

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: «**РОЗРОБКА ДОДАТКУ ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З
ПІДГОТОВКИ ДО ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ.
СПЕЦ ЧАСТИНА: РОЗРОБКА VASCEND ЧАСТИНИ ЗАСОБАМИ C#**»

Виконав: студент 4 курсу, групи ПД-42
спеціальності

121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Дубей Р.Р

(прізвище та ініціали)

Керівник Гаманюк І.М

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність - 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

О.В. Негоденко

“ _____ ” _____ 2023 року

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

Дубей Роман Русланович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання. Спец частина: Розробка WACKEND-частини засобами C#»

Керівник роботи Гаманюк І.М.,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “24” лютого 2023 року №26.

2. Строк подання студентом роботи 01.06.2023.

3. Вихідні дані до роботи:

- 3.1. Аналіз аналогів програмного забезпечення;
- 3.2. Порівняння додатку з аналогами;
- 3.3. Науково-технічна література;

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

4.1. Розробка структури додатку за допомогою мови програмування C# та фреймворку ASP.NET Core

4.3. Програмна реалізація додатку

4.4. Приклади використання та тестування застосунку

4.5. Висновки

5. Перелік графічного матеріалу.

5.1. Аналоги

5.2. Програмні засоби реалізації

5.3. Інструменти використані для реалізації

6. Дата видачі завдання: 19.12.2022

КАЛЕНДАРНИЙ ПЛАН

/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	При мітка
	Підготовка розділу 1	04.03.2023	Вик онано
	Підготовка розділу 2	04.03.2023 - 25.03.2023	Вик онано
	Підготовка розділу 3	01.04.2023 – 18.04.2023	Вик онано
	Висновки, оформлення роботи	19.04.2023 – 23.04.2023	Вик онано
	Розробка демонстраційних матеріалів	24.04.2023 – 10.05.2023	Вик онано
	Здача роботи	02.06.2023	

Студент _____ Дубей Р.Р.

(підпис) (прізвище та ініціали)

Керівник роботи _____ Гаманюк І.М.

(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 50 с., 18 рис., 13 джерел.

Актуальність даної роботи полягає у різкому збільшенні онлайн навчання, доступної користувачу для навчання в онлайні у вигляді веб додатку.

Об'єкт дослідження – вимагає використання C# для створення серверної частини програми, призначеної для допомоги студентам у підготовці до іспитів. Це вимагатиме вивчення продуктивності та масштабованості серверної частини C# з точки зору обробки кількох одночасних користувальницьких запитів, управління великими обсягами даних та забезпечення безпеки конфіденційної інформації.

Предмет дослідження – потребує вивчення кращих методів проектування та реалізації серверної архітектури для освітньої програми, яка орієнтована на надання студентам високоякісних курсів підготовки до іспитів. Це буде включати вивчення вимог програми, розробку відповідної архітектури, яка може масштабуватися і обробляти великі обсяги даних, а також реалізацію архітектури з використанням мови програмування C#.

Мета роботи – розробка серверної частини для освітньої програми, яка надає студентам курси підготовки до іспитів. Бекенд-система повинна бути спроектована так, щоб обробляти великі обсяги даних, забезпечувати ефективний та безпечний доступ до даних, а також забезпечувати масштабованість та надійність програми.

Робота виконувалася за допомогою методів конкретизації, моделювання та системного аналізу.

Суть розробки полягає в створенні конкурентно спроможного онлайн ресурсу за допомогою мов програмування C# на платформі ASP.NET Core і Razor, який буде задовольняти всі потреби користувача.

Наукова новизна полягає в створенні веб додатку за допомогою C# на платформі ASP.NET Core і Razor.

Галузь застосування: онлайн навчання.

ЗМІСТ

ВСТУП.....	10
1. ЗАСОБИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ПЗ	11
1.1 Аналіз предметної області і напрямків дослідження	11
1.2 Аналіз існуючих додатків	11
1.3 C#.....	17
1.4 Visual Studio	18
1.5 MySql.....	19
1.6..Net.....	20
1.7. ASP.NET.....	21
1.8 Entity Framework Core	22
1.9 SQL Server Management Studio	23
2. ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ	25
2.1 Обсяг та опис	25
2.2 Загальний опис	25
2.3 Опис архітектури додатку.....	26
2.4. Моделювання UML-діаграми	27
3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....	30
3.1. Розробка серверної частини.....	30
3.2 Тестування веб застосунку.....	36
3.3 Тестування веб застосунку.....	42
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	50

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

NN– Neural Network

OCR – Optical Character Recognition

LSTM - Long short-term memory

SDK – software development kit

API - Application Programming Interface

PaaS – Platform as a Services

GCNs for VRDS – Graph Convolution for Multimodal Information Extraction for Visually Rich Documents

cGAN – Conditional Generative Adversarial Networks

MLP – Multilayer perceptron

ReLU – rectified linear unit

JSON – Java Script Object Notation

GUI – Guide User Interface

QA – Quality Assurance

ПЗ – програмне забезпечення

ORM – Object-Relation Mapping

СУБД – Система Управління Базами Даних

ВСТУП

Сучасна освіта все більше переходить до використання засобів електронного навчання. Особливо актуальним стає підготовка до зовнішнього незалежного оцінювання, яке є обов'язковим для учнів 11 класу. Інтернет-ресурси, що допомагають у підготовці до зовнішнього незалежного оцінювання, стають все більш популярними серед учнів, вчителів та батьків.

Ціль даної дипломної роботи - розробка додатку, що дозволить організувати роботу з підготовки до зовнішнього незалежного оцінювання для учнів та вчителів шляхом створення спільної платформи для навчання та спілкування.

У роботі буде зосереджено увагу на розробці backend-частини додатку за допомогою мови програмування C# та системи управління базами даних MySQL. Описано архітектуру додатку та його функціонал. Також в роботі буде проведено аналіз сучасних засобів для організації навчання та підготовки до зовнішнього незалежного оцінювання, а також описано обмеження на розробку та реалізацію додатку.

Отже, ця дипломна робота буде корисною для всіх, хто зацікавлений у підготовці до зовнішнього незалежного оцінювання та розробці сучасних інтернет-ресурсів для навчання.

1. ЗАСОБИ ТА ТЕХНОЛОГІЇ ДЛЯ РОЗРОБКИ ПЗ

1.1 Аналіз предметної області і напрямків дослідження

Аналіз предметної області теми "Розробка додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання" може включати дослідження наступних напрямків:

- Вивчення вимог та критеріїв для підготовки до зовнішнього незалежного оцінювання відповідно до законодавства та вимог освітніх інституцій.
- Аналіз можливостей використання машинного навчання та штучного інтелекту для розробки персоналізованих курсів підготовки до зовнішнього незалежного оцінювання.
- Дослідження вимог та методів безпеки для розробки backend-частини додатку за допомогою мови програмування C#.
- Розгляд можливостей для інтеграції додатку з системами електронного навчання та іншими веб-сервісами.

Ці напрямки дослідження можуть допомогти визначити потреби користувачів та вимоги до додатку, а також допомогти в процесі розробки та вдосконалення додатку для оптимальної організації роботи курсів з підготовки до зовнішнього незалежного оцінювання.

1.2 Аналіз існуючих додатків

Для аналізу веб застосунків для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання було обрано наступні додатки:

1. Освіта.ua;
2. testportal.gov.ua;

Osvita.ua - сайт, що надає інформацію про освіту в Україні. Це один із найпопулярніших та всеосяжних освітніх порталів в Україні. На сайті

представлена інформація про навчальні заклади, програми та стипендії, доступні в Україні.

Osvita.ua був запущений у 2000 році Міністерством освіти та науки України. Сайт пропонує широкий спектр освітніх ресурсів, включаючи базу даних навчальних закладів, базу даних освітніх програм, базу даних стипендій та онлайн-бібліотеку. Веб-сайт також пропонує онлайн-курси, новини та статті на різні теми, пов'язані з освітою.

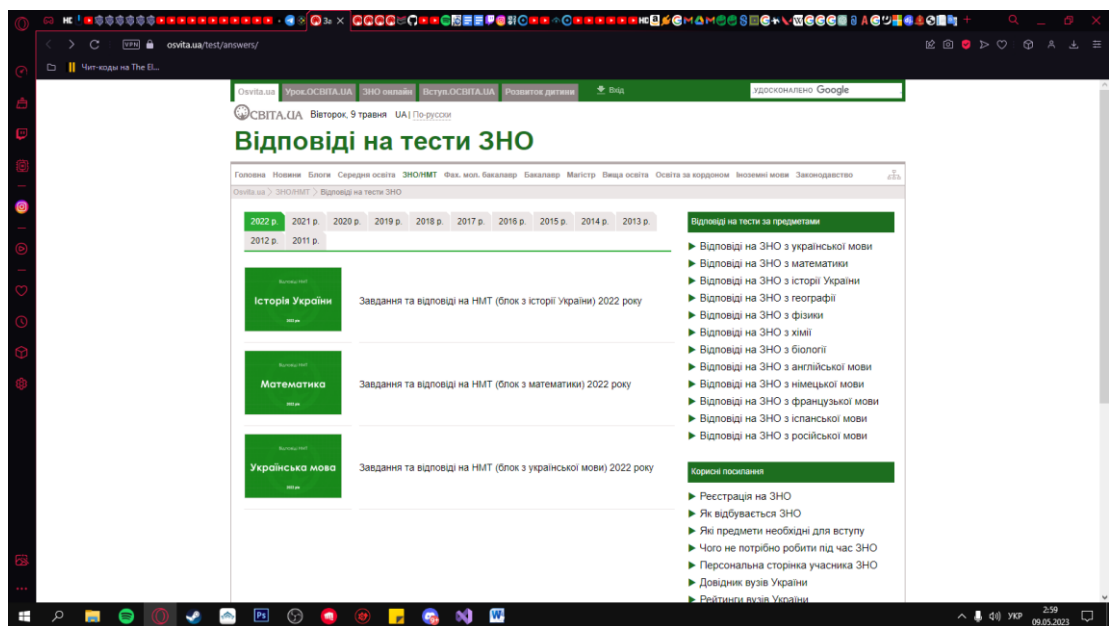


Рис 1.1. Веб-застосунок Освіта.юа

Переваги:

- Велика база даних: Osvita.ua має велику базу даних навчальних закладів, програм та стипендій, доступних в Україні. Це полегшує студентам пошук відповідних освітніх можливостей для них.
- Онлайн-бібліотека: Osvita.ua має онлайн-бібліотеку, яка надає доступ до широкого спектру освітніх ресурсів, включаючи книги, статті та дослідження.
- Онлайн-курси: Osvita.ua пропонує онлайн-курси на різні теми, пов'язані з освітою. Ці курси безкоштовні та доступні для всіх, хто має підключення до Інтернету.

