ua/student/tests>.
5. Тестова система «Test.edu.vn.ua» [Електронний ресурс] – Режим доступу до ресурсу: <https://test.edu.vn.ua/>
6. Тестова система «Тестування.укр» [Електронний ресурс] – Режим доступу до ресурсу: <https://xn--80adi8aaufcj8j.xn--j1amh/testelex/>.
7. Learn Java Script [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/>
8. CSS [Електронний ресурс] – Режим доступу до ресурсу: <http://ru.wikipedia.org/wiki/CSS>.

9. React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>.
10. JavaScript [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.javascript.ru/>.
11. Ленгсторф Дж. HTML и CSS для профессионалов / Дж. Ленгсторф. – К.: Диасофт, 2011. – 564с.
12. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов. Мощные инструменты для разработки современных веб-приложений / П. Лабберс, Б.Олберс, Ф. Салим. – СПб.: Питер, 2011. – 272 с.
13. SkillBox - дизайн сайта [Електронний ресурс] – Режим доступу до ресурсу: https://skillbox.ru/media/design/zachem_saytu_nuzhen_dizayn/.
14. Графічне оформлення [Електронний ресурс] – Режим доступу до ресурсу: http://uk.wikipedia.org/wiki/Графічне_оформлення.
15. Структура веб-сайту [Електронний ресурс] – Режим доступу до ресурсу: <http://www.vpu20.lviv.ua/images/library/informatyka/okg02..>
16. Печников В.М. Створення Web-сайтов без сторонньої допомоги /Печников В.М – М.: Тріумф, 2006
17. Реалізація веб-сайту [Електронний ресурс] – Режим доступу до ресурсу: <https://websait.uz.ua/website-development/etapy-rozrobky-sajtu/>.
18. Методи оцінювання студентів [Електронний ресурс] – Режим доступу до ресурсу: https://www.narodnaosvita.kiev.ua/?page_id=3381.
19. Веб-застосунок [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%BD%D0%BE%D0%BA>.
20. Бібліотека React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>.
21. Алгоритми та структури даних на JS [Електронний ресурс] – Режим доступу до ресурсу: <https://habr.com/ru/post/359192/>.

ДОДАТОК А

Технічне завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

ЗАТВЕРДЖУЮ
Завідувач кафедри ОТ
проф., д.т.н. Азаров О.Д.

" _____ " 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

“Комплекс спеціалізованих програмних засобів для здобуття професійних знань в галузі програмного забезпечення ”

08-23.МКР.024.00.000 ПЗ

Науковий керівник: доцент к.т.н.

Азаров О.Д. Прізвище І.Б.

Студент групи 1КІ-20м

Лабчук О.М. Прізвище І.Б.

ВНТУ 2022

1 Підставою для виконання магістерської кваліфікаційної роботи (МКР)

1.1 Важливим є актуальність дослідження у напрямку магістерської роботи, яка обумовлена тим, що технології сьогодні не стоять на місці. Кількість інструментів, які використовуються для створення веб-сайтів, зростає з кожним роком, деякі з них не підтримуються або не використовуються розробниками. Створені ресурси орієнтовані на студентів, які хочуть покращити свої знання в певній галузі або просто хочуть перевірити свої знання. Додаток також корисний для вчителів, які можуть легко створювати тести для перевірки своїх учнів. Системи тестування набувають сьогодні все більшої популярності, оскільки за їх допомогою можна швидко й об'єктивно оцінити рівень знань учня. Тести використовуються скрізь, від невеликих гуртків до вищих навчальних закладів.

2 Мета МКР і призначення розробки

2.1 Мета роботи — покращення якості тестування студентів, за рахунок розробленої спеціальної веб-системи, що забезпечує віддалений доступ користувачам;

2.2 Призначення розробки — підвищення ефективності тестування студентів та зручності користування системою.

3 Вихідні дані для виконання МКР

3.1 Проведення аналізу існуючих принципів та технологій навчальних систем;

3.2 Розробка структури та функціональної схеми створення тестів для перевірки знань та їх результатів;

3.3 На основі структурних та функціональних схем здійснено систему рейтингів для студентів;

3.4 Виконання розрахунків для доведення доцільності нової розробки з економічної точки зору;

4 Вимоги до виконання МКР

Конструкція пристрою повинна відповідати естетичним та ергономічним вимогам, повинна бути зручною в обслуговуванні та керуванні. Графічна та текстова документація повинна відповідати діючим стандартам України.

5 Етапи МКР та очікувані результати

Етапи роботи та очікувані результати приведено в Таблиці А.1.

