

Київський національний торговельно-економічний університет
Кафедра інженерії програмного забезпечення та кібербезпеки

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

Розробка веб-порталу сучасної поезії

Студента 4 курсу, 6 групи,
спеціальності 121 «Інженерія
програмного забезпечення»

Тарасова Дмитра
Володимировича

підпис студента

Науковий керівник
кандидат технічних наук,
доцент

Котенко Наталія
Олексіївна

підпис керівника

Гарант освітньої програми
кандидат технічних наук,
доцент

Цензура Микола
Олександрович

підпис керівника

КИЇВ – 2021

4. Консультанти роботи із зазначенням розділів, які консультують:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв

5. Зміст випускної кваліфікаційної роботи (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ

Теоретичні дані

Відомості про бренд

Технічне завдання

Висновок до розділу 1

РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ

Архітектура вебдодатку

Веб-Фреймворк ASP.NET

Структура файлів проекту

Структура системи

Висновок до розділу 2

РОЗДІЛ 3 РОЗРОБКА ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ

Середовище розробки

Технічні особливості

Висновок до розділу 3

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

6. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Строк виконання етапів роботи	
		за планом	фактично
1	2	3	4
1.	<i>Вибір теми випускної кваліфікаційної роботи</i>	21.09.2020	21.09.2020
2.	<i>Вступ та перелік літературних джерел</i>	14.12.2020	14.12.2020
3.	<i>Розділ 1. «ЗАГАЛЬНІ ВІДОМОСТІ»</i>	19.02.2021	19.02.2021
4.	<i>Розділ 2. «ПРОЕКТУВАННЯ ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ»</i>	05.03.2021	05.03.2021
5.	<i>Розділ 3. «РОЗРОБКА ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ»</i>	10.04.2021	10.04.2021
6.	<i>Висновки</i>	24.04.2021	24.04.2021
7.	<i>Здача випускної кваліфікаційної роботи на кафедрі (перша перевірка)</i>	14.05.2021	14.05.2021
8.	<i>Підготовка автореферату та презентації доповіді</i>	17.05.2021	17.05.2021
9.	<i>Попередній захист випускної кваліфікаційної роботи</i>	18.05.2021 - 21.05.2021	
10.	<i>Зовнішнє рецензування випускної кваліфікаційної роботи</i>	01.06.2021	
11.	<i>Здача прошого випускної кваліфікаційної роботи на кафедрі</i>	02.06.2021	
12.	<i>Публічний захист випускної кваліфікаційної роботи</i>		

7. Дата видачі завдання «20» жовтня 2020 р.

8. Науковий керівник випускної кваліфікаційної роботи

Котенко Н. О.

(прізвище, ініціали, підпис)

9. Гарант освітньої програми Цензура М. О.

(прізвище, ініціали, підпис)

10. Завдання прийняв до виконання студент Тарасов Д. В.

(прізвище, ініціали, підпис)

АНОТАЦІЯ

Дипломна робота на тему «Розробка веб-порталу сучасної поезії» містить 46 сторінок та 15 рисунків. Перелік списку використаних джерел налічує 10 посилань.

Мета дослідження: розробка веб-порталу сучасної поезії за допомогою мови гіпертекстової розмітки HTML, каскадної таблиці стилів CSS, розробка функціоналу вебдодатку за допомогою фреймворка ASP.NET. Дана система буде працювати в рамках певного підприємства і виконувати функції основного майданчика для розміщення творів.

Об'єкт дослідження: веб-портал сучасної поезії.

Предметом дослідження є технології розробки веб-порталу сучасної поезії.

В першому розділі представлено загальний огляд веб-порталу, проаналізовано веб-сайти брендів, що вже відомі на міжнародному ринку, їх переваги та недоліки. Також було розроблено технічне завдання з функціями, що мають бути реалізовані на веб-сайті та розроблені макети сторінок.

У другому розділі визначено напрямки та мови розробки веб-порталу, описано розробку проекту, представлено UML-діаграми послідовності, класів, діяльності та станів, які відображають взаємодію користувача з веб-сайтом.

У третьому розділі описано технічні аспекти створення інтерфейсу розробленого веб-порталу сучасної поезії, представлено фрагменти коду реалізації основних структурних одиниць проекту.

ANNOTATION

The thesis on the topic “Website development for modern art hosting” contains 46 pages and 15 drawings. The list of used sources includes 10 links.

The purpose of the study: analyze advantages and disadvantages of already existing art websites, develop a technical task with website layouts, UML diagrams of sequences, classes, activities and states, develop a website for hosting art objects.

Research subject: modern art webportal.

The study's subject is technology for developing a web portal of modern art.

The first section provides an overview of the web portal, analyzes the websites of brands that are already known in the international market, their advantages and disadvantages. A terms of reference were also developed with features to be implemented on the website and page layouts developed.

The second section defines the direction and languages of web portal development, describes project development, presents UML diagrams of sequences, classes, activities and states that show the user's interaction with the website.

The third section describes the technical aspects of creating the interface of the developed web portal of modern poetry, presents fragments of the code of implementation of the basic structures of the project units.

ЗМІСТ

ВСТУП.....	Error! Bookmark not defined.
РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ.....	Error! Bookmark not defined.
1.1 Теоретичні дані	Error! Bookmark not defined.
1.2 Відомості про бренд.....	Error! Bookmark not defined.
1.3 Технічне завдання	Error! Bookmark not defined.
1.4 Висновок до розділу 1.....	Error! Bookmark not defined.
РОЗДІЛ 2 ПРОЕКТУВАННЯ ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ.....	Error! Bookmark not defined.
2.1 Архітектура вебдодатку.....	Error! Bookmark not defined.
2.2 Веб-Фреймворк ASP.NET.....	Error! Bookmark not defined.
2.3 Структура файлів проекту	Error! Bookmark not defined.
2.4 Послідовність роботи системи	Error! Bookmark not defined.
2.5 Структура системи	Error! Bookmark not defined.
2.6 Висновок до розділу 2.....	Error! Bookmark not defined.
РОЗДІЛ 3 РОЗРОБКА ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ.....	Error! Bookmark not defined.
3.1 Середовище розробки	Error! Bookmark not defined.
3.2 Технічні особливості.....	Error! Bookmark not defined.
3.3 Висновок до розділу 3.....	Error! Bookmark not defined.
ВИСНОВКИ ТА ПРОПОЗИЦІЇ.....	Error! Bookmark not defined.
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	Error! Bookmark not defined.
ДОДАТКИ	Error! Bookmark not defined.

					<i>КНТЕУ 121 06-23.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>	<i>Розробка веб-порталу сучасної поезії</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		<i>20.10.2020</i>		<i>Зміст</i>	<i>2</i>	<i>46</i>
<i>Керівник</i>		<i>Котенко Н.О.</i>		<i>20.10.2020</i>		<i>Зміст</i>	Факультет інформаційних технологій, 4 курс, 6 група	
<i>Гарант</i>		<i>Жирова Т.О.</i>		<i>20.10.2020</i>				
<i>Розроб.</i>		<i>Тарасов Д.В.</i>		<i>20.10.2020</i>				

ВСТУП

На початку століття у повсякденне життя увійшла інфраструктура, яка дозволяє замовляти щотижнево товари з магазину, ділитися своїм життям у Facebook, надсилати електронною поштою листи та шукати найменшу кішку у світі. Це все завдяки інтернету. Одним із показників є кількість інформації, яка проходить через всесвітню павутину: близько п'яти екзобайт на день. Це еквівалентно 40 000 двогодинних фільмів із стандартною роздільною здатністю в секунду. Інтернет, який пов'язує мільярди людей у всьому світі, є основною опорою сучасного інформаційного суспільства. Глобальний рівень проникнення Інтернету сягає 60 відсотків, так станом на січень 2021 року в усьому світі було 4,66 мільярда активних користувачів Інтернету - 59,5% світового населення. І з загальної кількості 92,6 відсотка (4,32 млрд.) отримали доступ до Інтернету через мобільні пристрої. Також досить суттєву роль зіграла пандемія Covid-19, що призвела до неминучого сплеску використання цифрових технологій через норми соціального дистанціювання та загальнодержавні блокування. Людям та організаціям у всьому світі довелося пристосовуватися до нових способів роботи. На цьому фоні стає зрозуміло, що сьогодні більшість аспектів життя людини переходить в онлайн і робить інтернет найбільшим маркетплейс, мітингрум та культурним простіром. Світ мистецтва не є виключенням і також терміново переходить в Інтернет. Багато художніх закладів, таких як Лувр, Тейт і Мет, закрили свої двері, але завантажили величезну кількість вмісту в мережу, так як це дійсно вигідно і зручно для відвідувача, не кажучи про економію бюджету і часу. Веб-портал працює цілодобово і в автоматичному режимі без участі обслуговуючого персоналу та утримання величезних просторів, що є істотною економією і також приносить прибуток з досить простою бізнес моделлю – через хостинг реклами.

					<i>КНТЕУ 121 06-23.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.		14.12.2020	<i>Розробка веб-порталу сучасної поезії</i>	Стадія	Аркуш	Аркушів
Керівник		Котенко Н.О.		14.12.2020		В	3	46
Гарант		Цензура М.О.		14.12.2020		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Тарасов Д.В.		14.12.2020				
<i>Вступ</i>								

У порівнянні зі звичайним артплейсом, територія охоплення якого обмежується населенням міста або району, територія охоплення веб-порталу збільшується на всю Україну і українську аудиторію в інших країнах, або маштабується взагалі на весь світ з доповненням веб-порталу англійською версією.

На даний момент існує багато веб-порталів мистецтва. У кожного з них є красиве оформлення, достатньо багатий вибір різноманітних творів. Але в них є достатньо мінусів:

- постійна реклама (при вході на сторінку, на задньому фоні сторінки, при натисканні на будь-яку невідому кнопку);
- недостатня інформація про твір;
- відсутність будь-якого функціоналу, наприклад як додавання оцінки або стислого коментарю у формі тега;
- більшість сайтів неоптимізована, через що користувач змушений довго чекати завантаження сайту або навіть спостерігати пусту сторінку.

Так як в час розповсюдження коронавірусу веб-портали стали єдиною альтернативою перегляду різноманітних творів митців, таких як вірші, вирази, цифрові зображення, діджитал-інсталяції та інше, тема, обрана автором, є досить актуальною.

Мета випускної кваліфікаційної роботи: розробка веб-порталу сучасної поезії за допомогою мови гіпертекстової розмітки HTML, каскадної таблиці стилів CSS, розробка функціоналу вебдодатку за допомогою фреймворка ASP.NET. Дана система буде працювати в рамках певного підприємства і виконувати функції основного майданчика для розміщення творів.

Об'єктом дослідження є веб-портал сучасної поезії.

Предметом дослідження є технології розробки веб-порталу сучасної поезії.

Зм.	Аркуш	№ докум	Підпис	Дата

Завдання дослідження :

- аналіз веб-сайтів присвячених поезії та мистецтву;
- розробка технічного завдання;
- розробка макетів веб-порталу;
- розробка діаграм роботи веб-порталу;
- розробка бази даних порталу;
- розробка інтерфейсу веб-порталу.

Зм.	Аркуш	№ докум	Підпис	Дата

РОЗДІЛ 1 ЗАГАЛЬНІ ВІДОМОСТІ

1.1 Теоретичні дані

Веб-портал сучасного мистецтва та поезії - це прикладна система, побудована з використанням останніх веб технологій та архітектурних норм, для хостингу новітніх художніх та літературних творів. Подібно до звичайної галереї, електронний сервіс реалізує наступні основні функції: представлення арт об'єктів відвідувачу, обробку та збереження його оцінки та відгуку.

Основні проблеми реалізації інтернет-галереї лежать на стику технологій Інтернету і традиційного арт-центру. У арт-центрі відвідувачі звикли до того, що є можливість оглянути твір візуально, роздивитись, вийти в діалог з інсталяцією, отримати нативний досвід. В електронній версії він такої можливості позбавлений. Найчастіше візуальної інформації досить, але тут вступають в дію емоційні і психологічні чинники. Часто існують проблеми з сприйняттям творів, особливо якщо їх послання не досить зрозуміла.