- Новини та статті: Osvita.ua надає актуальні новини та статті про освіту в Україні. Це допомагає студентам, викладачам та дослідникам бути в курсі останніх подій у сфері освіти.
- Мультимовність: Osvita.ua доступний українською та англійською мовами. Це робить його доступним для ширшої аудиторії, зокрема для іноземних студентів, які зацікавлені у навчанні в Україні.

Недоліки:

- Застарілий дизайн.
- Маленький шрифт тексту на сайті.

Testportal.gov.ua — онлайн-майданчик для тестування та оцінки знань громадян України у різних галузях. Це державний сайт під управлінням Міністерства освіти і науки України, призначений для оцінки рівня знань студентів, претендентів та спеціалістів у різних галузях.

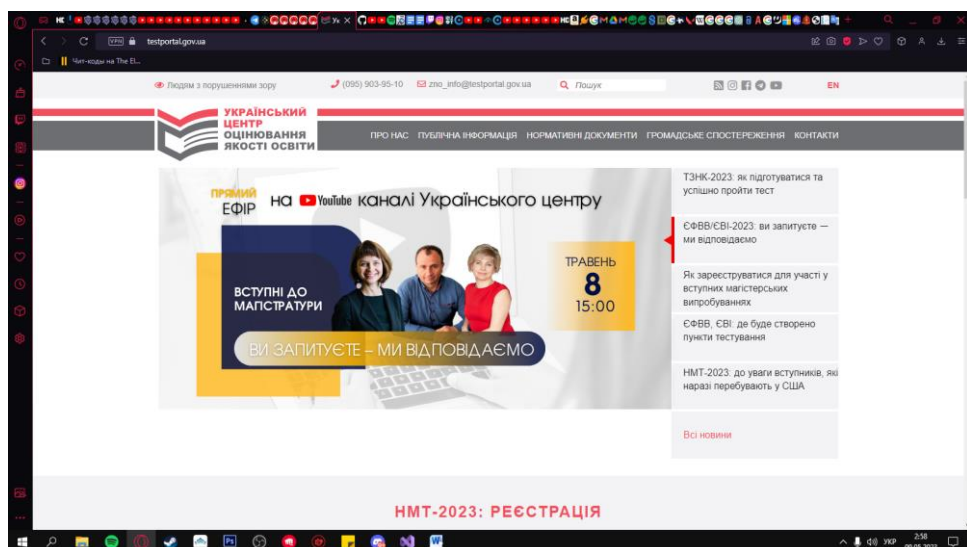


Рис.1.2. Веб-застосунок Testportal.gov.ua

Переваги:

- Єдина система тестування: Testportal.gov.ua є частиною Єдиної системи тестування в Україні. Ця система використовується для перевірки знань учнів у школах, коледжах та університетах по всій країні.
- Різні види тестів: Testportal.gov.ua пропонує різні тести, у тому числі тести для школярів, студентів вузів та шукачів. Тести охоплюють широке коло тем, включаючи математику, історію, літературу, іноземні мови та професійні навички.
- Результати тесту: Після завершення тесту Testportal.gov.ua надає миттєвий зворотний зв'язок результатів тесту. Це допомагає випробуваним виявити свої сильні та слабкі сторони та покращити свої знання та навички.
- Це полегшує людям можливість перевірити свої знання та покращити свої навички без будь-яких фінансових бар'єрів.
- Заходи боротьби з шахрайством: Testportal.gov.ua реалізує суворі заходи боротьби з шахрайством для забезпечення чесності тестів. Веб-сайт використовує передові технології для запобігання шахрайству та виявлення будь-якої підозрілої активності під час тестів.
- Зручний інтерфейс: Testportal.gov.ua має зручний інтерфейс, який спрощує навігацію та використання веб-сайту. Веб-сайт добре організований та візуально привабливий, що дозволяє випробуваним легше зосередитись на тестах.
- Мобільний додаток: Testportal.gov.ua також має мобільний додаток, який можна безкоштовно завантажити в App Store або Google Play. Це полегшує тестованим проходити тести зі своїх мобільних пристроїв у будь-який час та в будь-якому місці.

Недоліки:

- Застарілий дизайн.
- Багато непотрібної інформації.

Просте ЗНО

«Просте ЗНО» — популярний мобільний додаток, який надає українським студентам низку навчальних матеріалів для підготовки до іспитів зовнішнього незалежного тестування (ЗНО). Додаток розроблено, щоб допомогти школярам підготуватися до стандартизованих тестів з різних предметів, включаючи математику, українську мову та літературу, іноземні мови, історію, географію та біологію.

Додаток завоював величезну популярність серед студентів та викладачів завдяки широким навчальним ресурсам, зручному інтерфейсу та персоналізованим навчальним планам. З «Просте ЗНО» студенти можуть легко отримати доступ до великої бази навчальних матеріалів, включаючи відеоуроки, практичні іспити та навчальні посібники. Додаток пропонує різні режими та функції навчання, включаючи тестовий режим, режим практики та планувальник навчання.

Однією з найбільш примітних особливостей програми є функція персоналізованого навчального плану, яка дозволяє учням створювати навчальний план на основі своїх сильних та слабких сторін. Додаток також відстежує прогрес учнів та надає їм інформацію про їх моделі навчання, дозволяючи їм вносити корективи до свого навчального плану.

Ще однією важливою особливістю програми є можливість ставити запитання та отримувати відповіді від досвідчених наставників. Студенти можуть ставити запитання щодо певного предмета або теми та отримувати відповідь протягом кількох годин, що дозволяє їм прояснити будь-які сумніви, які вони мають перед іспитами.

Програма «Просте ЗНО» доступна безкоштовно на платформах iOS та Android, що робить його легко доступним для всіх студентів, які хочуть підготуватися до іспитів ЗНО. Додаток отримав позитивні відгуки як від студентів, так і від викладачів, які цінують його зручний інтерфейс, великі навчальні ресурси та персоналізовані навчальні плани. Насамкінець, «Просте ЗНО» є відмінним ресурсом для студентів, які готуються до іспитів ЗНО, і

зарекомендував себе як цінний інструмент, який допомагає студентам досягти своїх академічних цілей.

Переваги:

- **Комплексні учбові ресурси.** Додаток пропонує велику базу навчальних матеріалів, включаючи відеоуроки, практичні іспити та навчальні посібники, що робить його універсальним рішенням для студентів, які готуються до іспитів.
- **Індивідуальний план навчання:** функція персоналізованого плану навчання у додатку допомагає учням скласти план навчання на основі своїх сильних та слабких сторін. Ця функція також відслідковує прогрес учнів і дає уявлення про їхні моделі навчання, дозволяючи їм вносити корективи до свого навчального плану.
- **Зручний інтерфейс.** Інтерфейс програми інтуїтивно зрозумілий і простий у навігації, що дозволяє учням швидко знаходити потрібні навчальні матеріали.
- **Функція «Питання та відповіді».** Додаток дозволяє учням ставити запитання щодо певного предмета чи теми та отримувати відповіді протягом кількох годин, надаючи їм індивідуальний підхід до навчання.
- **Безкоштовний і доступний:** програма доступна безкоштовно на платформах iOS та Android, що робить її легко доступною для всіх студентів, які хочуть підготуватися до іспитів ZNO.

Недоліки:

- **Обмежене охоплення предмета:** хоча програма охоплює декілька предметів, які перевіряються на іспитах ZNO, вона може не забезпечувати всебічного охоплення всіх тем по кожному предмету. Учням, можливо, потрібно буде використовувати додаткові навчальні матеріали, щоб доповнити свою підготовку.

- Залежність від технологій: програма потребує надійного підключення до Інтернету, а учні повинні використовувати мобільний пристрій для доступу до навчальних матеріалів. Будь-які технічні проблеми, такі як повільне підключення до Інтернету або несправність пристрою, можуть порушити навчальний план учня.
- Обмежена інтерактивність: хоча програма пропонує функцію запитань і відповідей, вона може не надавати учням можливості взаємодіяти зі своїми однолітками чи вчителями в режимі реального часу, що може бути корисним при проясненні складних тем.
- Відсутність персоналізації: хоча програма пропонує функцію індивідуального плану навчання, вона може не надавати учням рівень налаштування, необхідний їм задоволення їх конкретних потреб у навчанні.
- Обмежений зворотний зв'язок: хоча програма відстежує прогрес учнів, вона може не надавати докладні відгуки про їх успішність або рекомендації щодо покращення.
- Мовні обмеження: в даний час додаток доступний лише українською мовою, що може обмежити його доступність для тих, для кого вона не є рідною, або для тих, хто не розмовляє українською.