Таблиця А.1 — Етапи МКР

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Огляд і аналіз існуючих методів та рішень розробки навчальних систем	15.09.2021	26.09.2021	Аналітичний огляд літературних джерел, задачі досліджень, розділ 1 ПЗ
2	Розробка методу та моделі системи тестування	27.09.2021	15.10.2021	Розділ 2
3	Розробка програмних засобів реалізації веб-сайту	16.10.2021	7.11.2021	Розділ 3
4	Підготовка економічної частини	8.11.2021	21.11.2021	Розділ 4
5	Апробація та впровадження результатів дослідження	8.11.2021	10.11.2021	Тези доповідей
6	Опублікування результатів досліджень	22.11.2021	30.11.2021	Стаття
7	Оформлення пояснювальної записки, графічного матеріалу і презентації	22.11.2021	30.11.2021	ПЗ, графіч. матеріал і презентація
8	Підготовка супроводжуючих документів, їх підписування, проходження нормоконтролю та тесту на плагіат	22.11.2021	30.11.2021	Оформлені документи

6 Матеріали, що подаються до захисту МКР

До захисту подаються: пояснювальна записка МКР, графічні і ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, відгук опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

7 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Екзаменаційної комісії, затвердженої наказом ректора.

8 Вимоги до оформлювання та порядок виконання МКР

8.1 При оформлювання МКР використовуються:

— ДСТУ 3008: 2015 «Звіти в сфері науки і техніки. Структура та правила оформлювання»;

— ДСТУ 8302: 2015 «Бібліографічні посилання. Загальні положення та правила складання»;

— міждержавний ГОСТ 2.104-2006 «Єдина система конструкторської документації. Основні написи»;

— Методичні вказівки до виконання магістерських кваліфікаційних робіт зі спеціальності 123 — «Комп'ютерна інженерія». Кафедра обчислювальної техніки ВНТУ 2022;

— документами на які посилаються у вище вказаних.

8.2 Порядок виконання МКР викладено в «Положення про кваліфікаційні роботи на другому (магістерському) рівні вищої освіти СУЯ ВНТУ-03.02.02-П.001.01:21».

ДОДАТОК Б

ЛІСТИНГ ПРОГРАМИ

Script.js

```

import React, { Component } from 'react'
import ActiveQuiz from '../components/ActiveQuiz/ActiveQuiz'
import FinishedQuiz from '../components/FinishedQuiz/FinishedQuiz'
import classes from './Quiz.module.css'
import Loader from '../components/UI/Loader/Loader'
import { connect } from 'react-redux'
import {
  fetchQuizById,
  quizAnswerClick,
  retryQuiz,
} from '../store/actions/actionsQuiz'

class Quiz extends Component {
  componentDidMount() {
    this.props.fetchQuizById(this.props.match.params.id)
    console.log(this.props.quiz)
  }
  componentWillUnmount() {
    this.props.retryQuiz()
    console.log(this.props.quiz)
  }
  render() {
    return (
      <div className={classes.Quiz}>
        <div className={classes.QuizWrapper}>
          <h1>{this.props.location.propsQuiz}</h1>
          {this.props.loading || !this.props.quiz ? (
            <Loader />
          ) : this.props.isFinished ? (
            <FinishedQuiz
              result={this.props.result}
              quiz={this.props.quiz}
              onRetry={this.props.retryQuiz}
            />
          ) : (
            <ActiveQuiz

```

```

        answers={this.props.quiz[this.props.activeQuestion].answers}
        question={this.props.quiz[this.props.activeQuestion].question}
        onAnswerClick={this.props.quizAnswerClick}
        quizLength={this.props.quiz.length}
        answerNumber={this.props.activeQuestion + 1}
        state={this.props.answerState}
      />
    )}
  </div>
</div>
)
}
}
function mapStateToProps(state) {
  return {
    result: state.quiz.result,
    isFinished: state.quiz.isFinished,
    activeQuestion: state.quiz.activeQuestion,
    answerState: state.quiz.answerState,
    quiz: state.quiz.quiz,
  }
}
function mapDispatchToProps(dispatch) {
  return {
    fetchQuizById: (id) => dispatch(fetchQuizById(id)),
    quizAnswerClick: (answerId) => dispatch(quizAnswerClick(answerId)),
    retryQuiz: () => dispatch(retryQuiz()),
  }
}
export default connect(mapStateToProps, mapDispatchToProps)(Quiz)

import React, { Component, Fragment } from 'react'
import Button from '../components/UI/Button/Button'
import { createControl, validate, validateForm } from '../form/formFramework'
import Input from '../components/UI/Input/Input'
import InputName from '../components/UI/InputName/InputName'
import classes from './QuizCreator.module.css'
import Select from '../components/UI/Select/Select'
import { connect } from 'react-redux'
import {
  createQuizName,

```

```

    createQuizQuestion,
    finishCreateQuiz,
} from '../..../store/actions/actionCreate'
import Loader from "../..../components/UI/Loader/Loader";

function createOptionControl(index) {
  return createControl(
    {
      label: `Відповідь ${index}`,
      errorMessage: 'Поле не може бути пустим',
      id: index,
    },
    {
      required: true,
    }
  )
}

function createFormControl() {
  return {
    question: createControl(
      {
        label: 'Введіть питання',
        errorMessage: 'Поле не може бути пустим',
      },
      { required: true }
    ),
    option1: createOptionControl(1),
    option2: createOptionControl(2),
    option3: createOptionControl(3),
    option4: createOptionControl(4),
  }
}

class QuizCreator extends Component {
  state = {
    isFormValid: false,
    rightAnswerId: 1,
    formControls: createFormControl(),
    nameTes: '',
    createTest: false
  }

  submitHandler = (event) => {

