

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра інженерії програмного забезпечення

## Пояснювальна записка

до бакалаврської кваліфікаційної роботи  
на ступінь вищої освіти бакалавр

на тему: «Проектування та розробка онлайн-курсу "Виготовлення декору для дому". Спец частина: Розробка backend-частини мовою C# на платформі ASP.NET Core та Razor»

Виконав: студент 4 курсу, групи ПД-42

спеціальності

121 Інженерія програмного  
забезпечення

(шифр і назва спеціальності)

.Каліш В.В

(прізвище та ініціали)

Керівник Яскевич В.О.

(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

**ЗАВДАННЯ**  
**НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

Каліш Владислав В'ячеславович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Проектування та розробка онлайн-курсу "Виготовлення декору для дому". Спец частина: Розробка backend-частини мовою C# на платформі ASP.NET Core та Razor»

Керівник роботи \_\_\_\_\_ Яскевич В.О., кандидат технічних наук.

(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 «1» червня 2023 року

3. Вихідні дані до роботи:

3.1. Публікації;

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

4.2. Програмна реалізація додатку

4.3. Приклади використання та тестування застосунку

4.4. Аналіз аналогів програмного забезпечення

5. Перелік графічного матеріалу

5.1. Результати проведеного опитування

5.2. Аналоги

5.3 Діаграми

5.4 Програмні засоби реалізації

6. Дата видачі завдання «25» лютого 2023 року

---

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів работ и	Примітка
1	Аналіз предметної області	24.02.2023	Виконано
2	Засоби розробки веб застосунку	26.02.2023 – 15.03.2023	Виконано
3	Розробка програмного забезпечення	15.03.2023 – 20.03.2023	Виконано
4	Висновки та оформлення роботи	04.05.2023 – 12.05.2023	Виконано
5	Попередній захист	1.06.2023	
6	Захист роботи		

Студент \_\_\_\_\_ Каліш В.В.  
( підпис ) ( прізвище та ініціали )

Керівник роботи \_\_\_\_\_ Яскевич В.О.  
( підпис ) ( прізвище та ініціали )





## РЕФЕРАТ

Текстова частина бакалаврської роботи 61 с., 41 рис., 11 джерел.

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

Мета роботи – поліпшення процесу навчання з виготовлення декору для дому, шляхом розробки та реалізації серверної частини онлайн-курсу з використанням мови C# на платформі ASP.NET Core.

Об'єкт дослідження – процес навчання з виготовлення декору для дому за допомогою онлайн-платформи та серверного додатку.

Предмет дослідження – розробка серверної частини для забезпечення доступності та функціональності онлайн-курсу з виготовлення декору для дому.

Робота виконувалася за допомогою методів моделювання та системного аналізу.

Суть розробки полягає в створенні конкурентно спроможного онлайн застосунку 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.....	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Аналіз предметної області і напрямки дослідження.....	9
1.2 Аналіз існуючих онлайн платформ.....	9
1.3 Моделювання UML-діаграми.....	17
2 ЗАСОБИ РОЗРОБКИ ВЕБ ЗАСТОСУНКУ.....	19
2.1 Мова програмування С#.....	19
2.2 Фреймворк ASP.NET.....	21
2.3 Середовище розробки Visual Studio.....	24
3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ.....	26
3.1 Розробка серверної-частини додатку.....	26
3.2 Розробка бази даних.....	39
3.2 Тестування веб застосунку.....	45
3.4 Тестування веб застосунку для онлайн курсів.....	48
ВИСНОВКИ.....	59
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	60

## ВСТУП

Онлайн-курси набирають великої популярності в останні роки, оскільки технології розвиваються і все більше людей мають доступ до інтернету. Ці курси пропонують зручний та гнучкий спосіб освоїти нові навички та отримати знання, не виходячи з власного будинку.

Об'єктом дослідження є навчальний курс з виготовлення декору для дому.

Онлайн застосунки надають доступ до широкого кола предметів, які можуть бути недоступні на місцевому рівні. Наприклад, якщо хтось живе у сільській місцевості або в місці з обмеженими освітніми ресурсами, він все одно може вивчати різні теми за допомогою онлайн-курсів. Крім того, онлайн-курси часто мають глобальне охоплення, поєднуючи студентів з викладачами та іншими учнями з усього світу.

Метою даної роботи є поліпшення процесу навчання з виготовлення декору для дому, шляхом розробки та реалізації серверної частини онлайн-курсу з використанням мови C# на платформі ASP.NET Core.

Онлайн-курси також можуть бути доступнішими, ніж традиційні очні курси, оскільки вони не вимагають фізичної присутності у приміщенні. Крім того, деякі онлайн-курси можуть пропонувати фінансову допомогу або стипендії, щоб зробити освіту більш доступною для тих, хто не має фінансових коштів для її оплати.

Основними завданнями цієї роботи є:

- Провести аналіз існуючих аналогів веб додатків. Функціональні можливості аналогічних програмних забезпечень, виявити переваги та недоліки кожного аналога.
- Спроекувати та розробити онлайн застосунок для онлайн курсів.
- Провести тестування та відладку розробленого продукту.



# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Аналіз предметної області і напрямки дослідження

Тема онлайн-курсів з декорування оселі - це широка та динамічна сфера, яка потребує постійних досліджень та аналізу останніх тенденцій, а також надання найкращих практик у сфері декорування.

Онлайн-курси пропонують поглиблений погляд на найпопулярніші стилі та тенденції домашнього декору, такі як мінімалізм, богемний стиль та скандинавський стиль. Вони також покажуть вам, як використовувати кольори, матеріали та текстури в кожному з цих стилів і як успішно інтегрувати їх у дизайн вашого інтер'єру. Крім того, існують онлайн-курси, присвячені інтер'єрам зі стійких та екологічних матеріалів, враховуючи зростаючий попит на екологічно чисті продукти.

Дослідження найефективніших маркетингових стратегій для онлайн-курсів також може допомогти охопити та залучити широку аудиторію. Це може включати аналіз тенденцій і взаємодії в соціальних мережах, методи оптимізації пошукових систем і цільові рекламні кампанії.

## 1.2 Аналіз існуючих онлайн платформ

Для аналізу веб застосунків для онлайн-курсів було обрано наступні додатки:

1. Coursera;
2. Khan Academy;
3. edX.

### 1.2.1 Аналіз веб застосунку Coursera

Coursera – провідний застосунок для онлайн-навчання, яка за останні роки зробила революцію у сфері освіти. Завдяки широкому вибору курсів, сертифікатів

та програм на здобуття ступеня Coursera стала популярним вибором для учнів у всьому світі, які прагнуть покращити свої знання та навички у різних галузях.[6]

Однією з унікальних особливостей Coursera є різноманітність пропонованих курсів. Учні можуть обирати більш ніж 6000 курсів з таких предметів, як бізнес, інформатика, наука про дані, соціальні та гуманітарні науки. Ці курси викладають професори та експерти з провідних університетів та організацій по всьому світу, включаючи Єльський університет, Університет Джона Хопкінса, Стенфордський університет та університет Мічігану. Платформа також співпрацює з компаніями та організаціями, пропонуючи програми професійного навчання та розвитку кар'єри, що робить її популярним вибором для людей, які прагнуть просунутися кар'єрними сходами.

Ще одним унікальним аспектом Coursera є наголос на доступності. Курси Coursera призначені для учнів усіх рівнів від початківців до фахівців. Платформа пропонує курси у різних форматах, включаючи відео-лекції, вікторини, оцінювані колегами завдання та інтерактивні форуми, що дозволяє учням навчатися у своєму власному темпі та за власним розкладом.

Відданість Coursera також відображена у її програмі фінансової допомоги, яка надає учням фінансову допомогу, щоб допомогти їм сплатити курси. Ця програма допомогла тисячам учнів по всьому світу отримати доступ до високоякісної освіти, яку вони не могли б собі дозволити.

Coursera також знаходиться в авангарді інновацій у галузі онлайн-освіти. Платформа розробила інноваційні інструменти та технології, які допомагають учням взаємодіяти з матеріалами курсу та співпрацювати з іншими учнями. Ці інструменти включають інтерактивні відео-лекції, автоматизовані системи оцінювання та складні інструменти аналізу даних, які допомагають викладачам відстежувати прогрес учнів та надавати персоналізований зворотний зв'язок.

Головну сторінку застосунку відображено на Рисунку.1.1

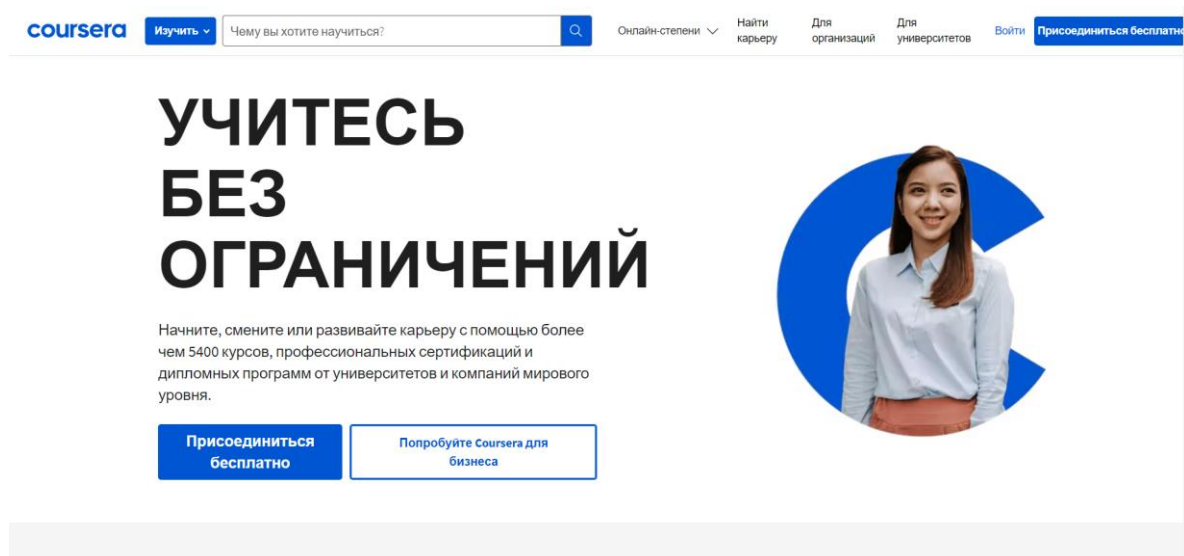


Рисунок. 1.1 - Платформа Coursera

### **Переваги:**

1. Доступ до високоякісних курсів найкращих університетів: Coursera пропонує курси, сертифікати та програми на здобуття ступеня від найкращих університетів та організацій по всьому світу. Учні можуть отримати доступ до високоякісного освітнього контенту, що викладається відомими професорами та експертами у своїх галузях;

2. Гнучкість та зручність. Курси Coursera розроблені так, щоб бути гнучкими та зручними, що дозволяє учням навчатися у своєму власному темпі та за власним розкладом. Учні можуть отримати доступ до матеріалів курсу та виконувати завдання з будь-якого місця, де є підключення до Інтернету, що дозволяє легко вписати навчання у своє зайняте життя;

3. Доступність: незважаючи на те, що деякі курси Coursera вимагають оплати сертифікатів або програм на отримання ступеня, багато курсів доступні безкоштовно. Учні також можуть подати заявку на фінансову допомогу, щоб покрити вартість курсів, що робить їх доступнішими для ширшого кола учнів;

4. Різноманітність курсів: Coursera пропонує курси з широкого кола предметів, від бізнесу та технологій до соціальних та гуманітарних наук. Це дозволяє учням легко знаходити курси, що відповідають їх інтересам та кар'єрним цілям.

### **Недоліки:**

1. Обмежена взаємодія з інструкторами: хоча курси Coursera пропонують деяку взаємодію з інструкторами через форуми або робочий час, рівень взаємодії часто обмежений у порівнянні з традиційними класними кімнатами;

2. Обмежена взаємодія з іншими учнями: хоча Coursera пропонує оцінювані колегами завдання та форуми, на яких учні можуть взаємодіяти один з одним, рівень взаємодії часто менше, ніж той, що може бути досягнутий у традиційній обстановці у класі;

3. Можливість академічної нечесності: дистанційний характер Coursera може утруднити перевірку того, що учні виконують завдання чи оцінки самостійно. Це може створити можливості для академічної нечесності;

4. Обмежена вартість сертифікатів: хоча Coursera пропонує сертифікати для деяких курсів, вартість цих сертифікатів може бути обмежена, оскільки вони можуть не визнаватись роботодавцями або навчальними закладами.

### **1.2.2 Аналіз веб застосунку Khan Academy**

Khan Academy - некомерційна освітня організація, заснована в 2008 році педагогом Салман Ханом. Організація надає безкоштовні онлайн-курси та навчальні відеоролики з широкого кола предметів, включаючи математику, природознавство, історію та інформатику. Основна мета Khan Academy полягає в тому, щоб надати безкоштовну освіту світового класу всім і скрізь.[7]

Однією з унікальних особливостей Khan Academy є орієнтованість на персоналізоване навчання. Платформа використовує адаптивні технології для створення індивідуального досвіду навчання кожного учня. У міру того, як учні виконують вправи та тести, платформа відстежує їх прогрес і надає зворотний зв'язок у тих галузях, де вони потребують покращення. Це дозволяє учням працювати у своєму власному темпі та зосередитися на тих сферах, де їм потрібна найбільша допомога.