Переваги хостингів поезії : Інтернет-галерея багато в чому виграє перед реальним музеєм або виставкою. Зменшується чисельність персоналу за рахунок скорочення обсягу взаємодії з відвідувачами, оренда дискового простору і розміщення «електронної експозиції» дешевше і простіше оренди виставкових приміщень або в утримання власних просторів. Досить часто фізичні виставки або арт-простори знаходяться у центрі міста і займають достить великий спейс, що обхдиться надто дорого в фінансуванні, будь-то за кошт спонсорів або міста.

Основні переваги Інтернет-порталів для споживачів - економія часу. Людині яка працює 6 днів на тиждень з 10 до 19,шноді не буває часу щоб відвідати заклади культури.

					<i>КНТЕУ 121 06-23.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.		19.02.2021	Розробка веб-порталу сучасної поезії	Стадія	Аркуш	Аркушів
Керівник		Котенко Н.О.		19.02.2021		P1	6	46
Гарант		Цензура М.О.		19.02.2021		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Тарасов Д.В.		19.02.2021				
					ТЕОРИТИЧНІ ВІДОМОСТІ			

Останнім часом в інтернет мережі стрімко набирають популярність так звані біржі, де можна придбати якийсь твір будь-то фоторафія, найяскравіший приклад Shutterstock чи трьохмірне зображення, так звана модель якоїсь сутності, які можуть використовуватися в ігровій індустрії при розробці ігор або при візуалізації будь-яких об'єктів, наприклад для окулярів доповненої реальності, як наприклад cgtrader.

По типу монетизації:

- біржа виступає вендором;
- митці виступають вендорами.

Біржа може мати в масиві тільки власні моделі, створені власним департаментом або притбані у підрядників або портал може бути маркетплейсом який сполучає підрядчика, іншими словами фрілансера, який виставляє на цифрову вітрину свої роботи за певну ціну. Також таким чином фрілансер може пропонувати свою роботу і розміщати на сайті своє портфоліо.

1.2 Відомості про бренд

При виборі назви інтернет платформи слід враховувати такі фактори:

- назва не має бути плагіатом;
- вона має бути лаконічною та якнайкраще передавати суть та філософію хостингу;
- назва має добре запам'ятовуватися.

“Photoalbum” – ідеально підходить оскільки за своєю сутністю одиницею на платформі буде зображення, іншими більш зрозумілими всім словами фотокартки, на яких можуть бути як вислови відомих людей так і живопис.

Наповнення порталу : спектром охоплення альбому є насправді досить велика множина різноманітних зображень, оскільки тема хостингу досить загальна та включає в себе досить великий масив сенсів, бо мистецтво для кожного може бути різним і сприйматися порізному.

Кожен митець бачить, відчуває, створює живе різним і розуміє мистецтво інакше за усіх. Це пов'язано з тим, що кожен має унікальний досвід, унікальні обставини та оточення в якому він ріс та знаходиться; відчуває інакші емоції у конкретний момент; на нього впливають унікальні обставини та призвели до певних дій або результатів на даний період; та звичайно з тим, що кожна людина то велечезний світ з мільонів різноманітних деталей, що називають також особистість. Тому і сприймається природньо мистецтво унікально кожною людиною. Митець пише свої твори так, як він бачить світ. Не можна сказати, що мистецтво а що ні, бо така оцінка, характеристика буде надто суб'єктивною та не відповідатиме дійсності. Хтось з великих цього світу казав, що мистецтво це все, що створено людиною. І звісно, наприклад, математики впевнені, що їх робота це мистецтво бо як ще інакше можна назвати ту магію чисел та безмежність думки, що створила абстракцію з усіма змінними або невідомими, що дозволяє їм вирішувати задачі, що стоять перед усім людством. То, що це, як не мистецтво. Мистецтво також є все у чому естетика, так наприклад, досить нещодавно стало мейнстрімом серед багатьох користувачів соціальної мережі інстаграм фотографувати та постити мало не звичайну каву, чорнобілі портрети, шафи родянських часів, будь-що в чому вони знаходять естетику. У мистецтва стільки втілень, скільки має життя.

Отже дуже приблизно, коротко та загально контент у альбомі у формі фотокарток, оскільки це як не єдина сутність або спосіб у який може бути поєднана різноманітність мистецтва, тож складається з:

- віршів;
- виразів;
- фотографій.

									Аркуш	
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-23.БР					10

Унікальність : кожен з всесвітньовідомих сервісів, незалежно від сфери його діяльності, приваблює своїх клієнтів однією серед трьох переваг, таких як:

- якість;
- ціна;
- зручність.

Наприклад : Pinchukartcentre приваблює клієнтів своєю ціною. Сервіс як і галерея є абсолютно безкоштовним, тож будь-який охочий може завітати за ознайомитись з експозицією. Проте сама експозиція обмежена тільки тою виставкою яка проходить у цей час у центрі, звісно справа йдеться про часи до пандемії ковід 19.

А ось головним конкурентом звісно виступає монополіст у галузі соціальна мережа Instagram від всесвітньо відомої компанії facebook що хизується безсумнівною якістю та різноманітністю контенту. Компанія з початку виходу на світовий ринок одразу знайшла прихильників саме через ідею, бо раніше у різноманітних мережах було занадто багато зайвої інформації. Тож поставивши головний акцент на фотографії у системі був досягнений успіх та захоплена, досить велика частка аудиторії конкурентів. Проте така популярна платформа має свої недоліки наприклад з кожним днем в ній все більше і більше реклами, яка зараз вже скрізь, як в сторіз так і в ленті.

Також Instagram вже почав втрачати, те що приваблювало багатьох та стало як не найважливішим чинником, що перетягнуло користувачів з раніше створених мереж. Можна навіть сказати, що інстаграм повністю втратив свою унікальну мінімалістичність, зараз інстапрофайл це вже далеко не фотоколекція просто з нікнеймом та коротким біо.

Інтернет-колекція photoAlbum переважатиме над іншими сайтами мінімалістичністю, прихильність до початкової концепції локонічності та маленька кількість реклами у поєднанні з різноманіттям.

									Аркуш
									11
Зм.	Аркуш	№ докум	Підпис	Дата					

Постачання : усі зображення, що будуть хотитись на платформі будуть завантажуватись з дозволу, за домовленістю або за запитом на розміщення самих авторів та юридичних осіб, власників. Політика редакції порталу досить категорично ставиться до прав на інтелектуальну власність, тож кожна одиниця, що розміщується на сайті буде детально перевірятись на плагіат та піратство, оскільки такі явища недопустимі у сучасному суспільстві та конфліктують, на погляд засновників з морально-етичної точки зору з загально людськими нормами та політикою прозорості та захисту прав на інтелектуальну власність, що впроваджується як загально світовий процес. Саме через це можливий сценарій при якому адміністратор сайту по клопотанню підписників та відвідувачів сайту або за власною ініціативою редакції робитиме запит до правовласника твору за дозволом на хостинг його роботи. Це допоможе уникнути зайвих проблем зі сторони закону, оскільки в інакшому випадку можуть прийти скарги від власників робіт через так звані страйки у гугл та інше, що насправді може призвести до грошового штрафу у кращому випадку або до видалення порталу у гіршому випадку

1.3 Технічне завдання

Предметом розробки є веб-портал сучасної поезії, з системою динамічного відображення та пошуку арт-об'єктів.

Призначення сайту:

- ознайомлення з новітніми творами сучасників;
- популяризації сучасного мистецтва;
- хостингу творів.

Ціль веб-платформи сучасного мистецтва – хостинг новітніх різноманітних арт-об'єктів , однак сайт також буде використовуватися для розміщення реклами, з монітезацією на меті.

									Аркуш	
Зм.	Аркуш	№ докум	Підпис	Дата	КНТЕУ 121 06-23.БР					12

Структура сайту : Будь-яка система має бути дружньою до користувачів, тобто бути наперед спроектованою з точки зору простоти у використанні та орієнтації для кінцевого споживача. Всі розділи, розташування інструментів, умовні позначення повинні бути інтуїтивно зрозумілими навіть дитині та знаходитися на логічному, легко доступному місці та по аналогії до інших. Тому структура сайту повинна бути лаконічна та зрозумілою для користувача. Тобто він не має «загубитися» на сайті. Щоб уникнути таких випадків треба продумати саме таку структуру (рис. 1.1).

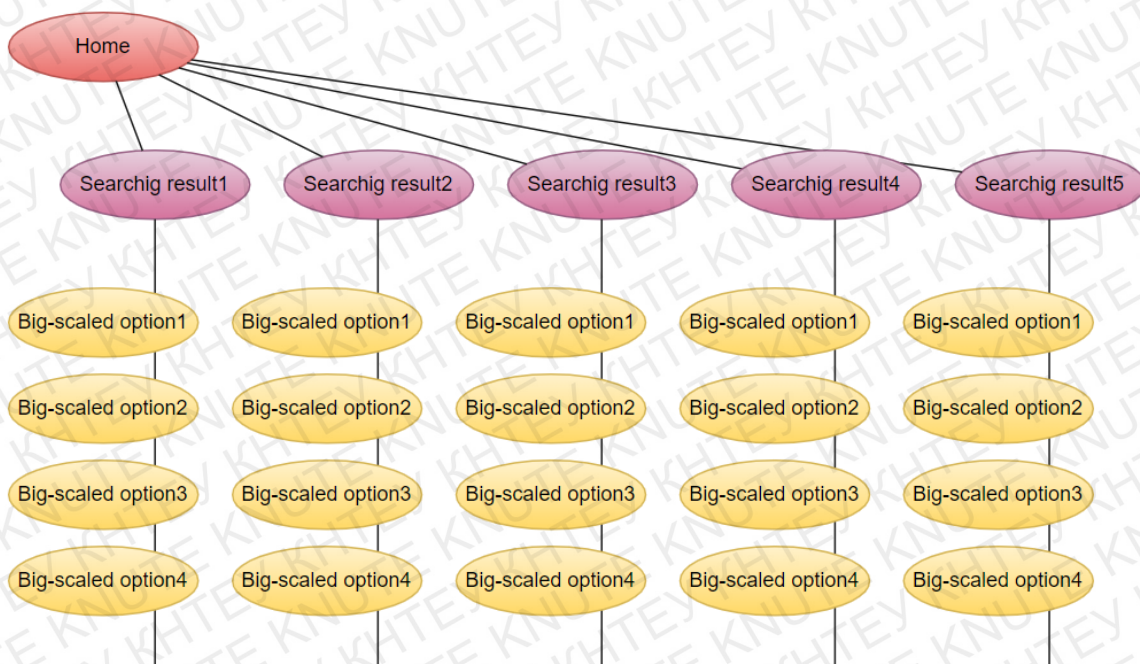


Рис. 1.1. Структура платформи сучасного мистецтва

UX дизайн (User Experience) – це те, який досвід або враження отримує користувач від роботи з вашим інтерфейсом. Простими словами – це структура самої веб-сторінки без графічно красиво відображених елементів. На основі UX дизайну створюється UI дизайн (рис. 1.2).