```



```
    event.preventDefault()
  }

addQuestionHandler = (event) => {
  event.preventDefault()
  const {
    question,
    option1,
    option2,
    option3,
    option4,
  } = this.state.formControls
  const questionItem = {
    question: question.value,
    id: this.props.quiz.length + 1,
    rightAnswerId: this.state.rightAnswerId,
    answers: [
      { text: option1.value, id: option1.id },
      { text: option2.value, id: option2.id },
      { text: option3.value, id: option3.id },
      { text: option4.value, id: option4.id },
    ],
  }
  this.props.createQuizQuestion(questionItem)
  this.setState({
    isValid: false,
    rightAnswerId: 1,
    formControls: createFormControl(),
  })
}

createQuizHandler = (event) => {
  event.preventDefault()
  console.log('fef')
  this.setState({
    isValid: false,
    rightAnswerId: 1,
    formControls: createFormControl(),
  })
  this.props.createQuizName(this.state.nameTes)
  this.props.finishCreateQuiz()
  console.log(this.nameTes)
```

```

}

changeHandler = (value, controlName) => {
  console.log(value)
  const formControls = { ...this.state.formControls }
  const control = { ...formControls[controlName] }
  control.value = value
  control.touched = true
  control.valid = validate(control.value, control.validation)
  formControls[controlName] = control

  this.setState({
    formControls,
    isValid: validateForm(formControls),
  })
}

createTestVisible = () => {
  this.setState({
    createTest: !this.state.createTest
  })
}

changeHandlerName = (value) => {
  // console.log(value)
  let oldNameTes = this.state.nameTest
  oldNameTes = value
  this.setState({
    nameTes: oldNameTes,
  })
  console.log(this.state.nameTes)
}

renderControl() {
  return Object.keys(this.state.formControls).map((controlName, index) => {
    const control = this.state.formControls[controlName]
    return (
      <Fragment key={controlName + index}>
        <Input
          value={control.value}
          valid={control.valid}
          touched={control.touched}
          label={control.label}
        />
      </Fragment>
    )
  })
}

```

```

        errorMessage={control.errorMessage}
        shouldValidate={!control.validation}
        onChange={(event) =>
            this.changeHandler(event.target.value, controlName)
        }
    />
    {index === 0 ? <hr /> : null}
</Fragment>
)
})
}
nameTest() {
    return (
        <InputName
            onChangeName={(event) => this.changeHandlerName(event.target.value)}
        />
    )
}
selectChangeHandler = (event) => {
    this.setState({
        rightAnswerId: +event.target.value,
    })
}
render() {
    const cls = [classes.Wrapper]
    if (this.state.createTest) {
        cls.push(classes.Show)
    }
    const select = (
        <Select
            label="Виберіть правельну відповідь"
            value={this.state.rightAnswerId}
            onChange={this.selectChangeHandler}
            options={[
                { text: 1, value: 1 },
                { text: 2, value: 2 },
                { text: 3, value: 3 },
                { text: 4, value: 4 },
            ]}
        />
    )
}

```

```

return (
  <div className={classes.QuizCreator}>
    <div className={classes.container}>
      <h1>У вас немає тестів</h1>
      <span>Створи свій власний тест</span>
      <div onClick={this.createTestVisible} className={this.state.createTest ?
classes.qurqle : classes.qurqleClose}>
        
      </div>
    </div>
    <div className={cls.join(' ')}>
      <h1>Створення теста</h1>
      <form onSubmit={this.submitHandler}>
        {this.renderControl()}
        {select}
        {this.nameTest()}
        <Button
          style={{ marginRight: '10px' }}
          type="primary"
          onClick={this.addQuestionHandler}
          disabled={!this.state.isFormValid}
        >
          Додати питання
        </Button>
        <Button
          type="success"
          onClick={this.createQuizHandler}
          disabled={this.props.quiz.length === 0}
        >
          Додати тест
        </Button>
      </form>
    </div>
  </div>
)
}
}
function mapStateToProps(state) {
  return {
    quiz: state.create.quiz,
  }
}

```

```

}

function mapDispatchToProps(dispatch) {
  return {
    createQuizName: (name) => dispatch(createQuizName(name)),
    createQuizQuestion: (item) => dispatch(createQuizQuestion(item)),
    finishCreateQuiz: () => dispatch(finishCreateQuiz()),
  }
}

export default connect(mapStateToProps, mapDispatchToProps)(QuizCreator)

class QuizList extends Component {
  renderQuizzes() {
    return this.props.quizzes.map((quiz) => {
      console.log(quiz)
      return (
        <li key={quiz.id}>
          <NavLink to={{ pathname: '/quiz/' + quiz.id, propsQuiz: quiz.name }}>
            {quiz.name}
          </NavLink>
        </li>
      )
    })
  }
  componentDidMount() {
    this.props.fetchQuizzes()
    console.log(this.quizitem)
  }
  render() {
    return (
      <div className={classes.QuizList}>
        <div className={classes WrapperQuizList}>
          <h1>Список тестів</h1>
          <div className={classes.Tests}>
            <div className={classes.Link}>
              <h1>Програмування</h1>
              {this.props.loading && this.props.quizzes.length !== 0 ? (
                <Loader />) : (
                  <ul>{this.renderQuizzes()}</ul>
                )}
            </div>
          </div>
        </div>
      )
    )
  }
}