Ще одна унікальна особливість Khan Academy – велика бібліотека навчальних відео. Відео створені командою експертів Khan Academy та охоплюють широке коло

тем, від базової арифметики до просунутого числення. Відеоролики розроблені таким чином, щоб бути захоплюючими та простими для розуміння, що робить складні концепції доступними для учнів різного віку та верств суспільства.

Платформа Khan Academy також включає інтерактивні вправи та вікторини, які дозволяють учням застосовувати отримані знання на практиці. Вправи розроблені так, щоб бути складними, але захоплюючими, і вони дають учням миттєвий зворотний зв'язок про свою роботу.

Крім безкоштовних онлайн-курсів та навчальних відео, Khan Academy також співпрацює зі школами та організаціями по всьому світу, щоб надавати рішення для індивідуального та змішаного навчання. Орієнтація організації на надання безкоштовної високоякісної освіти всім та скрізь зробила її популярним вибором як для учнів, так і для викладачів.

Головну сторінку Khan Academy відображено на Рисунку.1.2.

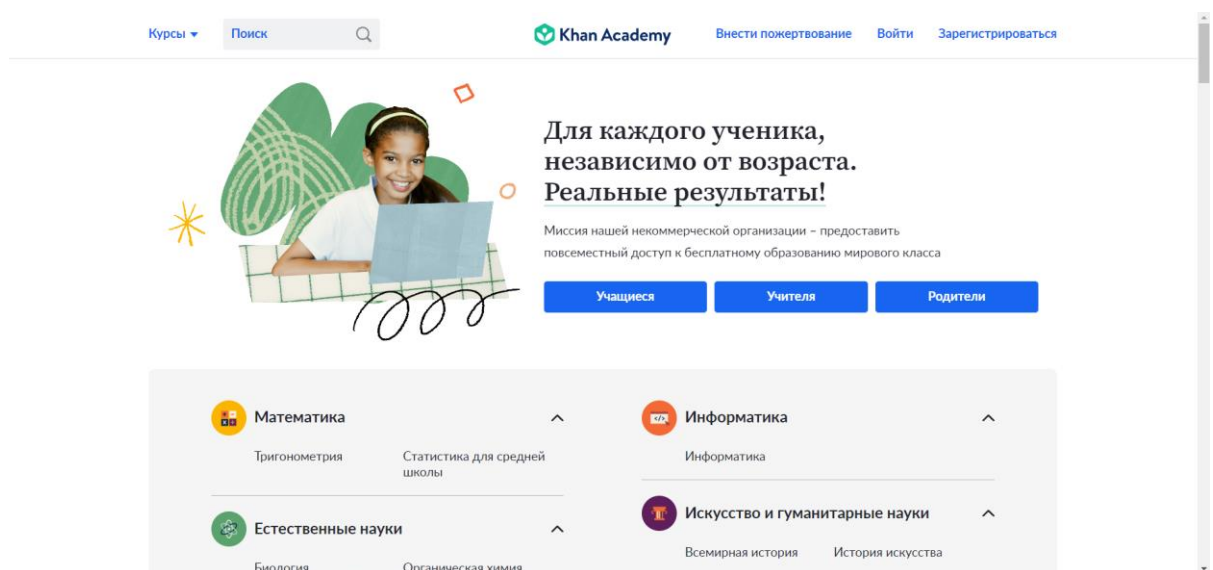


Рисунок. 1.2 - Платформа Khan Academy

### Преваги:

1. Безкоштовно та доступно: Khan Academy абсолютно безкоштовна та доступна для всіх, хто має підключення до Інтернету. Це робить його ідеальним ресурсом для учнів, які можуть не мати доступу до традиційних навчальних матеріалів;

2. Індивідуальне навчання: адаптивна технологія Khan Academy створює індивідуальний навчальний процес для кожного учня, дозволяючи їм працювати у своєму власному темпі та зосередитися на тих галузях, де їм найбільше потрібна допомога;

3. Широка бібліотека високоякісного контенту: Khan Academy пропонує велику бібліотеку навчальних відео, інтерактивних вправ і вікторин, що охоплюють широкий спектр предметів, від базової арифметики до обчислення;

4. Захоплюючий та простий для розуміння контент. Навчальні відеоролики Khan Academy розроблені таким чином, щоб бути захоплюючими та простими для розуміння, що робить складні концепції доступними для учнів будь-якого віку та походження.

### **Недоліки:**

1. Обмежена взаємодія з інструкторами: Khan Academy — це насамперед онлайн-платформа для самостійного навчання, що означає, що учні можуть мати обмежену взаємодію з інструкторами. Це може бути складним завданням для учнів, яким потрібно більше практичного керівництва та підтримки;

2. Обмежена соціальна взаємодія: Оскільки Khan Academy — це насамперед онлайн-платформа, у учнів можуть бути обмежені можливості спілкування з іншими учнями. Це може бути недоліком для учнів, які вважають за краще співпрацювати з іншими або отримують користь від соціальної взаємодії;

3. Обмежений охоплення деяких предметів: Хоча Khan Academy охоплює широке коло предметів, глибина і охоплення деяких предметів можуть бути обмежені. Учні, яким потрібна більш просунута або спеціалізована навчальна програма, можливо, знадобиться доповнити навчання додатковими ресурсами;

4. Надмірна залежність від технологій: адаптивні технології та онлайн-формат Khan Academy можуть підходити не всім учням. Деякі можуть віддати перевагу більш традиційним методам навчання або можуть не мати підключення до Інтернету.

### 1.2.3 Аналіз веб застосунку edX

«edX» - це некомерційна платформа для онлайн-навчання, заснована в 2012 році Гарвардським університетом та Массачусетським технологічним інститутом (MIT). Платформа пропонує широкий спектр онлайн-курсів та програм з різних предметів, включаючи інформатику, науку про дані, управління бізнесом, інженерію, гуманітарні науки та багато іншого. edX прагне надати якісну освіту студентам з усього світу, незалежно від їхнього місцезнаходження, походження чи фінансового стану.[8]

edX пропонує два типи курсів: курси для самостійного навчання та курси під керівництвом інструктора. Курси для самостійного навчання доступні для студентів у зручному для них темпі, у той час як курси під керівництвом інструктора мають конкретні дати початку та закінчення та вимагають, щоб студенти дотримувалися встановленого розкладу. Більшість курсів пропонуються безкоштовно, в той час як деякі курси потребують плати за підтверджений сертифікат.

Платформа співпрацює з провідними університетами та установами по всьому світу, щоб пропонувати своїм студентам високоякісну освіту. Деякі його партнери включають Гарвард, Массачусетський технологічний інститут, Берклі, Колумбію та інші. edX також пропонує програми MicroMasters, які є серією курсів для випускників, призначених для надання студентам навичок і знань, необхідних для досягнення успіху в їх кар'єрі.

Головну сторінку edX відображено на Рисунку.1.3.

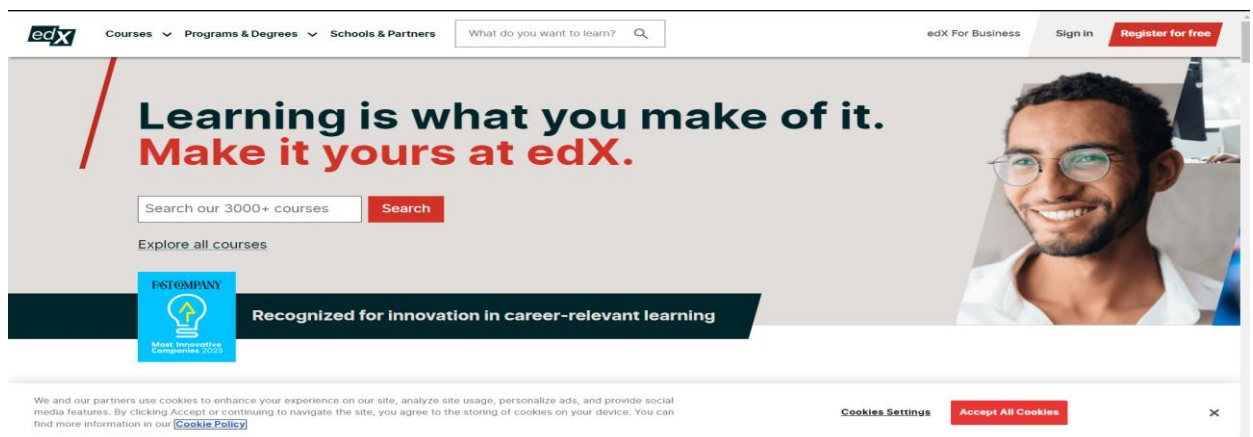


Рисунок. 1.3 - Платформа edX

## **Переваги:**

1. **Високоякісний контент:** edX надає доступ до широкого спектру курсів провідних університетів та інституцій з усього світу. Курси створюються та викладаються досвідченими інструкторами та експертами у предметній галузі, що забезпечує високу якість змісту;

2. **Гнучкість:** курси edX спроектовані так, щоб бути гнучкими, що дозволяє учням отримувати доступ до них з будь-якого місця, в будь-який час і з будь-якого пристрою. Це дозволяє студентам навчатися у своєму власному темпі та розкладі;

3. **Доступність:** багато курсів на edX безкоштовні, що робить його чудовим варіантом для людей, які хочуть навчатися, але не можуть дозволити собі традиційну вищу освіту;

4. **Сертифікація:** edX пропонує різні варіанти сертифікації від безкоштовних перевірених сертифікатів до професійних сертифікатів та програм мікромагістратури. Ці сертифікати можуть допомогти людям покращити свої навички, покращити своє резюме і просунутися кар'єрними сходами.

## **Недоліки:**

1. **Немає академічного акредитування:** хоча edX пропонує широкий спектр курсів від авторитетних навчальних закладів, більшість курсів не пропонують академічного кредиту. Це означає, що студенти не можуть використовувати свою курсову роботу edX для виконання вимог ступеня у більшості університетів;

2. **Потребує само мотивації та дисципліни:** курси edX призначені для самостійного вивчення, а це означає, що у студентів має бути мотивація та дисципліна, щоб виконувати курсову роботу самостійно. Без структури традиційного навчання у класі деяким учням може бути важко залишатися правильному шляху;

3. **Обмежені варіанти курсів:** хоча edX пропонує широкий спектр курсів, вибір може бути не таким широким, як у традиційних університетах. Крім того, не всі курси доступні будь-коли, що може обмежити можливості для студентів.



### 1.3 Моделювання UML-діаграми

UML - що означає Unified Modeling Language, є мовою візуального моделювання, що широко використовується в розробці програмного забезпечення.[10]

UML – це потужний інструмент для розробки програмного забезпечення, оскільки він може допомогти виявити потенційні проблеми на ранніх етапах процесу проектування та покращити взаємодію між членами групи. Він також досить гнучкий, щоб його можна було використовувати у різних методологіях розробки програмного забезпечення, включаючи agile, водоспад та спіраль.

UML використовується для представлення, проектування та документування програмних систем. Він надає стандартну нотацію для створення візуальних моделей програмних систем, які можуть допомогти у спілкуванні та розумінні складних проектів програмного забезпечення. UML включає безліч діаграм, кожна з яких є певною метою в процесі розробки програмного забезпечення.[11]

Діаграми використання є важливим інструментом для проектування та аналізу взаємодії користувачів та системи в рамках програмного продукту або послуги. Вони допомагають візуалізувати та описати функціональність програмного продукту або послуги з точки зору користувачів та систем, що з ними пов'язані.

Крім того, діаграми використання можуть бути корисним інструментом для комунікації між розробниками, проектним менеджером та замовником. Вони дозволяють візуалізувати та узгодити функціональність програмного продукту або послуги з різних точок зору та зробити її більш зрозумілою та доступною для всіх учасників проекту.

UML-діаграми для веб застосунку онлайн-курсів "Виготовлення декору для дому" представлені на рисунок. 1.4.