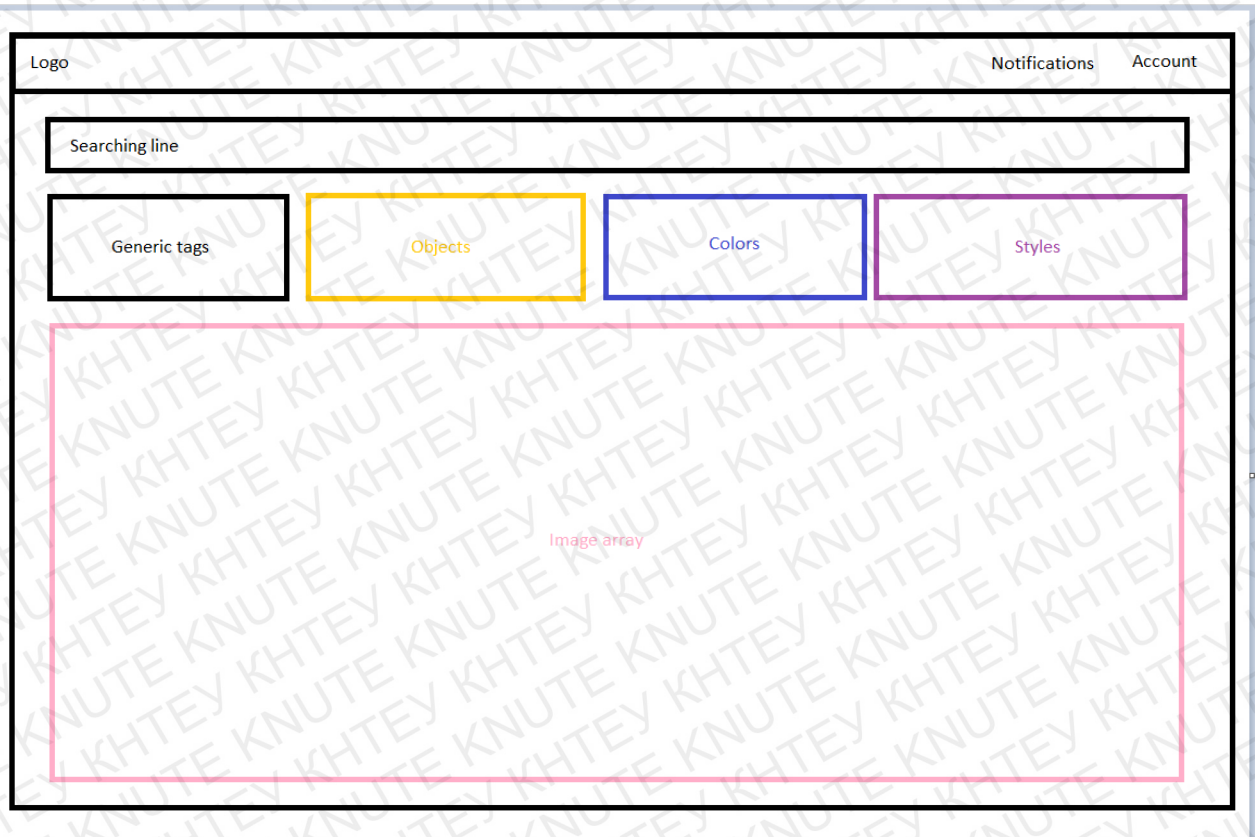


Рис. 1.2. UX дизайн сайту

UI дизайн сайту

UI дизайн (User Interface) – те, як виглядає інтерфейс і те, які фізичні характеристики набуває. Тобто, використовуючи графічні редактори (PhotoShop / Figma), створення візуального шаблону сайту (як він має виглядати в браузері) (рис. 1.3).

Зм.	Аркуш	№ докум	Підпис
			Дата

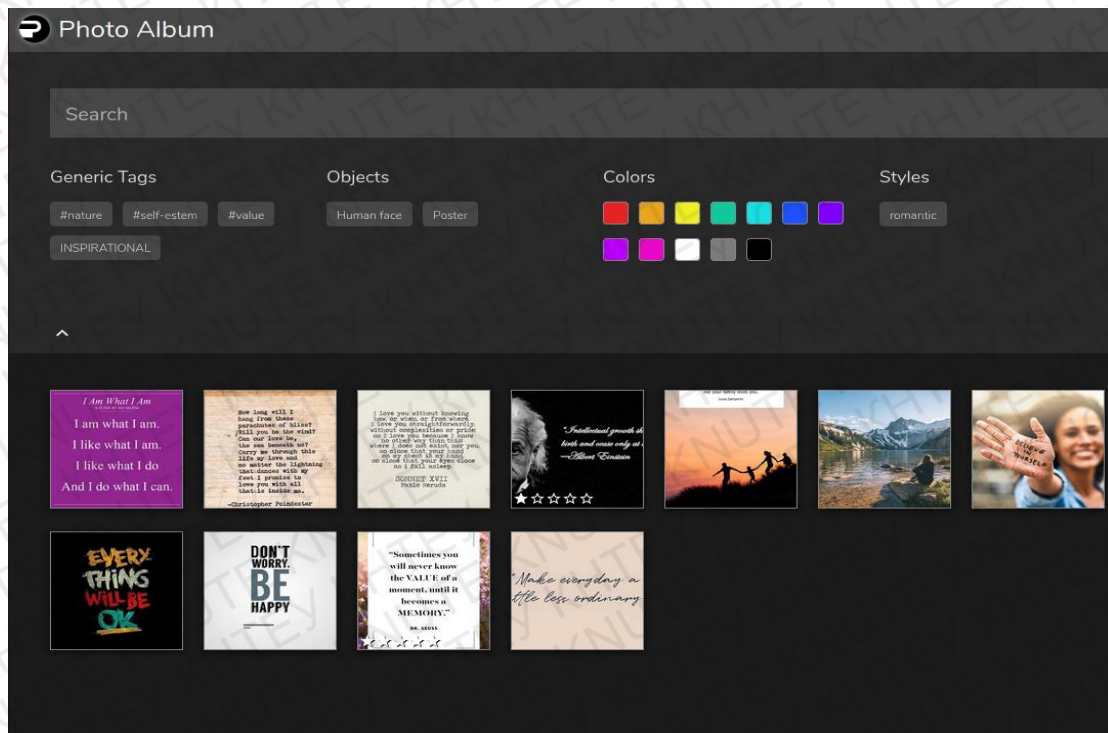


Рис. 1.3 UI дизайн сайту

Технічні вимоги :

Адаптивна верстка. Сайт має виглядати красивим як у десктопній версії, так і в мобільній. Усі блоки не мають вилазити за межі сторінки та не повинно бути зайвих відступів між ними. Також слід попрацювати над текстом, щоб він не був занадто великим або навпаки малим для обох версій;

Система пошуку по сайту. Щоб користувач за бажанням міг ввести назву об'єкту або асоційований тег у поле для вводу та, натиснувши кнопку, побачити список знайдених зображень.

Масштабування зображення. Для кожного зображення має бути можливість масштабування, під цим розуміється певний функціонал веб-сторінки який збільшує розмір зображення від маленького так званого попереднього перегляду 'preview' до повноцінного зображення майже на всю область екрану з діаграмою кольорів, прикріпленими тегами та областю з функціоналом додання нового тегу.

Зм.	Аркуш	№ докум	Підпис Дата

Оцінка зображення та додавання тегу. На сторінці окремого об'єкта після

гортання сторінки нижче мають динамічно відобразитися всі існуючі у базі даних теги, які додавали юзери під час перегляду даного зображення. Також має бути окреме поле, що реалізовуватиме можливість додання нових тегів та поле у якому юзер може залишити свою оцінку зображенню у формі кількості умовних одиниць, наприклад зірочок від однієї до п'яти, де одна – найгірше а п'ять найкраща оцінка.

Система розпізнання об'єктів та кольорів на зображенні. На сторінці масштабованого зображення у кінці сторінки має бути діаграма, що відображає кольори присутні на зображенні, також система має розпізнавати об'єкти на зображенні, щоб юзер при пошуку або сортуванні зображень міг застосувати спеціальний параметер як пошук за об'єктом, наприклад відвідувача порталу цікавлять лише зображення з людиною, тоді він задає системі такий параметер та система шляхом розпізнання всіх об'єктів на зображеннях повертає йому масив зображень з образом людини або юзера цікавить тільки зображення де привалює червоний колір, тоді такми самим чином система повертає йому тільки зображення, що мають червоний колір.

Слайдер, плавні переходи між сторінками, анімації. Для того щоб клієнту

було цікаво відвідувати сайт, треба додати анімації. Слайдер добре приваблює користувачів своєю анімацією, що сприяє позитивним чином на відвідуванність порталу. Також при завантаженні сторінки всі блоки мають плавно з'являтися на сайті, що неабияк порадує користувача.

Зміст сайту : дані сайту будуть динамічно підтягуватись з бази даних, яка буде розроблена розробником. Дані в базі даних будуть взяті з офіційних джерел або запроваджені авторами або особами, що мають права на інтелектуальну власність такого характеру.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата					16

Етапи створення сайту

- створення концепції арт-порталу, розробка та узгодження технічного завдання;
- розробка макета сайту, що включає всі графічні і інтерактивні елементи;
- програмування та підключення модулів управління;
- підготовка контенту - створення, оптимізація, узгодження;
- тестування сайту, при необхідності - внесення корективів;
- запуск.

Висновок до розділу 1

Отже, віртуальний хостинг сучасного мистецтва – ефективний спосіб розповсюдження та популяризації різноманітних творів, які на час пандемії стали недоступними для перегляду. Умовно сайти, тематикою яких є мистецтво можна поділити на три основних типи: віртуальна арт-галерея, веб-хостинг поезії, веб-фотоальбом. Кожен з них має свої переваги та недоліки, але найбільш зручним для відвідувачів буде веб-сайт, що поєднує у собі сутність усіх вище перелічених, оскільки не обмежує юзера за типом або характером контенту та дає можливість розсунути рамки звичайного розуміння та розподілу мистецтва за стилем, жанром, бекграундом та характером проблеми яку підіймає автор своїм твором.

У даному розділі було визначено суть порталу – хостинг сучасного мистецтва у всіх його проявах та його переваги. На сьогоднішній день немає жодного залу, галереї, музею, літературного вечору або публічного зачитування віршів, саме тому світ потребує віртуалізації такого роду заходів. Основною перевагою порталу являється саме різноманітність, сучасність та постійний апдейт, тому що відвідувач загалом цікавиться саме актуальністю мистецтва.

									Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата	<i>КНТЕУ 121 06-23.БР</i>				17

Основним джерелом контенту для сайту стануть молоді та амбіційні митці, котрі як не найкраще відчують сучасні проблеми та невирішенні питання, іншими словами що сьогодні на повісті дня.

Також у розділі було постановлене технічне завдання, за яким буде проходити розробка. Була визначена ціль веб-порталу сучасної поезії:

- для ознайомлення з новітніми творами сучасників;
- для популяризації сучасного мистецтва;
- для хостингу творів.

Була розроблена структура веб-сайту: легка для сприйняття користувачем та лаконічна. Були визначені та розроблені UX та UI дизайн. Були перераховані та роз'яснені такі вимоги до функціонування сайту:

- адаптивна верстка;
- система пошуку по сайту;
- масштабування зображення;
- оцінка зображення та додавання тегу;
- система розпізнання об'єктів та кольорів на зображенні;
- слайдер, плавні переходи між сторінками, анімації.

Також були розписані етапи розробки даного веб-порталу сучасного мистецтва.

Зм.	Аркуш	№ докум	Підпис	Дата

РОЗДІЛ 2

ПРОЕКТУВАННЯ ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ

2.1 Архітектура вебдодатку

Для того, щоб портал був оптимізований (тобто працював швидко та якісно), а також функціонував правильно треба продумати правильну архітектуру. Архітектура програмного забезпечення – це сукупність найважливіших рішень про організацію програмної системи. Архітектура включає:

- вибір структурних елементів і їх інтерфейсів, за допомогою яких складена система, а також їх поведінки в рамках співпраці структурних елементів;
- з'єднання обраних елементів структури і поведінки у все більш крупні системи;
- архітектурний стиль, який спрямовує всю організацію – всі елементи, їх інтерфейси, їх співпраця і їх з'єднання.

Документування архітектури програмного забезпечення спрощує процес комунікації між розробниками, дозволяє зафіксувати прийняті проектні рішення і надати інформацію про них експлуатаційного персоналу системи, повторно використовувати компоненти і шаблони проекту в інших.

