
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА

Кафедра «Інформаційні системи»

ДИПЛОМНА РОБОТА

на тему:

«Фотогалерея з штучним інтелектом»

Студент 4 курсу групи ПМі-44

Колтун Роман Михайлович

Науковий керівник:

Горlach Віталій Михайлович

Національна шкала _____

Кількість балів: _____ Оцінка: ECTS _____

Львів

2019

**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА
ФРАНКА**

Факультет прикладної математики та інформатики

Кафедра інформаційних систем

Освітньо-кваліфікаційний рівень бакалавр

Спеціальність 122 – комп'ютерні науки

«ЗАТВЕРДЖУЮ»

Зав. кафедрою проф. Шинкаренко Г.А.

« » вересня 20 19 р.

З А В Д А Н Н Я

НА ДИПЛОМНУ РОБОТУ СТУДЕНТА

Колтуна Романа Михайловича

(прізвище, ім'я, по батькові)

1. Тема роботи Фотогалерея з штучним інтелектом

керівник роботи Горlach Віталій Михайлович, канд. фіз.-мат. наук, доцент

затверджені Вченою радою факультету від «16» жовтня 20 19 р., № _____

2. Строк подання студентом роботи 10.06.2020

3. Вихідні дані до роботи _____

Література та інтернет-ресурси за тематикою роботи

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

– Огляд існуючих систем та технологій

– Постановка задачі

– Розробка моделі даних

– Проектування системи

– Тестування та опис

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

Ілюстрації моделі даних та інтерфейсу системи

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 23 вересня 2019 року

КАЛЕНДАРНИЙ ПЛАН

№	Найменування етапів дипломної (кваліфікаційної) роботи	Строк виконання етапів роботи	Примітка
1.	<i>Огляд існуючих систем та технологій</i>	<i>вересень</i>	
2.	<i>Постановка задачі</i>	<i>жовтень</i>	
3.	<i>Розробка моделі даних</i>	<i>листопад</i>	
4.	<i>Проектування системи</i>	<i>грудень-березень</i>	
5.	<i>Тестування та опис</i>	<i>березень-квітень</i>	
6.	<i>Оформлення дипломної роботи</i>	<i>травень</i>	

Студент _____
підпис

Колтун Р.М.

Керівник роботи _____
підпис

Горлач В.М.

ОЦІНОЧНИЙ ЛИСТ

дипломної роботи студента Колтуна Романа Михайловича
прізвище, ім'я, по-батькові
групи ПМi-44 факультету прикладної математики та інформатики

Відгук наукового керівника

Тематика

Комп'ютерне моделювання
Чисельні методи
Оптимізація процесів
Системне програмування
Бази даних
Навчальні програми
Веб-проектування
Інше

Зміст

Максимальна кількість балів

4	
4	
4	
4	

4	
---	--

Сума балів

--

Коментарі

Складність, повнота розкриття дослідження
Наукова новизна, елементи творчості
Самостійність виконання, систематичність роботи
Якість та складність програм

Оформлення

Стиль, грамотність Ілюстративний матеріал Відповідність вимогам до роботи

Науковий
керівник

доцент кафедри інформаційних систем,
канд. фіз.-мат. наук, В.М. Горlach

Оцінка рецензента ____б. Оцінка за захист ____б. СУМА БАЛІВ _____

Голова ЕК

РЕЦЕНЗІЯ
на дипломну роботу

дипломної роботи студента Колтуна Романа Миїайловича
прізвище, ім'я, по-батькові
групи ПМі-44 факультету прикладної математики та інформатики

Тематика

Комп'ютерне моделювання
Чисельні методи
Оптимізація процесів
Системне програмування
Бази даних
Навчальні програми
Веб-проектування
Інше

**Максимальна
кількість балів**

6	
6	
6	
6	

2	
2	
2	
Сума балів	

Зміст

Складність
Наукова новизна
Повнота розкриття дослідження
Якість та складність програм