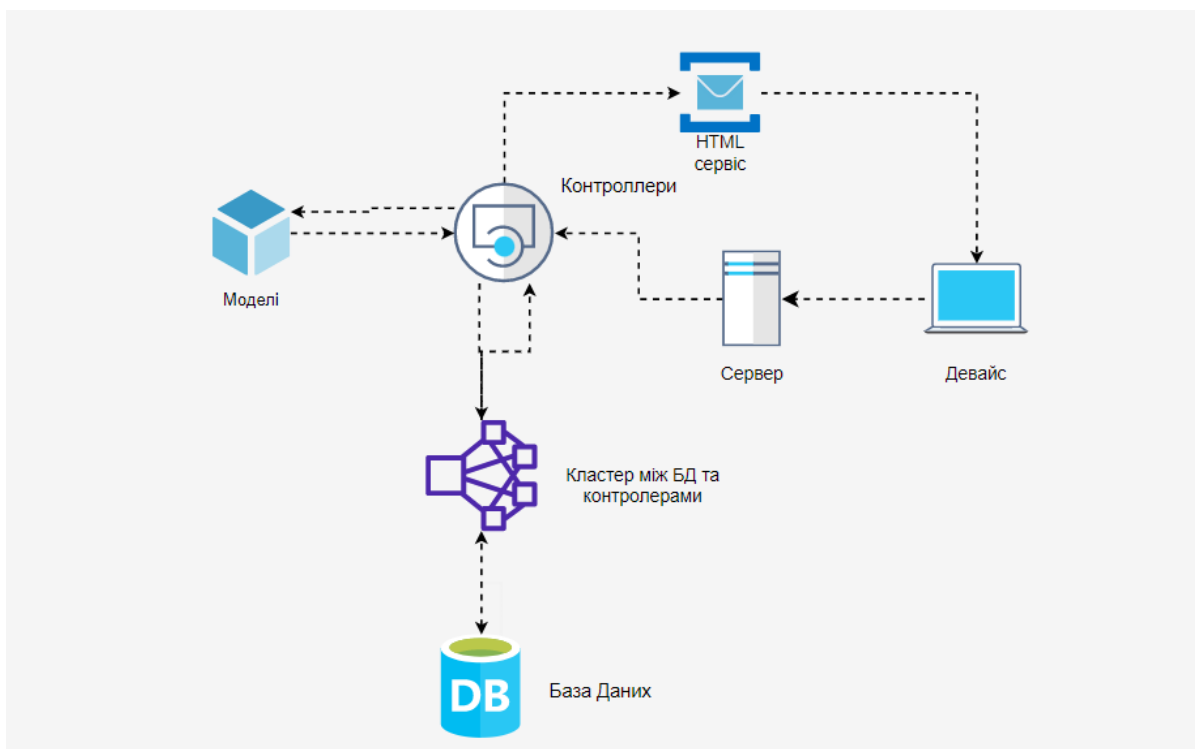


Рисунок. 1.4 – Діаграма архітектури

## 2 ЗАСОБИ РОЗРОБКИ ВЕБ ЗАСТОСУНКУ

### 2.1 Мова програмування C#

C# - це сучасна об'єктно-орієнтована мова програмування, розроблена Microsoft. Вперше був представлений в 2000 році як частина платформи Microsoft .NET і з тих пір став однією з мов програмування, що найбільш широко використовуються в світі.[9]

C# був розроблений, щоб бути простим, потужним та виразним. Його часто порівнюють з Java, оскільки він має багато спільного синтаксису та структури. Однак C# також знаходиться під сильним впливом C і C++ і включає функції, що полегшують розробку складних додатків.

Однією із ключових особливостей C# є його здатність працювати з платформою .NET. Цей фреймворк надає велику бібліотеку попередньо створених функцій та класів, які можна використовувати для розробки широкого спектру програм, від програмного забезпечення для настільних комп'ютерів до веб-додатків.

Ось кілька прикладів того, що можна зробити за допомогою C#:

1. Настільні програми: C# можна використовувати для розробки настільних програм, таких як Microsoft Office, Adobe Photoshop і навіть відеоігор. Однією з найпопулярніших ігор, розроблених за допомогою C#, є Minecraft;

2. Веб-програми: C# можна використовувати для розробки веб-додатків, таких як сайти інтернет-магазинів, платформи соціальних мереж та системи онлайн-банкінгу. Наприклад, Amazon.com та eBay.com використовують C# у своїх веб-додатках;

3. Мобільні програми: C# можна використовувати для розробки мобільних програм для iOS, Android і Windows Phone. Xamarin - популярний інструмент для розробки кроссплатформених мобільних додатків з використанням C#;

4. Штучний інтелект та машинне навчання: C# також можна використовувати для розробки програм штучного інтелекту та машинного навчання. Наприклад, платформа Accord.NET надає багато алгоритмів машинного навчання, які можна використовувати з C#.

Однією із сильних сторін C# є його простота. Мова включає безліч функцій, що спрощують написання чистого, читаного коду. Наприклад, C# включає автоматичне керування пам'яттю, що усуває необхідність у ручному керуванні пам'яттю та знижує ймовірність витоків пам'яті та інших поширених помилок програмування.

C# також включає ряд потужних функцій, що спрощують розробку складних додатків. Наприклад, C# включає підтримку інтерфейсів, що дозволяє розробникам визначати набір методів, які мають бути реалізовані будь-яким класом, що реалізує інтерфейс. Це спрощує розробку великих та складних додатків, розбиваючи їх на дрібніші та більш керовані компоненти.

Ще одна потужна функція C# - підтримка делегатів та подій. Делегати дозволяють розробникам визначати посилання на метод, який можна передати як параметр іншого методу. Події дозволяють розробникам створювати власні обробники подій, які можна використовувати для реагування на певні події у додатку.

C# також включає підтримку LINQ (Language Integrated Query), що дозволяє розробникам писати запити до широкого спектру джерел даних, включаючи бази даних, документи XML і колекції в пам'яті.

### **Менеджер пакетів**

NuGet – це менеджер пакетів для .NET, розроблений Microsoft. Це дозволяє розробникам легко керувати залежностями та бібліотеками для своїх проєктів .NET. NuGet надає простий спосіб пошуку та встановлення пакетів, а також управління оновленнями та залежностями.

Однією з основних переваг використання NuGet є те, що він полегшує процес управління залежностями в проектах .NET. Замість того, щоб вручну завантажувати та керувати залежностями, розробники можуть просто вказати потрібні їм пакети у файлі проекту, а NuGet подбає про все інше. Це може заощадити значну кількість часу та зусиль, особливо для великих проектів із багатьма залежностями.

NuGet також надає низку інших корисних функцій, у тому числі:

1. Керування версіями та залежностями: NuGet дозволяє розробникам вказувати певні версії пакетів та керувати залежностями між різними пакетами. Це може допомогти гарантувати, що в проектах використовуються сумісні версії пакетів і уникнути проблем із конфлікуючими залежностями;

2. Створення та публікація пакетів: NuGet надає інструменти для створення та публікації пакетів, дозволяючи розробникам легко ділитися своїми бібліотеками та компонентами з іншими учасниками спільноти .NET;

3. Розміщення приватних пакетів: NuGet підтримує розміщення приватних пакетів, що дозволяє організаціям розміщувати власні пакети та залежність всередині. Це може бути корисним для компаній, яким необхідно підтримувати контроль за своїми залежностями або пред'являти особливі вимоги до безпеки або відповідності вимогам.

Таким чином, NuGet – це потужний та корисний інструмент для розробників .NET. Спрощуючи процес управління залежностями та бібліотеками, NuGet може заощадити час та зусилля у проектах розробки. Крім того, його підтримка створення пакетів, управління версіями та приватного хостингу робить його цінним інструментом для спільного використання та розповсюдження пакетів у спільноті .NET.

## **2.2 Фреймворк ASP.NET**

ASP.NET — це платформа веб-застосунків, розроблена Microsoft, яка дозволяє розробникам створювати динамічні веб-додатки, керовані даними. Він

побудований на основі платформи .NET і дозволяє розробникам писати серверний код із використанням C# або Visual Basic .NET. ASP.NET надає безліч інструментів та бібліотек, які допомагають розробникам швидко та легко створювати потужні веб-програми.[1]

Однією з основних переваг використання ASP.NET є можливість відокремити рівень уявлення від рівнів бізнес-логіки та доступу до даних. Такий поділ забезпечує велику гнучкість і модульність дизайну програми, спрощуючи його обслуговування та масштабування у міру зростання програми. ASP.NET також підтримує широкий спектр моделей програмування, включаючи MVC (модель-представлення-контролер) та веб-API (інтерфейс прикладного програмування), які забезпечують додаткову гнучкість під час створення веб-додатків.

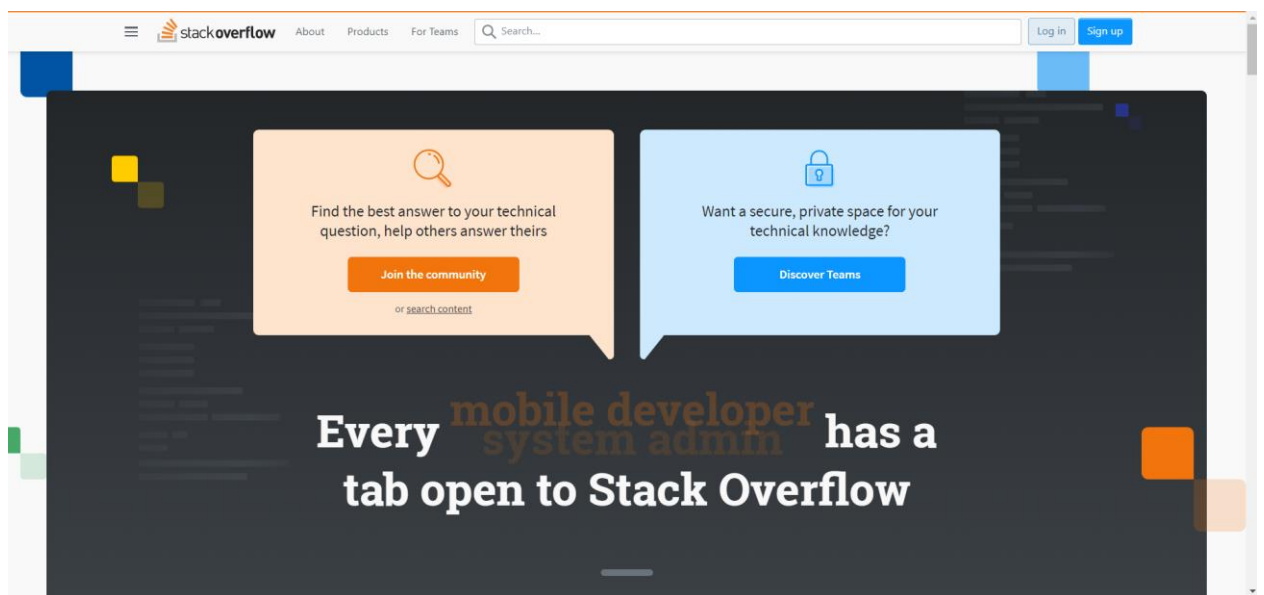


Рисунок. 2.1 - Приклад сайту створеного завдяки технології «ASP.NET»

Деякі приклади того, що можна зробити за допомогою ASP.NET:[4]

1. Створення платформи для блогів. Одним з найпопулярніших варіантів використання ASP.NET є створення платформи для ведення блогів. Цей тип програми зазвичай включає в себе такі функції, як автентифікація та авторизація користувачів, створення та редагування повідомлень у блогах, коментування та функції пошуку. Використовуючи вбудовані в ASP.NET функції автентифікації та

авторизації, розробники можуть швидко і легко створити безпечну систему входу в систему для своєї програми. Функції створення та редагування повідомлень у блогах програми можуть бути реалізовані за допомогою платформи ASP.NET MVC, яка забезпечує простий та інтуїтивно зрозумілий спосіб створення динамічних веб-сторінок;

2. Веб-сайт електронної комерції. Ще одним найпоширенішим варіантом використання ASP.NET є створення веб-сайтів електронної комерції. Цей тип програми зазвичай включає такі функції, як каталоги продуктів, кошик і обробка платежів. Використовуючи веб-API ASP.NET, розробники можуть легко інтегрувати свою програму зі сторонніми системами обробки платежів, такими як PayPal або Stripe. Функції каталогу продуктів та кошики покупок програми можуть бути створені з використанням уявлень ASP.NET Razor, які забезпечують чистий та лаконічний спосіб створення HTML-шаблонів;

3. Платформа соціальних мереж. Складнішим варіантом використання ASP.NET є створення платформи соціальних мереж. Цей тип програм зазвичай включає в себе такі функції, як профілі користувачів, стрічки новин, обмін повідомленнями та оновлення в реальному часі. Використовуючи бібліотеку SignalR ASP.NET, розробники можуть легко створювати системи обміну повідомленнями та повідомленнями в реальному часі для своїх програм. Профілі користувачів програми та функції стрічки новин можуть бути створені за допомогою платформи ASP.NET MVC, а функції обміну повідомленнями можуть бути створені за допомогою веб-API.

На додаток до цих прикладів ASP.NET надає широкий спектр інструментів та бібліотек, які допомагають розробникам створювати потужні та гнучкі веб-додатки. До них належать бібліотеки для доступу до даних, безпеки та оптимізації продуктивності, а також вбудована підтримка популярних веб-технологій, таких як HTML, CSS та JavaScript.[2]

## 2.3 Середовище розробки Visual Studio

«Visual Studio» - це потужне і популярне інтегроване середовище розробки (IDE), розроблене Microsoft. Він пропонує розробникам широкий спектр інструментів та функцій для створення програмних програм для різних платформ, включаючи Windows, iOS, Android та веб-додатки. Програмне забезпечення було вперше випущено в 1997 році і з тих пір стало одним із найбільш широко використовуваних середовищ розробки у світі.

Однією з найбільших переваг Visual Studio є його універсальність. Він підтримує широкий спектр мов програмування, включаючи C++, C#, Visual Basic, Python та JavaScript. Це означає, що розробники можуть використовувати Visual Studio для проектів, починаючи від невеликих сценаріїв і закінчуючи великомасштабними корпоративними програмами. Крім того, Visual Studio інтегрується з багатьма іншими популярними інструментами та платформами, включаючи GitHub, Azure та Unity.

