

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)
на тему:

«Вебдодаток для визначення віку людини»

Виконав студент групи ЗКНк-21-1
НАКОНЕЧНИЙ Нікіта Володимирович

Керівник роботи:
ЛЕВКІВСЬКИЙ Віталій Леонідович

Рецензент:
МАРЧУК Галина Вікторівна

Житомир – 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

Марина ГРАФ

«25» січня 2024 р.

ЗАВДАННЯ

на кваліфікаційну роботу освітнього ступеня «бакалавр»
за спеціальністю 122 «Комп'ютерні науки»
(освітня програма «Комп'ютерна графіка та розробка ігор»)

Здобувач вищої освіти: **НАКОНЕЧНИЙ Нікіта Володимирович**

Керівник роботи: **ЛЕВКІВСЬКИЙ Віталій Леонідович**

Тема роботи: **«Вебдодаток для визначення віку людини»**,

затверджена наказом закладу вищої освіти від **«25» січня 2024 р., №23/с**

Термін здачі закінченої роботи: **«10» червня 2024 р.**

Вихідні дані роботи (зазначається предмет і об'єкт дослідження): об'єктом дослідження є процес розробки веб-систем з використанням бібліотек комп'ютерного зору для аналізу зображень обличчя. Предмет дослідження: особливості розробки багатокористувацької веб-системи з API-модулем для визначення віку людини на основі аналізу фото обличчя.

Консультанти кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв

Календарний план

№ з/п	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка задачі, пошук та аналіз аналогічних розробок, опрацювання літературних джерел	25.01.2024 – 01.03.2023	Виконано
2	Планування структури програми	02.03.2024 – 20.03.2024	Виконано
3	Написання програмного коду	21.03.2024 – 01.05.2024	Виконано
4	Тестування роботи програми	02.05.2024 – 12.05.2024	Виконано
5	Оформлення пояснювальної записки	13.05.2024 – 31.05.2024	Виконано
6	Попередній захист	10.06.2024	Виконано

Здобувач вищої освіти **Нікіта НАКОНЕЧНИЙ**

Керівник **Віталій ЛЕВКІВСЬКИЙ**

РЕФЕРАТ

Кваліфікаційна робота для здобуття освітнього ступеня “бакалавр” складається з програмного вебдодатку для розпізнавання віку людини за фото та пояснювальної записки обсягом 65 сторінок друкованого тексту. Записка містить 24 рисунки та 11 таблиць. Список використаних джерел містить 15 найменувань.

Метою роботи є розробка вебдодатку з двома компонентами: багатокористувацького веб-сайту та API-модуля з можливостями розпізнавання віку людини за її обличчям на фото.

В роботі було обґрунтовано актуальність даної системи, сформульовано мету, об’єкт і предмет дослідження, сформульовано завдання до роботи. На етапі проектування та розробки було визначено функціональні вимоги системи, розроблено UML-діаграми, побудовано базу даних та здійснено опис розробки модуля для розпізнавання віку людини. В роботі наведено інструкцію щодо встановлення та використання даної системи. Описано помилки, що виникали на етапі встановлення та використання.

КЛЮЧОВІ СЛОВА: МАШИННЕ НАВЧАННЯ, МЕТОДИ КОМП’ЮТЕРНОГО ЗОРУ, РОЗПІЗНАВАННЯ ВІКУ, РОЗРОБКА ВЕБ-СИСТЕМ ТА API-СЕРВІСІВ.

					КН.КР.Б – 122 – 24 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Н.В. НАКОНЕЧНИЙ				Вебдодаток для визначення віку людини Пояснювальна записка	Літ.	Арк.	Архувів
Керівник	В.Л. ЛЕВКІВСЬКИЙ						4	63
Рецензент	Г.В. МАРЧУК					Житомирська політехніка, група ЗКНк-21-1		
Н. контр.								
Зав. каф.	М.С. ГРАФ							

ABSTRACT

The bachelor degree work consists of the software code of the website and the API module for recognizing the age of a person from a photo and an explanatory note of 65 pages of printed text. The note contains 24 figures and 11 tables. The list of used sources contains 15 items.

The purpose of the work is to develop a web system with two components: a multi-user website and an API module with the ability to recognize the age of a person by his face in a photo.

In the work, the relevance of this system was substantiated, the goal, object and subject of the research were formulated, the task was set, and the tasks for the work were formulated. At the stage of design and development, the functional requirements of the system were determined, UML diagrams were developed, a database was built, and a description of the development of the module for human age recognition was carried out. The work contains instructions for installing and using this system. Errors that occurred at the stage of installation and use are described.

KEYWORDS: MACHINE LEARNING, COMPUTER VISION METHODS, AGE RECOGNITION, DEVELOPMENT OF WEB SYSTEMS AND API SERVICES.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ЗАСОБІВ І ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБ-САЙТІВ ДЛЯ РОЗПІЗНАВАННЯ ВІКУ ЛЮДИНИ ЗА ФОТО	10
1.1. Постановка задачі розробки.....	10
1.2. Огляд та порівняння бібліотек комп'ютерного зору для вирішення задачі	13
1.3. Вибір і обґрунтування засобів розробки.....	18
Висновки до першого розділу.....	20
РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА ВЕБДОДАТКУ	22
2.1. Визначення функціональних вимог системи	22
2.2. Проектування багатокористувацької системи для розпізнавання віку	24
2.3. Проектування бази даних	31
2.4. Розробка API-модуля для розпізнавання віку.....	36
Висновки до другого розділу	40
РОЗДІЛ 3. ВСТАНОВЛЕННЯ, ВИКОРИСТАННЯ ТА ОСОБЛИВОСТІ ТЕСТУВАННЯ ВЕБДОДАТКУ	42
3.1. Встановлення та налаштування API-модуля для розпізнавання віку людини.....	42
3.2. Структура інтерфейсу та опис роботи з додатком	46
3.3. Тестування роботи системи	53
Висновки до третього розділу.....	55
ВИСНОВКИ.....	56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface

ML – Machine Learning

DL – Deep Learning

СКБД – Система керування базами даних

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. Розробка веб-сайту з API-модулем для розпізнавання віку людини за її фото має кілька важливих аспектів актуальності, які обумовлюють її значення в сучасному світі. До таких аспектів віднесемо *розвиток технологій штучного інтелекту та машинного навчання*: використання алгоритмів глибокого навчання дозволяє з високою точністю визначати вік людини за її зображенням; *безпека та контроль доступу*: можливість обмеження доступу до певних послуг або продуктів за віковим критерієм; *маркетинг та персоналізація*: визначення віку користувачів допомагає компаніям краще розуміти свою аудиторію і створювати персоналізовані пропозиції та рекламу, що підвищує ефективність маркетингових кампаній; *соціальні мережі та розважальні платформи*: використання технології для забезпечення безпеки неповнолітніх, автоматично виявляючи користувачів, які намагаються реєструватися, видаючи себе за старших.

Серед інших сфер застосування подібних систем можуть бути медицина та охорона здоров'я, розваги та ігри, освіта і навчання. Користь систем розпізнавання віку людини у медичних дослідженнях та для покращення обслуговування пацієнтів полягає, наприклад, для швидкої оцінки віку пацієнтів без необхідності звертатися до документів. У сфері онлайн-ігор можна застосовувати розпізнавання віку для надання контенту, відповідного віку користувача, що забезпечує кращий користувацький досвід та дотримання нормативних вимог. У освітніх платформах технологія може використовуватись для адаптації навчальних матеріалів під вік учнів, забезпечуючи більш ефективний процес навчання.

Враховуючи перелічені аспекти, розробка веб-сайту з API-модулем для розпізнавання віку людини за її фото має широке застосування і може задовольнити потреби різних сфер, забезпечуючи безпеку, персоналізацію та ефективність надання послуг.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

Розробка веб-сайту з API-модулем для розпізнавання віку людини за її фото має кілька основних цілей, які обумовлюють її значущість і корисність у різних сферах. Ось ключові цілі:

Точне визначення віку, що потребує використання передових технологій машинного навчання та комп'ютерного зору.

Автоматизація процесу перевірки віку при реєстрації на сайтах, які мають вікові обмеження (наприклад, сайти азартних ігор, алкогольні онлайн-магазини тощо).

Забезпечення *автоматичного контролю доступу* до контенту, що значно зменшує потребу в ручній перевірці.

Підвищення рівня безпеки користувачів, особливо дітей, шляхом автоматичного виявлення та запобігання доступу до контенту, не призначеного для їхнього віку. Це особливо актуально для соціальних мереж і платформ з віковими обмеженнями.

Загалом, основні цілі розробки системи для розпізнавання віку за фото спрямовані на підвищення точності, безпеки, автоматизації, персоналізації та ефективності у різних сферах, що робить цю технологію надзвичайно актуальною та корисною.

Основною метою роботи визначимо як розробка веб-сайту з API-модулем для розпізнавання віку людини за її фото, який забезпечує автоматизоване, точне та швидке визначення віку людини за її зображенням, з метою покращення безпеки, персоналізації, та ефективності у різних сферах застосування.

Для досягнення мети сформулюємо **завдання випускної кваліфікаційної роботи**:

- огляд і аналіз сучасних технологій для розробки веб-систем, що використовують з методи комп'ютерного зору;
- визначення актуальності, мети роботи, об'єкту і предмету дослідження, постановка задачі і визначення технічних вимог до розробки проекту;

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- пошук і порівняння бібліотек комп'ютерного зору. Вибір необхідної бібліотеки та інших засобів розробки;
- проектування системи: визначення функціональних вимог, розробка UML-діаграм;
- проектування системи: розробка структури реляційної бази даних;
- розробка API-модуля розпізнавання віку та веб-сайту для розпізнавання віку;
- розробка інструкцій щодо встановлення та користування.
- тестування веб-системи
- написання документації в формі пояснювальної записки та підготовка до захисту роботи

Об'єкт дослідження: процес розробки веб-систем з використанням бібліотек комп'ютерного зору для аналізу зображень обличчя.

Предмет дослідження: особливості розробки багатокористувацької веб-системи з API-модулем для визначення віку людини на основі аналізу фото обличчя.

Для розробки даної веб-системи було використано два стеки технологій. Для розробки багатокористувацького веб-сайту – мову розробки PHP, фреймворк Yii2 та реляційну базу даних MySQL. Для розробки модуля API для аналізу зображення – мову програмування Python, бібліотеки Flask і DeepFace.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ЗАСОБІВ І ТЕХНОЛОГІЙ РОЗРОБКИ ВЕБ-САЙТІВ ДЛЯ РОЗПІЗНАВАННЯ ВІКУ ЛЮДИНИ ЗА ФОТО

1.1. Постановка задачі розробки

В даному підрозділі визначимо теоретичне і практичне значення розробки багатокористувацької веб-системи для розпізнавання віку, здійснимо постановку основної задачі роботи та наведемо завдання, реалізувавши які, ми виконаємо основну задачу.

Теоретичне значення розробки веб-системи для розпізнавання віку людини по фото може включати такі аспекти, як

- *сприяння в цілому розвитку методів машинного навчання і комп'ютерного зору.* Наприклад, поліпшення точності, швидкості та ефективності моделей, що можуть бути застосовані в інших сферах, таких як розпізнавання облич, емоцій, статі та інших характеристик.
- *використання у психологічних і соціальних дослідженнях,* зокрема вивчення того, як алгоритми розпізнавання віку інтерпретують візуальні дані, може дати цінні інсайти в психології сприйняття віку людини. Це може сприяти кращому розумінню соціальних і когнітивних аспектів сприйняття зовнішності;
- *допомога у формулюванні принципів і стандартів у питаннях етики та конфіденційності,* забезпечуючи захист персональних даних і права користувачів;
- *сприяння розвитку нових міждисциплінарних підходів та методологій,* оскільки веб-система інтегрує знання з різних галузей, таких як інформатика, математика, психологія та соціальні науки;

Іншими теоретичними аспектами цінності роботи можуть бути сприяння розробці і тестуванню нових алгоритмів; розвитку методів роботи з великими даними: оптимальне зберігання, обробки і аналізу великих наборів даних;

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

розробці стандартів і норм для наборів даних та моделей для оцінки їх ефективності та якості.