1.3 C#

C# (*C Sharp*) - це об'єктно-орієнтована мова програмування, розроблена компанією Microsoft для розробки програмного забезпечення, що працює на платформі .NET. Вона була випущена в 2000 році і стала однією з ключових мов для розробки Windows-додатків, веб-додатків, мобільних додатків та інших типів програмного забезпечення.

C# є однією з найпопулярніших мов програмування, оскільки вона поєднує в собі простоту вивчення, високу продуктивність та безпеку. Її синтаксис схожий на синтаксис C++ та Java, що полегшує її вивчення програмістам, які вже мають досвід роботи з цими мовами.

C# має вбудовану підтримку багатьох функцій, таких як маніпулювання рядками, робота з файлами та мережеві операції, що дозволяє програмістам швидко розробляти додатки з високою функціональністю. Крім того, C# має вбудовану підтримку об'єктно-орієнтованого програмування, що дозволяє розробникам створювати додатки з легко змінюваним та розширюваним кодом.

Для розробки back-end частини додатків з використанням C# рекомендується використовувати фреймворк .NET, який надає широкий спектр функцій для створення веб-додатків та сервісів. .NET дозволяє програмістам розробляти програмне забезпечення для різних платформ, включаючи Windows, Linux та MacOS. Крім того, .NET має вбудовану підтримку безпеки, що є дуже важливою для back-end частини додатків, яка містить чутливу інформацію.

1.4 Visual Studio

Visual Studio є інтегроване середовище розробки (IDE) від Microsoft для розробки програмного забезпечення під різні платформи, включаючи Windows, macOS, Android та інші. Воно надає розробникам можливість створювати, тестувати та налагоджувати різні типи додатків, включаючи веб-додатки, мобільні додатки, десктопні додатки, ігри та інші.

Visual Studio має широкий набір інструментів, що дозволяє розробникам створювати якісні програми за короткий час. Це включає в себе засоби для автоматизації тестування, контролю версій, відлагодження, профілювання та інші. Крім того, Visual Studio підтримує різні мови програмування, включаючи C#, Visual Basic, C++, Python, JavaScript, інші.

Visual Studio дозволяє розробникам підключатись до різних сервісів та розширень, які полегшують розробку додатків. Наприклад, Azure DevOps дозволяє керувати процесом розробки та спільно працювати над проектом, а Visual Studio Marketplace містить багато різних розширень для Visual Studio.

У контексті розробки backend-частини додатку за допомогою C# використання Visual Studio дозволяє розробникам швидко створювати та тестувати код, використовуючи різні інструменти для налагодження та профілювання. Крім того, Visual Studio має вбудовану підтримку для популярних фреймворків, таких як .NET та ASP.NET, що полегшує розробку та розгортання додатків

1.5 MySql

MySQL - це відкрита реляційна база даних, яка використовує мову запитів SQL для зберігання і управління даними. Вона є однією з найпопулярніших баз даних у світі та часто використовується у розробці веб-додатків як backend-частина.

MySQL надає різноманітні можливості, такі як швидкість обробки великих об'ємів даних, висока надійність, простота у використанні та можливість розширення. Вона підтримує велику кількість типів даних, включаючи числа, рядки, дати та більш складні типи, такі як JSON.

MySQL є добре затвердженою системою управління базами даних, яка підтримується багатьма різними програмними засобами, в тому числі Visual Studio. З її допомогою можна створювати, зберігати та оновлювати дані в базі даних, а також виконувати різноманітні запити для отримання потрібної інформації.

1.6..Net

.NET — це безкоштовне програмне середовище з відкритим вихідним кодом, розроблене Microsoft, яке використовується для створення широкого спектру програм для Windows, веб-платформ та мобільних платформ. Вперше він був представлений у 2002 році і відтоді став одним із найпопулярніших фреймворків для розробки програмного забезпечення у світі.

Платформа .NET надає набір бібліотек, середовища виконання та інструментів, які розробники можуть використовувати для створення та розгортання програм для настільних комп'ютерів, Інтернету та мобільних пристроїв. Він підтримує кілька мов програмування, включаючи C#, F# і Visual Basic, і включає безліч функцій, що спрощують розробникам створення надійних, масштабованих і безпечних додатків.

Однією з ключових переваг .NET є те, що він надає середовище CLR, яке керує виконанням коду та забезпечує безпеку та надійність додатків. CLR відповідає за керування пам'яттю, обробку винятків та надання інших важливих служб, які допомагають зробити програми .NET більш ефективними та безпечними.

.NET також включає потужну бібліотеку класів, яка надає широкий спектр готових функцій та інструментів, які розробники можуть використовувати для більш швидкого та ефективного створення додатків. Ця бібліотека включає широкий спектр класів і функцій, які можна використовувати для роботи з даними, доступу до веб-сервісів, створення інтерфейсів і багато іншого.

Крім повної платформи .NET, Microsoft також розробила .NET Core, кросплатформенну версію .NET, яку можна використовувати для створення програм для Windows, macOS і Linux. .NET Core спроектований так, щоб бути більш модульним та легким, ніж повна платформа .NET, що робить його добре придатним для хмарних додатків та мікросервісів.

1.7. ASP.NET

ASP.NET — це платформа веб-застосунків, розроблена Microsoft, яка дозволяє розробникам створювати динамічні серверні веб-додатки. Вперше він був випущений у 2002 році і з того часу став популярним вибором для створення надійних та масштабованих веб-додатків.

ASP.NET побудований на основі .NET Framework, яка надає багатий набір бібліотек та інструментів для створення веб-застосунків. Він підтримує широкий спектр мов програмування, включаючи C#, Visual Basic.NET та F#. ASP.NET також підтримує різні моделі веб-розробки, включаючи веб-форми, MVC (модель-представлення-контролер) та веб-API.

Однією з ключових особливостей ASP.NET є можливість доставки динамічного веб-контенту за допомогою сценаріїв на стороні сервера. Це дозволяє розробникам створювати веб-сторінки, які можна налаштувати на основі введення користувача або даних, вилучених з бази даних. ASP.NET також включає безліч вбудованих елементів керування та компонентів, що спрощують створення інтерактивних веб-сторінок з багатьма функціональними можливостями.

Ще однією важливою особливістю ASP.NET є підтримка веб-служб та інтеграція з іншими технологіями. Наприклад, ASP.NET можна використовувати для створення RESTful API, які можуть використовуватись іншими програмами. Він також має вбудовану підтримку автентифікації та авторизації, що спрощує захист веб-застосунків.

В останні роки Microsoft також представила ASP.NET Core, кросплатформну версію платформи з відкритим вихідним кодом, призначену для роботи у Windows, Linux та macOS. ASP.NET Core включає низку нових функцій та покращень, у тому числі підвищену продуктивність, спрощене налаштування та покращену підтримку хмарних розгортань.

В цілому, ASP.NET — це потужне та гнучке середовище веб-розробки, яке надає розробникам широкий спектр інструментів та можливостей для створення сучасних веб-додатків. Незалежно від того, чи створюєте ви веб-

сайт для малого бізнесу або великомасштабний корпоративний додаток, ASP.NET це надійний вибір, який допоможе вам виконати роботу.

1.8 Entity Framework Core

Entity Framework Core - це платформа об'єктно-реляційного зіставлення (ORM) з відкритим вихідним кодом для додатків .NET, яка надає потужний і гнучкий спосіб доступу до даних, що зберігаються в реляційних базах даних, та управління ними. Це сучасна та полегшена версія Entity Framework, вперше представлена Microsoft у 2008 році.

Entity Framework Core (EF Core) дозволяє розробникам працювати з базами даних, використовуючи набір строго типізованих класів C#, які представляють таблиці бази даних та їх зв'язку. Це спрощує написання коду бази даних, який можна обслуговувати, тестувати і який менш схильний до помилок. За допомогою EF Core розробники можуть писати код, який залежить від базової схеми бази даних та мови запитів, що спрощує перемикання між різними платформами чи версіями баз даних.

Однією з ключових особливостей EF Core є можливість створювати схеми бази даних із класів C# за допомогою процесу, відомого як розробка коду. Такий підхід дозволяє розробникам зосередитися на моделі предметної сфери свого застосування, а не на деталях схеми бази даних. EF Core використовує набір угод та анотацій даних, щоб вивести схему бази даних із класів C#, або розробники можуть явно налаштувати схему за допомогою методів Fluent API.

EF Core також підтримує розробку, орієнтовану на базу даних, де він може перетворити схему бази даних класи C#, які можна використовувати для доступу до даних. Цей підхід корисний під час роботи з існуючими базами даних або застарілими системами.

Ще одна важлива особливість EF Core – підтримка міграції. Міграції дозволяють розробникам змінювати схему бази даних з часом, створюючи набір сценаріїв на основі коду, які можна застосовувати до бази даних. EF

Core відстежує поточний стан схеми бази даних і застосовує необхідні міграції, щоб привести схему у відповідність до поточного коду.