Visual Studio призначена для оптимізації процесу розробки та полегшення написання коду розробниками. Він включає широкий спектр інструментів і функцій, у тому числі автодоповнення коду, засоби налагодження і вбудований компілятор. Функція завершення коду, наприклад, може допомогти розробникам писати код швидше, пропонуючи варіанти завершення коду в міру їх введення. Інструменти налагодження можуть допомогти розробникам швидко та ефективно знаходити та виправляти помилки в коді.

Ще однією значною перевагою Visual Studio є його потужна підтримка спільноти. Платформа має велику та активну спільноту розробників, які діляться порадами, рекомендаціями та найкращими практиками. Спільнота надає доступ до безлічі ресурсів, включаючи посібники, документацію та зразки коду, які можуть бути надзвичайно корисними для нових розробників.

Visual Studio також пропонує безліч додаткових функцій для досвідчених розробників. Наприклад, платформа підтримує спільну роботу та командний розвиток. Він включає інструменти для контролю версій, перевірки коду і



безперервної інтеграції і розгортання. Ці інструменти можуть допомогти командам працювати більш ефективно, зменшити кількість помилок та забезпечити своєчасну доставку коду.

Однією з найвизначніших особливостей Visual Studio є Visual Studio Marketplace. Це великий репозиторій розширень та надбудов, які можна завантажити та встановити безпосередньо в IDE. Ці розширення можуть додавати до платформи нові функції, такі як підтримка нових мов, генератори коду та засоби налагодження.

Visual Studio також постійно розвивається, регулярно випускаються нові функції та оновлення. Наприклад, остання версія Visual Studio, Visual Studio 2022, включає покращену продуктивність, нові засоби налагодження та розширену підтримку розробки мобільних додатків.

## 3 РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

### 3.1 Розробка серверної-частини додатку

При розробці веб-порталу, необхідно забезпечити, щоб користувач при відкриванні посилання на портал завжди попадав на головну сторінку, тому потрібно налаштувати посилання таким чином, щоб вони вказували на URL головної сторінки порталу. Це можна здійснити за допомогою методу, розміщеного в контролері головної сторінки, який відповідає за цю функцію (див. рисунок. 3.1).

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

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

Під час розробки головної сторінки порталу, на ній необхідно розмістити дві важливих кнопки (див. рисунок. 3.2):

- Кнопку «Придбати курс», яка, після натискання, відкриє сторінку з переліком доступних до покупки курсів;
- Кнопку «Дізнатись більше», яка, після натискання відкриє сторінку з інформацією про курси.



Рисунок. 3.2 – Головні кнопки головної сторінки «Придбати курс» та «Дізнатись більше»

Створення меню навігації є необхідним для кожного порталу, для цього потрібно створити кнопки у правому верхньому куті шапки порталу, яке буде

відображатись на кожній сторінці порталу (див. рисунок. 3.3), за допомогою яких користувач зможе перейти на сторінку з курсами, повернутися на головну сторінку порталу, переглянути інформацію про курс та про вчителя курсу, також зареєструватися або увійти у вже створений аккаунт.

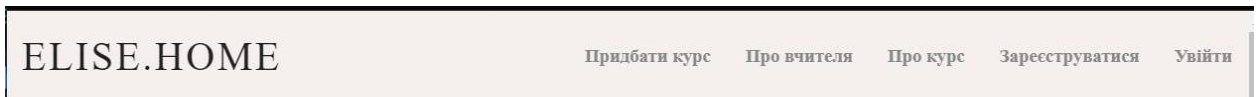


Рисунок. 3.3 – Меню навігації

Для цих кнопок необхідно створити методи в контролерах, які повернуть відповідні сторінки. Частина цих методів зображена на рисунках 3.4, 3.5, 3.6. Всі кнопки з меню навігації мають посилання на відповідні методи контролерів.

```
[HttpGet]
0 references
public IActionResult CoursesList()
{
    var CursesList = new List<DB.Curses>();

    if (User.IsInRole("User") || User.Identity.IsAuthenticated==false)
    {
        CursesList = DBService.SelectAllActiveCourses(db).ToList();
    }
    if (User.IsInRole("Admin"))
    {
        CursesList = DBService.SelectAllCourses(db).ToList();
    }

    foreach (var img in CursesList)
    {
        img.File = Convert.ToBase64String(img.MainPhoto);
    }

    return View("CoursesList", CursesList);
}
```

Рисунок. 3.4 - Метод в контролері, який повертає сторінку з доступними курсами

```
0 references
public async Task OnGetAsync(string returnUrl = null)
{
    ReturnUrl = returnUrl;
    ExternalLogins = (await _signInManager.GetExternalAuthenticationSchemesAsync()).ToList();
}
```

Рисунок.3.5 - Метод в контролері, який повертає сторінку для реєстрації

```

0 references
public IActionResult AboutTeacher()
{
    return View("AboutTeacher");
}

```

Рисунок. 3.6 – Метод в контролері, який повертає сторінку «Про вчителя»

Після натискання на кнопку «Придбати курс» користувача буде перенесено на сторінку, на якій будуть відображені всі доступні на порталі курси для покупки (рис. 3.7), які реалізовані за допомогою методів у відповідних контролерах.

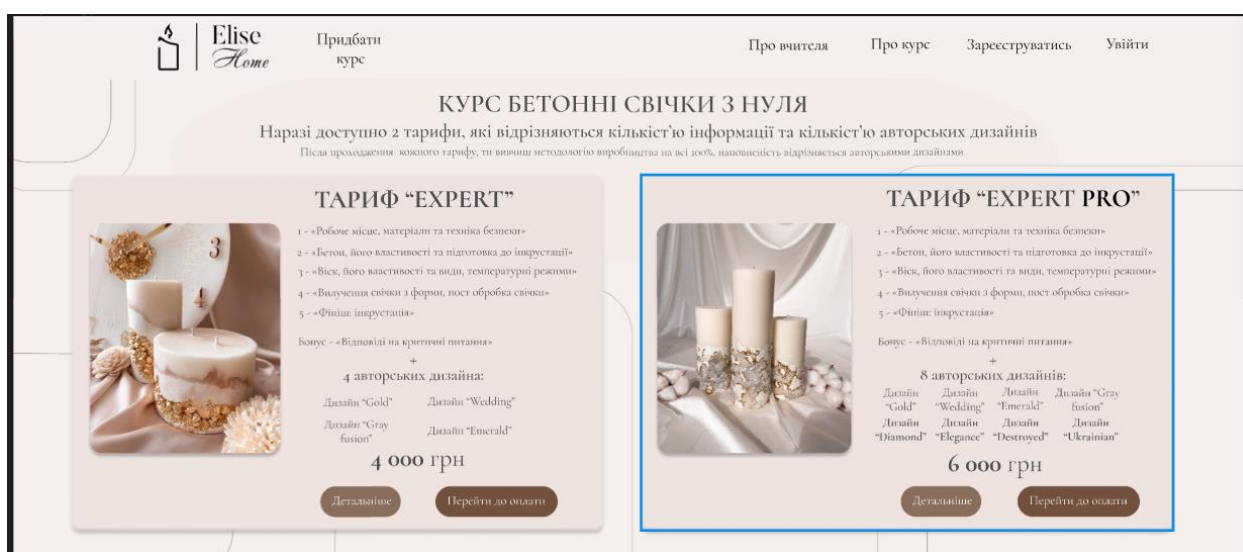


Рисунок.3.7 – Сторінка з доступними для покупки курсів

Якщо користувач натисне на один з доступних курсів, або на кнопку «Детальніше», то з'явиться сторінка з детальною інформацією про курс, як показано на рис. 3.8. Метод для відображення інформацій з курсу відображений на рисунку. 3.9 та рисунок. 3.10.

Придбати курс

[Про вчителя](#) | [Про курс](#) | [Зареєструватись](#) | [Увійти](#)

## ТАРИФ "ЕКСПЕРТ PRO"

### Зміст курсу

- Блок 1 - «Робоче місце, матеріали та техніка безпеки»**
  - Робоче місце, як його облаштувати і де воно повинно бути.
  - Матеріали які знадобляться, де купити, як обрати стартовий набір початківця
  - Техніка безпеки роботи з матеріалами, обладнання та екіпіровка для захисту здоров'я
- Блок 3 - «Віск, його властивості та види»**
  - Відомості вісків, який краще підходить та чому
  - Гіпти. Як їх бувають, чим відрізняються, який краще обрати та як тестувати воск з гіптом
  - Температурні режими. Що це таке, навіщо, і як їх дотримуватись
  - Заливаємо віск у форму, усі етапи до застигання
- Блок 5 - «Інкрустація»**
  - Інкрустація, як закріпити каміни усі методи
- Блок 2 - «Бетон та його властивості»**
  - Відомості форм, яку краще обрати
  - Як зробити форму власноруч
  - Підготовка до інкрустації
  - Замішуємо бетон, що додати, яка консистенція
  - Заливаємо бетон у форму
- Блок 4 - «Видачування свічки з форми, пост обробка свічки»**
  - Як дістати свічку з форми. Що робити, якщо свічка «не дістається»
  - Небезпечний метод (для свічки): як дістати свічку з форми, якщо сусід вкрав комунальні зручності
  - Пост обробка бетону і воску після видачування свічки з форми.
- Блок 6 Бонус - «Відповіді на критичні питання»**
  - Що впливає на відколи та як їх уникати
  - Плями на воску та звідки вони беруться
  - Чому свічка не горить
  - Недоліки - погано чи вигідно мистецтва
  - Як порозуміти співвартість свічки
  - Через що можуть з'явитись «запалки» на зовнішніх стінках воску

### 8 авторських дизайнів:

4 авторських дизайнів тільки для тарифу PRO:

[Перейти до оплати](#)

Рисунок. 3.8 – Сторінка з інформацією про курс

```

public IActionResult Course(Guid id)
{
    var ExistsCourseInUser = false;
    if (User.Identity.IsAuthenticated)
    {
        var user = DBService.FindUser(db, User.Identity.Name);
        ExistsCourseInUser = DBService.IsCourseBought(db, id, user.id);
    }

    #region TEST
    var course = DBService.FindCourse(db, id);
    course.IsBought = ExistsCourseInUser;

    if (ExistsCourseInUser == true)
    {
        List<string> filesByteList = DBService.SelectCourseFilesByCourseIdAndType(db, id, "mp4");
        //List<string> ff = DBService.SelectTT(db, id, "mp4");
        var files = new List<string>();
        foreach (var file in filesByteList)
        {
            files.Add(file);
        }
        course.Files = files;
    }
    else
    {
        List<string> filesByteList = DBService.SelectCourseFilesByCourseIdAndType(db, id, "img");
    }
}

```

Рисунок. 3.9 - Метод для відображення інформації з курсу

```
80 List<string> filesByteList = DBService.SelectCourseFilesByCourseIdAndType(db, id, "img");
81
82 var files = new List<string>();
83
84 foreach (var file in filesByteList)
85 {
86     files.Add(file);
87 }
88
89 //course.File =course.MainPhoto;
90
91 course.Files = files;
92
93
94 var listQuestions = DBService.SelectListQuestionsOfTupleByCourse(db, id.ToString());
95 if (listQuestions.Count > 0)
96 {
97     var list = new List<Tuple<string, string, int, string, int>>();
98     foreach (var item in listQuestions)
99     {
100         var t = new Tuple<string, string, int, string, int>(item.Question, item.AnswerFirst, Convert.ToInt32(item.IsFirstTrue), item.AnswerSecond, Convert.ToInt32(item.IsSecondTrue));
101         list.Add(t);
102     }
103     course.Questions = list;
104 }
105
106 #endregion
107
108 return View("Course", course);
109 }
```

Рисунок. 3.10. - Продовження методу для відображення інформації з курсу

Якщо користувач натисне кнопку «Перейти до оплати» на сторінці з усіма доступними курсами або на сторінці з інформацією про курс, відбудеться перевірка авторизації користувача - якщо користувач авторизувався він буде перенесений до сервісу оплати курсу, як тільки оплата буде здійснена, користувач матиме доступ до відео з курсу, де він зможе отримати інформацію у форматі відео, Рисунок.3.11.

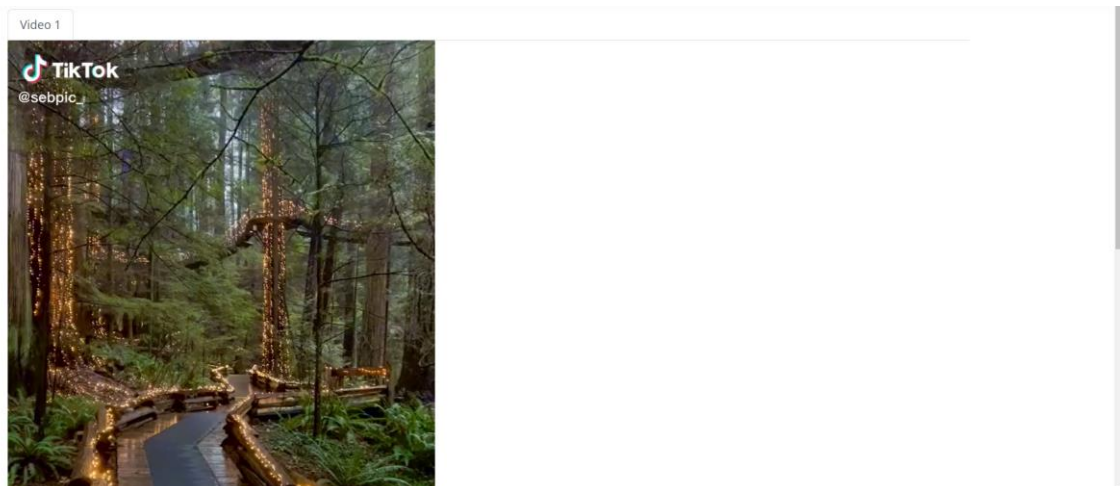


Рисунок.3.11 – Відео формат курсів

Якщо користувач не авторизувався йому буде запропоновано зареєструватися або увійти у вже існуючий акаунт.