Таким чином, теоретичне значення розробки веб-системи для розпізнавання віку людини по фото полягає у сприянні науковим дослідженням, розвитку технологій і розумінню етичних і соціальних аспектів використання таких систем.

Система для розпізнавання віку людини по фото може мати *практичні застосування* в різних сферах. Серед них такі застосування як персоналізація сервісів, безпека і контроль доступу, маркетинг і аналітика, розважальні та освітні платформи, контроль та моніторинг у громадських місцях, підтримка програм лояльності.

Говорячи про персоналізацію сервісів, зауважимо, що такі системи можуть використовуватись для адаптації контенту чи послуг відповідно до віку користувача. Наприклад, інтернет-магазини можуть пропонувати товари, які найбільш ймовірно будуть цікавими для певної вікової групи, або стрімінгові платформи можуть рекомендувати відеоконтент, що підходить для відповідної вікової категорії.

В напрямку безпеки і контролю доступу, розпізнавання віку може використовуватись для обмеження доступу до контенту чи послуг, що призначені для певних вікових груп. Це може бути корисно в онлайн-іграх, соціальних мережах, платформах для відеоконтенту або в системах продажу алкоголю чи інших товарів з віковими обмеженнями.

В сфері маркетингу і аналітики компанії можуть використовувати технології розпізнавання віку для збору даних про своїх клієнтів з метою проведення маркетингових досліджень та оптимізації рекламних кампаній. Це допоможе краще розуміти цільову аудиторію та підвищувати ефективність маркетингових стратегій.

Має місце використання таких сервісів в розважальних та освітніх платформах, наприклад, для адаптації навчальних матеріалів чи ігрового

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

контенту відповідно до вікової категорії користувачів, що сприяє підвищенню залученості та ефективності навчання.

Розпізнавання віку може сприяти контролю та моніторингу у громадських місцях. У громадських місцях, таких як торгові центри чи спортивні об'єкти, системи розпізнавання віку можуть використовуватись для автоматичного контролю віку відвідувачів, що може допомогти в управлінні натовпом або дотриманні вікових обмежень для певних заходів.

У роздрібній торгівлі та сфері послуг системи розпізнавання віку можуть використовуватись для персоналізації пропозицій у програмах лояльності, надаючи клієнтам спеціальні знижки або бонуси, що відповідають їх віковій категорії.

Таким чином, розробка системи для розпізнавання віку людини по фото має широкий спектр застосувань, що може підвищити ефективність роботи різних сервісів та забезпечити більш персоналізований підхід до користувачів.

Визначивши практичну значущість розробки та впровадження застосунку для розпізнавання, здійснимо постановку основної задачі даної роботи.

Основна задача роботи: розробити багатокористувацьку веб-систему, яка дозволяє користувачам завантажувати фотографії та отримувати прогнозовані результати розпізнавання віку з використанням технологій машинного навчання.

Для виконання основної задачі роботи сформулюємо наступні підзавдання:

1. Аутентифікація користувачів:

- реалізувати реєстрацію та авторизацію користувачів;
- врахувати необхідність захисту даних та конфіденційність користувачів;

2. Користувацький веб-інтерфейс:

- розробити зручний і інтуїтивно зрозумілий інтерфейс для завантаження фотографій;
- забезпечити можливість перегляду історії завантажених зображень та отриманих результатів;

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Завантаження зображень:

- розробити функції завантаження зображень;
- реалізувати обробку та збереження завантажених зображень на сервері;

4. Аналіз зображень:

- використати бібліотеку DeerpFace для розпізнавання обличчя та прогнозування віку;

5. Відображення результатів:

- реалізувати механізм відображення отриманих результатів розпізнавання віку користувачам; забезпечити точність та зрозумілість поданих результатів;
- реалізувати можливість збереження отриманих результатів аналізу;

6. Функції адміністратора:

- забезпечити адміністраторам можливість управління користувачами та перегляду статистики використання системи;

Отже в даному підрозділі ми визначили теоретичне і практичне значення роботи, здійснили постановку задачі та формулювання підзавдань. Таким чином результатом виконаної роботи є веб-система, що дозволяє користувачам завантажувати свої фотографії для розпізнавання віку з використанням DeerpFace, переглядати результати та історію розпізнавань, з функціями реєстрації, аутентифікації, зручним інтерфейсом та адміністративним модулем.

1.2. Огляд та порівняння бібліотек комп'ютерного зору для вирішення задачі

Для вирішення задачі розпізнавання віку людини за фото розглянемо три сучасні поширені бібліотеки, проаналізуємо їх, порівнявши функціональні можливості, виявивши переваги і недоліки.

Бібліотека OpenCV [4]. OpenCV — це найстарша та найпопулярніша бібліотека комп'ютерного зору з відкритим кодом. Це кросплатформна бібліотека, яка підтримує Windows, Linux, Android і macOS і може

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

використовуватися різними мовами, такими як Python, Java, C++ тощо. Використовує модель CUDA для GPU. Розроблений компанією Intel і на даний момент її можна безкоштовно використовувати за ліцензією BSD з відкритим кодом. Бібліотека містить деякі моделі, які можна перетворити на моделі TensorFlow, також володіє інструментами для 2D і 3D функцій.

Можливості і використання для задач комп'ютерного зору:

- Попередня обробка зображень та виділення ознак
- Інтеграція передтренованих моделей або власних моделей глибокого навчання
- Розширена документація та підтримка спільноти;
- Розпізнавання облич
- Розпізнавання жестів
- Розуміння руху
- Взаємодії людини з комп'ютером
- Виявлення об'єктів
- Сегментація та розпізнавання

Переваги:

- Широко використовується і добре документована;
- Підтримка різних мов програмування, таких як C++, Python та Java;
- Відмінна для задач попередньої обробки зображень;

Недоліки:

- Не спеціалізується на глибокому навчанні, тому моделі визначення віку необхідно інтегрувати з інших фреймворків;
- Для реалізації та навчання моделей потребує розуміння і написання програмного коду вручну.

Фреймворк TensorFlow [5]. Створений командою GoogleBrain та випущена у листопаді 2015 року. Призначена для полегшення процесу створення моделей штучного інтелекту. TensorFlow має індивідуальні рішення: *TensorFlow.js* – бібліотека JavaScript для навчання та розгортання моделей у браузері та на Node.js; *TensorFlow Lite* – полегшена бібліотека для розгортання моделей на

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

мобільних і вбудованих пристроях. Також володіє платформою *TensorFlow Hub* для створення, навчання, спільного і повторного використання моделей глибокого навчання (такі як BERT і Faster R-CNN), пошуку готових до розгортання моделей для власних проєктів штучного інтелекту.

Можливості і використання для задач комп'ютерного зору:

- Виявлення та класифікація об'єктів на зображеннях
- Сегментація зображень (розділення зображення на семантичні області)
- Розпізнавання облич, ідентифікація осіб
- Оптичне розпізнавання символів (OCR)

Переваги:

- Сучасні можливості глибокого навчання
- Розширена документація та навчальні матеріали
- Добре інтегрується з іншими інструментами та сервісами Google

Недоліки:

- Більш складний у вивченні для початківців
- Може бути надмірним для такої задачі як визначення віку

Бібліотека PyTorch [6]. PyTorch – бібліотека машинного навчання з відкритим вихідним кодом для Python, розроблена в дослідницькою групою AI Facebook. Використовує динамічні обчислення, що забезпечує більшу гнучкість у створенні складних архітектур. Pytorch використовує основні концепції Python, такі як класи, структури та умовні цикли, і сумісний із C++. Бібліотека підтримує як CPU, так і GPU обчислення.

Можливості і використання для задач комп'ютерного зору:

- Моделі оцінки зображення
- Сегментація зображення
- Класифікація зображень
- Динамічні обчислювальні графи: Дозволяє більше гнучкості та легше налагодження.
- Передтреновані моделі: Доступ до багатьох передтренованих моделей через TorchVision.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

- Спільнота та екосистема: Сильна спільнота та зростаюча екосистема.

Переваги:

- Легший у вивченні та використанні, ніж TensorFlow, для багатьох користувачів
- Відмінно підходить для досліджень та експериментів
- Забезпечує динамічні обчислення, що може бути більш інтуїтивно зрозумілим

Недоліки:

- Менш розвинена екосистема порівняно з TensorFlow
- Може бути повільнішим при виконанні робочих процесів, якщо не оптимізований належним чином

Фреймворк DeepFace [7]. Deepface – це легкий фреймворк для розпізнавання облич і аналізу атрибутів обличчя (вік, стать, емоції та раса) для Python. Це гібридний фреймворк для розпізнавання облич, що охоплює найсучасніші моделі: VGG-Face, FaceNet, OpenFace, DeepFace, DeepID, ArcFace, Dlib, SFace і GhostFaceNet.

Можливості і використання для задач комп'ютерного зору:

- Використання сучасних передтренованих моделей
- Інтеграція з іншими бібліотеками Python (Keras, TensorFlow)
- Комплексний аналіз облич, включаючи визначення віку

Переваги:

- Висока точність визначення віку
- Добре підходить для досліджень та розробок
- Підтримує кілька фреймворків та моделей

Недоліки:

- Може бути ресурсомістким
- Потребує значної обчислювальної потужності для навчання та інференсу

Бібліотека Dlib [8]. Dlib — це сучасний інструментарій C++, що містить алгоритми машинного навчання та інструменти для створення складного програмного забезпечення на C++ для вирішення прикладних задач штучного

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

інтелекту. Використовується як у промисловості, так і в наукових колах у широкому діапазоні областей, включаючи робототехніку, вбудовані пристрої, мобільні телефони та великі високопродуктивні обчислювальні середовища.

Можливості і використання для задач комп'ютерного зору:

- Машинне навчання: Надає набір алгоритмів машинного навчання.
- Виявлення облич: Потужні інструменти для виявлення та вирівнювання облич.
- Передтреновані моделі: Містить передтреновані моделі для розпізнавання облич, які можна адаптувати для визначення віку.

Переваги:

- Високоякісне виявлення та вирівнювання облич.
- Легко використовувати з Python.
- Добре підходить для початкового тестування та прототипування моделей.

Недоліки:

- Обмежена підтримка глибокого навчання порівняно з іншими фреймворками.
- Може вимагати інтеграції з іншими бібліотеками для більш просунутих задач визначення віку.

Таким чином, визначимо, залежно від конкретних вимог які бібліотеки підходять для нашого проекту:

- Для простоти та легкості використання: Dlib
- Для можливостей глибокого навчання: TensorFlow або PyTorch.
- Для традиційних задач комп'ютерного зору та попередньої обробки: OpenCV
- Для балансу між легкістю використання та продуктивністю: DeepFace

Отже, в даному підрозділі нами було здійснено огляд сучасних бібліотек і фреймворків для машинного зору, зокрема для розв'язку задачі визначення віку людини за фото і визначено залежно від конкретних вимог які бібліотеки підходять для нашого проекту. Виходячи з умов і цілей нашого проекту, було вирішено використати фреймворк DeepFace, який безпосередньо призначений

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

для розпізнавання облич та їх атрибутів (в т.ч. віку), а також є продуктивним та легким у використанні.