Оформлення

Стиль, грамотність
Ілюстративний матеріал
Відповідність вимогам до роботи

Коментарі

Рецензент

Зміст

Вступ.....	7
Розділ 1. Постановка задачі та огляд технологій.....	8
1.1. Постановка задачі.....	8
1.2. Огляд технологій.....	8
1) Microsoft .NET.....	8
2)ASP.NET MVC	9
3)ADO.NET Entity Framework.....	10
4) Microsoft SQL Server.....	11
5)ASP.NET CORE.....	12
6) Microsoft Azure.....	13
7)Face Recognition	13
8) HTML	14
9)CSS.....	15
Розділ 2. Програмна реалізація	16
2.1. База даних проекту.....	16
2.2. Опис контролерів системи	17
Розділ 3. Сценарій роботи системи	18
3.1. Реєстрація та вхід в систему	18
3.2. Завантаження.....	18
Висновки	22
Список використаної літератури	23

Вступ

Фотографії є важливою частиною нашого життя. Фотографії це наші спогади і кожна людина бажає запам'ятати певні моменти її життя.

У всесвітній павутині є багато соціальних мереж та додатків в яких можна зберігати свої фото. На полицях магазинів може знайти альбоми, де ми також можемо зберігати свої фото. Та в більшості додатки і звісно альбоми не запам'ятовують лиць, тому я написав цю галерею із використанням штучного інтелекту для запам'ятовування обличчя людини, що дозволить підписати старих знайомих та знайти їх достатньо швидко.

Веб застосунок буде працювати на платформі .NET. Клієнтська частина веб застосунку буде розроблена за допомогою новітніх технологій, бази даних реалізовані системою керування MS SQL. Власна UI частина без використання сторонніх фреймворків та шаблонізаторів, оригінальні дизайни. Все це забезпечує унікальність продукту візуально і створює необхідний функціонал.

Розділ 1. Постановка задачі та огляд технологій

1.1. Постановка задачі

Наша ціль: створити зручну галерею із використання штучного інтелекту, а саме Face recognition:

- Реєстрація
- Вхід
- Завантаження фотографії
- Виявлення і зберігання параметрів обличчя
- Інтуїтивний та зрозумілий інтерфейс.

1.2. Огляд технологій

1) Microsoft .NET — програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Одною з ідей .NET є сумісність служб, написаних різними мовами. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET — крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проєкті Mono (в рамках угоди між Microsoft з Novell), DotGNU.

Захист авторських прав відноситься до створення середовищ виконання (CLR — Common Language Runtime) для програм .NET. Компілятори для .NET випускаються багатьма фірмами для різних мов вільно.

.NET поділяється на дві основні частини — середовище виконання (по суті віртуальна машина) та інструментарій розробки.

Середовища розробки .NET-програм: Visual Studio .NET (C++, C#, J#), SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки .NET-програм. Застосовні програми також можна розроблювати в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-кода дозволяє отримати крос-платформовість на рівні скомпільованого проекту (в термінах .NET: збірка), а не на рівні сирцевого тексту, як, наприклад, в С. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора.

Слід зазначити, що один з перших JIT-компіляторів для Java був також розроблений фірмою Microsoft (тепер в Java використовується досконаліша багаторівнева компіляція — Sun HotSpot). Сучасна технологія динамічної компіляції дозволяє досягнути аналогічного рівня швидкодії з традиційними «статичними» компіляторами (наприклад, C++) і питання швидкодії часто залежить від якості того чи іншого компілятора.

2)ASP.NET MVC являє собою платформу для створення сайтів і веб-додатків з використанням патерну (або шаблону) MVC (model - view - controller).