EF Core підтримує широкий спектр баз даних, включаючи Microsoft SQL Server, MySQL, PostgreSQL та SQLite. Він також підтримує бази даних у пам'яті та може використовуватися з хмарними службами баз даних, такими як база даних SQL Azure.

Загалом EF Core – це потужна та гнучка платформа ORM, яка спрощує роботу з реляційними базами даних у додатках .NET. Він надає сучасний та ефективний спосіб доступу до даних та управління ними, який допомагає розробникам писати найкращий код бази даних та підвищувати загальну якість своїх додатків.

1.9 SQL Server Management Studio

SQL Server Management Studio (SSMS) - це інструмент з графічним інтерфейсом користувача (GUI), розроблений Microsoft, який використовується для управління і адміністрування баз даних Microsoft SQL Server. Це потужний та всеосяжний інструмент, який дозволяє розробникам та адміністраторам баз даних виконувати широкий спектр завдань, пов'язаних з управлінням базами даних, включаючи проектування баз даних, виконання запитів та обслуговування.

SSMS надає багатий та інтуїтивно зрозумілий інтерфейс для роботи з базами даних SQL Server. Він включає безліч інструментів і функцій, що спрощують створення і зміна об'єктів бази даних, таких як таблиці, уявлення, процедури, що зберігаються, і тригери. Він також включає потужний редактор запитів, який забезпечує підсвічування синтаксису, автодоповнення коду та функції налагодження, що допомагають розробникам більш ефективно писати складні SQL-запити.

Однією з ключових особливостей SSMS є можливість управління та моніторингу продуктивності баз даних SQL Server. Він включає низку інструментів моніторингу продуктивності, які дозволяють адміністраторам

баз даних відстежувати продуктивність екземплярів SQL Server та виявляти потенційні вузькі місця продуктивності. Він також включає різні інструменти обслуговування та оптимізації, які можна використовувати для оптимізації продуктивності баз даних SQL Server.

SSMS також включає ряд функцій безпеки і аудиту, які дозволяють адміністраторам баз даних керувати безпекою бази даних і перевіряти активність бази даних. Він включає потужний майстер безпеки, який можна використовувати для створення та управління обліковими записами та ролями бази даних, а також надає безліч функцій аудиту, які дозволяють адміністраторам відстежувати зміни в об'єктах бази даних та діях користувачів.

На додаток до своїх основних функцій SSMS також включає ряд надбудов та розширень, які можна використовувати для розширення його функціональності. Наприклад, він включає підтримку конструктора пакетів SQL Server Integration Services (SSIS), який дозволяє розробникам створювати та керувати пакетами SSIS для інтеграції та перетворення даних.

Загалом SQL Server Management Studio – це потужний комплексний інструмент, що надає багатий набір функцій для управління та адміністрування баз даних Microsoft SQL Server. Незалежно від того, чи ви є розробником або адміністратором бази даних, SSMS є важливим інструментом для роботи з базами даних SQL Server.

2. ВИМОГИ ТА ОЦІНКА ЯКОСТІ СИСТЕМИ

2.1 Обсяг та опис

Опис: Розробка додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання передбачає створення повноцінної системи, яка дозволяє керувати курсами, студентами, навчальними матеріалами та тестами. Для реалізації цієї задачі необхідна розробка не тільки frontend-частини, яка забезпечує інтерфейс для користувача, але й backend-частини, яка забезпечує роботу з даними та бізнес-логіку додатку.

У даній роботі розглянута розробка backend-частини засобами мови програмування C#. Ця мова програмування дозволяє розробляти високопродуктивні та надійні застосунки для платформи .NET Framework. Для розробки використовується середовище програмування Visual Studio, яке забезпечує зручну роботу з кодом, дозволяє використовувати різноманітні інструменти для тестування та налагодження додатку.

Для зберігання даних використовується система управління базами даних MySQL. MySQL є потужною та добре відомою системою, яка дозволяє зберігати великі обсяги даних та швидко їх опрацьовувати. Більшість провайдерів хостингу підтримують MySQL, що забезпечує зручну розгортання та налагодження додатку на серверах.

Обсяг: Описано процес розробки backend-частини додатку засобами мови програмування C#, середовища Visual Studio та системи управління базами даних MySQL

2.2 Загальний опис

Тема "Розробка додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання" охоплює проектування та розробку програмного забезпечення, що дозволяє підготувати користувачів до проходження зовнішнього незалежного оцінювання в Україні. У роботі буде використовуватися мова програмування C# для розробки back-end частини

додатку, а також MySQL для збереження даних. Основна мета розробки полягає в підготовці користувачів до здачі іспиту, забезпеченні доступу до необхідних матеріалів, тренуванні тестів та наданні результатів.

2.3 Опис архітектури додатку

Архітектура додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання включає три рівні: front-end, back-end та базу даних.

Front-end рівень відповідає за інтерфейс користувача та комунікацію з back-end рівнем. Для розробки front-end частини буде використано мову програмування JavaScript та фреймворки React або Angular.

Back-end рівень відповідає за обробку запитів користувача та доступ до бази даних. Для розробки back-end частини буде використано мову програмування C# та фреймворк ASP.NET Core.

База даних використовується для зберігання інформації про користувачів та результати навчання. Для управління базою даних буде використано систему керування базами даних MySQL.

Така архітектура дозволяє розділити рівні відповідальності та забезпечити гнучкість та масштабованість додатку.

Обмеження на розробку та реалізацію

Обмеження на розробку та реалізацію додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання можуть включати:

Обмеження з боку функціональності, дизайну та інші вимоги, які необхідно врахувати при розробці додатку.

Обмеження з боку технічних можливостей: розробка додатку повинна враховувати технічні можливості платформи, на якій буде запускатися додаток, а також розмір дискового простору та обсяг доступної оперативної пам'яті.

Обмеження з боку безпеки: необхідно забезпечити безпеку даних, зокрема шифрування персональної інформації користувачів, захист від хакерських атак та інших загроз.

Обмеження з боку доступності: додаток повинен бути доступний для користувачів з різних пристроїв та платформ, зокрема з мобільних пристроїв та планшетів.

Обмеження з боку часу: необхідно визначити часові рамки на розробку та випуск додатку, а також на його тестування та внесення необхідних змін.

2.4. Моделювання UML-діаграми

Уніфікована мова моделювання (UML) – це мова візуального моделювання, яка використовується у розробці програмного забезпечення для представлення програмних систем та їх компонентів, взаємозв'язків та взаємодій. Схеми UML надають стандартизовану нотацію та синтаксис для документування та передачі інформації про проекти програмного забезпечення, спрощуючи розробникам, зацікавленим сторонам та кінцевим користувачам розуміння програмних систем та обмін інформацією про них.

Існує кілька типів діаграм UML, які можна використовувати для представлення різних аспектів програмної системи, зокрема:

1. Діаграми класів. Діаграми класів використовуються для представлення класів, інтерфейсів та відносин між ними у програмній системі. Вони показують атрибути та методи кожного класу, а також відносини між класами, такі як спадкування, композиція та асоціація.

2. Діаграми варіантів використання: Діаграми варіантів використання використовуються для представлення взаємодій між суб'єктами та програмною системою. Вони показують різні варіанти використання або сценарії, в яких може використовуватися система, а також дійових осіб або користувачів, які взаємодіють із системою.

3. Діаграми послідовності. Діаграми послідовності використовуються для представлення взаємодій між об'єктами програмної системи в часі. Вони показують послідовність подій та повідомлень, якими обмінюються об'єкти у конкретному сценарії.

4. Діаграми дій. Діаграми дій використовуються для представлення потоку дій чи процесів у програмній системі. Вони показують кроки, пов'язані з конкретним процесом чи діяльністю, і як вони взаємопов'язані.

5. Діаграми кінцевого автомата: Діаграми кінцевого автомата використовуються для представлення станів та переходів об'єкта у програмній системі. Вони показують різні стани об'єкта та події або тригери, які викликають його перехід між станами.

6. Діаграма компонентів. Діаграми компонентів використовуються для представлення компонентів та залежностей програмної системи. Вони показують різні компоненти у системі і те, як вони пов'язані та залежать один від одного.

7. Діаграми розгортання. Діаграми розгортання використовуються уявлення фізичного розгортання програмної системи. Вони показують різні апаратні та програмні компоненти, що беруть участь у розгортанні системи, та способи їх підключення.

Діаграми UML широко використовуються у розробці програмного забезпечення для документування, аналізу та передачі інформації про проекти програмного забезпечення. Вони забезпечують спільну мову та позначення для розробників програмного забезпечення, зацікавлених сторін та кінцевих користувачів, спрощуючи розуміння програмних систем та спілкування з ними.