```

```

<div className={classes.Link}>
  <h1>Веб-дизайн</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>

  <h1>Маркетинг</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>
  <h1>Seo</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>
  <h1>Менеджмент</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>
  <h1>Копирайтинг</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>

```

```

<div className={classes.Link}>
  <h1>SMM</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>
  <h1>Фото и відео обробка</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
<div className={classes.Link}>
  <h1>Іноземні мови</h1>
  {this.props.loading && this.props.quizes.length !== 0 ? (
    <Loader />
  ) : (
    <ul>{this.renderQuizes()}</ul>
  )}
</div>
</div>
</div>
</div>
)
}
}
function mapStateToProps(state) {
  return {
    quizes: state.quiz.quizes,
    loading: state.quiz.loading,
    quizitem: state.quiz.quizName,
  }
}
function mapDispatchToProps(dispatch) {
  return {
    fetchQuizes: () => dispatch(fetchQuizes()),
  }
}

```

```
}

```

```
export default connect(mapStateToProps, mapDispatchToProps)(QuizList)

```

Style.css

```
.Quiz {
  display: flex;
  justify-content: center;
  align-items: center;
  flex-grow: 1;
  width: 100%;
  background: #f2f2f2;
  background-image: url("../img/Vector.svg");
  background-size: contain;
  height: 100vh;
}

```

```
.Quiz h1 {
  color: #000;
  margin-left: 10px;
  font-family: Domine;
  font-size: 24px;
}

```

```
.QuizWrapper {
  width: 600px;
  max-width: 1200px;
}

```

```
@media (max-width: 520px) {
  .Quiz h1 {
    color: #000;
    margin-left: 10px;
    font-size: 14px;
  }
  .Quiz {
  }
}

```

```
.QuizCreator {
  display: flex;
  background-color: #f2f2f2;
  background-image: url("../img/Vector.svg");
}

```



```
background-size: contain;
align-items: center;
width: 100%;
min-height: 100vh;
flex-direction: column;
}
.container {
margin-top: 100px;
display: flex;
flex-direction: column;
align-items: center;
min-height: 100%;
}
.container h1 {
font-size: 30px;
}
.qurqle img{
width: 30px;
transform: rotate( 180deg);
}
.qurqleClose img{
width: 30px;
transform: rotate(0deg);
}
.qurqle{
margin-top: 20px;
border: 2px solid #000;
border-radius: 50%;
width: 50px;
height: 50px;
display: flex;
justify-content: center;
align-items: center;
cursor: pointer;
}
.qurqleClose{
cursor: pointer;
margin-top: 20px;
border: 2px solid #000;
border-radius: 50%;
width: 50px;
```

```
    height: 50px;
    display: flex;
    justify-content: center;
    align-items: center;
}
.QuizCreator > div {
    max-width: 600px;
    width: 100%;
    padding: 0 20px;
}

.QuizCreator h1 {
    text-align: center;
    font-size: 30px;
}
.QuizCreator form {
    /* background: #eeeeee; */
    box-shadow: 1px 0 2px rgb(0 0 0 / 20%);
    background-color: #fff;
    padding: 15px;
}
.Wrapper {
    transform: translateY(-1100px);
    transition: transform 0.7s ease-in;
}
.Wrapper.Show {
    transform: translateY(0px);
}
@media (max-width: 520px) {
    .container {
        margin-top: 30px;
    }
    .container h1 {
        margin: 0px 0px 10px 0px;
        font-size: 20px;
    }
    .container span {
        font-size: 14px;
    }
    .QuizCreator h1 {
        font-size: 20px;
    }
}
```

```
    }  
  }  
  .QuizList {  
    display: flex;  
    justify-content: center;  
    min-height: 100vh;  
    /* background: linear-gradient(90deg, #1089a2 0%, #23c3a6 100%); */  
    /*background: linear-gradient(180deg, #fff 0%, #0d8b74 100%);*/  
    background: #f2f2f2;  
    background-image: url("../img/Vector.svg");  
    background-size: contain;  
    width: 100%;  
  }  
  .WrapperQuizList {  
    margin-top: 100px;  
    padding-left: 10px;  
    padding-right: 10px;  
    max-width: 1200px;  
  
  }  
  .QuizList h1 {  
    color: #000;  
    text-align: left;  
    font-family: Domine;  
    font-size: 24px;  
    max-width: 960px;  
    margin-left: 10px;  
  }  
  .Tests {  
    display: flex;  
    flex-wrap: wrap;  
    justify-content: center;  
    max-width: 960px;  
  }  
  .Link {  
    /*border: 1px solid #000;*/  
    /*border-radius: 5px;*/  
    padding: 20px;  
    text-align: left;  
    width: 300px;  
    margin: 0px 10px;
```

```
margin-bottom: 20px;
background-color: #fff;
box-shadow: 1px 0 2px rgb(0 0 0 / 20%);
}
.Link img {
width: 30px;
}
.Link h1 {
margin: 0;
font-size: 18px;
text-align: left;
font-family: Domine;
}

.QuizList ul {
list-style: none;
margin: 0;
padding: 0;
}

.QuizList li {
margin-bottom: 10px;
}

.QuizList li a {
text-decoration: none;
color: #000;
font-family: "Segoe UI";
}

.QuizList li a:hover {
opacity: 50%;
}
@media (max-width: 997px) {
.QuizList .Tests {
max-width: 640px;
}
}
@media (max-width: 678px) {
.QuizList .Tests {
max-width: 320px;
}
```

```
    }  
  }  
  @media (max-width: 520px) {  
    .WrapperQuizList {  
      margin-top: 20px;  
    }  
    .QuizList h1 {  
      font-size: 20px;  
    }  
    .QuizList {  
      background-size: auto;  
    }  
  }  
  .Auth {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    background-color: #f2f2f2;  
    background-image: url("../img/Vector.svg");  
    background-size: contain;  
    width: 100%;  
    flex-grow: 1;  
    height: 100vh;  
  }  
  
  .Auth > div {  
    width: 100%;  
    max-width: 600px;  
    padding: 0 20px;  
  }  
  
  .Auth h1 {  
    margin: 0;  
    padding-bottom: 20px;  
    text-align: center;  
    color: #bdbdbd;  
    font-size: 25px;  
  }  
  
  .WrapperForm {  
    box-shadow: 0 3px 0 rgb(0 0 0 / 5%);
```

```
background-color: #fff;
/*border-radius: 5px;*/
padding: 15px;
}
.SpanInfo a {
color: #2884f6;
}
```