Правильна архітектура має такі характерні риси:

- ефективність системи;
- гнучкість системи;
- можливість розширення системи;
- масштабування процесу розробки;
- тестування системи;
- повторне використання та простий супровід.

					<i>КНТЕУ 121 06-23.БР</i>			
<i>Зм.</i>	<i>Аркуш</i>	<i>№ докум</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Зав. каф.</i>		<i>Криворучко О.В.</i>		05.03.2021	<i>Розробка веб-порталу сучасної поезії</i>	<i>Стадія</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Керівник</i>		<i>Котенко Н.О.</i>		05.03.2021		<i>P2</i>	<i>19</i>	<i>46</i>
<i>Гарант</i>		<i>Цензура М.О.</i>		05.03.2021				
<i>Розроб.</i>		<i>Тарасов Д.В.</i>		05.03.2021				
					ПРОЕКТУВАННЯ ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ	Факультет інформаційних технологій, 4 курс, 6 група		

При розробці архітектури інтернет-хостингу мистецтва, до уваги було взято три головних аспекти: залучення до проекту бази даних для спрощення з роботою інформацією, використання ASP.NET а також мову гіпертекстової розмітки (HTML) та каскадну таблицю стилів (CSS)

В розробці систем, не залежно від її складності, першим із критеріїв якості стоїть доступність. Тобто структура розробленої системи має бути зрозумілою для інших розробників, тому що більшість проектів потребують оновлень та корективів, та дуже часто проекти створюються і редагуються не одним і тим самим розробником.

Одним із найдієвіших способів вирішення великої задачі (розробки вебпорталу мистецтва) – це розбиття її на менші задачі, а їх в свою чергу – на ще менші задачі, і так до тих пір поки задачі не стануть досить простими. На щастя це не лише єдине вірне існуюче рішення, а і універсальне рішення, що вирішує великий спектр задач забезпечуючи пониження складності, воно також забезпечує гнучкість системи, дає зручні можливості для масштабування та покращує її стійкість.

2.2 Веб-Фреймворк ASP.NET

Платформа ASP.NET Core представляє технологію від компанії Microsoft, призначену для створення різного роду веб-додатків: від невеликих веб-сайтів до великих веб-порталів і веб-сервісів.

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна. Розробка над платформою почалася ще в 2014 році. Тоді платформа умовно називалася ASP.NET vNext. У червні 2016 року вийшов перший реліз платформи. А в листопаді 2020 року побачила версія ASP.NET Core 5.0, яка власне і буде охоплена в поточному керівництві. ASP.NET Core тепер повністю є opensource-фреймворком.

Зм.	Аркуш	№ докум	Підпис	Дата

Фреймворк тепер має свій легкий контейнер для впровадження залежностей, і більше немає необхідності застосовувати сторонні контейнери, такі як Autofac, Ninject. Хоча при бажанні їх також можна продовжувати використовувати. В якості інструментарію розробки ми можемо використовувати останні випуски Visual Studio, починаючи з версії Visual Studio 2015. Крім того, ми можемо створювати додатки в середовищі Visual Studio Code, яка є крос-платформною і може працювати як на Windows, так і на Mac OS X і Linux. Для обробки запитів тепер використовується новий конвеєр HTTP, який заснований на компонентах Katana і специфікації OWIN. А його модульність дозволяє легко додати свої власні компоненти. Якщо підсумувати, то можна виділити наступні ключові відмінності ASP.NET Core від попередніх версій ASP.NET:

- новий легкий і модульний конвеєр HTTP-запитів;
- можливість розгорнути додаток як на IIS, так і в рамках свого власного процесу;
- використання платформи .NET Core і її функціональності;
- поширення пакетів платформи через NuGet;
- інтегрована підтримка для створення та використання пакетів NuGet;
- єдиний стек веб-розробки, що поєднує Web UI і Web API;
- конфігурація для спрощеного використання в хмарі;
- вбудована підтримка для впровадження залежностей;
- можливість розширення;
- кросплатформеність: можливість розробки і розгортання додатків ASP.NET на Windows, Mac і Linux;
- розвиток як open source, відкритість до змін;
- ці та інші особливості і можливості стали основою для нової моделі програмування.

									Аркуш
									22
Зм.	Аркуш	№ докум	Підпис	Дата					

2.3 Структура файлів проекту

При написанні будь-якого програмного забезпечення, велику увагу слід приділити файловій структурі, адже правильна файлова структура – перша ознака професіоналізму. Бо якщо не дотримуватися правил по її формуванню, то проект перетвориться в безлад, та розібратися в ньому буде важко не тільки іншим розробникам, а ще і самому власнику даного проекту.

Звичайний вебсайт складається з таких складових: .html, .css, .js документи та інші файли такі як картинки (.jpg, .png) та шрифти (.woff, .ttf). Для кожного з них мають бути створені окремі папки, а html файл має лежати у корені папки (сайту). Кожен компонент має дві складові: це js та scss файл. Але логіка, що кожен тип файлів має бути в окремій папці зберігається.

Кожен вебсайт починається із HTML файла, він в свою чергу складається з елементів мови розмітки (тегів), кожен з яких має свою роль та використовується с окремих випадках. Та при розробці масштабних проектів (таких як інтернет-га, соціальна мережа, блог тощо), тегів стає досить багато, а схожість блоків дуже велика, саме для вирішення таких проблем були створені JavaScript фреймворки: AngularJS розроблений компанією Google, ReactJS компанією Facebook та VueJS розробником Еваном Ю. Кожен з них має свою специфіку, але вирішують вони одну й ту саму проблему.

Розробка на фреймворку передбачає собою створення окремих компонентів які будуть заміняти десятки, а то і сотні строчок коду. Наприклад у нас на сторінці є такий блок з зображенням (рис. 2.1):

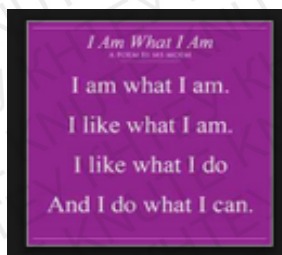


Рис. 2.1 Блок з зображенням

Зм.	Аркуш	№ докум	Підпис	Дата

Так він буде представлений у вигляді коду:

```
<div class = "artimage">  
      
    <div class = "artimage_block-content">  
        </ div>  
    </ div>
```

Для відображення даного блоку з товаром було використано 5 строчок коду, а якщо на сторінці треба відобразити 100 таких товарів? То вже буде складно оперувати з таким об'ємом коду, не кажучи вже про те що буде якщо потрібно підправити ці блоки.

Файлова структура веб-додатку (рис. 2.2):

Name	Date modified	Type	Size
.idea	3/9/2021 6:58 PM	File folder	
PhotoAlbumBLL	3/8/2021 5:56 PM	File folder	
PhotoAlbumDAL	3/9/2021 7:02 PM	File folder	
WebPhotoAlbum	3/9/2021 8:56 PM	File folder	
WebPhotoAlbum.sln	3/8/2021 7:10 PM	Visual Studio Solution	3 KB

Рис. 2.2 Файлова структура веб-додатку

Структура регулювання інтернет-порталу мистецтва : на даному етапі слід розібратись хто буде виконувати яку роль використовуючи веб-додаток. Є такі користувачі додатку:

- відвідувач – особа яка щойно зайшла на сторінку. Основними функціями веб-сайту, якими відвідувач буде користуватися – це сторінка реєстрації до системи. Відвідувач має ввести логін та пароль, ці данні будуть збережені у системі, після цього етапу відвідувач стає користувачем системи;
- користувач системи – особа яка переглядає арт-об'єкти інтернет порталу мистецтва, після розгорнення зображення на весь екран, користувач може оцінити зображення або додати тег-коментар.

Основні функції користування – це додавання коментарів або оцінка зображення;

- менеджер (адмін) – особа яка адмініструє дії користувача під час замовлення. Адміністратор узгоджує нові коментарі, додає щойно зареєстрованим або користувачам, що достатній час використовують систему права адміністратора; наповнює каталог навими зображеннями; узгоджує умови хостингу творів з митцями. Ось графічне відображення даної схеми. (рис. 2.3)

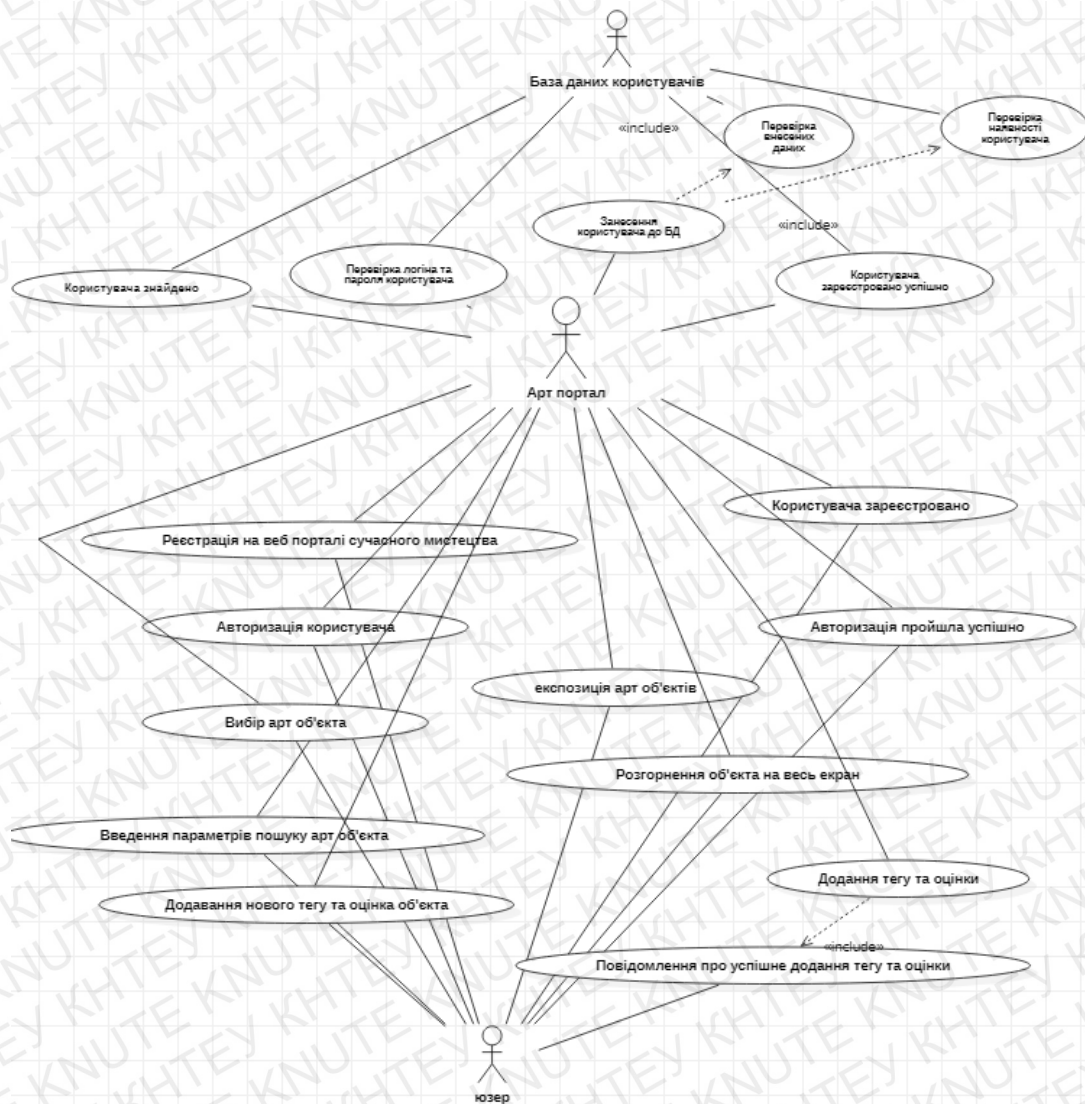


Рис. 2.3 Структура регулювання інтернет порталу мистецтва

Зм.	Аркуш	№ докум	Підпис	Дата
-----	-------	---------	--------	------

2.5 Послідовність роботи системи.