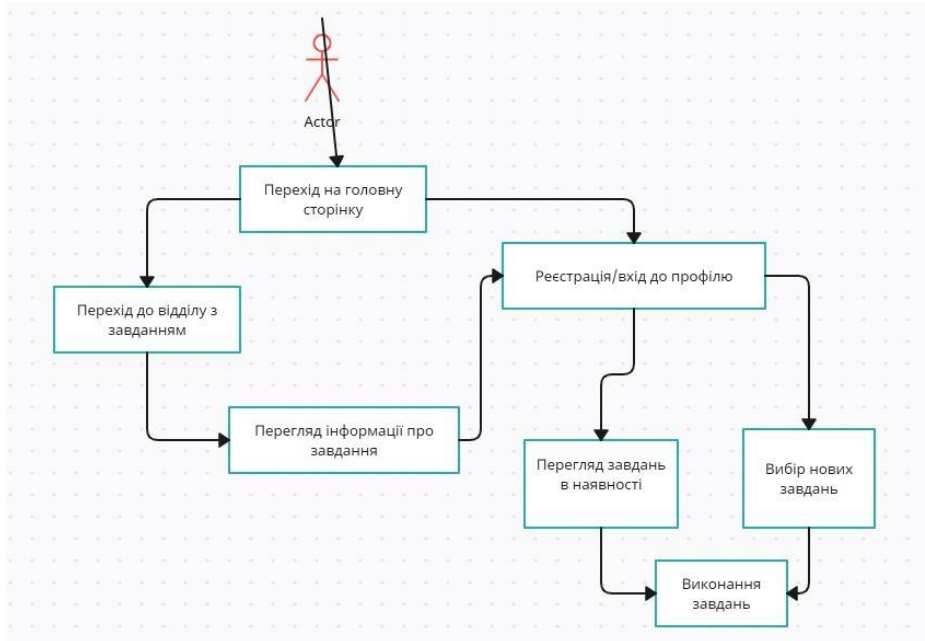


Рис 2.1 – UML-діаграма для застосунку організації роботи курсів з підготовки до зовнішнього незалежного оцінювання.

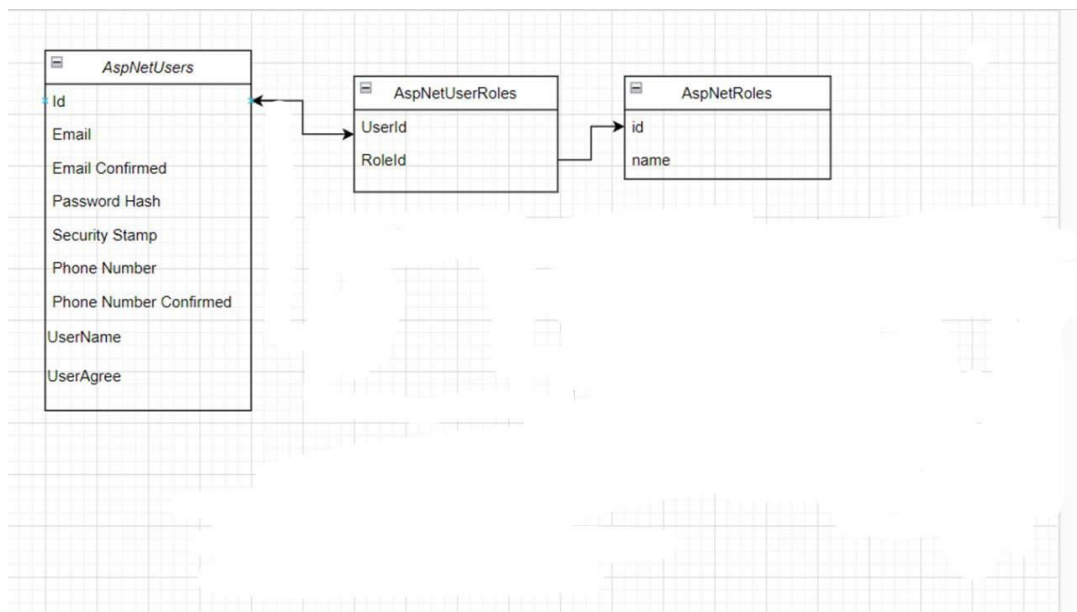


Рис 2.2 – Структура бази даних.

3. РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Розробка серверної частини

Після відкриття веб-порталу в браузері, користувач автоматично перенаправляється на головну сторінку порталу, за це відповідає метод який знаходиться в контролері головної сторінки, рис. 3.1.

```
0 references  
public IActionResult Index()  
{  
    return View();  
}
```

Рис. 3.1 – Метод в контролері, який повертає головну сторінку порталу

Після відкриття головної сторінки за стосунку, рис 3.2, ми можемо побачити кнопки «Реєстрація» та «Вхід», які зображена на рис. 3.3. Також можна побачити панель швидкого доступу з переходом до інформації на сторінці, рис 3.4.

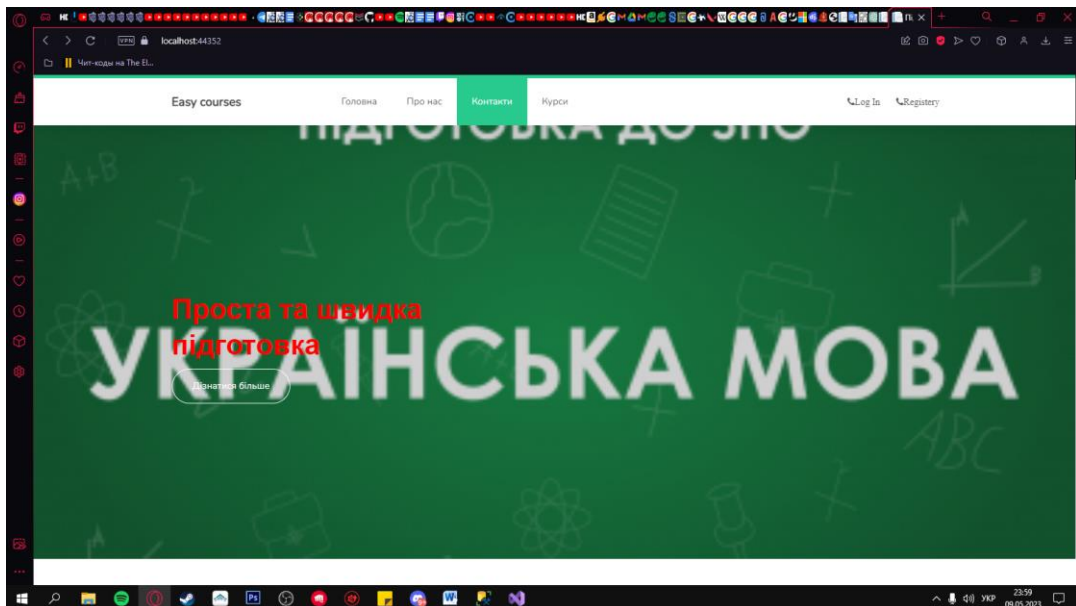


Рис 3.2 – Головна сторінка за стосунку

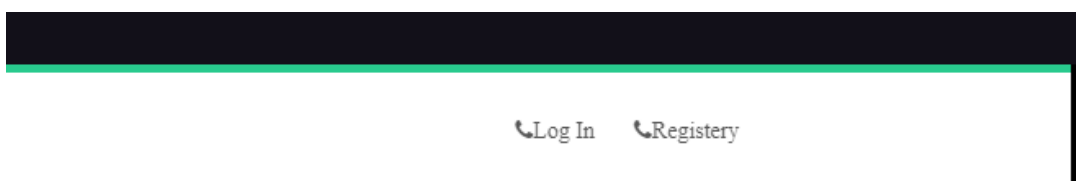


Рис 3.3 – Посилання на вхід та реєстрацію.

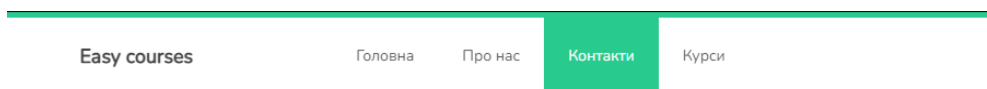


Рис 3.4 – Панель швидкого переходу.

Коли користувач натискає кнопку "Зареєструватись", він переходить на вкладку, де може створити свій аккаунт. На вкладці "Зареєструватися" можна побачити поля для введення електронної пошти, пароля та підтвердження пароля, а також кнопку "Зареєструватися", які зображені на Рис. 3.5.

A registration form with three input fields and a button. The first field is labeled "Адрес электронной почты" (Email address). The second field is labeled "Пароль" (Password). The third field is labeled "Подтверждение пароля" (Confirm password). Below the fields is a blue button with the text "Зареєструватися" (Register).

Рис 3.5 – форма реєстрації.

Після того, як користувач введе свої дані та натисне кнопку "Зареєструватися", його інформація буде додана до таблиці з обліковими записами користувачів, метод для обробки інформацій користувача відображено на Рис.3.6. А сам користувач буде перенаправлений на головну сторінку порталу. На цій сторінці буде відображене ім'я користувача та кнопка "Вийти", які зображені на Рис. 3.7.


```

public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl = "https://tender_msd-spirits.com";
    ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser
        {
            UserName = Input.Email,
            Email = Input.Email,
            UserAgree = false,

            IsActive = true,
            IsActiveComment = " ",

            EmailConfirmed = true,
        };
        var result = await _userManager.CreateAsync(user, Input.Password);
        if (result.Succeeded)
        {
            var newuser = db.AspNetUsers.First(x => x.UserName == Input.Email);

            db.SaveChanges();
            var newRoleForUser = new DB.AspNetUserRoles
            {
                UserId = newuser.id,
                RoleId = "2"
            };
            var res = DBService.AddRoleToUser(newRoleForUser, db);
            await _signInManager.SignInAsync(user, isPersistent: false);
            return RedirectToAction("Index", "Home");
            //}
        }
        foreach (var error in result.Errors)
        {
            ModelState.AddModelError(string.Empty, error.Description);
        }
    }
}