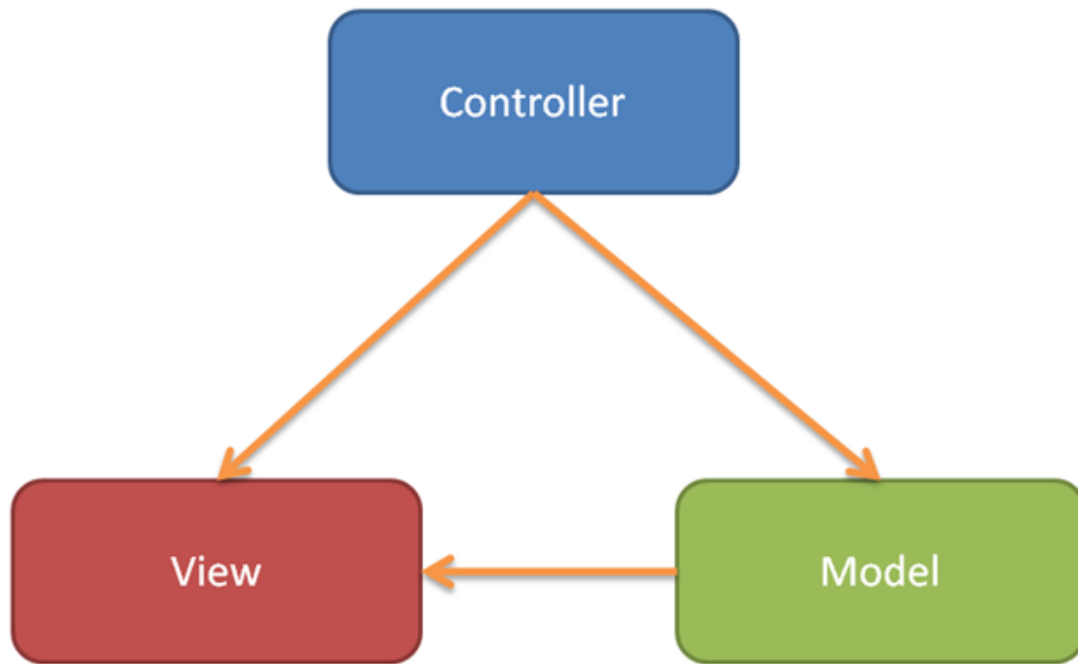
Шаблон MVC, що лежить в основі нової платформи, має на увазі взаємодію трьох компонентів: контролера (controller), моделі (model) і уявлення (view). Що ж являють ці компоненти?

Контролер (controller) представляє клас, з якого власне і починається робота програми. Цей клас забезпечує зв'язок між моделлю і представленням. Отримуючи вводяться користувачем дані, контролер виходячи з внутрішньої логіки при необхідності звертається до моделі і генерує відповідне подання.

Подання (view) - це власне візуальна частина або призначений для користувача інтерфейс програми - наприклад, html-сторінка, через яку користувач, що зайшов на сайт, взаємодіє з веб-додатком.

Модель (model) представляє набір класів, що описують логіку використовуваних даних.

Загальну схему взаємодії спрощено можна представити таким



чином:

ASP.NET MVC є в деякому роді конкурентом для традиційних веб-форм і має в порівнянні з ними наступні переваги:

Поділ відповідальності. У MVC додаток складається з трьох частин: контролера, уявлення і моделі, кожна з яких виконує свої специфічні функції. В результаті додаток буде легше підтримувати модифікувати в майбутньому. В силу поділу відповідальності додатки mvc мають кращу тестованого. І ми можемо тестувати окремі компоненти незалежно один від одного.

Відповідність протоколу HTTP. Додатки MVC на відміну від веб-форм не підтримують об'єкти стану (ViewState). Ясність і простота платформи дозволяють домогтися більшого контролю над роботою додатка.

Гнучкість. Ви можете налаштовувати різні компоненти платформи на свій розсуд. Змінювати будь-якої частини конвеєра роботи MVC або адаптувати його до своїх потреб і потреб.

3)ADO.NET Entity Framework - об'єктно-орієнтована технологія доступу до даних, «object-relational mapping» (ORM) рішення для .NET Framework від Microsoft. Надає можливість взаємодії з об'єктами як за допомогою LINQ у вигляді LINQ to Entities, так і з використанням Entity SQL. Для полегшення побудови веб-рішень використовується як ADO.NET Data Services (Astoria), так і зв'язка з Windows

Communication Foundation, і Windows Presentation Foundation, що дозволяє будувати багаторівневі додатки, реалізуючи один з шаблонів проектування MVC, MVP або MVVM. Спочатку була частиною платформи .NET, але з релізом Entity Framework 6 вона відокремила від платформи.

Підходи в Entity Framework.