ІЛЮСТРАТИВНА ЧАСТИНА

**КОМПЛЕКС СПЕЦІАЛІЗОВАНИХ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ ЗДОБУТТЯ
ПРОФЕСІЙНИХ ЗНАНЬ В ГАЛУЗІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

Комплекс спеціалізованих засобів для
здобуття професійних знань в галузі
програмного забезпечення

Виконав:
Лабчук О.М.

Науковий керівник:
Азаров О.Д.

Рисунок В.1 – Титульний слайд

Комплекс спеціалізованих засобів для
здобуття професійних знань в галузі
програмного забезпечення

- **Мета роботи:** покращення якості навчання студентів, заохочення працювати з даним ресурсом використовуючи зручний інтерфейс та можливість реалізації багатьох функцій на сайті.
- **Об'єкт дослідження:** процес обробки та оцінювання знань студентів.
- **Предмет дослідження:** навчальна система для студентів.



Рисунок В.2 – Мета, об'єкт, предмет дослідження

- **Наукова новизна:** Подальшого розвитку отримав метод тестування знань студентів який навідміну від існуючих забезпечує використання нових завдань та формування оцінки у відсотковій шкалі, що дозволяє адаптувати запропонований метод під будь-які вимоги до шкали оцінювання та підвищує універсальність його використання.
- Подальшого розвитку дістали модель тестування веб-системи яка навідміну від існуючих використовує уніфіковані підходи до підбору питань та шкали оцінювання і забезпечує цінний доступ користувачів за рахунок використання веб-середовища, що розширює коло потенційних користувачів.
- **Практична цінність:** Можливість проведення тестового контролю знань в середовищі веб-системи в онлайн режимі

Рисунок В.3 – Наукова новизна та практична цінність отриманих результатів

Структура магістерської кваліфікаційної роботи

- Аналіз можливих методів реалізації системи та постановка задач дослідження
- Розробка структури та алгоритмів веб застосунку
- Розробка системи тестування знань студентів
- Тестування веб застосунку

Рисунок В.4 – Структура роботи

Аналоги

- [Testorium](#)

- На Урок

- [Test.edu.vn](#)

- [Тестування.укр](#)



Testorium
.net



ТЕСТУВАННЯ.УКР

Рисунок В.5 – Порівняльний аналіз аналогів

Засоби розробки

На сьогоднішній день існує не мало методів розробки веб сайтів. Основні з них:

- Ручна розробка за допомогою HTML;
- за допомогою програмних засобів розробки сайтів;
- За допомогою інструментальних систем таких як CMS;
- З використанням популярних на цей час фреймворків;
- На SaaS-платформах у CLOUD.

Проаналізувавши основні методи розробки було вибрано метод з використанням фреймворків. За основну мову програмування було взято Java Script та її популярну бібліотеку React.js, яка дозволяє створювати великі веб-застосунки які в свою чергу змінюють дані без перезавантаження сторінки.