```

Рис.3.6. Метод для обробки інформації користувача після реєстрації

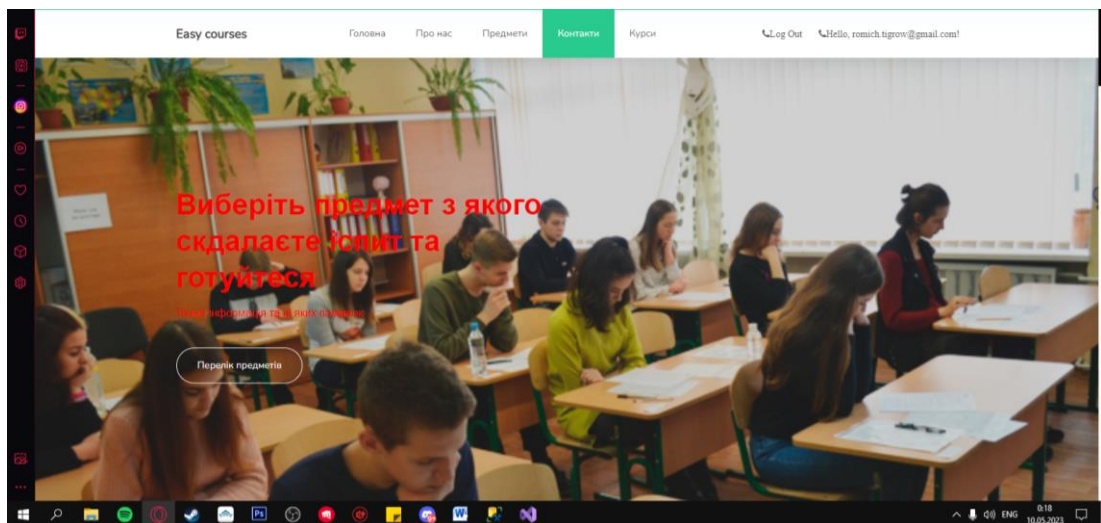


Рис 3.7. – головна сторінка користувача після реєстрації.

Коли користувач натискає кнопку "Увійти", він буде перенаправлений на сторінку, де можна увійти в свій обліковий запис. На цій сторінці користувач може побачити поля для введення електронної пошти, поля для введення пароля від облікового запису та кнопку "Увійти", які зображені на Рис. 3.8.

[Easy courses](#) [Головна](#) [Про нас](#)

Адрес електронної пошти

Пароль

Запам'янтати мене

[Зареєструвати нового користувача](#)

Рис 3.8 – Сторінка входу в особистий профіль.

Коли користувач успішно увійде до свого облікового запису, метод для обробки введених даних користувача для входу відображено на Рис.3.9, його буде перенаправлено на головну сторінку порталу, де буде відображатись ім'я користувача та кнопка "Вийти", які зображені на Рис.3.10.

```
0 references
public async Task<IActionResult> OnPostAsync(string returnUrl = null)
{
    returnUrl = returnUrl ?? Url.Content("~/");
    if (ModelState.IsValid)
    {
        // This doesn't count login failures towards account lockout
        // To enable password failures to trigger account lockout, set lockoutOnFailure: true
        var result = await _signInManager.PasswordSignInAsync(Input.Email, Input.Password, Input.RememberMe, lockoutOnFailure: false);
        if (result.Succeeded)
        {
            _logger.LogInformation("User logged in.");
            return LocalRedirect(returnUrl);
        }
        if (result.IsLockedOut)
        {
            _logger.LogWarning("User account locked out.");
            return RedirectToPage("./Lockout");
        }
        else
        {
            ModelState.AddModelError(string.Empty, "Invalid login attempt.");
            return Page();
        }
    }

    // If we got this far, something failed, redisplay form
    return Page();
}
```

Рис 3.9. Метод для входу користувача в особистий профіль

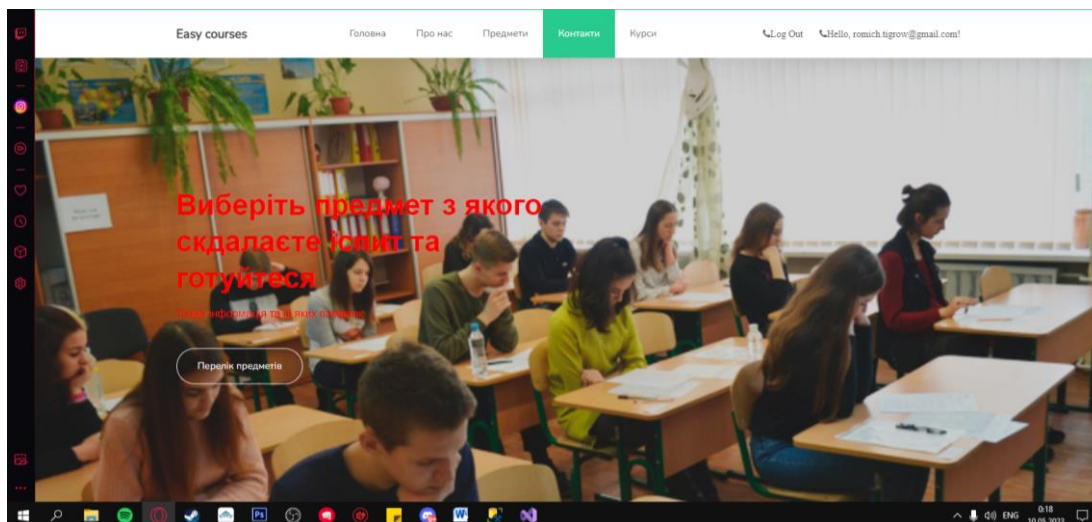


Рис 3.10. – Головна сторінка застосунку після входу

Після входу в особистий профіль на панелі швидкого доступу з'являється пункт “Предмети”, Рис 3.11. Натиснувши на цей пункт користувач переходить до переліку предметів, Рис 3.12.

Предмети

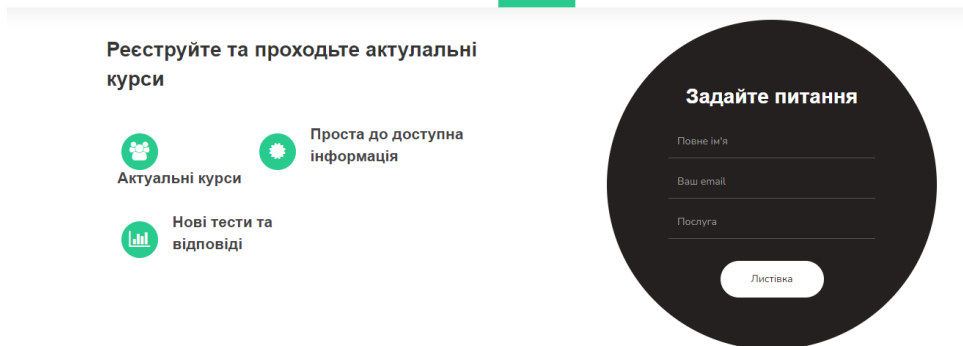
Рис 3.11 – Пункт “Предмети”.



Рис 3.12 – Список предметів.

Після натискання на один з предметів користувача перенесе на сторінку де він може переглянути завдання та відповіді.

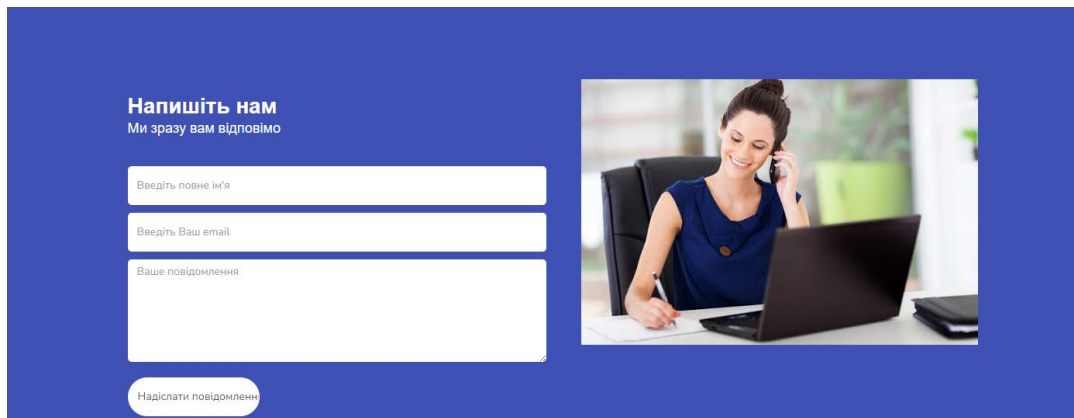
Якщо у користувача виникають питання він може звернутися напряду через відповідну форму Рис 3.13.



The image shows a user interface with a light gray background. On the left, there is a section titled "Реєструйте та проходите актуальні курси" (Register and take current courses). Below this title are three green circular icons with white text: "Актуальні курси" (Current courses), "Проста до доступна інформація" (Simple and accessible information), and "Нові тести та відповіді" (New tests and answers). On the right, there is a large black circular button with the text "Задайте питання" (Ask a question). Below this button are three input fields: "Повне ім'я" (Full name), "Ваш email" (Your email), and "Послуга" (Service). At the bottom of the black circle is a white button with the text "Листівка" (Postcard).

Рис 3.13 – Форма зв'язку з відповідних питань.

Якщо користувач хоче зв'язатися з адміністратором то він може написати свою електронну пошту, ім'я та повідомлення на певній формі Рис 3.14.