Спочатку з найпершої версії Entity Framework підтримував підхід Database first, який дозволяв по готовій базі даних згенерувати модель EDMX. Потім ця модель використовувалася для підключення до бази даних. Пізніше був доданий підхід Model First. Він дозволяв створити вручну за допомогою візуального редактора моделі EDMX, і по ній створити базу даних. Починаючи з 5.0 добавився підхід Code First. Його суть - спочатку пишеться код моделі на C #, а потім по ньому генерується база даних. При цьому модель EDMX вже не використовується. В даному проекті був використаний саме останній підхід Code First.

4) Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI / ISO щодо структурованої мови запитів SQL із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства. Багато років вдало конкурує з іншими системами керування базами даних.

SQL Server має вбудовану підтримку .NET Framework. Завдяки цьому, процедури бази даних, що зберігаються, можуть бути написані на будь-якій мові платформи .NET з використанням повного набору бібліотек, доступних для .NET Framework. На відміну від інших процесів, .NET Framework виділяє додаткову пам'ять і будує засоби керування SQL Server, не використовуючи вбудовані засоби Windows. Це підвищує продуктивність порівняно із загальними алгоритмами Windows, оскільки алгоритми розподілу ресурсів спеціально налагоджені для використання у структурах SQL Server.

5)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 грудні 2019 року побачила версія ASP.NET Core 3.1, яка власне і буде охоплена в поточному керівництві.

ASP.NET Core тепер повністю є opensource-фреймворком. ASP.NET Core може працювати поверх крос-платформної середовища .NET Core, яка може бути розгорнута на основних популярних операційних системах: Windows, Mac OS, Linux. І таким чином, за допомогою ASP.NET Core ми можемо створювати крос-платформні додатки. І хоча Windows як середовище для розробки і розгортання програми досі превалює, але тепер вже ми не обмежені тільки цією операційною системою. Тобто ми можемо запускати веб-додатки не тільки на ОС Windows, але і на Linux і Mac OS. А для розгортання веб-додатки можна використовувати традиційний IIS, або крос-платформний веб-сервер Kestrel. Завдяки модульності фреймворка всі необхідні компоненти веб-додатки можуть завантажуватися як окремі модулі через пакетний менеджер Nuget. Крім того, на відміну від попередніх версій платформи немає необхідності використовувати бібліотеку System.Web.dll. ASP.NET Core включає в себе фреймворк MVC, який об'єднує функціональність MVC, Web API і Web Pages.

У попередніх версії платформи дані технології реалізувалися окремо і тому містили багато дублюючої функціональності. Зараз же вони об'єднані в одну програмну модель ASP.NET Core MVC. А Web Forms повністю пішли в минуле. Крім об'єднання вищезазначених технологій в одну модель в MVC був доданий ряд додаткових функцій. Однією з таких функцій є тег-хелпери (tag helper), які дозволяють більш органічно поєднувати синтаксис html з кодом C#. ASP.NET Core характеризується розширюваністю. Фреймворк побудований з набору щодо незалежних компонентів. І ми можемо або використовувати вбудовану реалізацію

цих компонентів, або розширити їх за допомогою механізму спадкування, або зовсім створити і застосовувати свої компоненти зі своїм функціоналом.

6) Microsoft Azure- хмарна платформа та інфраструктура корпорації Microsoft, призначена для розробників застосунків хмарних обчислень (cloud computing) і покликана спростити процес створення онлайн-додатків

Платформа виготовлена з групи із трьох технологій, що забезпечують спеціалізований набір можливостей для розробників. Більше того, платформу Windows Azure можна використовувати в додатках, що працюють локально на комп'ютерах користувачів і додатків, які працюють в хмарі.

Платформа Windows Azure складається з таких компонентів:

- Windows Azure — надає середовище виконання для додатків, заснованих на операційних системах, та на Windows Server, а також місця для зберігання даних. Система працює на віртуальних машинах за допомогою, аналогічної технології Hyper-V.
- Обчислення — відповідає за обчислення розміщення додатків.
- Зберігання — відповідає за зберігання даних в хмарі.
- SQL Azure — надає можливість використовувати реляційну базу даних для запуску в хмарі.
- Платформа Windows Azure AppFabric — компонент, який забезпечує додаткову функціональність у вигляді послуг.