Коли користувач натискає кнопку "Зареєструватися", він переходить на сторінку, де може створити новий акаунт (рисунок. 3.12). На сторінці "Зареєструватися" можна побачити форму на якій зображені поля для введення електронної пошти, паролю та підтвердження паролю, а також кнопку «Зареєструватися» або кнопку «Увійти», що понизу форми, якщо він загадав, що у нього вже є акаунт.

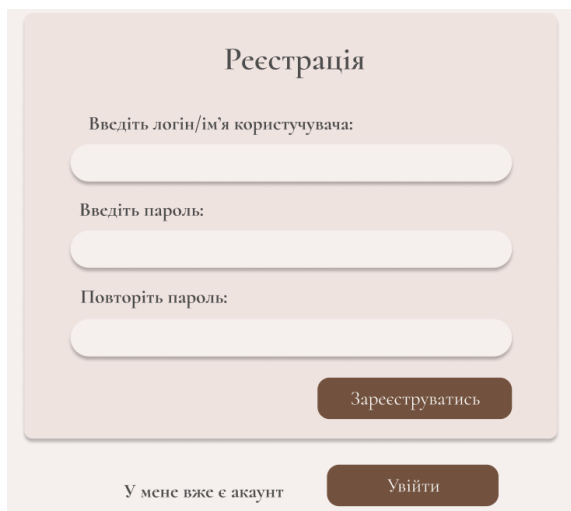


Рисунок. 3.12 – Форма реєстрації

При розробці форми для реєстрації, на формі було розміщено усі необхідні поля для надання користувачем потрібних даних.

Для того щоб, користувач надав придатні для використання дані, які обов'язково будуть захищені та збережені, при натисканні користувачем на кнопку "Зареєструватися" на формі, відбуваються наступні дії:

1. Збір введених користувачем даних з форми реєстрації.
2. Валідація введених даних для перевірки їх коректності та повноти. Це включає перевірку правильності формату електронної пошти, достатньої довжини пароля, відповідності вимогам до введення інших обов'язкових полів та ін.
3. Перевірка унікальності облікового запису, наприклад, перевірка, чи не зареєстрований користувач з такою ж електронною поштою або ім'ям.
4. Шифрування пароля або застосування хеш-функцій для безпечного зберігання парольного хешу у базі даних.

5. Запис введених даних про нового користувача до бази даних, створення облікового запису та призначення унікального ідентифікатора користувача.

Метод, за допомогою якого відбувається перевірка унікальності облікового запису, тобто якщо користувача з такою ж електронною поштою не зареєстрований (його не має в базі даних) то відбувається запис нового користувача в базу даних та після цього йому видається роль за замочуванням, зображений на рисунку. 3.13.

```
0 references
public async Task<ActionResult> 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);
        }
    }

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

Рисунок. 3.13 – Метод для обробки інформації користувача після реєстрації

Після успішного завершення реєстрації, користувача буде переадресовано на головну сторінку сайту з відповідним повідомленням про успішну реєстрацію. На цій сторінці буде відображене ім'я користувача та кнопка "Вийти", які зображені на Рисунку. 3.14.



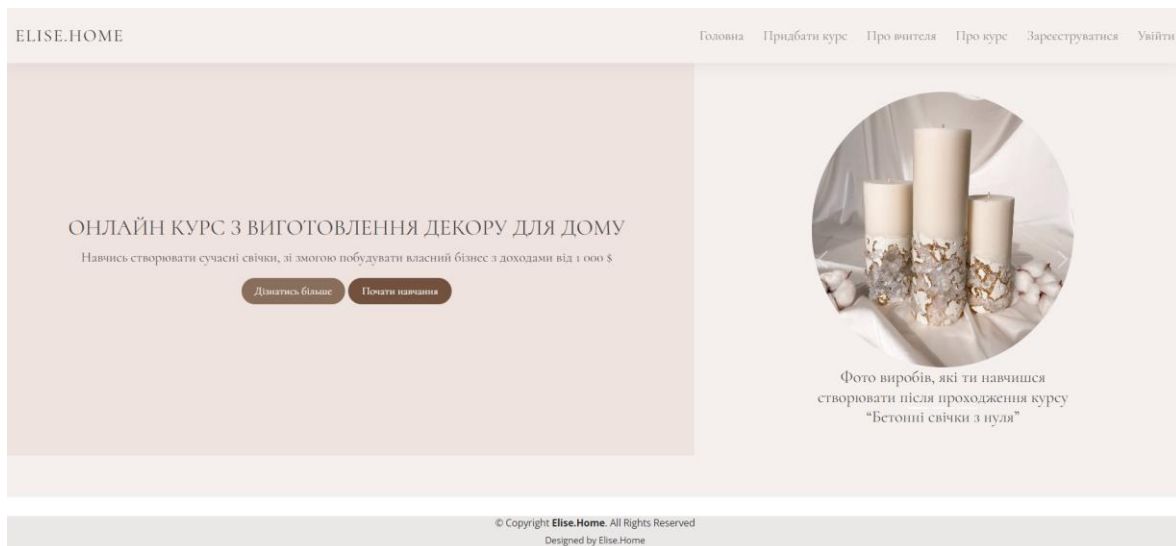


Рисунок.3.14 - Головна сторінка порталу після реєстрації користувача

Коли користувач натискає кнопку "Увійти", його буде перенесено на сторінку, де можна увійти у вже створений обліковий запис. На цій сторінці користувач може побачити форму з полями для введення електронної пошти, поля для введення пароля від облікового запису та кнопку "Увійти", які зображені на Рисунок. 3.15.

The image shows a login form titled "Вхід" (Login). The form is set against a light beige background. It contains two input fields: the first is labeled "Введіть логін/ім'я користувача:" (Enter login/username) and the second is labeled "Введіть пароль:" (Enter password). Below the input fields is a dark brown button with the text "Увійти" (Login). At the bottom of the form, there is a link "У мене ще немає акаунту" (I don't have an account yet) and a dark brown button with the text "Зареєструватись" (Register).

Рисунок. 3.15 – Форма входу

При розробці форми для входу в існуючий акаунт, на формі було розміщено усі необхідні поля для надання користувачем потрібних даних.

Для того щоб, користувач надав правильні дані, при натисканні користувачем на кнопку «Увійти» на формі, відбуваються наступні дії:

1. Збір введених користувачем даних з форми входу (зазвичай це електронна пошта або ім'я користувача та пароль);
2. Перевірка введених даних на валідність і коректність;
3. Перевірка збігу введених даних зі збереженими в базі даних обліковим записом користувача;
4. При успішній аутентифікації створення сесії для авторизації, переадресація користувача на головну сторінку сайту після входу;
5. При невдалій спробі входу відображення відповідного повідомлення про помилку.

Коли користувач успішно увійде до свого облікового запису, його буде перенаправлено на головну сторінку порталу, де буде відображатись ім'я користувача та кнопка "Вийти", які зображені на Рисунок.3.16.

Метод для обробки введених даних користувача для входу відображено на Рисунок.3.17.

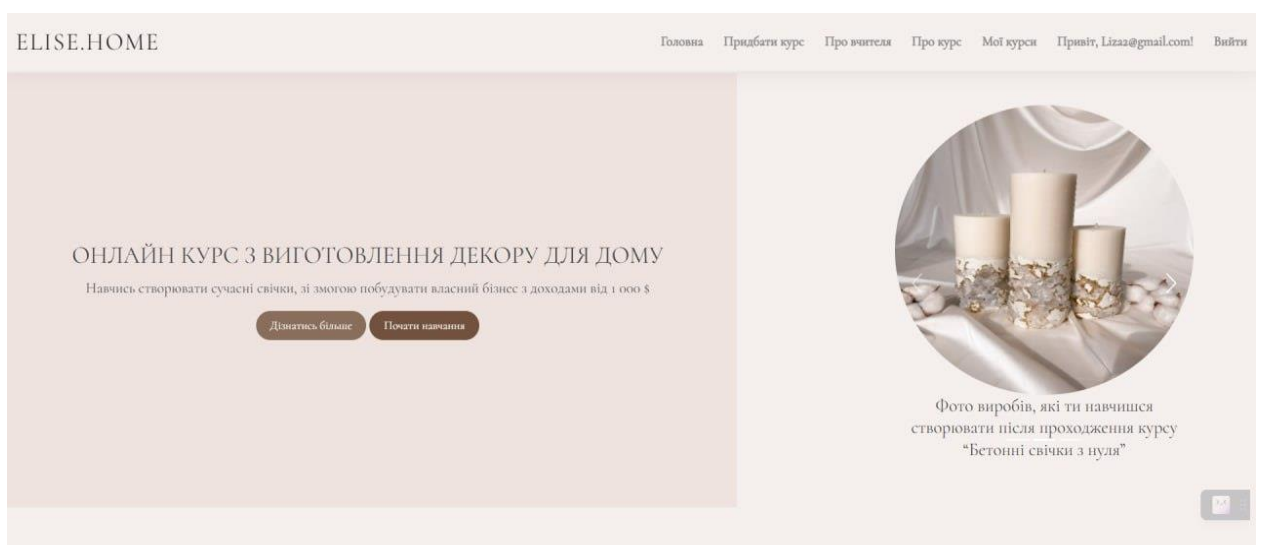


Рисунок. 3.16 – Головна сторінка порталу після входу користувача в акаунт

```
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();
}
```

Рисунок. 3.17 - Метод для входу користувача в особистий профіль

При вході в обліковий запис з роллю адміністратора, у користувача з'являється можливість переглянути сторінку зі списком усіх існуючих користувачів порталу завдяки появі кнопки «Користувачі» (рисунок. 3.18). Метод, який повертає список користувачів порталу зображений на рисунок.3.19.

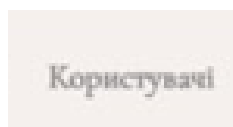


Рисунок. 3.18 – Кнопка «Користувачі»

```
0 references
public IActionResult UsersList()
{
    var usersList = DBService.SelectAllUsers(db);
    return View("UsersList", usersList);
}
```

Рисунок. 3.19 - Метод який повертає сторінку з користувачами

Після переходу адміністратора на вкладку з користувачами, з'явиться таблиця зі списком імен користувачів та кнопкою "Змінити", як показано на рисунок. 3.20.

Email	Save
admin3@gmail.com	Change
caixiao	Change
kalishvlad3@gmail.com	Change
OfficerV	Change

Рисунок. 3.20 – сторінка з користувачами порталу

Після натискання адміністратором на кнопку «Змінити», на екрані з’явиться інформація про користувача, включаючи доступні ролі та куплені курси. Адміністратор може змінити роль користувача та його куплені курси. (рисунок.3.21). Метод, який повертає інформацію користувача для редагування зображено на рисунок. 3.22.

Change User: admin3@gmail.com

Is Active

Course in User

Roles in User

Save

[Back to users page](#)

Рисунок. 3.21 – Сторінка з інформацією про користувача

```

{
    var user = DBService.FindUserById(db, id); //db.AspNetUsers.FirstOrDefault(x => x.Id == id);
    var roles = new Dictionary<string, string>();

    var listOfRoles = DBService.SelectAllRoles(db);
    foreach (var role in listOfRoles.Select(t => new { t.Id, t.Name }).ToDictionary(t => t.Id, t => t.Name))
    {
        roles.Add(role.Key, role.Value);
    }
    ViewBag.UserName = user.UserName;
    ViewBag.Roles = roles;

    var courseListForView = new List<string, string, string>();
    var courseList=DBService.SelectAllCourses(db);
    var courseListInBuyer = DBService.SelectCoursesListByBuyer(db,user.id);

    foreach (var t in courseList)
    {
        if (courseListInBuyer.Any(x => x.id == t.id))
        {
            courseListForView.Add((t.id.ToString(), t.Name, "selected"));
        }
        else
        {
            courseListForView.Add((t.id.ToString(), t.Name, ""));
        }
    }

    ViewBag.CourseList = courseListForView;

    var listRolesUserForView = new List<string, string, string>();
    var userRolesList=DBService.SelectRolesByUserId(db,user.id).Select(x=>x.RoleId).ToList();
    var listRoles=DBService.SelectAllRoles(db);
    foreach (var t in listRoles)
    {
        if (userRolesList.Contains(t.Id)) listRolesUserForView.Add((t.Id, t.Name, "selected"));
        else listRolesUserForView.Add((t.Id, t.Name, ""));
    }