На даному етапі слід зрозуміти послідовність дій кожного з діючих об'єктів (клієнта(відвідувача), системи (веб-хостингу), адміністратора), які приймають участь у роботі порталу. Отже маємо таку послідовність дій цих об'єктів:

- адміністратор наповнює портал;
- система розпізнає об'єкти, кольори та інше на зображеннях;
- клієнт заходить на сайт;
- система перевіряє чи є він у базі даних, якщо ні система відображає вікно реєстрації;
- система відображає каталог;
- клієнт задає параметри пошуку;
- система відображає результат пошуку;
- клієнт обирає об'єкт, що його зацікавив;
- система масштабує зображення;
- клієнт оцінює зображення;
- клієнт додає тег-коментар;
- адміністратор цензурує тег-коментар.

Дана послідовність у виді діаграми (рис 2.4)

Зм.	Аркуш	№ докум	Підпис	Дата

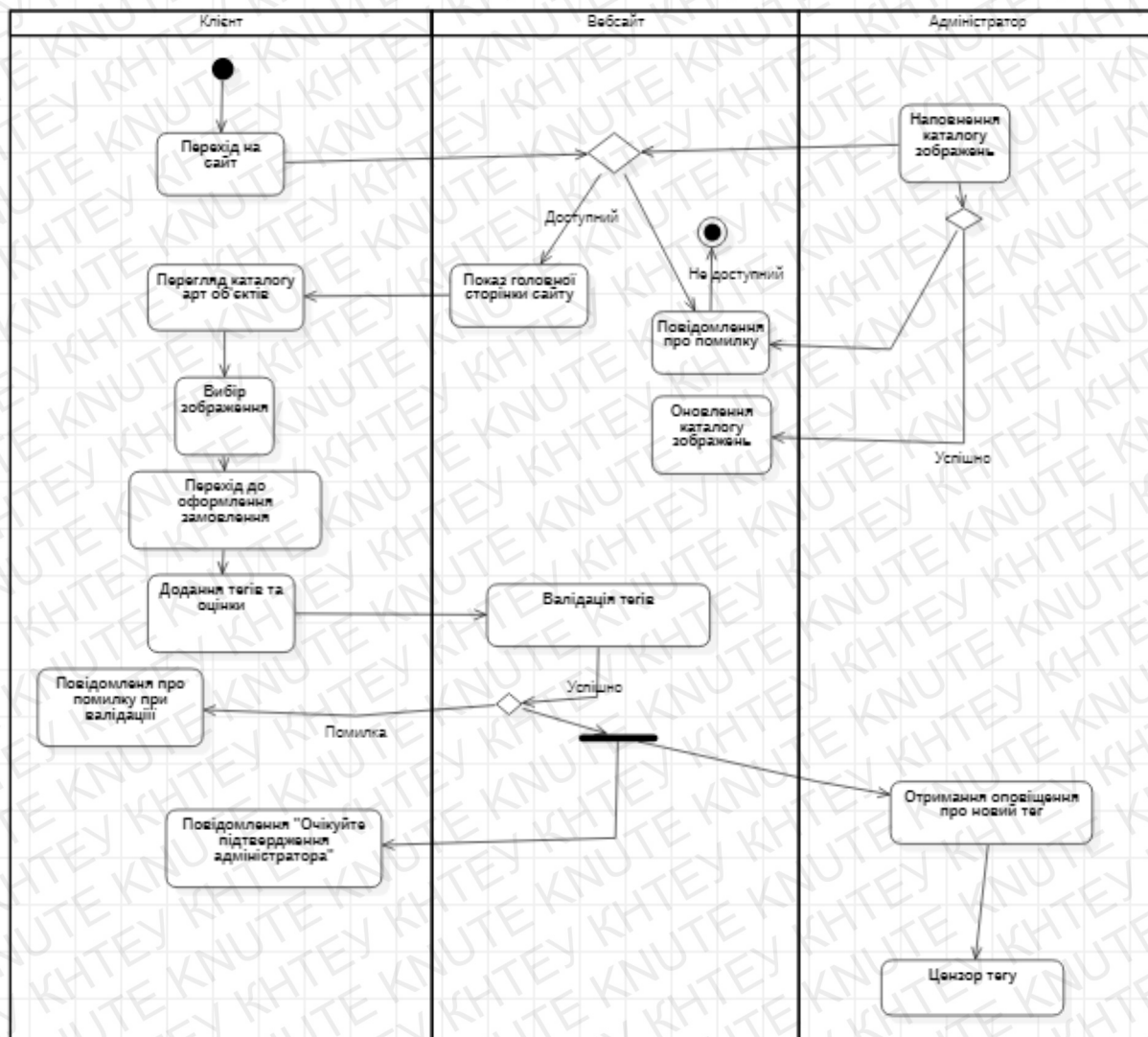


Рис. 2.4 Послідовність роботи системи

2.6 Структура системи.

Однією зі складових процесу розробки продуктів програмного забезпечення є розробка архітектури програмного забезпечення. Існує чимало визначень поняття архітектури програмного забезпечення. Загалом всі формулювання можна поділити на дві групи. Перша група пов'язана з розділенням програмної системи на великі складові частини, друга - з формуванням певних конструктивних рішень, які з розвитком системи не мають змінюватись. Існує також думка від авторитетних фахівців у програмній інженерії, що архітектура програмного забезпечення є у великій мірі суб'єктивним поняттям.

Зм.	Аркуш	№ докум	Підпис	Дата
-----	-------	---------	--------	------

Якщо поглянути на це питання дуже спрощено та у рамках конкретного проекту, що розробляється, то архітектуру ПЗ можна розглядати як утверджену у команді розробників точку зору щодо результатів проектування програмної системи. Проте існує стандарт IEEE 1471 щодо визначення цього поняття. Архітектура -це базова організація системи, втілена в її компонентах, їх відносинах між собою і з оточенням, а також принципи, що визначають проектування і розвиток системи. У цьому стандарті також визначаються такі значущі терміни: Система- це набір компонентів, об'єднаних для виконання визначеної функції або набору функцій. Термін «система» охоплює окремі застосування, системи в традиційному розумінні, підсистеми, системи систем, лінійки продуктів, родини продуктів, цілі корпорації та інші агрегації, що мають відношення до цієї теми. Система існує для виконання однієї або більше місій в своєму оточенні.

Оточення, або контекст, визначає хід і обставини економічних, експлуатаційних, політичних і інших впливів на систему. Місія - це застосування або діяльність, для якої одна або кілька зацікавлених сторін планують використовувати систему відповідно до певного набору умов. Зацікавлена сторона -це фізична особа, група або організація (або її категорії), які зацікавлені в системі або мають пов'язані з нею завдання. Відповідно до стандарту IEEE 1471 можна сформулювати метамодель для визначення архітектури. Загальна структура системи та його складових має такий вигляд (рис 2.5).

Зм.	Аркуш	№ докум	Підпис	Дата

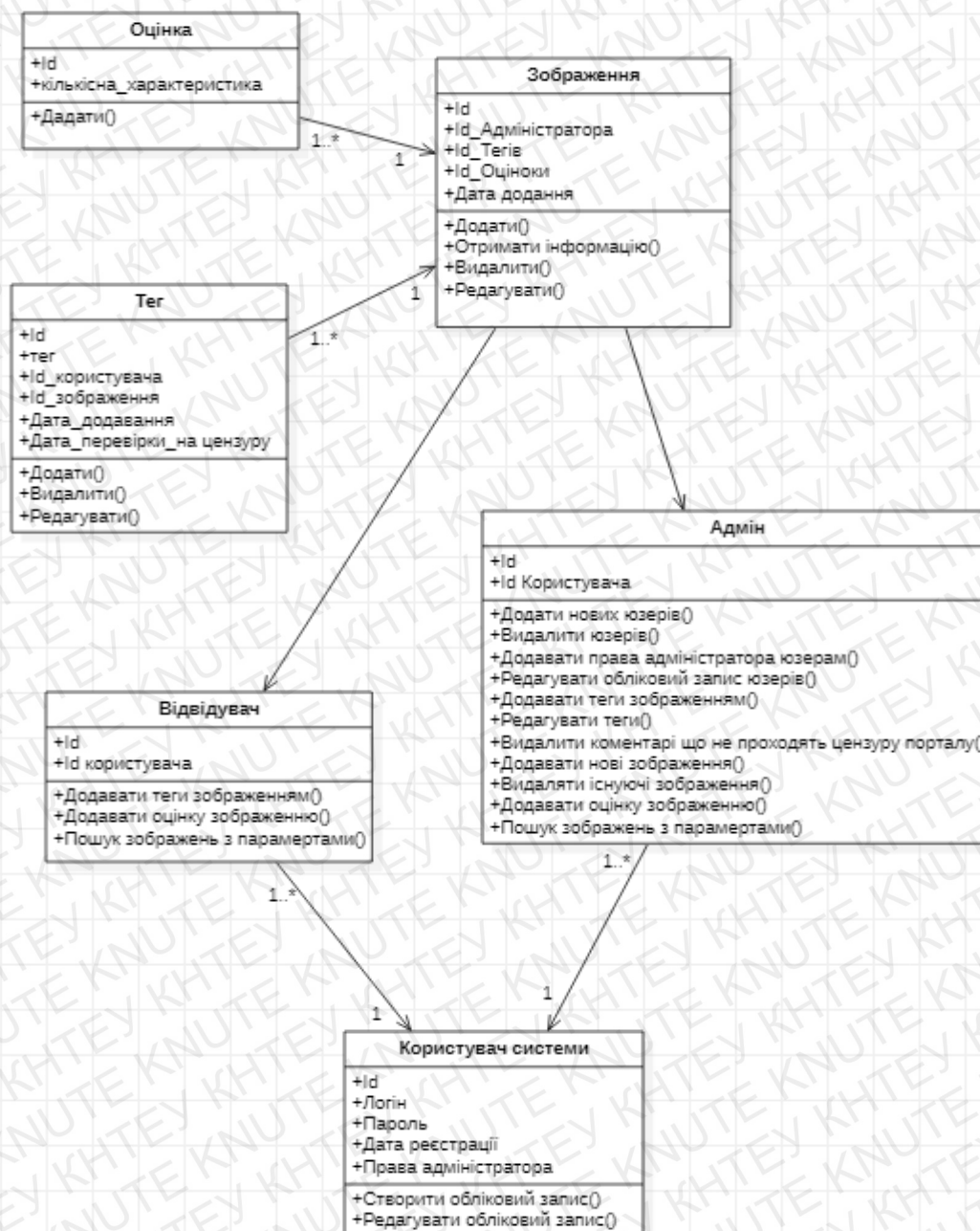


Рис. 2.5 Структура системи

Схема роботи клієнта з інтернет-порталом : гловна діюча особа під час користування веб-додатком – це клієнт (користувач сервісу). Щоб надати клієнту можливість зручно користуватися можливостями сайту треба поставити розробника на місце клієнта та виявити: яка послідовність дій буде у нього під час користування порталом у ролі відвідувача арт-порталу.

Зм.	Аркуш	№ докум	Підпис	Дата
-----	-------	---------	--------	------

Цей етап дуже важливий, так як саме зручність у користуванні. На схемі видно послідовність дій відвідувача, що користується інтернет-порталом. (рис. 2.6).