7)Face Recognition Система розпізнавання обличчя - це технологія, здатна ідентифікувати або перевіряти людину з цифрового зображення або відеокадру з джерела відео. Існує кілька методів, в яких працюють системи розпізнавання обличчя, але в цілому вони працюють, порівнюючи вибрані риси обличчя із заданого зображення з обличчями в базі даних . Він також описаний як додаток на основі біометричного штучного інтелекту, який дозволяє однозначно ідентифікувати людину шляхом аналізу шаблонів, заснованих на фактурі та формі обличчя людини

Незважаючи на те, що спочатку це форма комп'ютерних застосувань , останнім часом спостерігається ширше використання на мобільних платформах та інших технологіях, таких як робототехніка. Зазвичай він використовується як контроль доступу в системах безпеки і може порівнюватися з іншими біометричними характеристиками, такими як відбитки пальців або системи розпізнавання райдужної оболонки ока . [4] Хоча точність системи розпізнавання обличчя як біометричної технології нижча за розпізнавання райдужної оболонки та розпізнавання відбитків пальців , вона широко застосовується завдяки безконтактному та неінвазивному процесу. [5] Останнім часом він також став популярним як інструмент комерційної ідентифікації та маркетингу.[6] Інші додатки включають в себе вдосконалене взаємодія людина-комп'ютер, відеоспостереження, автоматичну індексацію зображень та відео-базу даних

8) HTML (англ. HyperText Markup Language)— мова розмітки гіпертексту) — це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет

Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа.

Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегамі, написаними з використанням кутових дужок. Теги, такі як і безпосередньо вводять вміст на сторінку. Інші теги, такі як `` `<input />` `<p>` оточують і надають інформацію про текст документа і можуть включати інші теги як піделементи. Браузери не показують теги HTML, але використовують їх для інтерпретації вмісту сторінки.

HTML може вбудовувати програми, написані на мові сценаріїв, наприклад JavaScript, що впливає на поведінку та вміст веб-сторінок. Включення CSS визначає вигляд і компонування вмісту. World Wide Web Consortium (W3C), які

супроводжують як HTML і CSS стандартів, заохочує використання CSS над явним презентаційним HTML з 1997 року.[1]

HTML впроваджує засоби для:

- створення структурованого документа шляхом позначення структурного складу тексту: заголовки, абзаци, списки, таблиці, цитати та інше;
- отримання інформації із Всесвітньої мережі через гіперпосилання;
- створення інтерактивних форм;
- включення зображень, звуку, відео, та інших об'єктів до тексту.

9)CSS (англ. Cascading Style Sheets) — це спеціальна мова стилю сторінок[en], що використовується для опису їхнього зовнішнього вигляду. Самі ж сторінки написані мовами розмітки даних.

CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript[1].

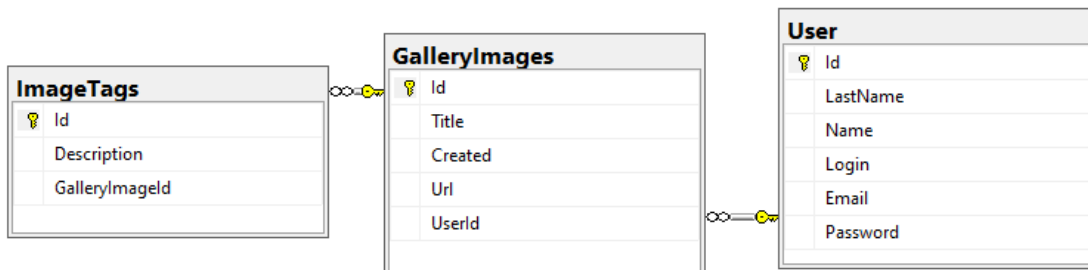
Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі. CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі — сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для принтерів, мобільних пристроїв тощо.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

Розділ 2. Програмна реалізація

2.1. База даних проекту



Таблиці:

«**User**» – містить основну інформацію про користувачів системи.

Id	Ідентифікатор користувача
Email	Електронна адреса
Password	пароль
Name	Ім'я
Login	логін
Last Name	Прізвище

«**GalleryImages**» - таблиця фото.

Title	Назва
Created	Дата створення
Url	Шлях
UserId	Ідентифікатор юзера

«**ImageTags**» - містить можливі теги.