    ViewBag.UserRoles = listRolesUserForView;
}

```

Рисунок. 3.22 - Метод, який повертає інформацію користувача для редагування

Адміністратор може створювати та редагувати курси на сторінці «Існуючі курси». Для створення нового курсу адміністратор повинен натиснути кнопку «Створити», а для редагування вже існуючого курсу - кнопку «Змінити». Ці опції можна побачити на рисунках 3.23, 3.24, 3.25. Методи, які відповідають за створення та редагування курсів зображені на рисунках 3.26, 3.27.

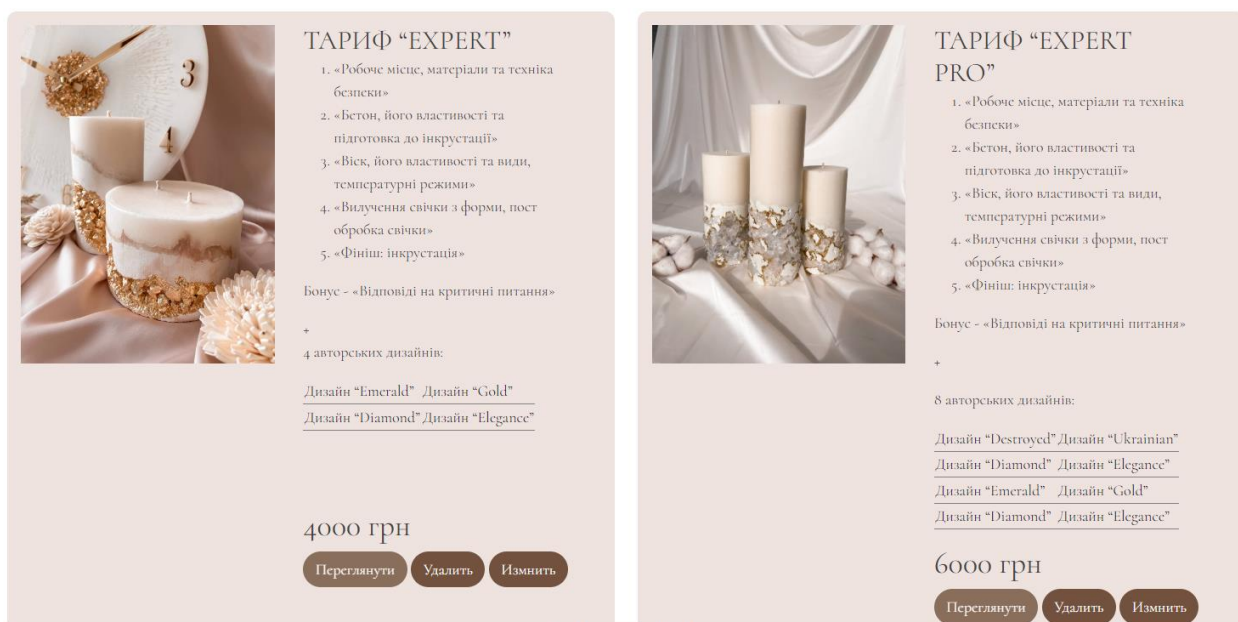


Рисунок.3.23 – Сторінка «Існуючі курси» для адміністратора

Create course

Name

Note

Price

Main Photo  Файл не выбран

Photo and Video  Файл не выбран

Level Access

[Main page](#)

Рисунок. 3.24 – Сторінка створення нового курсу

## Change Course " Тест "

Name  Note

Price

Main Photo

Photo and Video

Level access

[Main page](#)

Рисунок. 3.25 – Сторінка для змін вже існуючого курсу

```
0 references
public ActionResult CreateCourse()
{
    if (!User.IsInRole("Admin"))
        return RedirectToAction("CoursesList");

    //var user = DBService.FindUser(db, User.Identity.Name);

    ViewBag.CourseAccessLevel = DBService.SelectAccessLevels(db);

    return View("CreateCourse", new DB.Courses { id = Guid.NewGuid() });
}

[HttpPost]
0 references
public ActionResult CreateCourse(DB.Courses supply, IFormFile file, IList<IFormFile> filesList)
{
    if (!ModelState.IsValid) return View("CreateCourse");

    var guid = Guid.NewGuid();

    var course = new DB.Courses
    {
        id = guid,
        IsActive = supply.IsActive,
        Price = supply.Price,
        Name = supply.Name,
        Note = supply.Note,
    };

    var fff = Utils.UtilsData.FormFileToDB(file);
    course.MainPhoto = fff.Array;

    DBService.InsertNewCourse(course, db);

    var listCoursesFile = new List<CoursesFiles>();
    foreach (var f in filesList)
    {
        var fileT = new CoursesFiles();
        fileT.Courses_Id = guid;
        fileT.FileName = f.FileName;

        var ff = Utils.UtilsData.FormFileToDB(f);
        fileT.FileData = ff.Array;
    }
}
```

Рисунок. 3.26 - Метод для створення нового курсу

```

[HttpGet]
// [Authorize(Roles = "Admin")]
public ActionResult EditCourse(Guid id)
{
    ViewBag.CourseAccessLevel = DBService.SelectAccessLevels(db);
    var p = DBService.FindCourse(db, id);

    return View("EditCourse", p);
}

[HttpPost]
public ActionResult EditCourse(Courses courses, IFormFile file, IList<IFormFile> fileList)
{
    if (User.IsInRole("Admin"))
    {
        var id = courses.id;
        var course = DBService.FindCourse(db, id);
        try
        {
            if (file != null)
            {
                var fff = Utils.UtilsData.FormFileToDB(file);
                course.MainPhoto = fff.Array;
            }

            var price = courses.Price;

            course.Name = courses.Name;
            course.Note = courses.Note;
            course.IsActive = courses.IsActive;
            course.Price = Convert.ToDecimal(price);

            var res = DBService.UpdateCourse(course, db);
            if (fileList.Count() > 0 && fileList != null)
            {
                var listCoursesFile = new List<CoursesFiles>();
                foreach (var f in fileList)
                {
                    var fileT = new CoursesFiles();
                    fileT.Courses_Id = id;
                    fileT.FileName = f.FileName;

                    var ff = Utils.UtilsData.FormFileToDB(f);
                    fileT.FileData = ff.Array;
                }
            }
        }
    }
}

```

Рисунок. 3.27 - Метод для редагування вже існуючого курсу

Якщо закрити портал та відкрити портал знову то данні про вхід користувача в обліковий запис збережуться.

### 3.2 Розробка бази даних

Для веб застосунку була розроблена база даних, в якій зберігається вся інформація про користувачів, курси та покупців курсів. Структура бази даних складається з 7 таблиць:

- AspNetRoles;
- AspNetUsers;
- AspNetUserRoles;
- Buyers;
- Curses;
- Courses;
- CourseAccessLevels.

Діаграма структури бази даних - є графічним зображенням об'єктів бази даних та зв'язків між ними. Вона відображає структуру та організацію даних в базі даних та допомагає зрозуміти взаємозв'язки між об'єктами.

У діаграмі структури бази даних відображені таблиці, які відповідають таблицям бази даних. Кожна таблиця містить поля. Зв'язки між таблицями відображають взаємозв'язки між таблицями.

Діаграма структури бази даних є важливим інструментом для проектування та розробки баз даних. Вона дозволяє розуміти структуру та організацію даних. Діаграма структури бази даних може бути використана для узгодження замовником та розробниками структури, а також для документування бази даних та забезпечення її якості та стабільності.

Структуру бази даних розроблюваного порталу відображено на рисунок 3.28.

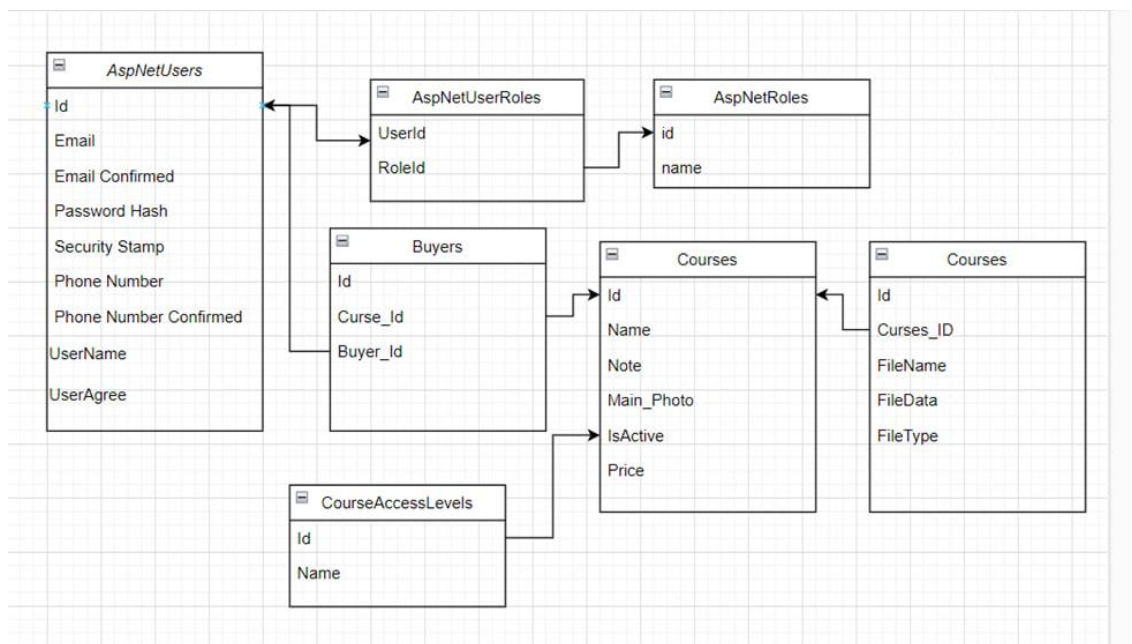


Рисунок. 3.28 - Структура бази даних для веб порталу

### AspNetUsers

В таблиці AspNetUsers зберігається інформація про користувачів, така як електронна пошта, пароль, ім'я користувача і т.д, структура та поля таблиці AspNetUsers відображена на рисунок. 3.29.



id (PK, nvarchar(128), He NULL)
Email (nvarchar(256), NULL)
EmailConfirmed (bit, He NULL)
PasswordHash (nvarchar(max), NULL)
SecurityStamp (nvarchar(max), NULL)
LockoutEndDateUtc (datetime, NULL)
LockoutEnabled (bit, He NULL)
AccessFailedCount (int, He NULL)
UserName (nvarchar(256), He NULL)
UserAgree (bit, He NULL)
NormalizedUserName (nvarchar(256), NULL)
NormalizedEmail (nvarchar(256), NULL)
LockoutEnd (datetimeoffset(7), NULL)
IsActive (bit, He NULL)
IsActiveComment (nvarchar(max), NULL)
ConcurrencyStamp (nvarchar(max), NULL)
PhoneNumber (nvarchar(max), NULL)
PhoneNumberConfirmed (bit, He NULL)
TwoFactorEnabled (bit, He NULL)
TelegramId (bigint, NULL)

Рисунок. 3.29 - Структура та поля таблиці AspNetUsers

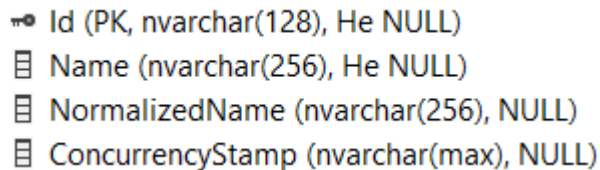
Таблиця AspNetUsers має 20 полів. Кожне поле відповідає за відповідну інформацію. Наприклад, поле PasswordHash відповідає за зберігання паролю в форматі хешу. Або поле EmailConfirmed відповідає, чи підтверджена електронна пошта користувача.

### **AspNetRoles**

В таблиці AspNetRoles зберігається інформація про ролі, які доступні у веб застосунку. Наразі існують тільки дві ролі:

- Користувач;
- Адміністратор.

Структура та поля таблиці AspNetRoles відображена на рисунок. 3.30.



- Id (PK, nvarchar(128), Not Null)
- Name (nvarchar(256), Not Null)
- NormalizedName (nvarchar(256), Null)
- ConcurrencyStamp (nvarchar(max), Null)

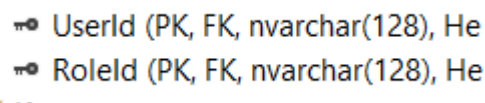
Рисунок. 3.30 - Структура та поля таблиці AspNetRoles

Поле *Id* - це ключ за яким зберігається ім'я ролі в полі *Name*.

Поле *Id* має тип даних *NVARCHAR*. *NVARCHAR* – тип даних, який зберігає інформацію в текстовому форматі. В полі *Id* ключ зберігається в форматі *GUID*.

### AspNetUserRoles

В таблиці AspNetUserRoles, зберігається інформація про ролі користувачів у веб-застосунку. Після того, як користувач проходить реєстрацію, для нього надається роль «Користувач», а дані записуються в таблицю AspNetUserRoles. Кожного разу коли користувач заходить в особистий профіль - запитом до таблиці, іде перевірка, яку роль має користувач. Структура та поля таблиці відображенні на рисунок .3.31.



- UserId (PK, FK, nvarchar(128), Not Null)
- RoleId (PK, FK, nvarchar(128), Not Null)

Рисунок. 3.31 - Структура та поля таблиці AspNetUserRoles

Таблиця AspNetUserRoles має два поля – *UserId* та *RoleId*.

У типі даних поля *UserId* - *NVARCHAR*, зберігається *GUID*-ключ користувача з таблиці AspNetUsers. Поле *RoleId* також містить *GUID*-ключ, що відповідає таблиці AspNetRoles. Крім того, поля *UserId* та *RoleId* є вторинними ключами для таблиць AspNetUsers та AspNetRoles відповідно. При записі інформації до таблиці AspNetUserRoles проводиться перевірка наявності відповідних даних в таблицях за допомогою ключів.

## Curses

Таблиця Curses містить дані про курс, такі як: назва, опис, ціна, головна фотографія і т.д. При створенні нового курсу адміністратор вводить необхідну інформацію, що стосується курсу, після чого, ці дані зберігаються в таблицю Curses. Потім, дана інформація відображається на сторінці, на якій розміщений список інших курсів, а також на сторінці самого курсу, структура та поля таблиці відображені на рисунок. 3.32.