1.3. Вибір і обґрунтування засобів розробки

В даному підрозділі визначимо засоби розробки та обґрунтування їх вибору. В таблиці 1.2 наведемо перелік засобів розробки для двох застосунків, з яких складається веб-система.

Таблиця 1.1

Мови та бібліотеки для розробки веб-системи розпізнавання віку людини

Назва застосунку	Мова або інструмент розробки	Призначення
Веб-сайт “Розпізнавання віку людини за фото”	PHP 8.0	Мова розробки серверних сценаріїв
	MySQL (MariaDB)	Реляційна СКБД
	Yii2	PHP-фреймворк для розробки веб- орієнтованих інформаційних систем
API-застосунок для розпізнавання	Python 3.12	Мова розробки
	Flask 3.0.3	Веб-фреймворк для розробки веб- застосунків та API- сервісів
	deepface 0.0.92	Бібліотека для створення і використання моделей

		глибинного навчання по розпізнаванню облич та їх властивостей і частин
--	--	---

Обґрунтуємо використання PHP-фреймворку Yii2. Вибір даного фреймворку пояснюється в першу чергу наявністю багатьох функціональних можливостей та компонентів: аутентифікація користувачів, авторизація, робота з формами, валідація даних тощо. Це знижує час розробки і дозволяє зосередитися на основній функціональності системи. Даний фреймворк володів високою продуктивністю та оптимізацією коду, що дозволяє створювати швидкі та ефективні веб-додатки і є важливим для забезпечення гарного користувацького досвіду у багатокористувацькій системі. Серед інших позитивних сторін – наявність вбудованих засобів для безпеки (захист від XSS, CSRF та SQL-ін'єкцій), проста інтеграція з MySQL.

Рішення щодо вибору СКБД MySQL ґрунтується в першу чергу на широкому використанні та підтримці, наявності великої спільноти, великої кількості ресурсів для навчання та документації. MySQL має ряд значних переваг: висока продуктивність і масштабованість, підтримка транзакцій, надійність.

Для API-сервісу, що призначений для застосування алгоритму розпізнавання облич, рекомендується вибрати стек технологій, що на сьогодні є найпоширенішими: мова програмування Python, фреймворк Flask і бібліотека DeepFace.

Python є однією з найпопулярніших мов програмування завдяки своїй простоті і читабельності. Це знижує поріг входу для нових розробників і прискорює процес розробки. Python володіє великою екосистемою бібліотек і фреймворків для різних завдань. Для даної роботи ми використовуємо саме бібліотеки Python для машинного навчання, аналізу даних і веб-розробки.

Flask є легким і гнучким веб-фреймворком, що дозволяє швидко розробляти веб-сайти або веб-служби. Flask підходить як для невеликих

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

проектів, так і для великих систем, що дозволяє легко масштабувати додаток в міру його розвитку.

DeerFace – бібліотека з реалізацією алгоритмів розпізнавання облич. Забезпечує доступ до передових моделей глибокого навчання для розпізнавання облич і віку, що робить її ідеальним вибором для завдань, пов'язаних з аналізом зображень. DeerFace легко інтегрується з наявними Python-застосунками, що спрощує розробку і впровадження функціональності розпізнавання віку.

Таким чином для розробки багатокористувацької системи для розпізнавання віку по фото людини було вибрано 2 стеки технологій відповідно для розробки багатокористувацького веб-сайту та API-застосунку для розпізнавання. PHP, Yii2 та MySQL забезпечують надійну та швидку роботу серверної частини веб-сайту з безпечним управлінням даними користувачів, тоді як Python, Flask і DeerFace дозволяють легко реалізувати функціональність розпізнавання облич та віку з високою точністю.

Висновки до першого розділу

В першому розділі роботи було визначено теоретичне і практичне значення роботи, здійснено постановку задачі та формулювання підзавдань на виконання проекту. Результатом виконаної роботи є веб-система, що дозволяє користувачам завантажувати свої фотографії для розпізнавання віку з використанням DeerFace, переглядати результати та історію розпізнавань, з функціями реєстрації, аутентифікації, зручним інтерфейсом та адміністративним модулем.

До теоретичної складової роботи віднесено огляд сучасних бібліотек і фреймворків для машинного зору, зокрема для розв'язку задачі визначення віку людини за фото і визначено залежно від конкретних вимог які бібліотеки підходять для нашого проекту. Виходячи з умов і цілей нашого проекту, було вирішено використати фреймворк DeerFace, який безпосередньо призначений для розпізнавання облич та їх атрибутів (в т.ч. віку), а також є продуктивним та

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

легким у використанні. Таким чином для розробки багатокористувацької системи для розпізнавання віку по фото людини здійснено вибір і обґрунтовано 2 стеки технологій відповідно для розробки багатокористувацького веб-сайту та API-застосунку для розпізнавання.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА ВЕБДОДАТКУ

2.1. Визначення функціональних вимог системи

В даному підрозділі визначимо функціональні вимоги для користувачів, адміністраторів та системи, а також основні взаємодії акторів з системою та наведемо діаграму варіантів використання.

Спочатку визначимо функціональні вимоги для користувачів:

- 1) реєстрація та аутентифікація. Включають в себе такі функції:
 - реєстрація нового користувача
 - вхід в систему для зареєстрованих користувачів
 - відновлення паролю
- 2) управління профілем:
 - редагування профілю користувача
 - зміна паролю
- 3) розпізнавання віку:
 - завантаження фото для розпізнавання
 - обробка фото з використанням алгоритмів розпізнавання віку
 - отримання результатів розпізнавання віку
- 4) історія розпізнавань:
 - перегляд історії завантажених фото та отриманих результатів
 - видалення записів з історії розпізнавань
- 5) повідомлення та оповіщення:
 - отримання повідомлень про успішне розпізнавання
 - повідомлення про помилки та невдалі спроби розпізнавання.

Функціональні вимоги для адміністратора наступні:

- 1) управління користувачами:
 - перегляд списку користувачів;
 - редагування профілю користувачів;
 - видалення користувачів;
- 2) здійснення моніторингу та перегляд статистики:

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

- перегляд загальної статистики використання системи: кількість користувачів та їх завантажених фото;
- перегляд статистики розпізнавань: кількість успішних та невдалих спроб;

Технічні функції системи:

1) функції безпеки:

- захист даних користувачів;
- шифрування збережених фото та історії розпізнавань;
- захист від несанкціонованого доступу;

2) функції масштабування та продуктивності:

- підтримка багатокористувацької взаємодії з системою;
- забезпечення високої швидкості обробки запитів та фото;

3) інтерфейс користувача:

- зручний та інтуїтивно зрозумілий інтерфейс для користувачів та адміністраторів;
- адаптивний дизайн для роботи на різних пристроях (комп'ютери, смартфони, планшети);

4) внутрішня інтеграція даної системи з API-застосунком для визначення віку людини по фото

Здійснимо розробку, опис та вивід діаграми варіантів використання. В системі передбачено наявність 2 акторів: “Адміністратор” та “Користувач”. Адміністратори системи унаслідують роль користувачів, тобто володіють в тому числі і всіма функціями та можливостями користувачів. На рисунку 2.1 наведено діаграму варіантів використання системи:

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

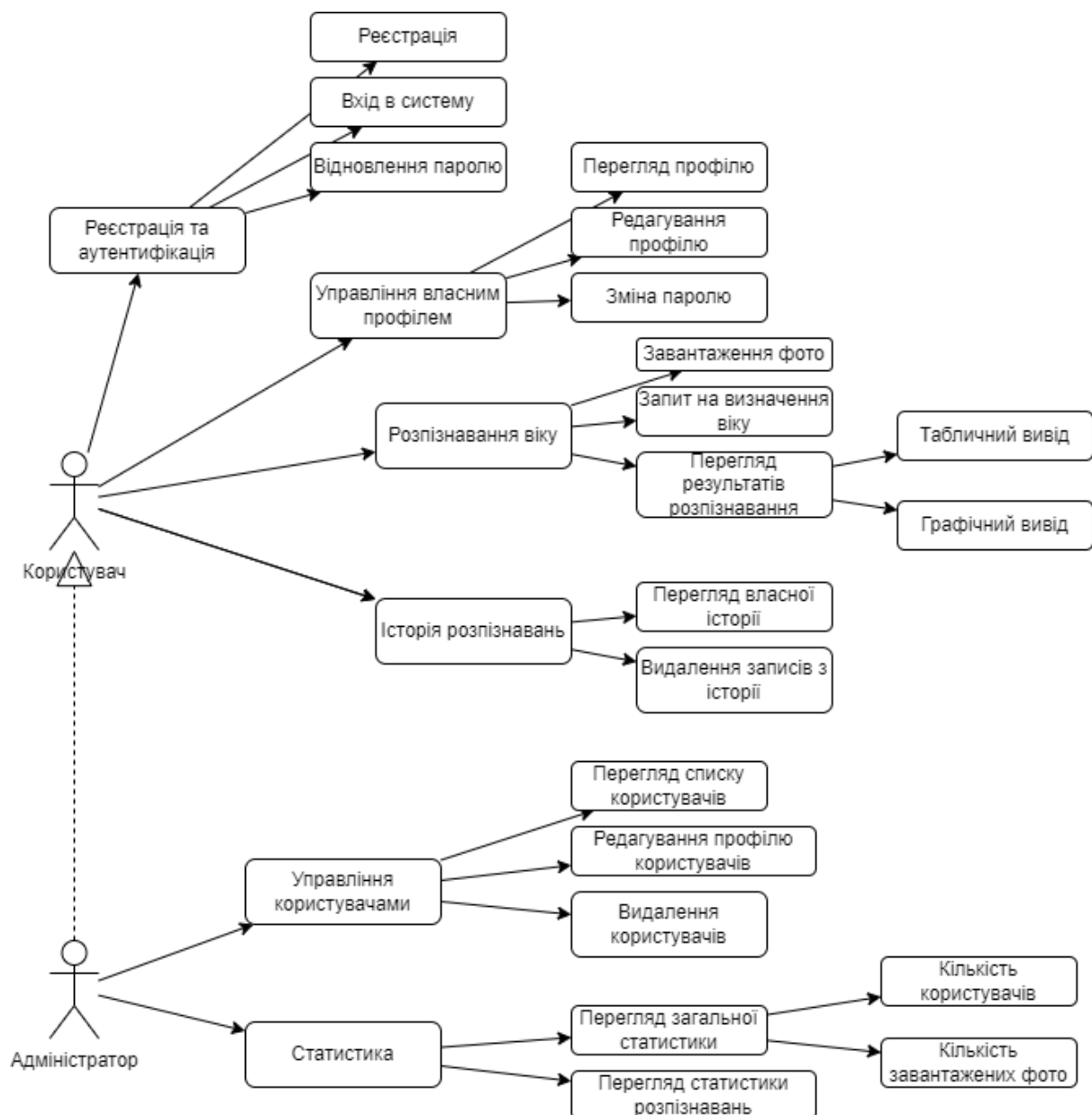


Рис. 2.1. UML-діаграма варіантів використання веб-системи для розпізнавання віку по фото

2.2. Проектування багатокористувацької системи для розпізнавання віку

В даному підрозділі побудуємо основні UML-діаграми для розробки веб-системи:

Таблиця 2.1

UML-діаграми та їх основне призначення для побудови веб-системи

Діаграма	Основне призначення	Опис
Варіантів	Визначення вимог	Забезпечує загальне уявлення про

використання	і функціональності	те, які функції повинна підтримувати система, і хто буде взаємодіяти з нею
Активності	Моделювання робочих процесів	Детально описує робочі процеси та потоки даних в системі, допомагаючи оптимізувати їх
Послідовності	Визначення взаємодії	Детально описує порядок взаємодії між об'єктами, що важливо для розробки та налагодження програмного забезпечення
Класів	Структурування системи	Надає чітку структуру системи, визначаючи, як дані будуть зберігатися і оброблятися, що є основою для об'єктно-орієнтованого програмування

Діаграма активності для багатокористувацької веб-системи для розпізнавання віку по фото може відображати основні кроки, які виконують користувачі та адміністратори під час роботи з системою. Вона також може показати потоки даних та основні операції системи.