The image shows a contact form on a blue background. The form is titled "Напишіть нам" (Write to us) with the subtitle "Ми зразу вам відповімо" (We will answer you immediately). There are three input fields: "Введіть повне ім'я" (Enter full name), "Введіть Ваш email" (Enter your email), and "Ваше повідомлення" (Your message). Below the input fields is a white button with the text "Надіслати повідомлення" (Send message). To the right of the form is a photograph of a woman in a blue dress sitting at a desk, talking on a mobile phone while looking at a laptop.

Рис 3.14 – Форма зв'язку з адміністратором.

Якщо користувач захоче зв'язатися з розробником застосунку він може натиснути на посилання в низу сторінки Рис 3.15.

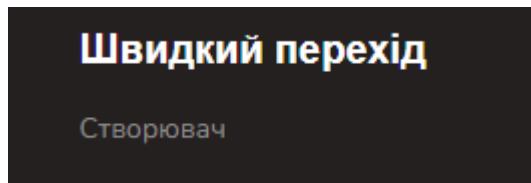


Рис 3.15 – Швидкий перехід для зв'язку з створювачем.

3.2 Тестування веб застосунку

Тестування веб-додатків - це процес перевірки, чи відповідає програма, розроблена для портативних пристроїв, певним вимогам, які встановлені для веб-браузерного тестування. Веб-додаток є програмою, яка працює у веб-браузері. У всіх процесах розробки програмного забезпечення тестування веб-додатків є обов'язковим перед випуском програми на ринок. Цей крок забезпечує належне функціонування кінцевого продукту протягом усього його життєвого циклу.

Види тестування веб застосунків

Unit тестування

Модульне тестування - це тип тестування програмного забезпечення, спрямований на перевірку функціональності окремих одиниць коду у відриві від решти системи. Зазвичай це виконується розробниками в процесі розробки, перш ніж код буде інтегрований у більшу систему.

Метою модульного тестування є виявлення та виправлення помилок та багів у коді якомога раніше в процесі розробки. Тестуючи окремі блоки коду ізольовано, розробники можуть швидко і легко виявляти проблеми до того, як їх буде складніше виправити зі зростанням кодової бази.

Модульні тести зазвичай є автоматизовані тести, призначені для багаторазового запуску в міру зміни коду. Зазвичай вони пишуться з використанням середовища тестування, такого як JUnit для Java або NUnit для .NET, яке надає набір інструментів для створення та запуску тестів.

Процес створення модульного тесту складається з кількох кроків. Спочатку розробник пише тестовий код, який визначає вхідні значення для модуля, що тестується, і очікуваний результат. Потім розробник запускає тест, який зазвичай включає компіляцію коду і виконання тестового прикладу.

Якщо тест не пройдено, розробник може змінити код і повторно запустити тест, доки він не пройде.

Модульне тестування може бути складним, особливо під час роботи зі складними системами чи залежностями. У деяких випадках розробникам може знадобитися використовувати фіктивні об'єкти або інші методи для імітації поведінки зовнішніх залежностей або компонентів, які ще не доступні.

Незважаючи на ці проблеми, модульне тестування є невід'ємною частиною сучасної розробки програмного забезпечення. Це допомагає розробникам виявляти помилки на ранній стадії, знижує ризик внесення нових помилок при зміні існуючого коду та забезпечує захист від змін коду. Інвестуючи в модульне тестування, розробники можуть створювати більш надійні, зручні у супроводі та високоякісні програмні системи.

Інтеграційне тестування

Інтеграційне тестування – це тип тестування програмного забезпечення, спрямований на перевірку взаємодії між різними модулями чи компонентами програмної системи. **Мета інтеграційного тестування – переконатися, що окремі модулі працюють разом, як очікується, та що система загалом відповідає своїм функціональним вимогам.**

Інтеграційне тестування зазвичай виконується після модульного тестування та перед тестуванням системи. Він включає тестування точок інтеграції між різними модулями, а також потоку даних і зв'язку між ними. Інтеграційне тестування може виконуватися з використанням різних методів, включаючи підходи «згори донизу», «знизу догори» та гібридні підходи.

При низхідному підході тестування починається з модуля найвищого рівня в системі і продовжується до найнижчого рівня. Цей підхід дозволяє розробникам спочатку протестувати найбільш важливі та складні частини системи, але може зажадати використання заглушок або фіктивних об'єктів для імітації поведінки модулів нижчого рівня.

При висхідному підході тестування починається з модуля найнижчого рівня і продовжується до найвищого рівня. Цей підхід дозволяє розробникам тестувати окремі модулі ізольовано, перш ніж інтегрувати їх у більшу систему, але може вимагати використання драйверів або тестових наборів для імітації поведінки модулів вищого рівня.

У гібридному підході тестування починається з комбінації підходів «згори донизу» та «знизу догори», при цьому тестування просувається в обох напрямках, поки всі модулі не будуть інтегровані та протестовані разом.

Інтеграційне тестування можна додатково поділити на різні типи залежно від обсягу та характеру тестування. До них відносяться:

1. Інтеграційне тестування «великого вибуху»: включає одночасну інтеграцію всіх модулів системи та їхнє тестування в цілому.
2. Інкрементне інтеграційне тестування: воно включає інтеграцію модулів по одному і їх ізольоване тестування перед їх інтеграцією в більшу систему.
3. Інтеграційне тестування зверху вниз: воно включає спочатку тестування модулів найвищого рівня, а потім поступову інтеграцію модулів нижчого рівня, поки не буде протестована вся система.
4. Висхідне інтеграційне тестування: воно включає спочатку тестування модулів найнижчого рівня, а потім поступову інтеграцію модулів вищого рівня, поки не буде протестована вся система.

Інтеграційне тестування може бути складним завданням, особливо у складних системах з безліччю взаємозалежних компонентів. Щоб забезпечити успіх інтеграційного тестування, важливо мати чітко визначений план інтеграції, чіткі вимоги та повне розуміння архітектури системи. Інтеграційне тестування також має бути доповнено іншими типами тестування, такими як модульне тестування, системне тестування та приймальне тестування, щоб гарантувати, що система відповідає всім вимогам та має високу якість.

Функціональне тестування

Функціональне тестування – це тип тестування програмного забезпечення, спрямований на перевірку функціональності програмної системи. Мета функціонального тестування – переконатися, що програмна система відповідає заданим вимогам і точно та ефективно виконує намічені завдання.

Функціональне тестування може виконуватися на різних етапах життєвого циклу розробки програмного забезпечення, у тому числі на етапі аналізу вимог, на етапі проектування та розробки та на етапі обслуговування.

Тестування безпеки

Тестування безпеки – це процес оцінки безпеки програмної системи, програми або мережної інфраструктури для виявлення вразливостей та потенційних загроз, якими можуть скористатися зловмисники. Він включає тестування програмного забезпечення та систем з точки зору зловмисника для виявлення потенційних слабких місць у системі безпеки та оцінки рівня захисту, що забезпечується програмним забезпеченням та системами.

Мета тестування безпеки — переконатися, що система захищена та може протистояти атакам та спробам використання її вразливостей. Це найважливіший аспект розробки програмного забезпечення та управління мережевою інфраструктурою, оскільки він допомагає виявляти потенційні ризики безпеки та вживати заходів щодо їх усунення до того, як зловмисники зможуть їх використати.

Тестування безпеки включає низку методів та методологій, таких як оцінка вразливостей, тестування на проникнення, перевірка коду, аудит безпеки та аналіз ризиків. Кожен метод має свої сильні та слабкі сторони та підходить для різних типів сценаріїв тестування безпеки.

Оцінка вразливості включає виявлення потенційних уразливостей у програмному забезпеченні або мережній інфраструктурі шляхом сканування системи на наявність відомих уразливостей та помилок конфігурації. Тестування на проникнення включає моделювання атаки на систему для

виявлення вразливостей і слабких місць, які можуть бути використані зловмисниками. Перевірка коду включає перевірку коду програми для виявлення потенційних недоліків безпеки та слабких місць. Аудит безпеки включає перевірку засобів управління безпекою та політик організації, щоб переконатися, що вони ефективні для захисту від загроз безпеки.

Аналіз ризиків включає виявлення потенційних ризиків та оцінку їх ймовірності та впливу на систему. Це допомагає організаціям розставити пріоритети у своїх зусиллях щодо безпеки та виділити ресурси для усунення найбільш серйозних ризиків.

Тестування безпеки – це безперервний процес, який необхідно виконувати регулярно, щоб переконатися, що система залишається безпечною та захищеною від загроз, що виникають. Для організацій важливо інвестувати у тестування безпеки, щоб захистити свої системи та дані від потенційних порушень безпеки та кібератак.

Найкращі методи тестування веб додатків

- **Тест на зручність використання.** Тестування зручності використання важливе для веб-додатків, оскільки воно гарантує, що користувачі можуть легко переміщатися програмою і виконувати необхідні дії.
- **Тест на продуктивність.** Тестування продуктивності має вирішальне значення для веб-застосунків, щоб гарантувати, що програма може впоратися з очікуваним навантаженням без збоїв або уповільнення.
- **Тестування безпеки.** Тестування безпеки необхідно для веб-додатків, щоб виявляти та усувати потенційні уразливості безпеки, які можуть бути використані зловмисниками.
- **Тестування у кількох браузерях та на різних пристроях.** Веб-програми слід тестувати в різних браузерях і різних пристроях, щоб переконатися, що вони сумісні з різними платформами.

- Виконання регресійного тестування. Регресійне тестування є важливим для того, щоб переконатися, що зміни, внесені до програми, не впливають на існуючі функції та функції.
- Тестування за допомогою користувачів: кінцеві користувачі можуть надати цінні відгуки про зручність використання та функціональність веб-програми.