Рисунок В.6 – Порівняння засобів розробки

Модель формування тесту

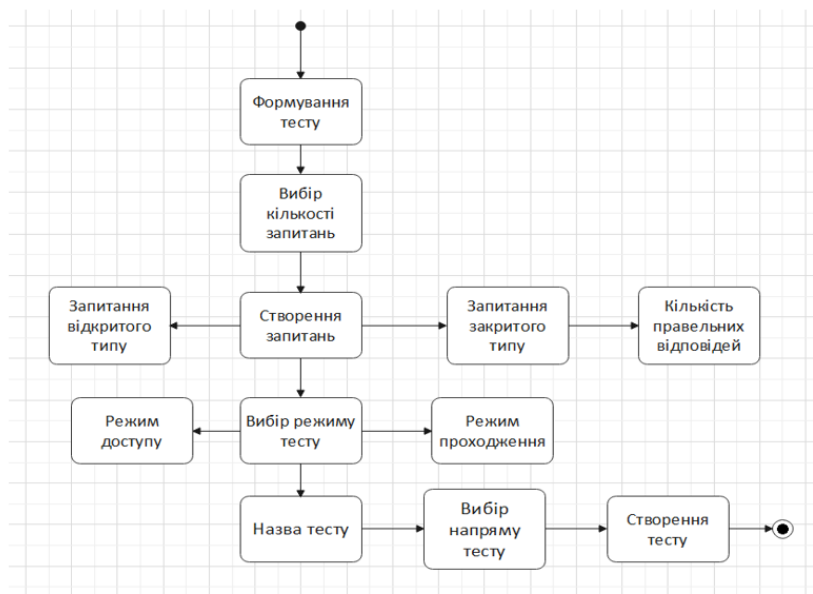


Рисунок В.7 – Модель формування тесту

ER-модель

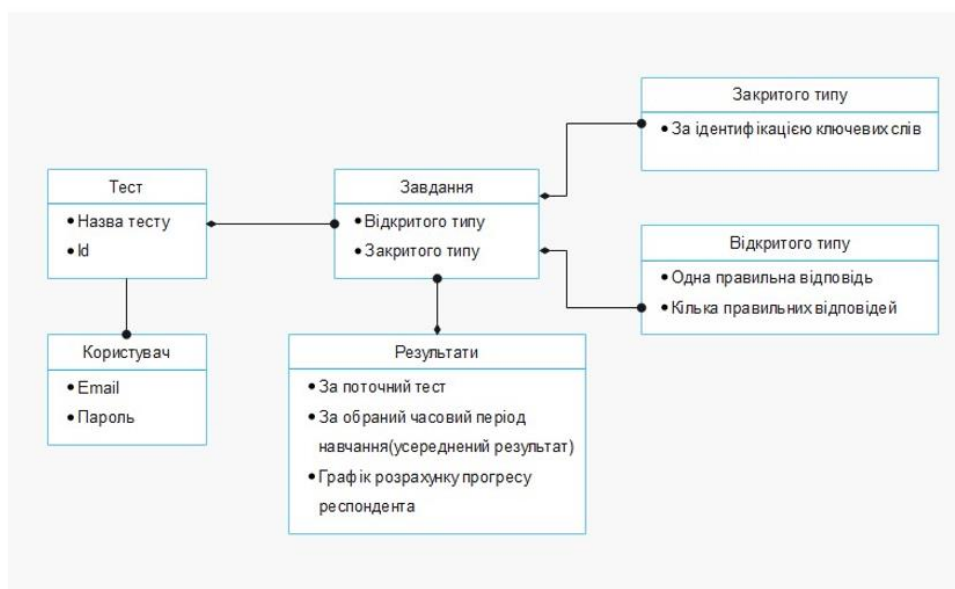


Рисунок В.8 – ER-модель

Головна сторінка системи

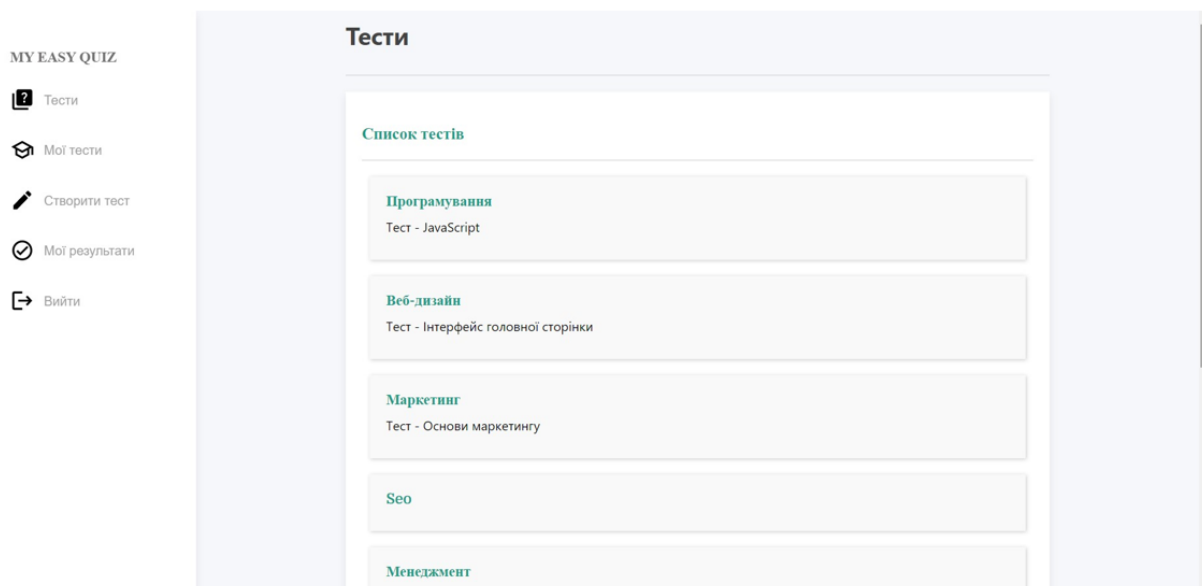


Рисунок В.9 – Головна сторінка системи

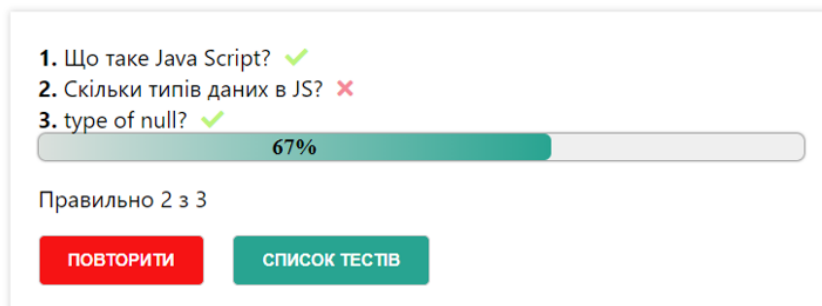
Тестування веб-застосунку

The screenshot shows the login form for the 'MY EASY QUIZ' system. The form is titled 'Вхід' (Login) and includes the following elements: an 'Email' input field containing 'labchuk.s66gmail.com' with a red error message 'Введіть коректний email' (Enter correct email) below it; a 'Пароль' (Password) input field with masked characters '.....' and a red error message 'Введіть правильний пароль' (Enter correct password) below it; a green 'УВІЙТИ' (Login) button; and a link at the bottom that reads 'Ще не зареєстровані? Обов'язково [зареєструйтесь](#)' (Not registered yet? You must [register](#)).