Опишемо основні кроки щодо взаємодії зареєстрованого користувача з системою (рис. 2.2). Користувач заповнює форму реєстрації та створює обліковий запис. Після реєстрації користувач вводить свої облікові дані для здійснення входу в систему. Аутентифікований користувач з допомогою візуальних інструментів завантажує фото для розпізнавання віку. Алгоритми системи обробляють фото та визначають вік користувача на фото. Якщо отримано успішні результати аналізу, система генерує відповідь з даними аналізу. При неуспішних результатах генерується відповідь з даними про помилку. Відповідь відправляється користувачеві та відображається структуровано у вигляді таблиці та графічно на самому зображенні. Оброблене зображення записується в історію. Історія дає можливість переглянути дані щодо попередніх розпізнавань. Користувач може видаляти записи з власної історії розпізнавань. Вихід з системи завершує відповідну сесію.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

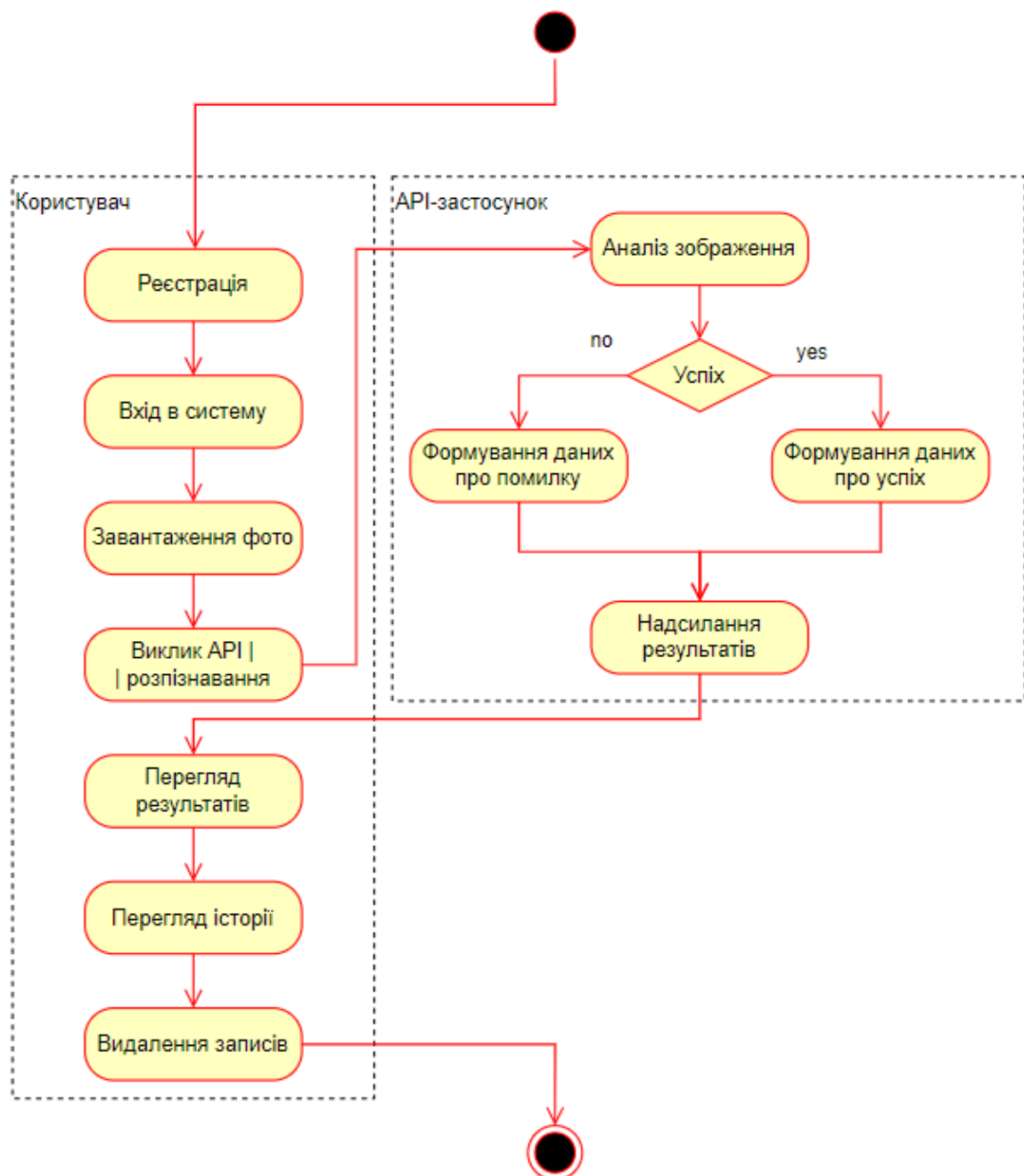


Рис. 2.2. Діаграма активності для зареєстрованого користувача системи

Опишемо основні кроки щодо взаємодії адміністратора з системою (рис. 2.3). Адміністратор вводить свої облікові дані та здійснює вхід в систему. В системі для адміністратора є можливість управління користувачами, а саме перегляд списку користувачів, редагування профілю, видалення користувачів. Окремо визначений набір діяльностей для перегляду статистики, що стосується системи в цілому (кількість користувачів, кількість завантажених і проаналізованих фото), так і перегляд статистики розпізнавань. Після

завершення роботи адміністратор здійснює вихід з системи, завершуючи відповідну сесію.

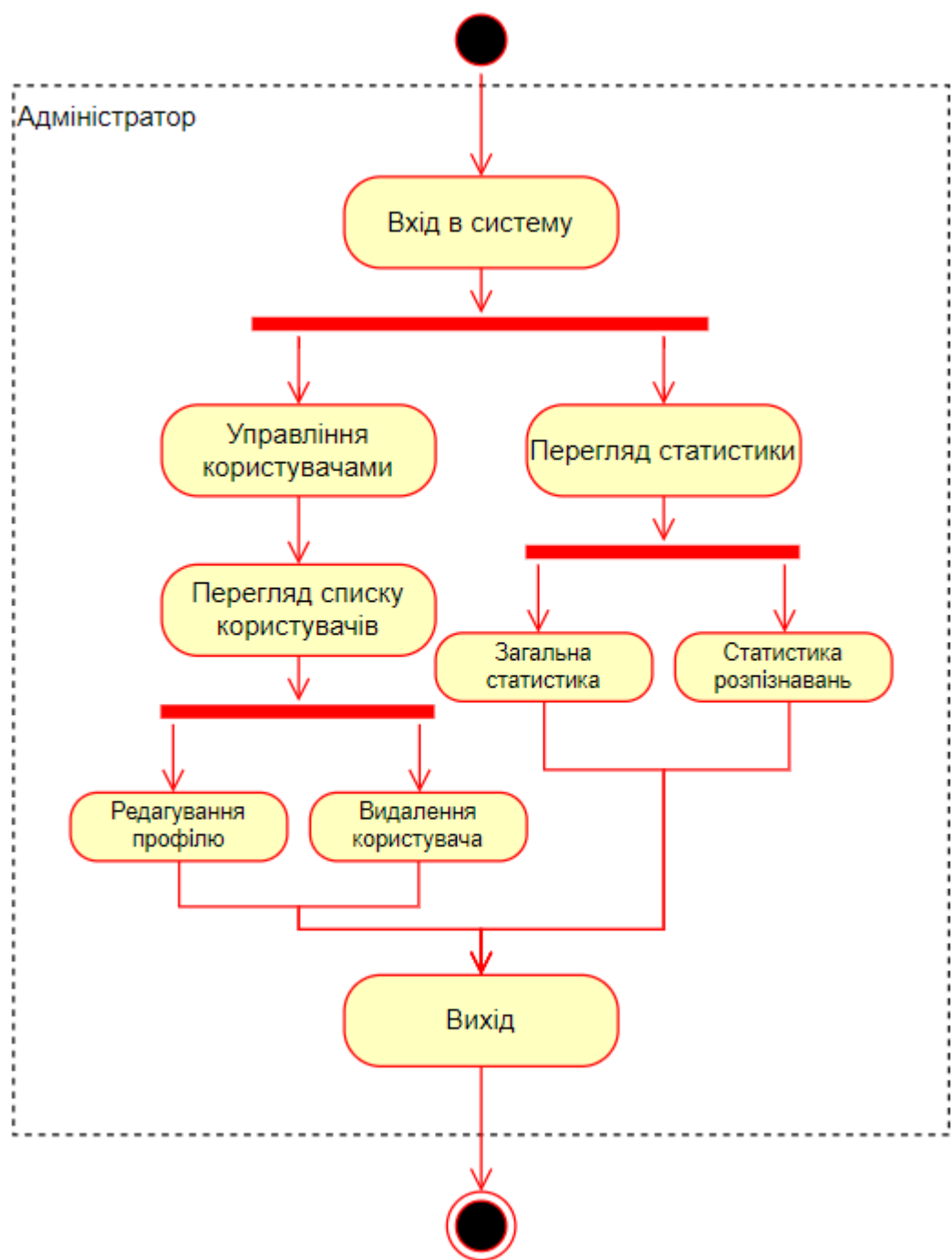


Рис. 2.3. Діаграма активності для адміністратора системи

Наступним етапом проектування веб-системи для аналізу зображень і розпізнавання віку людини є діаграма послідовності, яка показує, як взаємодіють різні частини системи під час виконання певного процесу. Для багатокористувацької веб-системи для розпізнавання віку по фото діаграма

послідовності може включати взаємодії між такими компонентами: користувач, веб-інтерфейс, сервер системи, API-застосунок для розпізнавання та база даних.