Id	Ідентифікатор
Description	Опис
GalleryImageId	Ідентифікатор галереї

2.2. Опис контролерів системи

«**AccountController**» – Перший контролер проекту, який виконує наступні дії:

Login user
Register user
Authenticate
Logout

«**GalleryController**» – контролер виконує такі дії:

Detail
Delete

«**ImageController**» – контролер Виконує такі дії:

Upload New Image

«**PeopleController**» – контролер виконує такі дії:

Create
Detect

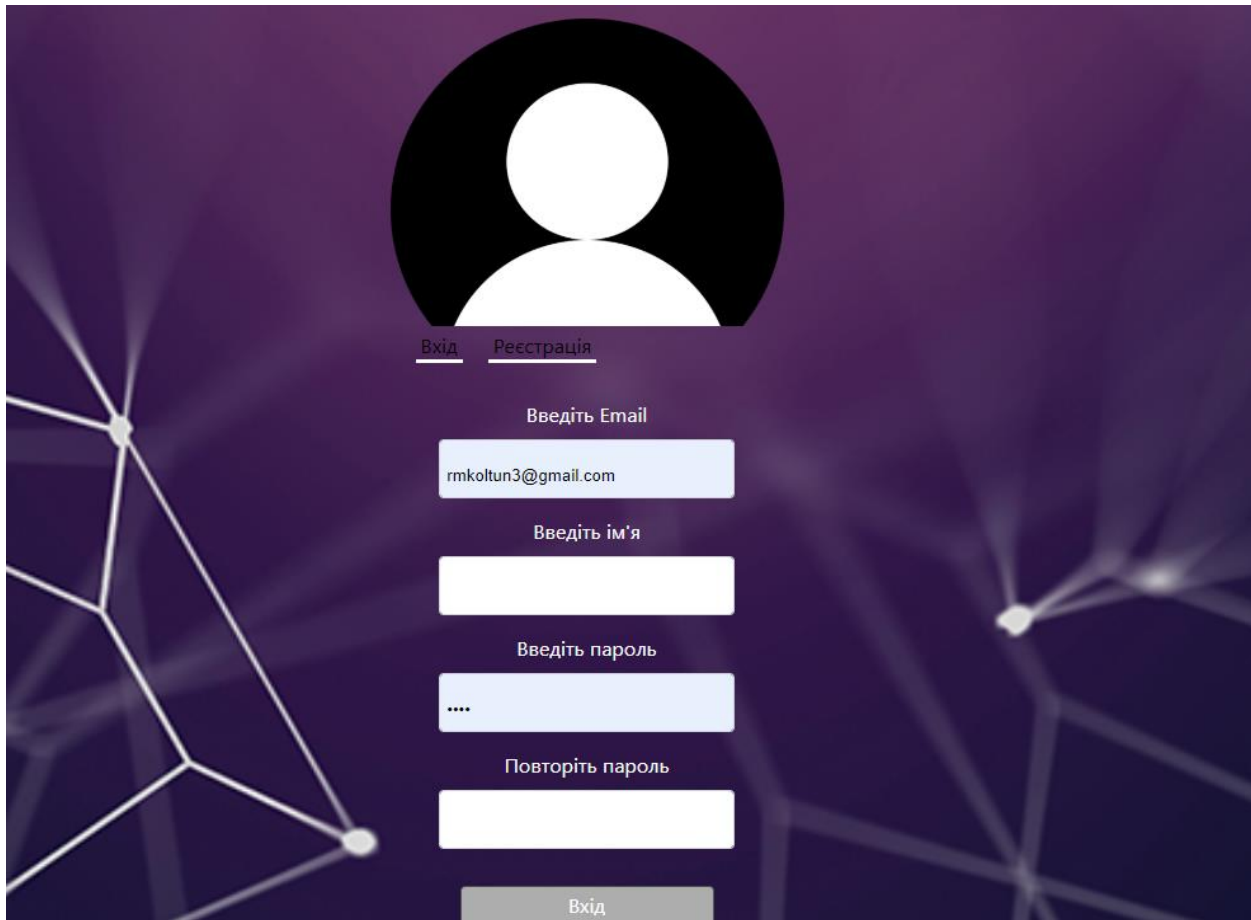
Розділ 3. Сценарій роботи системи

3.1. Реєстрація та вхід в систему

Користувач має можливість зареєструватися в системі ввівши наступні дані:

Обов'язкові:

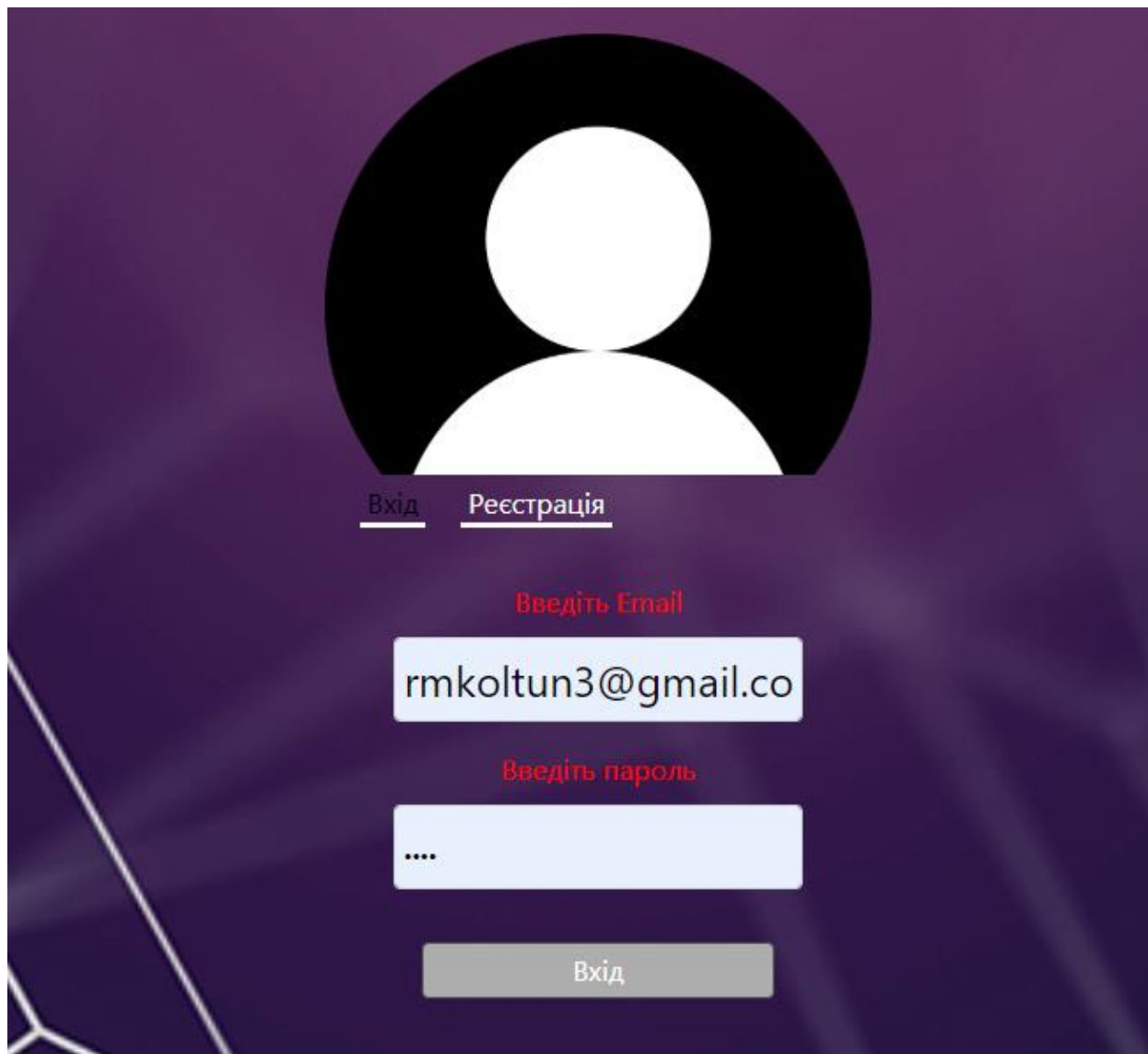
- Логін (буде використовуватись при коментуванні як ім'я користувачі)
- Емейл (потрібен для входу)
- Пароль та підтвердження паролю (потрібні для входу)