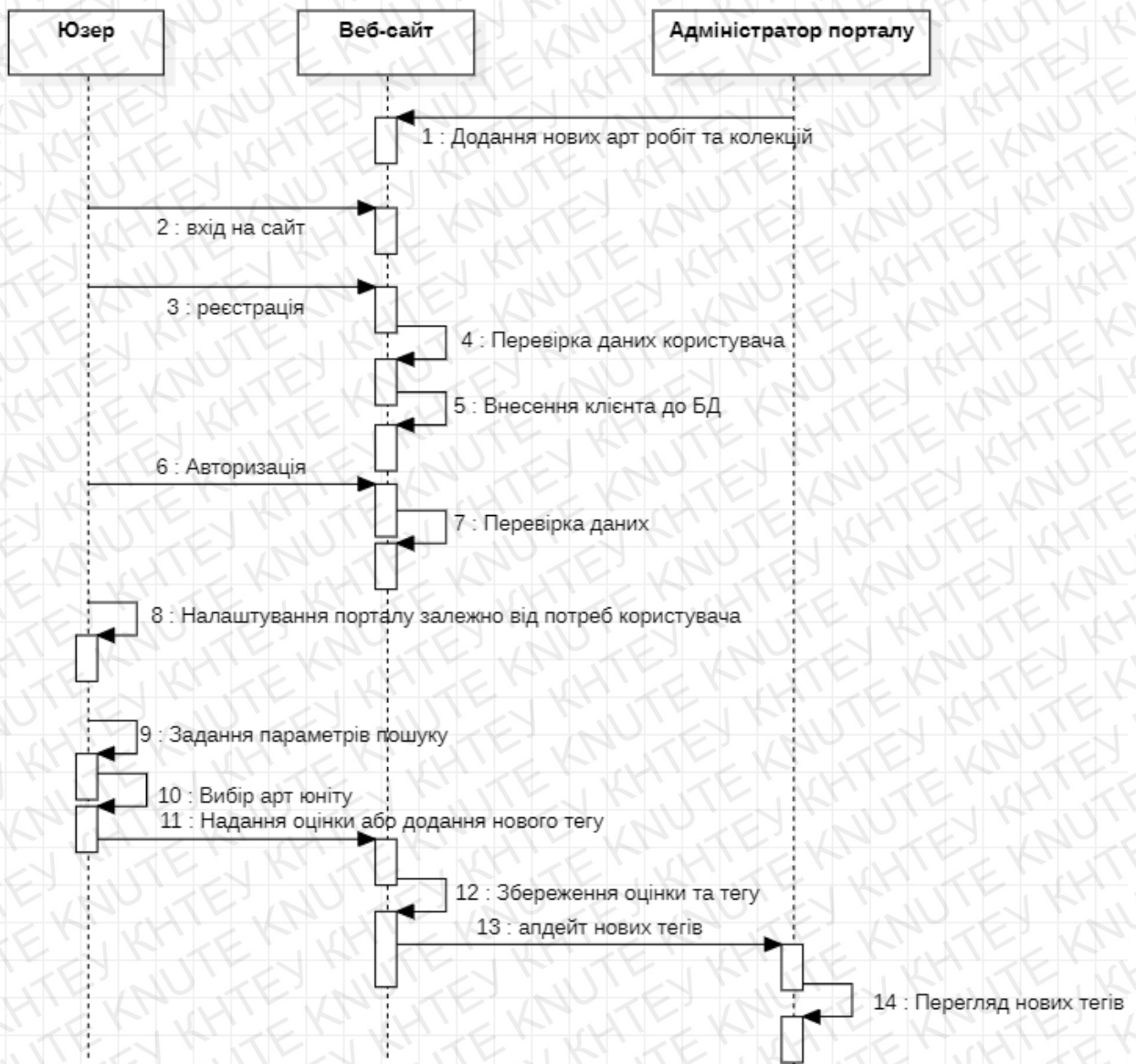


Рис. 2.6 Схема взаємодії у інтернет-порталі

Висновок до розділу 2

Найважливіший та найскладніший етап будь-якого проекту – це початок. Формування технічного завдання та вибір архітектури додатку мають великий вплив на подальшу розробку додатку, саме тому варто до цього етапу підходити серйозно із розумінням та баченням фінального продукту.

Саме після вирішення архітектури додатку був створений та підготовлений до активної стадії розробки проект, що дозволило розробити користувацький інтерфейс засобами середовища розробки Visual Studio Code.

Зм.	Аркуш	№ докум	Підпис	Дата

КНТЕУ 121 06-23.БР

Аркуш

31

РОЗДІЛ 3 РОЗРОБКА ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ

3.1 Середовище розробки

Розробка на фреймворку ASP.NET виконується написанням коду на мові програмування C#, та середовищем розробки даних файлів є текстовий редактор. Корпорація Майкрософт рекомендує використовувати в динамічному коді програми code-behind model, яка розміщує цей код у окремому файлі або в спеціально позначеному тегу. Файли коду, як правило, мають імена типу «MyPage.aspx.cs» або «MyPage.aspx.vb», а файл сторінки — MyPage.aspx (таке ж ім'я, як і у файлу сторінки (ASPX), але з розширенням, що визначає сторінку мови). Ця практика використовується у Visual Studio та інших IDE.

VisualStudio— частково безкоштовний текстовий редактор, що має безліч плагінів для прискореної розробки. Також в ньому є вбудовані функції для виявлення та виправлення синтаксичних помилок у коді. Окрім корисних функцій, даний текстовий редактор має набір тем, які змінюють інтерфейс та шрифти для зручного користування редактором (рис. 3.1).

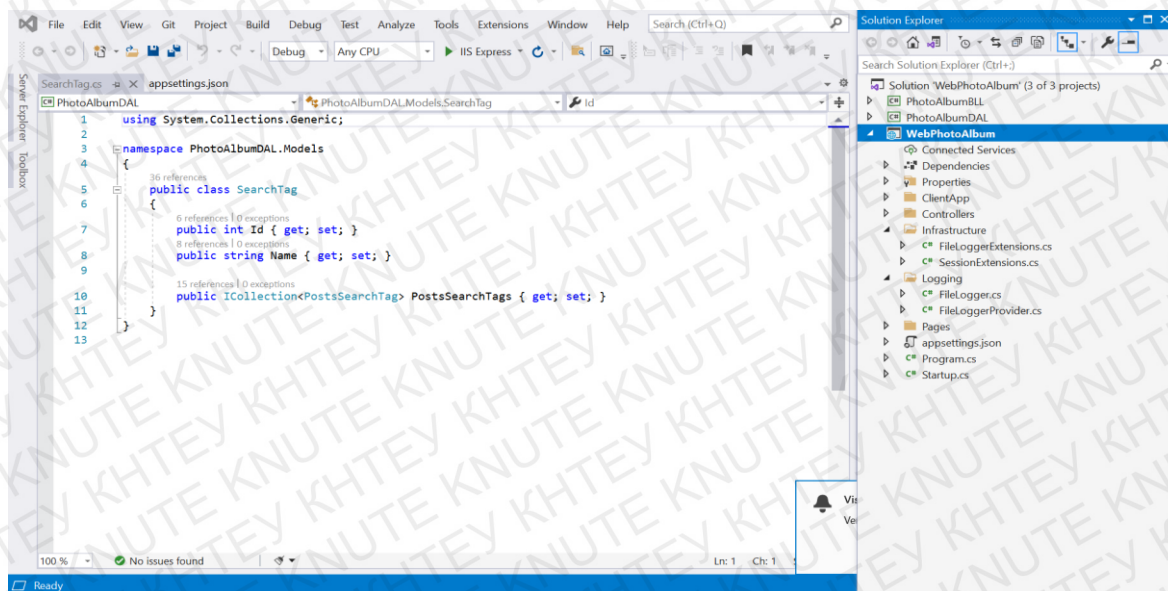


Рис. 3.1. Інтерфейс VisualStudio

					<i>КНТЕУ 121 06-23.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.		10.04.2021	<i>Розробка веб-порталу сучасної поезії</i>	Стадія	Аркуш	Аркушів
Керівник		Котенко Н.О.		10.04.2021		P3	32	46
Гарант		Цензура М.О.		10.04.2021		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Тарасов Д.В.		10.04.2021				
					РОЗРОБКА ВЕБ-ПОРТАЛУ СУЧАСНОЇ ПОЕЗІЇ			

3.2 Технічні особливості

Загалом можна виділити основні одиниці, що складають модель ASP.NET - форми HTML, динамічний призначений для користувача інтерфейс, модель подій ASP.NET, автоматична зворотна відправка даних, стан представлення.

Форми HTML : Найбільш простим способом відправки клієнтських даних сервера є використання дескриптора <form>. Усередині дескриптора <form> можна розміщувати дескриптори <input>, що представляють інші базові елементи призначеного для користувача інтерфейсу, на зразок кнопок, текстових полів, вікон списків і перемикачів. У ASP.NET всі елементи управління поміщені в окремий дескриптор <form>. Цей дескриптор позначений атрибутом runat = "server", який дозволяє йому працювати на сервері. ASP.NET не допускає створення веб-форм, що містять більше одного серверного дескриптора <form>, хоча можна створювати сторінки, які виконують відправлення інформації інших сторінок, з використанням технології міжсторінкових відправки.

Динамічний призначений для користувача інтерфейс : Безсумнівно, модель елементів управління полегшує вилучення інформації про форму. Ще більш примітно те, що вона спрощує життя при необхідності додавання інформації на сторінку. Майже всі властивості веб-елемента управління можна читати і встановлювати. Це означає, що, наприклад, властивість текстового поля Text можна встановити так само просто, як і прочитати його.

Модель подій ASP.NET: (event-driven model). У цій моделі розробник спочатку додає в веб-форму елементи управління, а потім вирішує, на які події він хоче реагувати. Кожен обробник подій є окремий метод, завдяки чому код сторінок виглядає акуратно і організовано. Автоматична зворотна відправка даних : В ASP.NET дії клієнта відбуваються на стороні клієнта, а серверна обробка здійснюється на веб-сервері.

Це означає, що відповідь на подію завжди тягне за собою певні накладні витрати. Тому швидко генеруються події в світі ASP.NET є абсолютно непрактичними. Щоб домогтися в інтерфейсі певного ефекту, для обробки швидких подій на кшталт переміщень курсору миші можна скористатися спеціальним елементом управління ASP.NET з вбудованими можливостями подібного роду. Однак код бізнес-логіки повинен обов'язково виконуватися тільки в безпечної багатофункціональної середовищі сервера. В ASP.NET веб-елементи управління доповнюють модель засобом автоматичної зворотної відправки. За допомогою цього засобу елементи управління введенням можуть генерувати різні події, на які ваш серверний код може негайно відповідати.

Стан представлення : Останнім компонентом в моделі ASP.NET є механізм стану представлення (view state). Цей механізм вирішує ще одну проблему, яка виникає через те, що HTTP не підтримує станів - втрату інформації про зміни. Кожен раз, коли сторінка відправляється назад сервера, ASP.NET отримує всі дані, які користувач ввів в будь-якому з містяться в дескрипторі <form> елементів управління <input>. Після цього ASP.NET завантажує веб-сторінку в початковому стані і налаштовує її відповідно до цих новими даними. Проблема полягає в тому, що в динамічній веб-формі код може змінювати набагато більше деталей. Всі дії мають на увазі зміна вихідного стану сторінки. Однак жодне з них не відбивається в даних, що пересилаються форми. А це означає, що після кожної зворотної відправки даних ця інформація буде втрачатися.

Зазвичай проблема відсутності підтримки станів долається за рахунок використання простих cookie-наборів, сеансових cookie-наборів та інших обхідних прийомів. Однак усі такі механізми вимагають застосування заходів власної розробки (на які часом іде чимало зусиль). Для усунення цього обмеження в ASP.NET є свій інтегрований механізм серіалізації стану.