```
id (PK, uniqueidentifier, He NULL)
Name (nvarchar(max), He NULL)
IsActive (int, He NULL)
Note (nvarchar(max), NULL)
Price (money, NULL)
MainPhoto (varbinary(max), NULL)
```

Рисунок. 3.32 - Структура та поля таблиці Curses

## Courses

В таблиці Courses зберігаються відео матеріали до самого курсу. При внесенні інформації про курс також вносяться навчальні відео матеріали самого курсу, які будуть відображатися у користувача після покупки курсу, структура та поля таблиці відображенні на рисунок. 3.33.

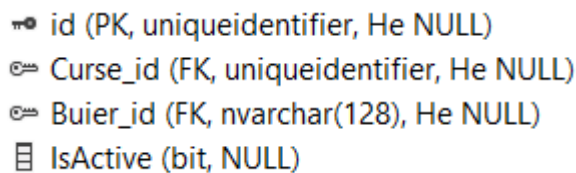
```
id (PK, uniqueidentifier, He NULL)
Courses_id (FK, uniqueidentifier, He NULL)
FileName (nvarchar(max), NULL)
FileData (varbinary(max), NULL)
FileType (varchar(10), NULL)
```

Рисунок. 3.33 - Структура та поля таблиці Courses

В таблиці зберігається інформація про ім'я файлу відео, поле – FileName, сам файл у форматі VARBINARY - FileData та тип, по якому можна буде швидко знайти відповідний запис – FileType.

## Buyers

В таблиці Buyers зберігається інформація про користувачів, які купили відповідний курс. При покупці курсу після проходження оплати в таблицю записуються дані користувача та курсу, структура та поля таблиці відображенні на рисунок. 3.34.



- id (PK, uniqueidentifier, Not Null)
- Course\_id (FK, uniqueidentifier, Not Null)
- Buyer\_id (FK, nvarchar(128), Not Null)
- IsActive (bit, Null)

Рисунок. 3.34 - Структура та поля таблиці Buyers

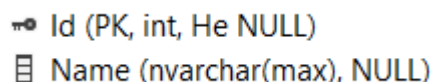
Поля Course\_id та Buyer\_Id зберігають ключі курсів та користувачів.

Поле Course\_id зберігає інформацію про курс. В цьому полі зберігається ключ, який має курс та при записі інформації до таблиці Buyers перевіряється - чи існує курс з відповідним ключем в таблиці Course.

Поле Buyer\_Id зберігає інформацію про користувача. В цьому полі зберігається ключ користувача, який записується при реєстрації нового користувача. При записі інформації до таблиці Buyers перевіряється чи існує користувач з відповідним ключем у таблиці ASPNETUsers.

## CourseAccessLevels

Таблиця CourseAccessLevels зберігає інформацію про тип доступу користувача до курсу. Тобто, буде курс відображений для користувача чи ні. При створенні нового курсу, адміністратор обирає відповідний тип з таблиці CourseAccessLevels, який буде записано до таблиці Course. у поле IsActive, структура та поля таблиці відображенні на рисунок. 3.35.



- Id (PK, int, Not Null)
- Name (nvarchar(max), Null)

### 3.2 Тестування веб застосунку

Тестування веб-додатків - це процес перевірки, чи відповідає програма, 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.

Функціональне тестування програмного забезпечення оцінює функції, функціональність та потоки користувачів програми з метою переконатися, що вони працюють належним чином. Цей тип тестування включає перевірку різних компонентів програми, щоб гарантувати їх правильну роботу та відповідність вимогам. Функціональне тестування може виявляти помилки у коді, відсутність функцій або неправильні розрахунки, що допомагає забезпечити належну функціональність кінцевого продукту. Таке тестування є обов'язковим етапом у всіх процесах розробки програмного забезпечення, яке гарантує, що програма працює належним чином та задовольняє потреби користувачів.

Операційне тестування є видом тестування програмного забезпечення, яке має на меті перевірити продуктивність та надійність програми в різних умовах експлуатації. В рамках цього виду тестування тестують здатність програми працювати з різними навантаженнями, обсягами трафіку та умовами навколишнього середовища, щоб забезпечити її оптимальну роботу.

Операційне тестування дозволяє виявити проблеми, пов'язані з продуктивністю програми, наприклад, повільним часом відгуку, збоями або іншими помилками, які можуть вплинути на взаємодію користувача. Проведення тестування дозволяє розробникам переконатися, що програма може впоратися з очікуваним навантаженням та працювати надійно у різних умовах експлуатації.

Тестування на безпеку є одним з типів тестування програмного забезпечення, який має на меті оцінити функції безпеки та виявити вразливості програми. Під час такого тестування перевіряється наявність потенційних загроз безпеці програми та вразливостей, таких як SQL-ін'єкції, міжсайтові скрипти та інші порушення безпеки, які можуть призвести до компрометації безпеки програми.

Тестування безпеки може допомогти виявити проблеми безпеки додатка до його випуску, запобігти витоку даних, системним збоям та іншим серйозним інцидентам безпеки, які можуть завдати шкоди репутації організації та призвести до фінансових втрат. Розробники, проводячи тестування на безпеку програми, можуть переконатися, що вона захищена від потенційних загроз та вразливостей.

Регресійне тестування – це тип тестування програмного забезпечення, який перевіряє, що внесені зміни або оновлення до програми не мають впливу на її функціональність. Для цього повторно тестуються функції або можливості програми, які були вже протестовані, щоб переконатися, що вони працюють належним чином після внесення будь-яких змін або оновлень.

Регресійне тестування допомагає виявляти дефекти або проблеми, які можуть виникнути в додатку через зміни або оновлення, внесені у процесі розробки. Тестуючи раніше перевірені можливості або функції програми, розробники можуть переконатися, що вони продовжують працювати належним чином, та що загальна якість та функціональність програми не постраждають.

Найкращі методи тестування веб додатків:

- Для ефективного тестування веб додатків необхідно встановити чітку стратегію та цілі тестування перед початком тестів.
- Тестування інтерфейсу користувача потрібно проводити на реальних браузерах, а не тільки на емуляторах або симуляторах.
- Тестування слід проводити як на початку процесу розробки, так і поетапно, щоб виявити проблеми на ранній стадії.
- Важливо встановлювати пріоритетність певних пристроїв та браузерів для тестування, щоб забезпечити практичність.

- Для забезпечення повноцінного тестування необхідно проводити достатню кількість тестів на продуктивність, навантаження та безпеку.

### 3.4 Тестування веб застосунку для онлайн курсів

Таблиця 3.1 = Тестування веб застосунку

<b>ID</b>	<b>Назва</b>	<b>Кроки</b>	<b>Очікуваний результат</b>	<b>Реальний результат</b>
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 «Придбати курс» на головній сторінці	1. Відкриття порталу. 2. Натискання на кнопку «Придбати курс» на головній сторінці	Перехід на сторінку зі списком курсів	Застосунок перенаправляє користувача на сторінку зі списком курсів
4	Перевірка кнопки «Придбати курс» в правому верхньому куті	1. Відкриття порталу 2. Натискання на кнопку «Придбати курс»	Перехід на вкладку с курсами	Застосунок перенаправляє на сторінку зі списком курсів.
5	Перевірка кнопки «Головна»	1. Відкриття порталу. 2. Натискання на кнопку «Головна»	Відкриття головної сторінки порталу.	Застосунок перенаправляє на головну сторінку порталу.



6	Перевірка кнопки «Зареєструватися»	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Натиснути кнопку «Зареєструватися»</li> </ol>	Відкриття сторінки для реєстрацій	Застосунок перенаправляє на сторінку для реєстрацій
7	Введення даних для реєстрації на вкладці «Зареєструватися»	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Відкрити вкладка “Зареєструватис”</li> <li>3. Ввести поштову скриньку, пароль та підтвердити пароль</li> </ol>	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	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Натискання на кнопку «Увійти»</li> <li>3. Ввести данні</li> </ol>	При введенні не правильних даних повинно з’явитися повідомлення про некоректні	При введенні некоректних даних з’являється повідомлення про не вірні

		для входу в свій особистий профіль	дані. При введенні коректних даних, повинно увійти в профіль користувача	данні. 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	Перехід на головну сторінку порталу	Перехід на головну сторінку.

		<p>3. Увійти в особистий профіль</p> <p>4. Вийти з особистого профілю</p>		
11	Перевірки відображення списку курсів для покуп	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p>	<p>Відображення списку курсів які доступні для покупки.</p> <p>Відображення картинка курсу.</p>	<p>Після переходу на сторінку с курсами.</p> <p>Відображається список курсів та картинка курсу</p>
12	Перевірка вмісту курсу	<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>	<p>Відображення інформацій стосовно курсу та головної картинки</p>	<p>Відображення інформацій стосовно курсу та головної картинки</p>
13	Перевірка кнопки «Перейти до оплати» на сторінці курсу	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється</p>	<p>З'являється сторінка для покупки курсу, де відображено інформацію про</p>	<p>З'являється сторінка для покупки курсу, де відображено інформацію про</p>

		<p>головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p> <p>4. Натиснути на картинку курсу зі списку</p> <p>5. Натиснути кнопку «Перейти до оплати»</p>	<p>курсу, назву та ціну</p>	<p>курсу, назву та ціну</p>
14	<p>Перевірка вкладки «Мої курси»</p>	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку «Мої курси»</p>	<p>Перехід на сторінку де відображається список придбаних курсів.</p>	<p>Перехід на сторінку де відображається список придбаних курсів.</p>
15	<p>Перевірка відображення інформацій курсу після покупки на вкладці «Придбати курс»</p>	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p>	<p>Перехід на сторінку з відображенням інформації курсу в відео форматі</p>	<p>Перехід на сторінку з відображенням інформації курсу в відео форматі</p>

		4. Натиснути на картинку курсу зі списку		
16	Перевірка відображення інформацій курсу після покупки на вкладці «Мої курси»	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Зачекати поки відкриється головна сторінка порталу</li> <li>3. Перейти на вкладку «Мої курси»</li> <li>4. Натиснути на картинку курсу зі списку</li> </ol>	Перехід на сторінку з відображенням інформації курсу в відео форматі	Перехід на сторінку з відображенням інформації курсу в відео форматі.
17	Перевірка переключення між відео інформацією	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Зачекати поки відкриється головна сторінка порталу</li> <li>3. Перейти на вкладку «Мої курси»</li> <li>4. Натиснути на картинку курсу зі списку</li> </ol>	Відображення нового відео.	Відображення нового відео.

		5. Переключити на вкладці з інформацію на інше відео		