The screenshot shows a web interface for user registration and login. At the top, there is a large black circle with a white person icon inside. Below it are two tabs: 'Вхід' (Login) and 'Реєстрація' (Registration), with 'Реєстрація' being the active tab. The form contains the following fields and labels:

- Label: 'Введіть Email' (Enter Email). Input field: 'rmkoltun3@gmail.com'.
- Label: 'Введіть ім'я' (Enter name). Input field: (empty).
- Label: 'Введіть пароль' (Enter password). Input field: '....'.
- Label: 'Повторіть пароль' (Repeat password). Input field: (empty).
- At the bottom, there is a 'Вхід' (Login) button.

Користувач має можливість увійти в систему за допомогою свого логіну (емейлу введеного при реєстрації) та паролю. Для подання цього функціоналу клієнту реалізована наступна форма



Після входу у свій обліковий запис користувач може завантажити фотографії де він має вказати назву людини на цій фотографії у полі Title та може поставити теги. Після цього користувач може переглянути галерею в зручному для нього вигляді, також користувач може вийти з свого облікового запису.

3.2 Завантаження В програмі можна завантажити фото. При завантаженні потрібно вказати назву та теги.

Title

Порошенко Петро Олекс

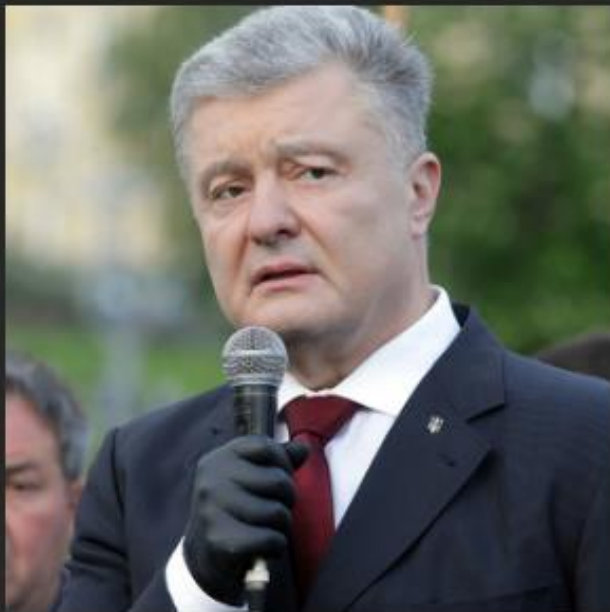
Tags

Петро 1

Browse for Image

Submit

Після завантаження фотографії face recognition розпізнає обличчя та зберігає його Azure назву



Title: Порошенко Петро Олексійович AzureName: Порошенко Петро Олексійович

[Back to Gallery](#)

[Delete](#)

Після завантаження ще фотографій із тим самим користувачем і будь-якою назвою, Azure назва обличчя буде така ж сама як і при першому збереженні



Title: Порошенко AzureName: Порошенко Петро Олексійович

[Back to Gallery](#)

[Delete](#)

Висновки

Дана робота присвячена розробці додатку на платформі .NET. Під час написання даної програми було використано чимало передових технологій, як для інтерфейсу користувача, поліпшення продуктивності за стосунку так і для його публікації.

Розроблений сайт повністю відповідає поставленим вимогам. Основною метою якого було запам'ятовування лиць на фотографіях і виявлення їх в подальших зображеннях.

Список використаної літератури

1. <http://htmlbook.ru/> - технічна підтримка з HTML5, CSS3.
2. <http://www.w3schools.com/> - технічна підтримка з HTML5, CSS3.
3. <https://learn.javascript.ru/> - підтримка з JS.
4. «CLR via C#: (4th Edition)» by Jeffrey Richter – підтримка з C#.
5. MSDN - .net підтримка.
6. <https://stackoverflow.com/> - підтримка під час розробки.
7. <https://metanit.com/> - підтримка з ASP.NET MVC.
8. https://uk.wikipedia.org/wiki/Microsoft_Azure
9. https://en.wikipedia.org/wiki/Facial_recognition_system
10. <https://uk.wikipedia.org/wiki/CSS>
11. <https://uk.wikipedia.org/wiki/HTML>