Рисунок В.10 – Валідація форми

Тестування веб-застосунку

Java Script



1. Що таке Java Script? ✓
2. Скільки типів даних в JS? ✗
3. type of null? ✓

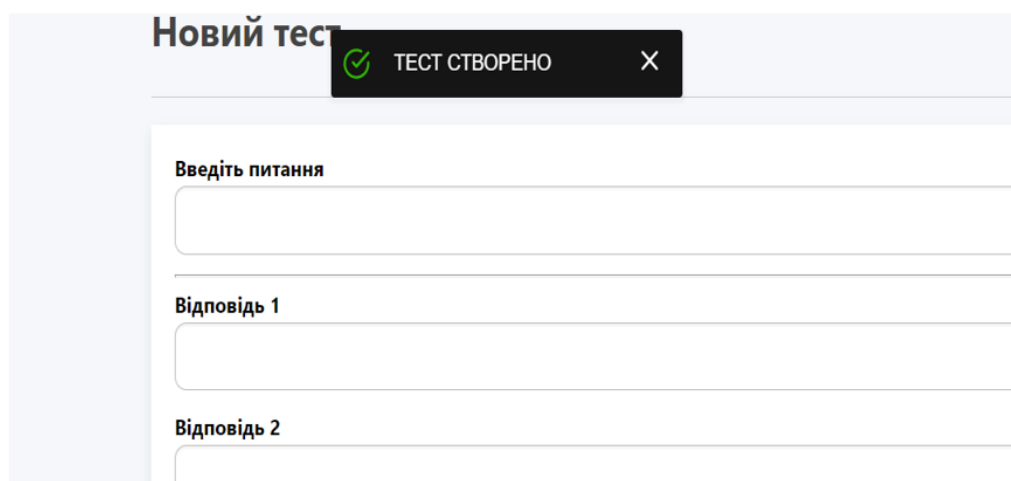
67%

Правильно 2 з 3

ПОВТОРИТИ СПИСОК ТЕСТІВ

Рисунок В.11 – Результати тестування

Тестування веб-застосунку



Новий тест

✓ ТЕСТ СТВОРЕНО ✗

Введіть питання

Відповідь 1

Відповідь 2

Рисунок В.12 – Алерт створення тесту

Тестування веб-застосунку

Новий тест

Python
Програмування РЕДАГУВАТИ

Тип питання: Одна правильна Бали: 1

Введіть питання ЗОБРАЗИТИ

Відповідь 1 ЗОБРАЗИТИ

Відповідь 2 ЗОБРАЗИТИ

Відповідь 3 ЗОБРАЗИТИ ✕

Відповідь 4 ЗОБРАЗИТИ ✕

[Додати варіант відповіді](#)
Виберіть правильну відповідь: 1 Додати питання

Новий тест

Python
Програмування РЕДАГУВАТИ

Тип питання: Одна правильна Бали: 1

Введіть питання ЗОБРАЗИТИ

Відповідь 1 ЗОБРАЗИТИ

Відповідь 2 ЗОБРАЗИТИ

Відповідь 3 ЗОБРАЗИТИ ✕

Відповідь 4 ЗОБРАЗИТИ ✕

Відповідь 5 ЗОБРАЗИТИ ✕

Відповідь 6 ЗОБРАЗИТИ ✕

[Додати варіант відповіді](#)
Виберіть правильну відповідь: 1 Додати питання

Рисунок В.13 – Перевірка валідності форми

Тестування веб-застосунку

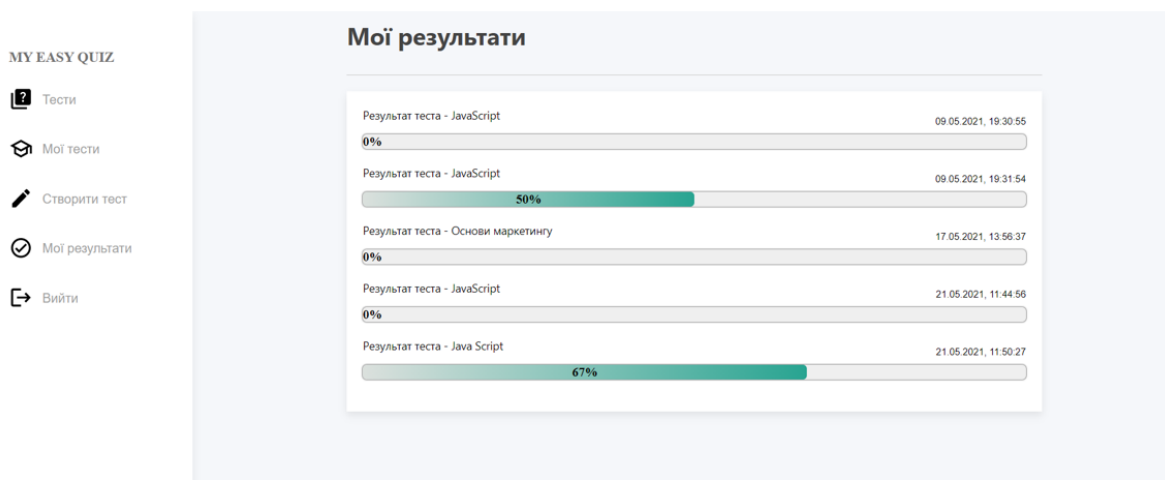


Рисунок В.14 – Результати на сторінці результатів

Тестування веб-застосунку

Результат пройденого тесту

Назва тесту: Основи Java Script
ПІБ студента: Лабчук Олександр Михайлович
Група студента: ІПІ-176
Дата проходження: 15:17:54 12-04-2021

Оцінка: **4**

Набрано балів: 38
Максимальна кількість балів: 50
Зроблено спроб: 1
Кількість правильних відповідей: 22
Питань всього: 28

Шкала оцінок:

"5"	: 90%
"4"	: 75% - 89%
"3"	: 60% - 74%
"2"	: 55% - 59%
"1"	: 20% - 54%
"0"	: 19%

[Пройти тест знову](#)

Рисунок В.15 – Запис результатів у PDF-файл

Висновки

- розроблено алгоритм авторизації та входу в систему;
- розроблено модуль формування тесту;
- розроблено модуль тестування з ідентифікований рівнем складності самого тесту та підбором відповідних питань;
- розроблено модуль формування оцінки по завершенню проходження тесту;
- перевірено функціонал розробленого засобу шляхом тестування.

Рисунок В.16 – Висновки

ДОДАТОК Г
**ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ НА
НАЯВНІСТЬ ТЕКСТОВИХ ЗАСОЗИЧЕНЬ**

Назва роботи: Комплекс спеціалізованих програмних засобів для здобуття професійних знань в галузі програмного забезпечення

Тип роботи: _____ магістерська кваліфікаційна робота _____
(БДР, МКР)

Підрозділ _____ кафедра обчислювальної техніки _____
(кафедра, факультет)

Показники звіту подібності Unichesk

Оригінальність _____ 80% _____ Схожість _____ 20% _____

Аналіз звіту подібності (відмітити потрібне):

- 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 виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- 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 _____ Захарченко С.М. _____
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichesk щодо роботи.

Автор роботи _____ Лабчук О.М. _____
(підпис) (прізвище, ініціали)

Керівник роботи _____ Азаров О.Д. _____
(підпис) (прізвище, ініціали)