На рисунку 2.4 наведено діаграму послідовності взаємодій між даними компонентами. Здійснимо детальний опис діаграми:

Взаємодії по додаванню нового зображення з фото:

- 1) користувач через веб-інтерфейс завантажує фото з метою подальшого розпізнавання віку;
- 2) веб-інтерфейс отримує фото та відправляє запит на збереження до сервера;
- 3) сервер отримує фото та зберігає його у своїй файловій системі;
- 4) сервер надсилає запит в БД для додавання нового запису про фото;
- 5) БД додає запис про фото і надсилає підтвердження на сервер;
- 6) сервер генерує сторінку з переглядом фото і надсилає її в веб-інтерфейс;
- 7) веб-інтерфейс виводить сторінку з фото для перегляду в браузері;

Взаємодії по аналізу зображення та визначення віку по фото:

- 8) користувач надсилає запит на аналіз даного фото;
- 9) веб-інтерфейс надсилає запит серверу;
- 10) сервер надсилає фото і запит на його аналіз API-сервісу;
- 11) API-сервіс обробляє фото, визначає вік та повертає результат на сервер;
- 12) сервер отримує результат від API та зберігає результат у базі даних, надсилаючи відповідних запит у БД;
- 13) сервер відправляє результат даних розпізнавання на веб-інтерфейс;
- 14) веб-інтерфейс виводить результат розпізнавання користувачу;

Взаємодії по перегляду історії розпізнавань:

- 15) користувач через веб-інтерфейс запитує історію своїх розпізнавань;
- 16) веб-інтерфейс відправляє запит на сервер для отримання історії;
- 17) сервер запитує і отримує історію розпізнавань з бази даних та відправляє історію розпізнавань на веб-інтерфейс;
- 18) веб-інтерфейс відображає історію розпізнавань користувачу.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

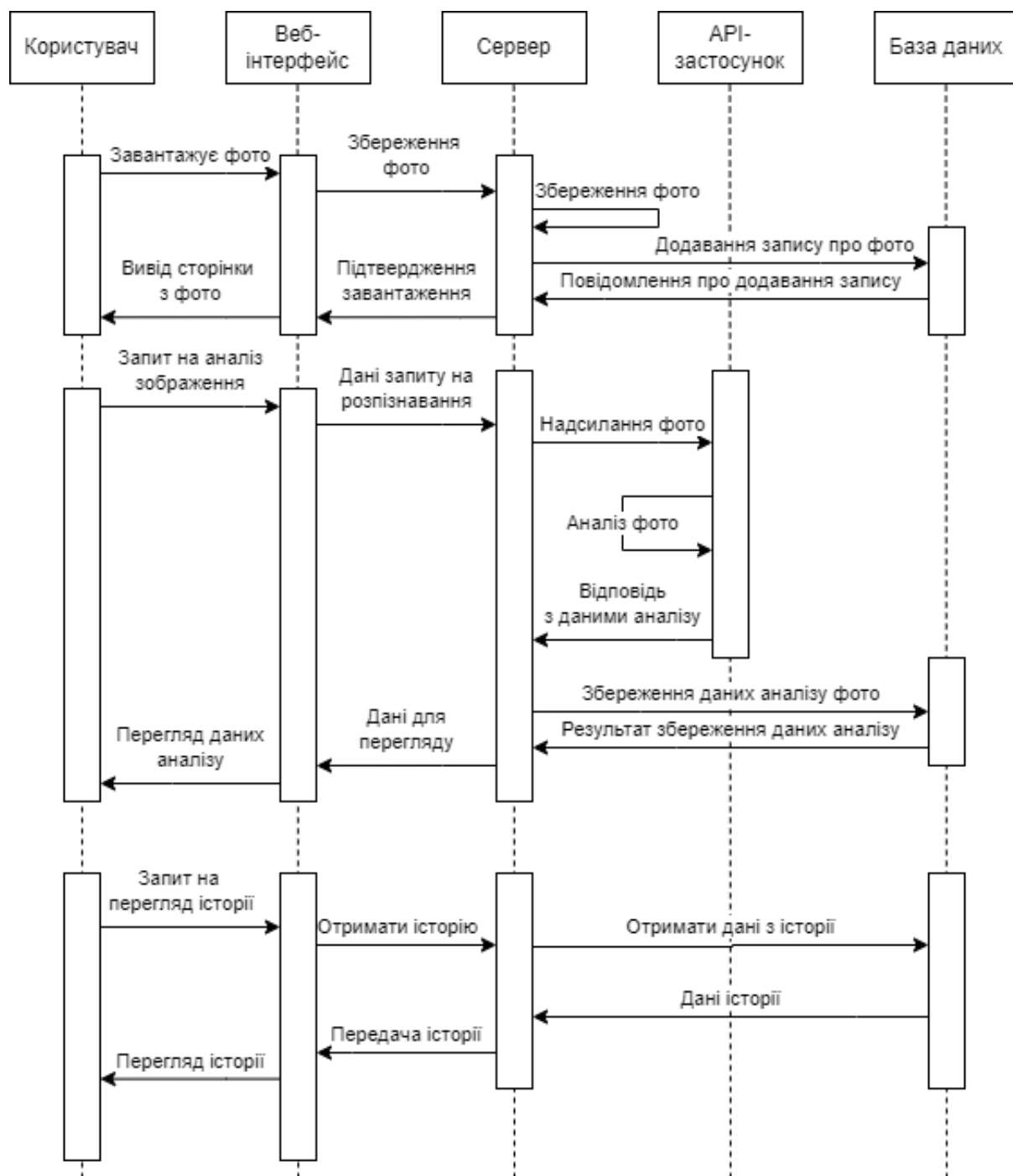


Рис. 2.4. Діаграма послідовності веб-системи для розпізнавання віку

Отже, нами було спроектовано та наведено діаграму послідовності, що демонструє взаємодію між користувачем, веб-інтерфейсом, сервером, API розпізнавання та базою даних при завантаженні фото, розпізнаванні віку та перегляду історії розпізнавань.

Діаграма класів – визначає основні класи, їх атрибути, методи та зв'язки між ними. На рисунку 2.5 наведено діаграму основних класів системи.

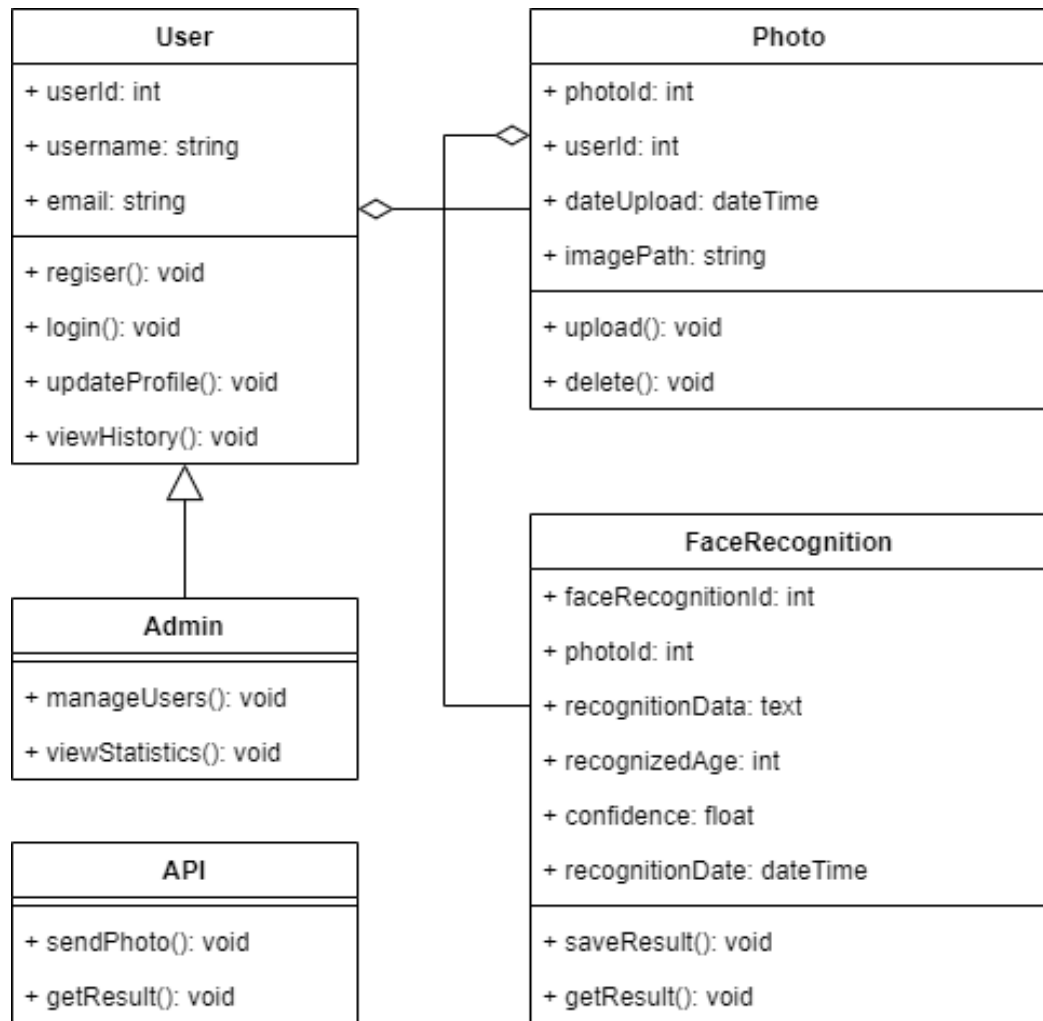


Рис. 2.5. Діаграма класів веб-системи для розпізнавання віку

Наведемо детальний опис діаграми класів:

- **User (Користувач).** Клас зберігає інформацію про користувача та забезпечує функції для реєстрації, входу, оновлення профілю та перегляду історії.
- **Admin (Адміністратор).** Клас наслідується від User і додає можливості управління користувачами, перегляду статистики та налаштування параметрів системи.
- **Photo (Фото).** Клас представляє завантажені фото користувачів, містить дані про фото та методи для завантаження та видалення фото.
- **FaceRecognition (Результат аналізу облич).** Клас зберігає результати розпізнавання віку, включаючи вік, точність розпізнавання та дату розпізнавання. Має методи для збереження та отримання результатів.

- API (API для розпізнавання). Клас здійснює взаємодію з зовнішнім API для розпізнавання віку, містить методи для відправки фото на розпізнавання та отримання результатів.

Отже, нами було здійснено розробку, ілюстрацію та опис UML-діаграм варіантів використання, активностей, послідовності та класів. Використання цих діаграм у процесі розробки веб-системи для розпізнавання віку по фото обличчя допомагає забезпечити структурований підхід до проектування, розробки та тестування системи. Вони сприяють кращому розумінню вимог, оптимізації процесів, коректній реалізації функціональності та ефективній взаємодії між компонентами системи.

2.3. Проектування бази даних

Одним із основних компонентів веб-системи є база даних, призначена для постійного зберігання даних про користувачів, завантажених фотографій, результатів аналізу, системи авторизації. В даному підрозділі здійснимо розробку структури бази даних: таблиць, їх полів, індексів та зв'язків між таблицями. В таблиці 2.2 наведено перелік таблиць та їх призначення в системі.

Таблиця 2.2

Основні таблиці бази даних системи та їх призначення

Назва таблиці	Призначення та опис
user	зберігає дані про зареєстрованих користувачів
photo	зберігає дані про завантажені фотографії (зображення)
face_recognition	зберігає дані про результати аналізу зображення для кожного обличчя окремо
auth_item	елементи системи ролей та дозволів
auth_assignment	зберігає дані про призначення

	ролей і дозволів користувачам системи
auth_item_child	підпорядкування ролей та дозволів
auth_rule	зберігає дані про правила для додаткових перевірок при наданні доступу

В середовищі RНРMyAdmin створимо базу даних за схемою, поданою на рисунку 2.6. Дана схема визначає, як дані будуть організовані, зберігатися і взаємодіяти один з одним.