18	Перевірка відкриття сторінки для створення нового курсу	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Зачекати поки відкриється головна сторінка порталу</li> <li>3. Перейти на вкладку з курсами</li> <li>4. Натиснути кнопку «Створити»</li> </ol>	Відкривається сторінка для створення нового курсу.	Відкривається сторінка для створення нового курсу.
19	Перевірка введення інформацій для створення нового курсу	<ol style="list-style-type: none"> <li>1. Відкриття додатку</li> <li>2. Зачекати поки відкриється головна сторінка порталу</li> <li>3. Перейти на вкладку з курсами</li> <li>4. Натиснути кнопку «Створити»</li> <li>5. Ввести всі необхідні данні</li> </ol>	Створення нового курсу.	Створення нового курсу.

		для створення нового курсу 6. Натиснути кнопку створити		
20	Перевірка рівня доступності «Відображати для адміністратора» під час створення курсу	1. Відкриття додатку 2. Зачекати поки відкриється головна сторінка порталу 3. Перейти на вкладку з курсами 4. Натиснути кнопку «Створити» 5. Ввести всі необхідні данні для створення нового курсу 6. Обрати «Відображати для адміністратора». 7. Натиснути кнопку «Створити»	Після створення нового курсу з рівнем доступу «Відображати для адміністратора». Новий курс не буде відображатися в списку курсу в користувачів окрім адміністратора	Після створення нового курсу з рівнем доступу «Відображати для адміністратора». Новий курс не буде відображатися в списку курсу в користувачів окрім адміністратора
21	Перевірка рівня доступності «Відкритий для	1. Відкриття додатку 2. Зачекати поки відкриється	Після створення нового курсу з рівнем доступності	1. Відкриття додатку 2. Зачекати поки відкриється

	всіх» під час створення курсу	<p>головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p> <p>4. Натиснути кнопку «Створити»</p> <p>5. Ввести всі необхідні данні для створення нового курсу</p> <p>6. Обрати «Відкритий для всіх».</p> <p>7. Натиснути кнопку «Створити»</p>	«Відкритий для всіх», новий курс буде відображатися у всіх користувачів.	<p>головна сторінка порталу</p> <p>3. Перейти на вкладку з курсами</p> <p>4. Натиснути кнопку «Створити»</p> <p>5. Ввести всі необхідні данні для створення нового курсу</p> <p>6. Обрати «Відображати для адміністратора».</p> <p>7. Натиснути кнопку «Створити»</p>
22	Надання користувача прав адміністратора	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з користувачами</p>	Після входу користувача в особистий профіль у користувача з'являться права адміністратора.	Після входу користувача в особистий профіль у користувача з'являться права адміністратора



		<p>4. На певному користувачі натиснути кнопку «Змінити»</p> <p>5. Перехід на сторінку з даними про користувача.</p> <p>6. Вибір ролі адміністратора для користувача</p> <p>7. Натиснути кнопку «Зберегти»</p>		
23	Надання користувача прав користувача	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з користувачами</p> <p>4. На певному користувачі натиснути кнопку «Змінити»</p> <p>5. Перехід на сторінку з даними про користувача.</p>	Після входу користувача в особистий профіль у користувача з'являться права користувача.	Після входу користувача в особистий профіль у користувача з'являться права користувача.

		<p>6. Вибір ролі користувача для користувача</p> <p>7. Натиснути кнопку «Зберегти»</p>		
24	Відкриття нових курсів користувачу	<p>1. Відкриття додатку</p> <p>2. Зачекати поки відкриється головна сторінка порталу</p> <p>3. Перейти на вкладку з користувачами</p> <p>4. На певному користувачі натиснути кнопку «Змінити»</p> <p>5. Перехід на сторінку з даними про користувача.</p> <p>6. Вибір курсу для користувача</p> <p>7. Натиснути кнопку «Зберегти»</p>	<p>Після входу користувача в особистий профіль у користувача з'являться новий курс в списку курсів на вкладці «Мої курси», який для нього був відкритий.</p>	<p>Після входу користувача в особистий профіль у користувача з'являться новий курс в списку курсів на вкладці «Мої курси», який для нього був відкритий.</p>

## ВИСНОВКИ

У дипломній роботі була розроблена серверна частина веб додатку для онлайн курсів «Виготовлення декору для дому». Розроблена серверна частина виконує важливі функції, необхідні для ефективного функціонування сайту і надання якісних послуг користувачам.

Проведено аналіз ринку та потенційних користувачів онлайн застосунку. Визначено основні проблеми з якими зіштовхуються користувачі при роботі з іншими онлайн платформами.

Проведено аналіз існуючих програм, які забезпечують онлайн курси. Порівняно їх функціональні можливості та інші характеристики. Виявлено переваги та недоліки проаналізованих аналогів.

Розроблена архітектура онлайн застосунку, включаючи базу даних та логіку взаємодій між серверною частиною та інтерфейсом користувача. Реалізовано відображення списку доступних курсів та інформацію про них, покупку курсів та відображення відеоматеріалів курсів. Забезпечено можливість авторизації та аутентифікації користувачів.

Проведено тестування функціоналу та тестування безпеки сервісу.

Дана дипломна робота відповідає поставленим цілям та завданням.

Ця дипломна робота дала мені цінний досвід у розробці серверних частин веб-додатків та підвищила мої навички у використанні сучасних інструментів та технологій.

## 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. Coursera- [Електронний ресурс]. – Режим доступу:  
<https://www.coursera.org/>
7. khan academy- [Електронний ресурс]. – Режим доступу:  
<https://www.khanacademy.org/>
8. EdX- [Електронний ресурс]. – Режим доступу:  
<https://www.edx.org/>
9. 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/muL3rq9M7IS/edit>
11. All You Need to Know About UML Diagrams [Електронний ресурс] – Режим доступу до ресурсу: <https://tallyfy.com/uml-diagram/>

12. Методи дослідження в дипломній роботі [Електронний ресурс] – Режим доступу до ресурсу: <https://diplom.com.ua/uk/metodi-doslidzhennya-v-diplomnij-roboti/>

13. What is Web Testing [Електронний ресурс] – Режим доступу до ресурсу: <https://katalon.com/resources-center/blog/types-of-web-testing>

14. The Basics of Web App Test Automation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.leapwork.com/blog/web-application-testing-the-basics-of-web-app-test-automation>

## Додаток А



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ  
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ  
ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### Проектування та розробка онлайн-курсу "Виготовлення декору для дому". Спец частина: Розробка backend-частини мовою C# на платформі ASP.NET Core та Razor

Виконав студент 4 курсу  
групи ПД-42  
Каліш Владислав В'ячеславович  
Керівник роботи

К.т.н, доц, доцент кафедри ІПЗ Яскевич Владислав Олександрович  
Київ – 2023

### МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета** – поліпшення процесу навчання з виготовлення декору для дому, шляхом розробки та реалізації серверної частини онлайн-курсу з використанням мови C# на платформі ASP.NET Core.
- **Об'єкт** – процес навчання з виготовлення декору для дому за допомогою онлайн-платформи та серверного додатку.
- **Предмет** – розробка серверної частини для забезпечення доступності та функціональності онлайн-курсу з виготовлення декору для дому.

---

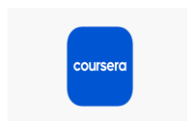
## ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Провести аналіз існуючих аналогів веб додатків. Функціональні можливості аналогічних програмних забезпечень, виявити переваги та недоліки кожного аналога.
2. Спроектувати та розробити онлайн застосунок для онлайн курсів.
3. Провести тестування та відладку розробленого продукту.

---

3

### АНАЛІЗ АНАЛОГІВ



Khan Academy

Характеристики	edX	Khan Academy	Coursera	<b>Розроблений портал</b>
Швидкий відгук програми	5,7	8	8	<b>5.9</b>
Захищенні данні користувачів	"+"	"-"	"+"	<b>"+"</b>
Зберігання кукі файлів користувача	"-"	"+"	"-"	<b>"+"</b>

4

## ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1. Система має забезпечувати реєстрації користувачів, їх авторизацію та аутентифікацію з використанням засобів ASP.NET Core Identity.
2. Система має забезпечувати додавання, редагування та видалення користувачів, а також робити їхню інформацію доступною для перегляду.
3. Система має забезпечувати додавання, редагування та видалення матеріалів курсу, таких як уроки, завдання, тести тощо.
4. Система має забезпечувати зберігання та взаємодії з базою даних для зберігання користувачів, матеріалів курсу та інших даних.

5

## ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

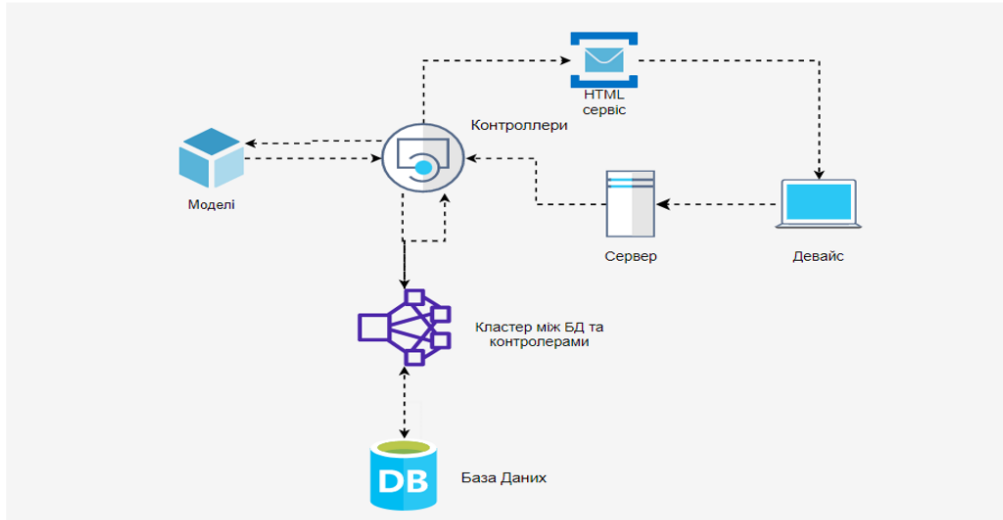


---

6

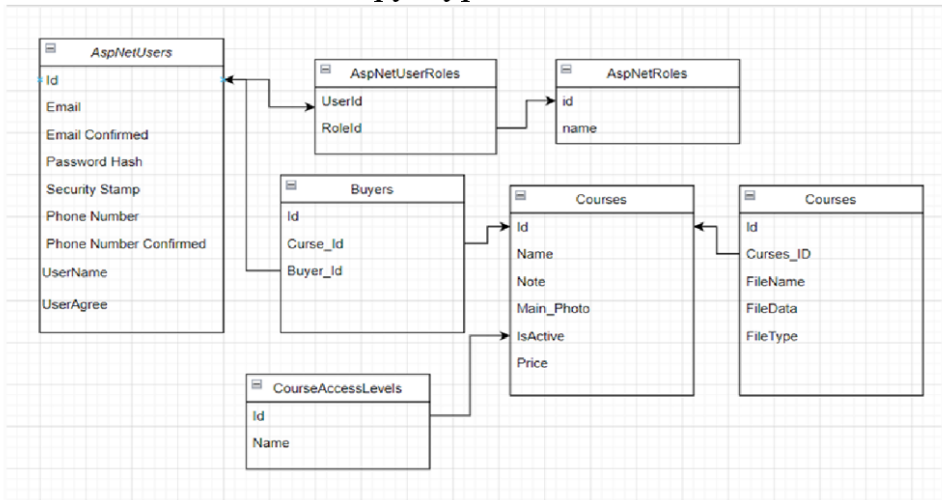


## Діаграма архітектури



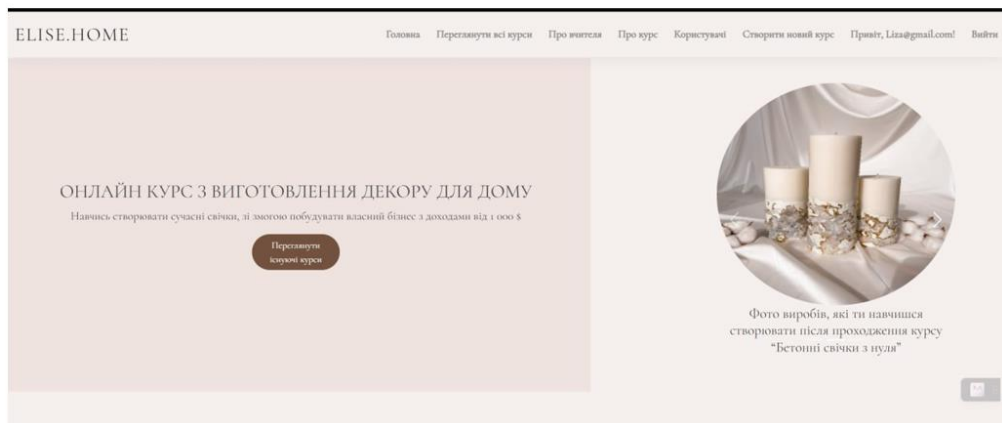
7

## Структура бази даних



8

# ЕКРАННІ ФОРМИ



Вигляд головної сторінки, для користувача з роллю адміністратора

9

# ЕКРАННІ ФОРМИ

The screenshot displays a login form titled 'ВХІД'. A red error message, 'Не вірно введені данні', is shown above the input fields. The form contains two input fields: 'Введіть логін/ім'я користувача:' with the value 'Liza@gmail.com' and 'Введіть пароль:'. A dark brown button labeled 'Увійти' is located at the bottom of the form.

Вигляд форми входу, при помилково введених даних, або їх відсутності

10

# ЕКРАННІ ФОРМИ

Сторінка для створення нового курсу на порталі

11

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. [Каліш В.В. Фреймворки Asp.Net Core та Java Spring / В.О. Яскевич, В.В. Каліш // Застосування програмного забезпечення в інфокомунікаційних технологія. Збірник тез. 20.04.2023, ДУТ, м. Київ – К.: ДУТ, 2023. – С.54](#)
2. [Каліш В.В. Значення back-end розробки в успіху сучасних веб-додатків / В.О. Яскевич, В.В. Каліш // III Всеукраїнська Науково-практична конференція «Сучасні інтелектуальні інформаційні технології в науці та освіті». Подано до друку](#)

12

## ВИСНОВКИ

1. Проведено аналіз ринку та потенційних користувачів онлайн застосунку. Визначено основні проблеми з якими зіштовхуються користувачі при роботі з іншими онлайн платформами.
2. Проведено аналіз існуючих програм, які забезпечують онлайн курси. Порівняно їх функціональні можливості та інші характеристики. Виявлено переваги та недоліки проаналізованих аналогів.
3. Розроблена архітектура онлайн застосунку, включаючи базу даних та логіку взаємодій між серверною частиною та інтерфейсом користувача. Реалізовано відображення списку доступних курсів та інформацію про них, покупку курсів та відображення відеоматеріалів курсів. Забезпечено можливість авторизації та аутентифікації користувачів.
4. Проведено тестування функціоналу та тестування безпеки сервісу.

**ДЯКУЮ ЗА УВАГУ!**