По суті, цей механізм працює так: після завершення виконання коду сторінки ASP.NET вивчає властивості всіх представлених на цій сторінці елементів управління. Якщо хоч якийсь з цих властивостей змінилося в порівнянні з тим, яким воно було в початковому стані, ASP.NET робить відповідну замітку в колекції "ім'я-значення". Потім ASP.NET бере всю зібрану інформацію і серіалізує її в рядок формату Base64, а потім вставляє цей рядок в розділ <form> сторінки як нове приховане поле. При наступній зворотного відправлення даної сторінки, ASP.NET виконує такі дії:

- відтворює об'єкт сторінки і її елементів управління на основі значень за замовчуванням, в результаті чого сторінка виявляється в тому ж стані, в якому була, коли було запитано про вперше;
- виконує десеріалізацію інформації про стан і оновлює всі елементи управління. В результаті сторінка повертається в стан, в якому перебувала перед останньою відправкою клієнту;
- наостанок проводиться настройка сторінки відповідно до відправленими даними форми. Наприклад, якщо користувач ввів новий текст в текстовому полі або ж зробив новий вибір у вікні списку, ця інформація розміститься в колекції Form і використовується ASP.NET для побудови відповідних елементів управління. Після цього сторінка відображає поточний стан при її перегляді користувачем;
- тепер в дію вступає ваш код обробки подій. ASP.NET генерує відповідні події, і код може реагувати зміною сторінки, переходом на нову сторінку або ж виконанням якоїсь іншої операції;

Використання стану подання є прекрасним рішенням, оскільки серверні ресурси можуть звільнятися після кожного запиту, допускаючи масштабованість сотень або тисяч запитів без уповільнення роботи сервера. Зберігання стану уявлення всередині сторінки збільшує її розміри.

Проте на отримання і відправку сторінки йде багато часу. При налаштуванні складних елементів управління з великою кількістю даних, таких як GridView, обсяг інформації про стан подання може істотно збільшитися. У таких випадках можна відключити стан уявлення для елемента управління, встановивши його властивість `EnableViewState` в `false`. Але в такому випадку доведеться повторно форматувати елемент управління при кожній зворотного відправлення. Ця привілейована інформація про стан подання відома як стан елемента управління, і відключити її не можна. Однак в добре розробленому елементі управління обсяг даних про стан елемента управління значно менше розміру всього стану уявлення. ASP.NET застосовує стан уявлення тільки до властивостей сторінок і елементів управління. ASP.NET не виконує ці ж дії зі змінними примірників і іншими даними, які ви можете використовувати. Однак, ви зможете поміщати інші типи даних в стан уявлення і витягувати цю інформацію вручну пізніше. На наступному малюнку показаний механізм наскрізних запитів сторінок, який об'єднує всі ці концепції (рис 3.2, рис 3.3).



Рис. 3.2. Перший запит

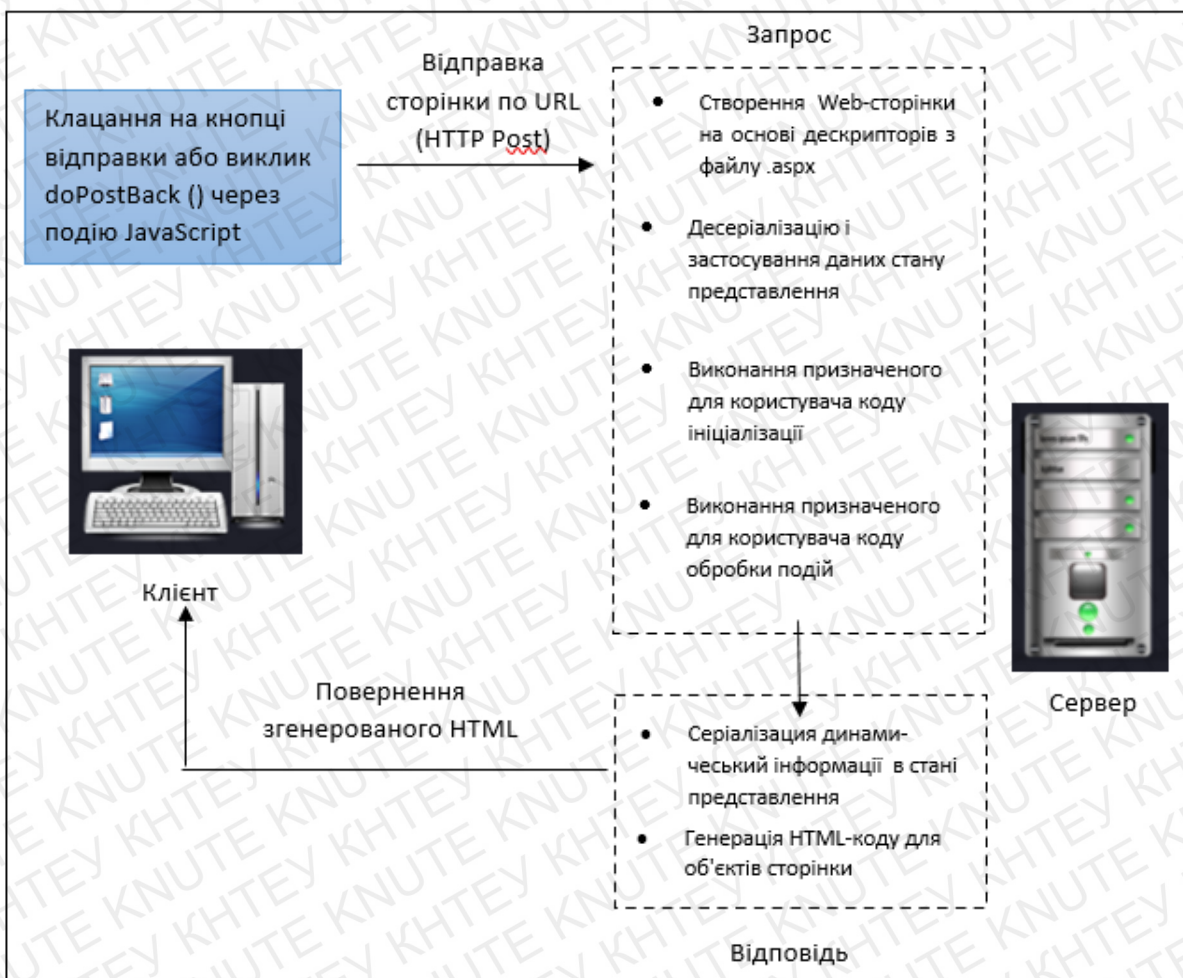


Рис. 3.3. Запрос зворотної відправки

Аналіз стану представлення : у HTML-коді для сторінки ASP.NET є поле вводу з інформацією про стан подання. Стан уявлення є не дуже хорошим місцем для зберігання секретної інформації, яку клієнт не повинен бачити, тому дані такого роду слід залишати на сервері. Крім цього, також не варто приймати на основі стану уявлення рішень, які зможуть скомпрометувати додаток в разі, якщо клієнт спеціально або випадково внесе в дані стану уявлення якісь зміни. Існує можливість посилити безпеку даних стану уявлення. Можна включити функцію автоматичного хешування для запобігання спроб змни стану уявлення, а можна навіть зашифрувати стан уявлення для виключення вірогідності його розшифровки.

Структура проекту : умовно еплікейшн ділиться на три частини (рис 3.4)

Data Access layer , Business logic layer та APP layer.

Name	Date modified	Type	Size
.idea	3/9/2021 6:58 PM	File folder	
PhotoAlbumBLL	3/8/2021 5:56 PM	File folder	
PhotoAlbumDAL	3/9/2021 7:02 PM	File folder	
WebPhotoAlbum	3/9/2021 8:56 PM	File folder	
WebPhotoAlbum.sln	3/8/2021 7:10 PM	Visual Studio Solution	3 KB

Рис.3.4. Файли проекту

Data Access layer – частина, основна задача якої полягає у прямій взаємодії з базою даних. Для цього є спеціальні класи – репозиторії(рис 3.6). Їх основний функціонал це create read update delete (crud) операції над таблицею. Для цього створюються аналогічні методи у репозиторії це createByKey, readByKey, selectAll, updateAll, deleteByKey – в нашому випадку. Кожний репозиторій має мати свій інтерфейс (рис 3.5), як контракт який між програмістом та програмним забезпеченням. Йдеться про задання певних методів, які будь-яка імплементація повинна мати, оскільки так сформоване ТЗ. В нашому випадку наш фото репозиторій успадковується від репозиторія, а сам репозиторій має CRUDметоди.

```
using PhotoAlbumDAL.Models;  
  
namespace PhotoAlbumDAL.Interfaces  
{  
    public interface IPhotoRepository : IRepository<Photo, int>, IAsyncRepository<Photo, int> { }  
}
```

Рис. 3.5. IPhotoRepository


```

using System;
using System.Collections.Generic;
using System.Threading.Tasks;

namespace PhotoAlbumDAL.Interfaces
{
    public interface IAsyncRepository<TEntity, TKey>
        where TEntity : class
    {
        Task<TEntity> GetByKeyAsync(TKey key);
        Task<IEnumerable<TEntity>> GetAllAsync();
        Task<IEnumerable<TEntity>> GetByConditionAsync(Func<TEntity, bool> predicate);
        Task CreateAsync(TEntity entity);
        Task UpdateAsync(TEntity entity);
        Task DeleteAsync(TEntity entity);
        Task DeleteByKeyAsync(TKey key);
    }
}

```

Рис.3.6. Repository

Business logic layer – суть цієї частини, у обробці готових даними з бази даних, які нам достав DAL. Тобто ми у цій частині описуємо бізнес логіку, яка потрібна для здійснення роботи нашого еплікейшина. Основні складові рівня бізнес логіки це сервіси. Наприклад, коли еплікейшн завантажує пости, вона визначає яким тег-коментарам відноситься фотокартка. В цьому випадку у сервіса є задача дістати пост та витягнути його коментар

APP layer – ділиться на дві частини : front-end Client App & back-end.

Головна частина back-end це контролер, основна його функція у наданні рестапп для фронтенду а рестапі відповідає всім методам які були описані в сервісах

У дипломній роботі для розпізнавання образів була використана TensorFlow з ImageNet датасетом. TensorFlow - відкрита програмна бібліотека для машинного навчання, розроблена компанією Google для вирішення завдань побудови і тренування нейронної мережі з метою автоматичного знаходження та класифікації образів, досягаючи якості людського сприйняття. Застосовується як для досліджень, так і для розробки власних продуктів Google.

Основний API для роботи з бібліотекою реалізований для Python, також існують реалізації для R, C Sharp, C ++, Haskell, Java, Go і Swift. Є продовженням закритого проекту DistBelief. Спочатку TensorFlow була розроблена командою Google Brain для внутрішнього використання в Google, в 2015 році система була переведена в вільний доступ з відкритою ліцензією Apache 2.0. Щоб навчити нейронну сітку на TensorFlow розпізнавати потрібні вам об'єкти, вам знадобиться кілька сотень зображень цих об'єктів. Ці зображення можна знайти і скачати з інтернету, або зробити власні фотографії.

Щоб спростити процес установки необхідного програмного забезпечення (особливо під ОС Windows), потрібно скористатися дистрибутивом Anaconda для Python. Потім, доведеться вручну відзначити об'єкти на фотографіях, наприклад, за допомогою утиліти LabelImg.

Функція витягання : щоб виконати розпізнавання або класифікацію зображень, нейронна мережа повинна виконати витяг ознак. Ознаки - це елементи даних, які представляють максимальний інтерес і які будуть передаватися по нейромережі. У конкретному випадку розпізнавання зображень такими ознаками є групи пікселів, такі як лінії і точки, які мережа буде аналізувати на наявність патерну. Розпізнавання ознак (або витяг ознак) - це процес вилучення відповідних ознак з вхідного зображення, щоб їх можна було проаналізувати. Багато зображення містять анотації або метадані, які допомагають нейромережі знаходити відповідні ознаки.

Витяг ознак за допомогою фільтрів : Перший шар нейронної мережі приймає всі пікселі в зображенні. Після того, як всі дані введені в мережу, до зображення застосовуються різні фільтри, які формують розуміння різних частин зображення. Це витяг ознак, яке створює «карти ознак». Цей процес вилучення ознак із зображення виконується за допомогою «сверточного шару», і згортка просто формує уявлення частини зображення.