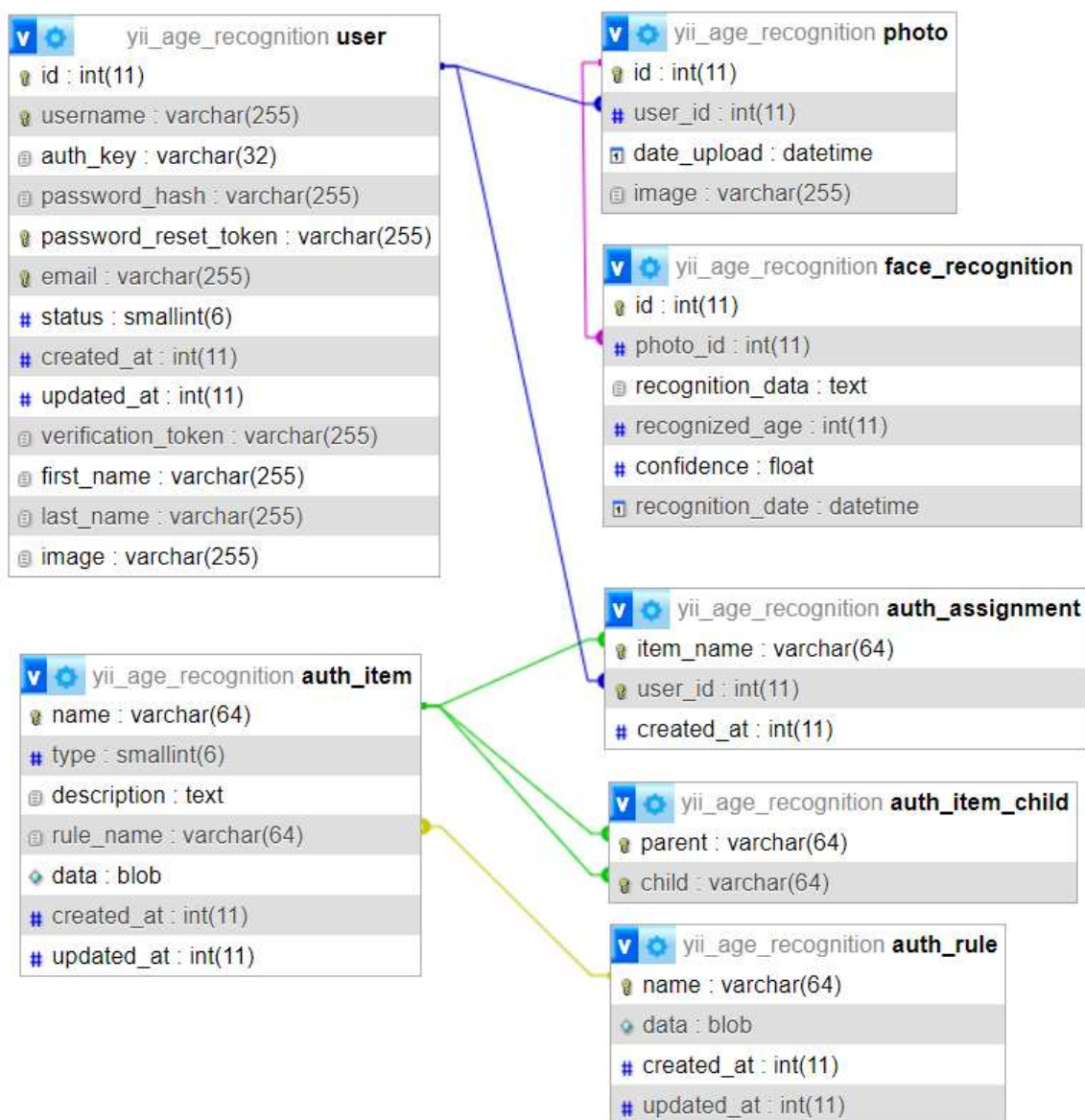


Рис. 2.6. Схема бази даних для веб-системи

Дана схема крім визначення організації та структури даних забезпечує також цілісність даних через використання ключів (первинних, зовнішніх), що гарантує, що всі записи пов'язані коректно, а також правильність введення даних через визначення обмежень (constraints) для полів. Наведемо детальний опис таблиць даних, їх структуру та призначення.

Таблиця 2.3

Таблиця даних про користувачів, *user*

Поле	Параметри поля	Опис
`id`	int(11) NOT NULL	Унікальний ідентифікатор користувача
`username`	varchar(255) NOT NULL	Логін користувача
`auth_key`	varchar(32) NOT NULL	Ключ авторизації засобами cookie (в даній роботі не використовується, оскільки авторизація відбувається засобами сесії)
`password_hash`	varchar(255) NOT NULL	Захешований пароль
`password_reset_token`	varchar(255) DEFAULT NULL	Ключ для відновлення пароль
`email`	varchar(255) NOT NULL	Електронна адреса
`status`	smallint(6) NOT NULL DEFAULT 10	Статус (активований, не активований)
`created_at`	int(11) NOT NULL	Дата і час реєстрації
`updated_at`	int(11) NOT NULL	Дата і час останнього оновлення даних
`verification_token`	varchar(255) DEFAULT NULL	Ключ для активації облікового запису
`first_name`	varchar(255) DEFAULT NULL	Ім'я
`last_name`	varchar(255) DEFAULT NULL	Прізвище
`image`	varchar(255) DEFAULT NULL	Зображення профілю

Таблиця 2.4

Таблиця даних про завантажені фото, *photo*

Поле	Параметри поля	Опис
`id`	int(11) NOT NULL	Унікальний ідентифікатор зображення
`user_id`	int(11) NOT NULL	Ідентифікатор користувача, що завантажив зображення
`date_upload`	datetime NOT NULL	Дата і час завантаження зображення
`image`	varchar(255) NOT NULL	Повне ім'я файлу зображення

Таблиця 2.5

Таблиця даних про аналіз облич на фото, *face_recognition*

Поле	Параметри поля	Опис
`id`	int(11) NOT NULL	Унікальний ідентифікатор даних аналізу
`photo_id`	int(11) NOT NULL	Ідентифікатор зображення з виявленим обличчям
`recognition_data`	text NOT NULL	JSON-дані, отримані як результат аналізу
`confidence`	float NOT NULL	Точність розпізнавання
`recognition_date`	datetime NOT NULL	Дата і час розпізнавання

Таблиця 2.6

Таблиця даних про ролі та дозволи, *auth_item*

Поле	Параметри поля	Опис
`name`	varchar(64) NOT NULL	Назва ролі або дозволу
`type`	smallint(6) NOT NULL	Тип елемента авторизації (роль або дозвіл)
`description`	text DEFAULT NULL	Опис ролі чи дозволу (необов'язкове поле)
`rule_name`	varchar(64) DEFAULT NULL	Посилання на правило, яке може бути додане до ролі чи дозволу
`data`	blob DEFAULT NULL	Може бути використаний для збереження серіалізованих даних про елемент авторизації

`created_at`	int(11) DEFAULT NULL	Дата і час створення елементу авторизації
`updated_at`	int(11) DEFAULT NULL	Дата і час останнього оновлення елементу авторизації

Таблиця 2.7

Таблиця даних про відношення між ролями і дозволами, *auth_item_child*

Поле	Параметри поля	Опис
`parent`	varchar(64) NOT NULL	Посилання на батьківський елемент авторизації
`child`	varchar(64) NOT NULL	Посилання на дочірній елемент авторизації

Таблиця 2.8

Таблиця даних про призначення елементів авторизації, *auth_assignment*

Поле	Параметри поля	Опис
`item_name`	varchar(64) NOT NULL	Назва ролі чи дозволу
`user_id`	int(11) NOT NULL	Ідентифікатор користувача, якому призначають даний елемент авторизації
`created_at`	int(11) DEFAULT NULL	Дата і час призначення

Таблиця 2.9

Таблиця даних про правила, *auth_rule*

Поле	Параметри поля	Опис
`name`	varchar(64) NOT NULL	Назва правила
`data`	blob DEFAULT NULL	Серіалізовані дані про правило
`created_at`	int(11) DEFAULT NULL	Дата і час створення правила
`updated_at`	int(11) DEFAULT NULL	Дата і час останнього оновлення про правило

В даному розділі було спроектовано та створено реляційну базу даних засобами RHPMyAdmin. Така база даних призначена для забезпечення зберігання, організації та швидкого доступу до даних, включаючи користувачів, завантажені фото та результати розпізнавання. Крім цього завдяки створеним індексам та зв'язкам база даних підтримує цілісність та безпеку даних, гарантуючи правильність зв'язків між різними типами інформації.

2.4. Розробка API-модуля для розпізнавання віку

В даному підрозділі наведемо процес розробки API-застосунку використовуючи Flask та DeepFace для розпізнавання віку по фотографії обличчя. Нагадаємо, що DeepFace – Python-бібліотека, що розроблена для розпізнавання облич, а також його характеристик (таких, як стать, вік, частини тощо).

Етап 1. Відкриємо середовище PyCharm і створимо новий проект AgeRecognitionApp. Середовище PyChart автоматично створить локальну папку .venv. Дана папка .venv використовується для створення та керування локальним віртуальним середовищем для проекту Python. Якщо дана папка не була створена, то її необхідно створити і активувати. Створення папки і віртуального середовища:

```
python -m venv .venv
```

Дана команда створить папку .venv у каталозі нашого проекту, що містить копію інтерпретатора Python і каталог site-packages, де будуть встановлені всі залежності. Для активації віртуального середовища виконаємо команду:

```
.venv\Scripts\activate
```

Після активації в командному рядку терміналу з'явиться слово (.venv), що інформує про роботу в віртуальному середовищі.

Етап 2. Після створення проекту і активації віртуального середовища, встановимо необхідні пакети використовуючи команду pip:

```
pip install Flask deepface
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

Flask забезпечує легку та гнучку веб-платформу для створення API-сервісу, що може обробляти запити HTTP. *DeepFace* надає алгоритми розпізнавання обличчя та аналізу, зокрема прогнозування віку, що є важливим для функціональності програми.

Дані пакети будуть встановлені в локальну папку `.venv` віртуального середовища.

Етап 3. Створення застосунку *Flask*. Наведемо процес створення *Flask*-застосунку, який може обробляти завантаження зображень і використовувати *DeepFace* для прогнозування віку на основі завантаженого зображення. Створимо файл `main.py`

Ось повний код із коментарями:

```
from flask import Flask, request, jsonify, render_template
from deepface import DeepFace

# Initialize the Flask application
app = Flask(__name__)

@app.route("/")
def home():
    return render_template('home.html')

@app.route('/predict_age', methods=['POST'])

def predict_age():
    # Check if an image is part of the POST request
    if 'file' not in request.files:
        return jsonify({"error": "No file part"}), 400
    # Get the image from the request
    file = request.files['file']
    if file.filename == '':
        return jsonify({"error": "No selected file"}), 400

    try:
        # Save the uploaded image
        file.save(file.filename)
        # Use DeepFace to analyze the image and predict the age
        analysis = DeepFace.analyze(file.filename, actions=['age'])
        # Return the result as a JSON response
        return jsonify({"age": analysis})
    except Exception as e:
        # Handle exceptions and return an error message
        return jsonify({"error": str(e)}), 500

if __name__ == "__main__":
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```
# Run the Flask app
app.run()
```

Наведемо пояснення до коду:

Імпорт бібліотек. Робимо імпорт необхідних бібліотек для програми: Flask для веб-фреймворку, DeepFace для розпізнавання віку і службові функції *request*, *jsonify*, *render_template*.

Ініціалізуємо екземпляр Flask-застосунку і налаштовуємо папку для завантаження.

Обробляємо index-маршрут, який повертає головну сторінку з формою для завантаження зображення, шаблон якої розробимо на наступному етапі.

Обробляємо маршрут `/predict_age`. Даний маршрут обробляє завантаження зображень. Він перевіряє, чи надано зображення, зберігає зображення і використовує DeepFace для прогнозування віку. Нарешті, він повертає прогнозований вік у відповіді JSON.

Таким чином даний код надає простий функціонал застосунку для передбачення віку людини за обличчям на фото, використовуючи бібліотеку DeepFace.

Етап 4. Розробка веб-сторінки для швидкої перевірки роботи системи в браузері. Створимо папку `templates` та в ній файл шаблону `index.html` з таким кодом:

```
<!doctype html>
<html>
<head>
  <title>Predict Age</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1, shrink-to-fit=no">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootst
rap.min.css" rel="stylesheet"
integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">
</head>
<body>
<div class="container">
  <h1>Завантажте зображення з обличчям людини</h1>
  <!-- Starts form section -->
  <div class="form-container ">
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        <form class="form-horizontal" enctype="multipart/form-data"
action="/predict_age" method="post">

        <div class="mb-3">
            <label for="file" class="form-label">File:</label>
            <input type="file" class="form-control" id="file"
name="file" placeholder="name@example.com">
        </div>
        <div class="col-auto">
            <button type="submit" class="btn btn-
primary">Predict</button>
        </div>
    </form>
    <!-- Ends form section -->
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstra
p.bundle.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsPlUyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>

```

Етап 5. Для запуску застосунку виконаємо команду:

```
.venv\Scripts\activate
```

Після активації в командному рядку терміналу з'явиться слово (.venv), що інформує про роботу в віртуальному середовищі.