3.3 Тестування веб застосунку

Таблиця 3.1

ID	Назва	Кроки	Очікуваний результат	Реальний результат
1	Перевірка роботи веб застосунку	1. Відкриття порталу	Додаток відкривається.	Додаток відкривається.
2	Перевірка логотипу на всіх сторінках	1. Відкриття порталу 2. Натискання на логотип в лівому верхньому куті	Перехід на головну сторінку порталу.	Перенаправляє на головну сторінку порталу
3	Перевірка кнопки «Предмети» на головній сторінці	1. Відкриття порталу. 2. Натискання на кнопку «Предмети» на головній сторінці	Перехід на розділ зі списком предметів	Застосунок перенаправляє користувача на розділ зі списком предметів
4	Перевірка кнопки «Про нас»	1. Відкриття порталу 2. Натискання на кнопку «Про нас»	Перехід на розділ з інформацією	Застосунок перенаправляє на розділ з інформацією.
5	Перевірка кнопки «Easy courses»	1. Відкриття порталу. 2. Натискання на кнопку «Easy courses»	Відкриття головної сторінки порталу.	Застосунок перенаправляє на головну сторінку порталу.

6	Перевірка кнопки «Реєстрація»	1. Відкриття додатку 2. Натиснути кнопку «Реєстрація»	Відкриття сторінки для реєстрацій	Застосунок перенаправляє на сторінку для реєстрацій
7	Введення даних для реєстрації на вкладці «Реєстрація»	1. Відкриття додатку 2. Відкрити вкладка «Реєстрація» 3. Ввести поштову скриньку, пароль та підтвердити пароль	При введенні некоректної поштової скриньку повинно з'явитися повідомлення з помилкою. При введенні коректних даних користувача зареєструє в застосунку	При введенні некоректної поштової скриньки з'являється повідомлення про не правильні данні. При введенні коректних даних користувача реєструє в застосунку та перенаправляє на головну сторінку
8	Перевірка входу в аккаунт користувача	1. Відкриття додатку 2. Натискання на кнопку «Увійти» 3. Ввести данні	При введенні не правильних даних повинно з'явитися повідомлення	При введенні некоректних даних з'являється повідомлення

		для входу в свій особистий профіль	про некоректні дані. При введенні коректних даних, повинно увійти в профіль користувача	про невірні дані. При введенні коректних даних, користувач входить в свій особистий профіль та перенаправляє на головну сторінку порталу.
9	Відображення імені користувача після входу в особистий профіль	1. Відкриття додатку в браузері 2. Увійти в особистий профіль 3. Зачекати поки перенаправить на головну сторінку порталу	Коли користувач увійде в особистий профіль повинно відобразитися ім'я користувача в правому верхньому куті .	Після того, як користувач увійде в особистий профіль, на всіх сторінка відображається ім'я користувача в правому верхньому куті
10	Перевірка виходу з особистого профіля користувача	1. Відкриття додатку 2. Зачекати поки відкриє головну сторінку 3. Увійти в	Перехід на головну сторінку порталу	Перехід на головну сторінку.

		особистий профіль 4. Вийти з особистого профілю		
11	Перевірки відображення списку предметів	1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку з предметами	Відображення списку предметів які доступні для проходження. Відображення картинка предмету.	Після переходу на сторінку з предметами. Відображається список предметів та картинка предметів
12	Перевірка вмісту предметів	1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку з предметами 4. Натиснути на картинку предмету зі списку	Відображення інформацій стосовно предмету та головної картинки	Відображення інформацій стосовно предмета та головної картинки
13	Перевірка кнопки «Листівка» на	1. Відкриття додатку 2. Зачекати поки	Заповнити поля та натиснути на кнопку Листівка	Заповнити поля та натиснути на кнопку Листівка

	сторінці	<p>відкриється</p> <p>головна сторінка порталу</p> <p>3. Перейти на вкладку про нас</p> <p>4. Натиснути на кнопку листівка</p>		
14	Перевірка вкладки «Курси»	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку «Курси»</p>	Перехід на сторінку де відображається список курсів.	Перехід на сторінку де відображається список курсів.
15	Перевірка відображення інформацій про курси після переходу на вкладці «Курси»	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p> <p>4. Натиснути на картинку курсу зі списку</p>	Перехід на сторінку з відображенням інформації курсу	Перехід на сторінку з відображенням інформації курсу

16	Перевірка відображення інформацій курсу після переходу на вкладці «Курси»	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку «Курси» 4. Натиснути на картинку курсу зі списку 	Перехід на сторінку з відображенням інформації курсу	Перехід на сторінку з відображенням інформації курсу
17	Перевірка переключення між інформацією	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку «Курси» 4. Натиснути на картинку курсу зі списку 5. Переключити на вкладці з інформацією 	Відображення нового тесту.	Відображення нового тесту.
18	Перевірка відкриття сторінки для створення	<ol style="list-style-type: none"> 1. Відкриття додатку 2. Зачекати поки відкриється 	Відкривається сторінка для створення нового курсу.	Відкривається сторінка для створення нового курсу.

	нового курсу	головна сторінка порталу 3. Перейти на вкладку з курсами 4. Натиснути на будь яке зображення.		
19	Перевірка введення інформацій для створення нового курсу	1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку з курсами 4. Натиснути на будь яке зображення 5. Ввести всі необхідні данні для створення нового курсу 6. Підтвердити створення	Створення нового курсу.	Створення нового курсу.

ВИСНОВКИ

У дипломній роботі було виконано розробку серверної частини веб додатку для організації роботи курсів з підготовки до зовнішнього незалежного оцінювання. В процесі виконання роботи було проаналізовано аналоги додатків: testportal.gov.ua, Освіта.ua, Просте ЗНО.

Під час дослідження було визначено переваги та недоліки аналогів, а також основний функціонал, який повинен містити застосунок.

На етапі моделювання діаграми для програмного забезпечення було поставлено вимоги до його функціоналу.

Створено структури та класи, які можуть використовуватись повторно для розширення функціоналу додатку.

Для досягнення поставленої мети було проаналізовано актуальність розроблюваного програмного забезпечення, досліджено та використано мову програмування С#, фреймворк ASP.NET Core та середу розробки Visual Studio, за допомогою яких було розроблено функціональні вимоги до застосунку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ASP.NET documentation - [Електронний ресурс]. – Режим доступу:
[https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0\](https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-6.0)
2. GitHub ASP.NET Core - [Електронний ресурс]. – Режим доступу:
<https://github.com/dotnet/aspnetcore>
3. Architectural principles- [Електронний ресурс]. – Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/architectural-principles>
4. Common web application architectures- [Електронний ресурс]. – Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures>
5. Common client-side web technologies- [Електронний ресурс]. – Режим доступу:
<https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-client-side-web-technologies>
6. documentation c#- [Електронний ресурс]. – Режим доступу:
<https://learn.microsoft.com/ru-ru/dotnet/csharp/>
10. UML diagram [Електронний ресурс] – Режим доступу до ресурсу:
<https://app.creately.com/d/muL3rq9M7lS/edit>
11. All You Need to Know About UML Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://tallyfy.com/uml-diagram/>
12. osvita.ua - [Електронний ресурс]. – Режим доступу:
<https://osvita.ua/test/answers/>
13. testportal.gov.ua - [Електронний ресурс]. – Режим доступу:
<https://testportal.gov.ua>

Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Проектування та розробка онлайн-курсу "РОЗРОБКА ДОДАТКУ
ДЛЯ ОРГАНІЗАЦІЇ РОБОТИ КУРСІВ З ПІДГОТОВКИ ДО
ЗОВНІШНЬОГО НЕЗАЛЕЖНОГО ОЦІНЮВАННЯ. СПЕЦЧАСТИНА:
РОЗРОБКА BACKEND-ЧАСТИНИ ЗАСОБАМИ C#

Виконав студент 4 курсу

групи ПД-42

Дубей Роман Русланович

Керівник роботи

Старший викладач кафедри ІПЗ Гаманюк Ігор Михайлович
Київ – 2023

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- Об'єкт дослідження – процес навчання для здачі зовнішнього незалежного оцінювання.
- Предмет дослідження – онлайн платформа для онлайн курсів підготовки до зовнішнього незалежного оцінювання.
- Мета роботи – Забезпечити користувачам зручну та ефективну платформу для підготовки.

АНАЛІЗ АНАЛОГІВ



	Освіта.юа	testportal.gov.ua
Великий вибір предметів	9	6
Простота використання	5	3
Взаємодія з користувачем	7	5
Доступність	8	6
Середня оцінка	7,25	5

ВИСНОВКИ

1. Результатом кваліфікаційної роботи бакалавра є застосунок "Виготовлення декору для дому", який включає в себе відеоуроки та інформаційні матеріали для створення декору різного типу та стилю. Застосунок розроблений з використанням мови програмування *C#* та платформи ASP.NET Core та Razor для створення backend-частини.
2. Для досягнення поставленої мети було розроблено серверну частину для онлайн-застосунку. Створенно методи для функціонування та відображення даних з бази даних.
3. Застосунок був перевірений в різних браузерах, зокрема Google Chrome, Mozilla Firefox, Opera, Microsoft Edge.
4. Створено базу даних де зберігаються данні користувачів та данні про курс
5. Створений застосунок відповідає заданій тематиці та готовий до використання.

Дякую за увагу!