Саме з цієї концепції згортки ми отримуємо термін "сверточное нейронна мережа" (Convolutional Neural Network, CNN) - тип нейронної мережі, найбільш часто використовуваний в класифікації і розпізнаванні зображень. Якщо ви хочете візуалізувати, як саме працює створення карт ознак, уявіть собі процес піднесення ліхтарика до зображення в темній кімнаті. Коли ви ковзати променем по картинці, ви дізнаєтеся про особливості зображення. Фільтр - це те, що мережа використовує для формування уявлення про зображення, і в цій метафорі світло від ліхтарика є фільтром. Ширина променя вашого ліхтарика визначає розмір фрагмента зображення, який ви переглядаєте за один раз, і нейронні мережі мають аналогічний параметр - розмір фільтра. Розмір фільтра впливає на те, скільки пікселів перевіряється за один раз. Загальний розмір фільтра, використовуваного в CNN, дорівнює 3, і він охоплює як висоту, так і завширшки, тому фільтр перевіряє область пікселів 3 x 3. У той час як розмір фільтра покриває висоту і ширину фільтра, глибина фільтра також повинна бути вказана.

Цифрові зображення відображаються у вигляді висоти, ширини і деякого значення RGB, яке визначає колір пікселя, тому відстежується «глибина» - це кількість колірних каналів, які має зображення. Зображення в градаціях сірого мають тільки 1 кольоровий канал, в той час як кольорові зображення мають глибину в 3 каналу. Для фільтра розміром в 3, застосованого до повнокольорове зображення, підсумкові розміри цього фільтра будуть 3 x 3 x 3. Для кожного пікселя, охопленого цим фільтром, мережа примножує значення фільтра на значення самих пікселів, щоб отримати числове подання цього пікселя. Потім цей процес виконується для всього зображення, щоб отримати повне уявлення. Фільтр переміщується по іншій частині зображення відповідно до параметром, званим «крок», який визначає, на скільки пікселів повинен бути переміщений фільтр після того, як він визначить значення у своїй поточній позиції.

Звичайний розмір кроку для CNN - 2. Кінцевим результатом усіх цих розрахунків є карта ознак. Цей процес зазвичай виконується з декількома фільтрами, які допомагають зберегти складність зображення.

Функції активації : Після того, як карта ознак зображення була створена, значення, що представляють зображення, передаються через функцію активації або шар активації. Функція активації приймає ці значення, які завдяки сверточное шару знаходяться в лінійній формі (тобто просто список чисел) і збільшує їх нелінійність, оскільки самі зображення є нелінійними. Типовою функцією активації, використовуваної для досягнення цієї мети, є випрямлення лінійна одиниця (ReLU), хоча є і деякі інші функції активації, які також іноді використовуються (ви можете прочитати про них тут).

Об'єднання шарів : Після активації дані відправляються через який об'єднує шар. Об'єднання «спрощує» зображення: бере інформацію, яка представляє зображення, і стискає її. Процес об'єднання в пул робить мережу більш гнучкою і здатною краще розпізнавати об'єкти і зображення на основі відповідних функцій. Коли ми дивимося на зображення, нас, як правило, хвилює не вся інформація (наприклад, що на задньому плані зображення), а тільки ознаки, які нас цікавлять - люди, тварини і т. Д. Аналогічно, який об'єднує шар в CNN позбудеться непотрібних частин зображення, залишивши тільки ті частини, які він вважає доречними, в залежності від заданого розміру об'єднує шару. Оскільки мережа повинна приймати рішення щодо найбільш важливих частин зображення, розрахунок йде на те, що вона вивчить тільки ті частини зображення, які дійсно представляють суть даного об'єкту. Це допомагає запобігти "перенавчання" - коли мережа занадто добре вивчає всі аспекти навчального прикладу і вже не може узагальнювати нові дані, оскільки враховує нерелевантні відмінності. Існують різні способи об'єднання значень, але найчастіше використовується максимальне об'єднання.

Максимальна об'єднання має на увазі взяття максимального значення серед пікселів в межах одного фільтра (в межах одного фрагмента зображення). Це відсіває $3/4$ інформації, за умови використання фільтра розміром 2×2 . Максимальні значення пікселів використовуються для того, щоб врахувати можливі спотворення зображення, а кількість параметрів (розмір зображення) зменшені, щоб контролювати перенавчання. Існують і інші принципи об'єднання, такі як середнє або сумарне об'єднання, але вони використовуються не так часто, оскільки максимальне об'єднання дає більшу точність.

Стиснення : останні шари нашої CNN - щільно пов'язані шари - вимагають, щоб дані були представлені в формі вектора для подальшої обробки. З цієї причини дані необхідно «звести воедино». Для цього значення стискаються в довгий вектор або стовпець послідовно впорядкованих чисел.

Навчання моделі : після того, як ваша модель створена, вам просто залишається створити екземпляр моделі і підігнати його до своїх даних для навчання. Найбільшу увагу при навчанні моделі приділяється кількості необхідного для навчання часу. Ви можете вказати тривалість навчання мережі, задавши кількість епох навчання. Чим довше ви тренуєте модель, тим вище її ефективність, але якщо використовувати занадто багато епох навчання - ви ризикуєте перенавчити модель. Вибір кількості епох для навчання - це те, що ви навчитеся визначати згодом, і, як правило, слід завжди зберігати ваги нейромережі між тренувальними сесіями, щоб вам не потрібно було починати спочатку після досягнення певного прогресу в навчанні.

Оцінка моделі : існує кілька кроків для оцінки моделі. Першим кроком є порівняння продуктивності моделі з набором перевірочних даних: тих даних, на яких модель не була навчена. Таким чином, ви перевірите роботу моделі з цим новим набором даних і проаналізуєте її ефективність за допомогою різних показників.

Існують різні метрики для визначення продуктивності моделі нейронної мережі, але найбільш поширеною є «точність», тобто кількість правильно класифікованих зображень, поділене на загальну кількість зображень у вашому наборі даних. Після того, як ви побачите точність моделі в перевірочному наборі даних, ви, ймовірно, знову повернетеся і до-навчіть мережу, використовуючи злегка підправлені параметри, оскільки навряд чи будете задоволені своєю ефективністю своєї мережі при першому тренуванні. Ви будете продовжувати налаштовувати параметри своєї мережі, повторно навчати її і вимірювати ефективність, поки не будете задоволені точністю мережі. Нарешті, ви перевірите ефективність мережі на тестовому наборі. Це ще один набір даних, який ваша модель ніколи не бачила раніше. Вся справа в тому, що всі зміни параметрів, які ви робили, донастраивая мережу при роботі з "перевірочним набором даних" в поєднанні з багаторазовим повторним тестуванням цього набору - могли привести до того, що ваша мережа вивчила деякі особливості набору, але при цьому вона не буде так само добре узагальнювати дані поза вибіркою. Саме тому слід надати мережі абсолютно нові тестові дані. Мета тестового набору - перевірити наявність проблем, таких як перенавчання, щоб бути більш впевненими в тому, що ваша модель дійсно придатна для роботи в реальному світі.

Висновок до розділу 3

У даному розділі були повністю розглянуті засоби реалізації розробленого інтерфейсу та логіки додатку. Також були розглянуті та реалізовані основні можливості фреймворку ASP.NET, а саме: ініціалізація ASP.NET проекту та створення правильної структури; принципи роботи даного фреймворка; створення логіки; відображення компонентів; створення навігації, завдяки якій і відбувається перехід між сторінками у додатку розробленого на ASP.NET та багато іншого.

Також було розроблено пошукову систему для сайту, котра дозволяє шукати об'єкт не лише по банальним критеріям, таким як тег-опис а й по розпізаному образу на зображенні або привалюючому кольору, завдяки використанню бібліотек комп'ютерного бачення. Був розроблений функціонал розгорнутого зображення, який відображає перелік тег-описів доданих користувачами до зображення та область сторінки, що дозволяє користувачу додавати новий опис.

					<i>КНТЕУ 121 06-23.БР</i>	Аркуш
Зм.	Аркуш	№ докум	Підпис	Дата		45

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

В ході виконання дипломної роботи було розглянуто питання процесу навчання завдяки використанню сучасних технологій та проаналізовані багато можливих способів та можливих інструментів для створення програмних навчальних додатків для різноманітних цільових платформ. Завдяки вибору веб-фреймворку ASP.NET був розроблений повноцінний веб портал сучасної поезії та мистецтва, котрий виділяється серед конкурентів адаптивністю, оптимізованістю та унікальними рисами такими як розумний пошук та різноманітністю контенту.

Під час розробки були задіяні такі технології:

- photoShop – графічний редактор для створення UX / UI веб-дизайну;
- starUML – для розробки діаграм, завдяки яким було створено структуру проекту;
- asp.net — технологія створення [веб-застосунків](#) і [веб-сервісів](#) від компанії [Майкрософт](#);
- openCV - Бібліотеки образів комп'ютерного зору.

У процесі навчання використано працюючий курс, що дає впевнений старт для початку розробки із застосуванням інструментів для розробки ASP.NET .

У подальшому розвитку додатку можливе додавання інших мов інтерфейсу та побудова більш раціонального у використанні інтерфейсу.

					<i>КНТЕУ 121 06-23.БР</i>			
м.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.		24.04.2021	<i>Розробка веб-порталу сучасної поезії</i>	Стадія	Аркуш	Аркушів
Керівник		Котенко Н.О.		24.04.2021		ВП	46	46
Гарант		Цензура М.О.		24.04.2021		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Тарасов Д.В.		24.04.2021				
<i>ВИСНОВКИ ТА ПРОПОЗИЦІЇ</i>								

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Майк Брінд та Імар Спанджаарс. Початок веб-сторінок ASP.NET з WebMatrix;
 2. Марк Дж. Прайс. С # 9 та .NET - Сучасна крос-платформна розробка. Створюйте інтелектуальні програми, веб-сайти та служби за допомогою Blazor, ASP.NET Core та Entity Framework Core, використовуючи Visual Studio Code;
 3. Нішант Найр, Рагіні Кумбхат Бхандарі. Створення мобільних додатків за допомогою Kendo UI Mobile та веб-API ASP.NET;
 4. Тамір Дрешер, Амір Цукер, Шей Фрідман. Повна повна стекова веб-розробка за допомогою ASP.NET Core Вивчіть наскрізну веб-розробку за допомогою провідних фронтенд-фреймворків, таких як Angular, React та Vue;
 5. Badrinarayanan Lakshmiraghavan. Pro ASP.NET Web API Security Securing ASP.NET Web API;
 6. Буді Курняван. Реальні .NET-додатки;
 7. Річард Шеліскі. Комп'ютерне бачення: алгоритми та програми;
 8. Саймон Принс. Комп'ютерне бачення: моделі, навчання та умовивід;
 9. Річард Хартлі та Ендрю Ціссерман. Геометрія декількох поглядів у комп'ютерному баченні;
 10. Курт ДеМагд, Ентоні Олівер, Натан Остендорп та Кетрін Скотт. Практичне комп'ютерне бачення з SimpleCV;
- Інтернет ресурси
- Photoshop: <https://www.adobe.com/ua/products/photoshop.html>;
- StarUML: <https://staruml.io/>
- ASP.NET : <https://dotnet.microsoft.com/apps/aspnet>.

					<i>КНТЕУ 121 06-23.БР</i>			
Зм.	Аркуш	№ докум	Підпис	Дата				
Зав. каф.		Криворучко О.В.		24.04.2021	<i>Розробка веб-порталу сучасної поезії</i>	Стадія	Аркуш	Аркушів
Керівник		Котенко Н.О.		24.04.2021		СВД	47	46
Гарант		Цензура М.О.		24.04.2021		Факультет інформаційних технологій, 4 курс, 6 група		
Розроб.		Тарасов Д.В.		24.04.2021				
					<i>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</i>			