```
python app.py
```

Успішний запуск і коректна робота застосунку підтверджується на рисунках 3.7 і 3.8:

```

* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [22/Jun/2024 23:35:04] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [22/Jun/2024 23:35:05] "GET /favicon.ico HTTP/1.1" 404 -

```

Рис. 2.7. Логування при успішному запуску застосунку

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

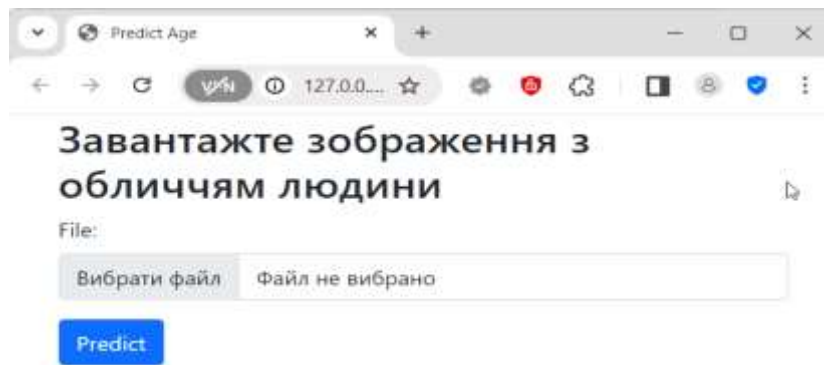


Рис. 2.8. Вигляд головної сторінки для швидкої перевірки роботи системи

Отже, в даному підрозділі ми навели приклад розробки API-застосунку, як компоненту веб-системи для розпізнавання віку по фото. Зокрема здійснено такі етапи як створення проекту і установка необхідних пакетів, створення застосунку Flask та html-шаблону головної сторінки для швидкої перевірки, запуск застосунку та перевірка його роботи.

Висновки до другого розділу

В другому розділі роботи здійснено розробку, ілюстрацію та опис UML-діаграм варіантів використання, активностей, послідовності та класів. Використання цих діаграм у процесі розробки веб-системи для розпізнавання віку по фото облич допомагає забезпечити структурований підхід до проектування, розробки та тестування системи. Вони сприяють кращому розумінню вимог, оптимізації процесів, коректній реалізації функціональності та ефективній взаємодії між компонентами системи.

Спроектовано та створено реляційну базу даних засобами PHPMyAdmin. Така база даних призначена для забезпечення зберігання, організації та швидкого доступу до даних, включаючи користувачів, завантажені фото та результати розпізнавання. Крім цього завдяки створеним індексам та зв'язкам база даних підтримує цілісність та безпеку даних, гарантуючи правильність зв'язків між різними типами інформації.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

Наведено приклад розробки API-застосунку, як компоненту веб-системи для розпізнавання віку по фото. Зокрема здійснено такі етапи як створення проекту і установка необхідних пакетів, створення застосунку Flask та html-шаблону головної сторінки для швидкої перевірки, запуск застосунку та перевірка його роботи.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						41
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. ВСТАНОВЛЕННЯ, ВИКОРИСТАННЯ ТА ОСОБЛИВОСТІ ТЕСТУВАННЯ ВЕБДОДАТКУ

3.1. Встановлення та налаштування API-модуля для розпізнавання віку людини

Етап 1. Налаштування локального середовища розробки

1. Операційна система: Microsoft Windows 11 Pro. Версія 10.0.22631 N/A Build 22631
2. Встановлення пакету Visual C++ Redistributable для коректної роботи бібліотек DeepFace і TensorFlow
3. Встановлення Python (Python 3.12.4). Для цього завантажимо дистрибутив python-3.12.4-amd64.exe з веб-сайту та виконаємо кроки майстра встановлення. При встановленні необхідно передбачити можливість виконання команди `python` в командному рядку операційної системи, вибравши “*Add python.exe to PATH*”. Рекомендується також вибрати наступні додаткові можливості: встановлення документації Python, пакетного менеджера *pip*, середовищ *tkinter* та *IDLE*. Після встановлення Python в командному рядку (cmd) можна перевірити роботу його команд:

```
C:\Users\username>python --version
```

```
C:\Users\username>pip --version
```

Очікуваний результат виводу:

```
Python 3.12.4
```

```
pip 24.0 from C:\Program Files\Python312\Lib\site-packages\pip (python 3.12)
```

4. Встановлення середовища програмування *PyCharm*.
5. Запуск середовища *PyCharm* і створення проекту *AgeRecognitionApp*. При цьому необхідно перевірити:
 - а. Правильно заданий шлях до інтерпретатора Python;
 - б. Наявність директорії `.venv` – віртуальне середовище Python з внутрішньою файловою структурою. Воно має підкаталог `bin`, який

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

містить посилання на інтерпретатор Python, а також підкаталоги, які зберігають пакунки, встановлені в даному проекті.

6. Встановлення пакетів Flask та DeepFace. В середовищі проекту створимо новий термінал і в корені проекту виконаємо команду

```
pip install Flask deepface
```

Модуль deepface використовує ряд залежностей, які автоматично теж будуть встановлені. Серед них є такі модулі як tensorflow і keras. Для перевірки роботи tensorflow можна виконати команду:

```
python -c "import tensorflow as tf; print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

Очікуваний результат виводу:

```
tf.Tensor(-248.85239, shape=(), dtype=float32)
```

Етап 2. Створення, запуск та тестування застосунку

7. Створимо файл app.py з програмним кодом нашого API-застосунку.
8. Здійснимо запуск застосунку. Для цього в терміналі середовища програмування виконаємо команду:

```
python app.py
```

Очікуваний результат успішного запуску застосунку в локальному середовищі наведений на рисунку:

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.0.2:5000
Press CTRL+C to quit
```

Рис. 3.1. Логи застосунку в робочому режимі (локальне середовище)

9. Протестуємо роботу застосунку. Для цього можна використати:
 - а. Надсилання POST-запиту на локальний сервер застосунку засобами інструмента командного рядка curl: `curl -i -X POST -H "Content-Type: multipart/form-data" -F "file=@images/image01.jpg" http://127.0.0.1:5000/predict_age`. Результат виконання запиту наведено на рисунку 3.2

```
1 C:\Users\...\PycharmProjects\AgeRecognitionApp>curl -i -X POST -H "Content-Type: multipart/form-data" -F "file=@images/image01.jpg" http://127.0.0.1:5000/predict_age
2 HTTP/1.1 200 OK
  Server: Werkzeug/3.0.3 Python/3.12.4
  Date: Fri, 21 Jun 2024 05:32:18 GMT
  Content-Type: application/json
  Content-Length: 129
  Connection: close
3 [{"age": [{"age": 31, "face_confidence": 0.91, "region": {"h": 228, "left_eye": [408, 179], "right_eye": [319, 166], "w": 228, "x": 251, "y": 83}]}]]
```

Рис. 3.2. Результат виконання запиту через curl: 1 - команда, 2 - отримані заголовки, 3 - тіло відповіді

б. При надсиланні POST-запиту на локальний сервер застосунку засобами розробленої веб-сторінки через браузер за адресою <http://127.0.0.1:5000>:

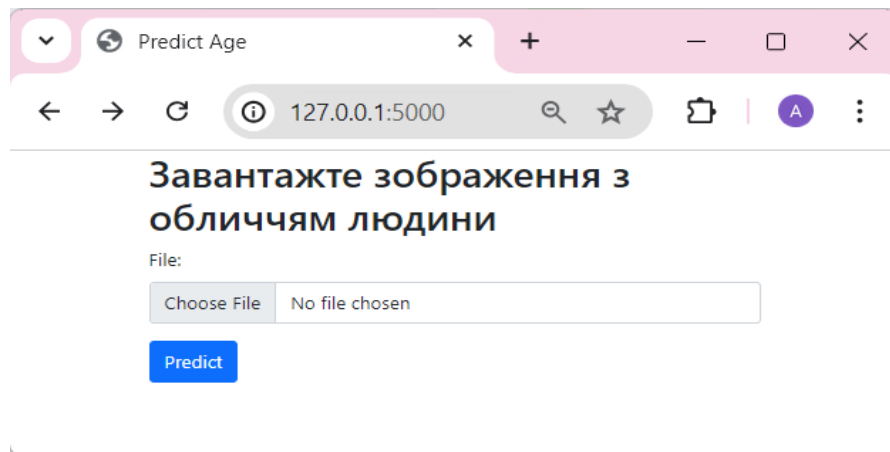


Рис. 3.3. Сторінка з формою для завантаження зображення людини.

Результат обробки запиту у вигляді отриманої відповіді з JSON-даними наведено на рисунку 3.4:

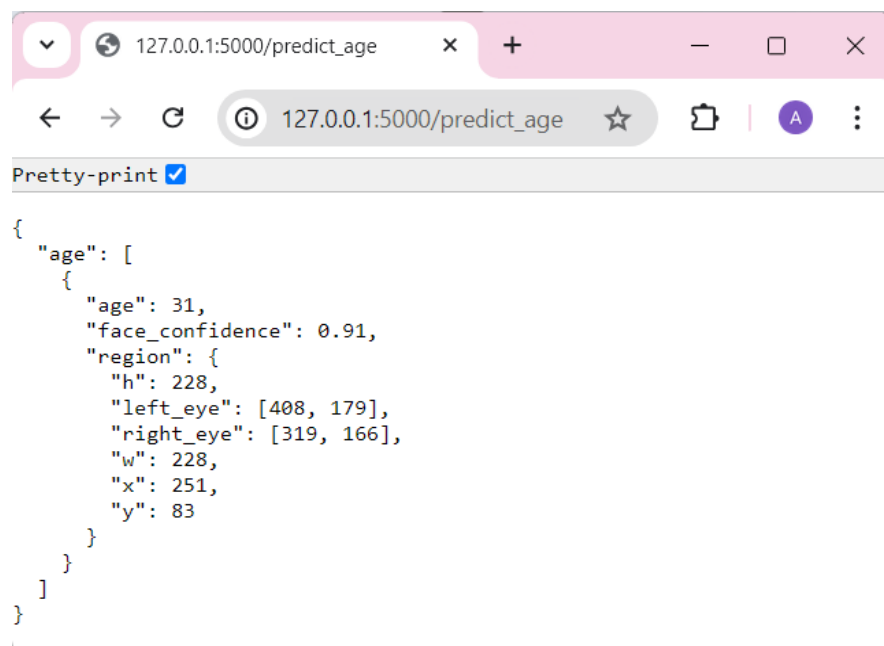


Рис. 3.4. Очікуваний результат – отриманий JSON з даними передбачення

Етап 3. Контейнеризація застосунку

10. Встановлення програмного забезпечення Docker та Docker Compose для контейнеризації застосунку. Для перевірки правильності встановлення виконаємо команди:

```
docker --version
```

```
docker-compose --version
```

```
C:\Windows\System32>docker --version
Docker version 26.1.4, build 5650f9b

C:\Windows\System32>docker-compose --version
Docker Compose version v2.27.1-desktop.1
```

Рис. 3.5. Вивід версій Docker та Docker Compose

11. Створення файлу *requirements.txt*
12. Створення файлу *Dockerfile*
13. Побудова образу Docker

```
docker build -t age-recognition-app .
```

14. Запуск контейнера Docker локально:

```
docker run -p 5000:5000 age-recognition-app
```

Про успішний запуск контейнера можна пересвідчитись з логування результатів запуску (рис. 3.6). Далі застосунок можна перевірити за попередніми тестами.

```
24-06-21 10:30:27 - Directory /root/.deepface created
24-06-21 10:30:27 - Directory /root/.deepface/weights created
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

Рис. 3.6. Логування процесу успішного запуску контейнера з застосунком.

В даному підрозділі було наведено кроки щодо встановлення та налаштування API-модуля для розпізнавання віку людини за фото, зокрема розглянуто етапи налаштування локального середовища розробки, створення, запуск, тестування та контейнеризації застосунку.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						45
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2. Структура інтерфейсу та опис роботи з додатком

В цьому підрозділі наведемо інструкції по користуванню веб-системою для розпізнавання віку по фото. Інструкції будуть розроблені та надані у послідовності кроків з наведеними ілюстраціями веб-сторінок.

Реєстрація користувача. Для отримання доступу до функцій системи, необхідно зареєструватись. Для цього виконаємо запит <http://age-recognition.local/site/signup>, або перейдемо за посиланням “Реєстрація”. На рисунку 3.7 зображено сторінку з формою реєстрації:

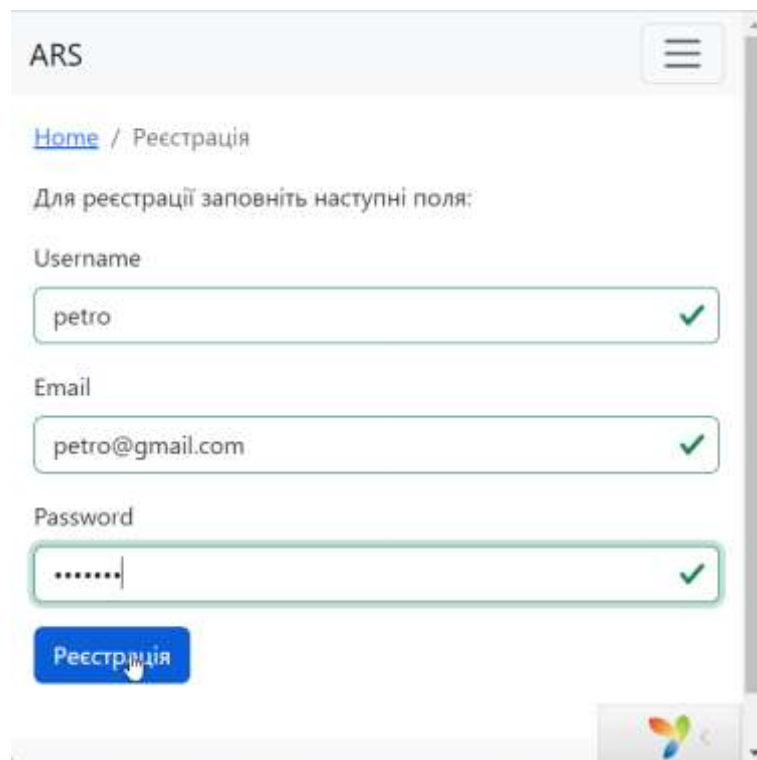


Рис. 3.7. Реєстрація користувача. Сторінка з формою реєстрації

Після додавання користувача в БД запит перенаправляється на головну сторінку з повідомленням про успішну реєстрацію:

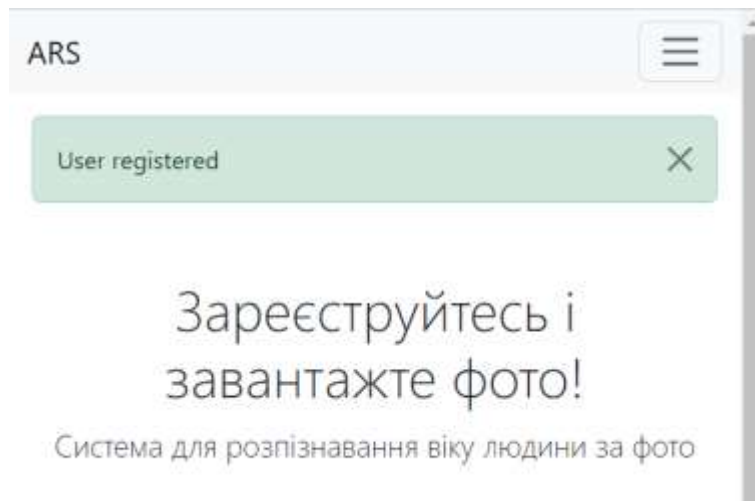


Рис. 3.8. Головна сторінка з повідомленням про успішну реєстрацію користувача

Вхід в систему. Для отримання доступу до функцій системи, необхідно здійснити вхід. Для цього потрібно перейти за посиланням “Вхід” (URL <http://age-recognition.local/site/login>)

Рис. 3.9. Сторінка з формою для входу користувача в систему

Після здійснення входу в систему, для користувача надається доступ на такі веб-сторінки та функції системи:

- Сторінка “Мій профіль”
- Сторінка “Редагування профілю”
- Сторінка “Мої фото (зображення)”
- Сторінка “Завантажити фото (зображення)”

- Сторінка “Перегляд фото (зображення)”
- Запит на видалення зображення на сторінках “Мої фото” і “Перегляд фото”
- Запит на аналіз зображення на сторінці “Перегляд фото”
- Запит на очистку результатів аналізу

Сторінка “Мій профіль” (рис. 3.10). На сторінці користувач може переглянути дані про себе: електронну адресу, логін, прізвище та ім’я, а також власне зображення (фото). В верхньому лівому куті сторінки присутнє посилання на сторінку для редагування профілю.



Рис. 3.10. Сторінка перегляду профілю користувача

Сторінка “Редагування профілю” (рис. 3.11). На даній сторінці є форма і користувач може змінити свої дані: логін, електронну адресу, прізвище та ім’я, а також зображення:

ARS 

[Home](#) / [Користувачі](#) / [ivan](#) / Оновлення

Оновлення профілю: ivan

Логін

Email

Ім'я

Прізвище

Фото

Choose File No file chosen

Save

Рис. 3.11. Сторінка з формою для редагування профілю користувача

Сторінка зі списком (історією) завантажених фото користувачем (рис. 3.12). На даній сторінці здійснено вивід всіх завантажених фото. На сторінці можна здійснювати пошук і сортування за датою завантаження та датою останнього розпізнавання. Користувач має можливість видалити зображення з усіма його даними (в тому числі, результатами розпізнавання), а також перейти на сторінку з деталями зображення, скориставшись відповідними кнопками дій. Зверху зліва сторінки є посилання на сторінку з формою для завантаження нової фотографії (рис. 3.13).

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

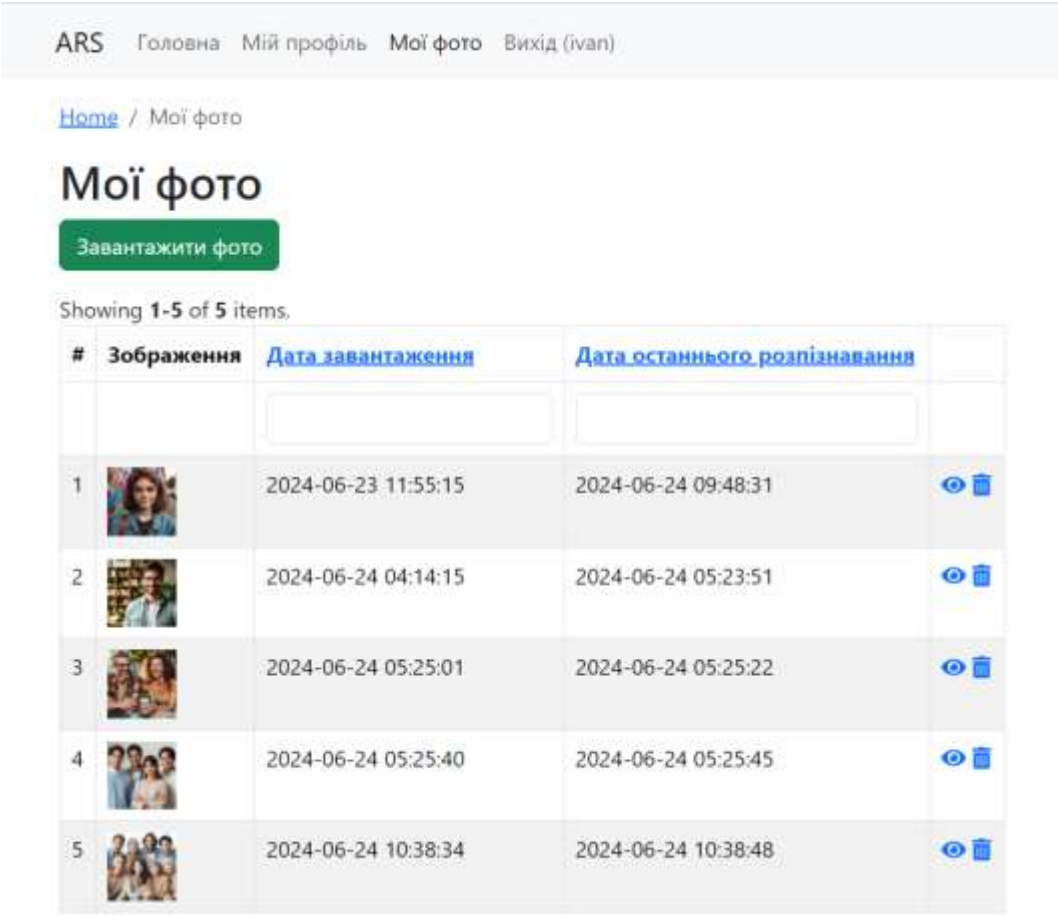


Рис. 3.12. Сторінка зі списком (історією) завантажених фото користувачем

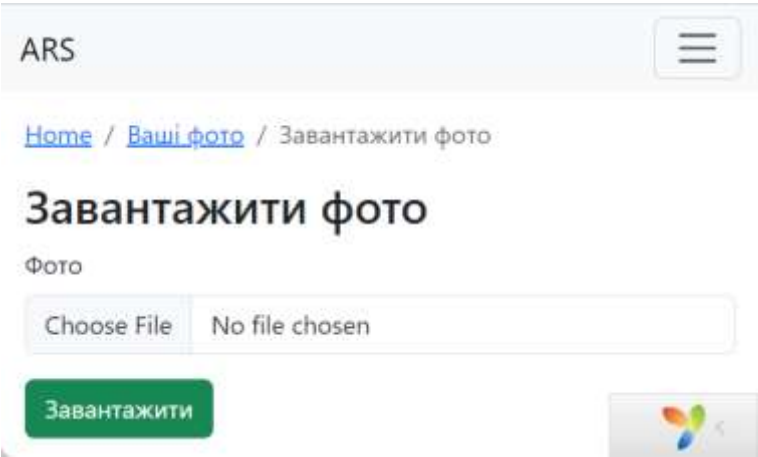


Рис. 3.13. Сторінка з формою для завантаження фото

Сторінка “Перегляд деталей фото” (рис. 3.14). На даній сторінці виводиться зображення та його інформація: дата завантаження, назва файлу. А також кнопки “Аналізувати зображення” та “Очистити результати”.

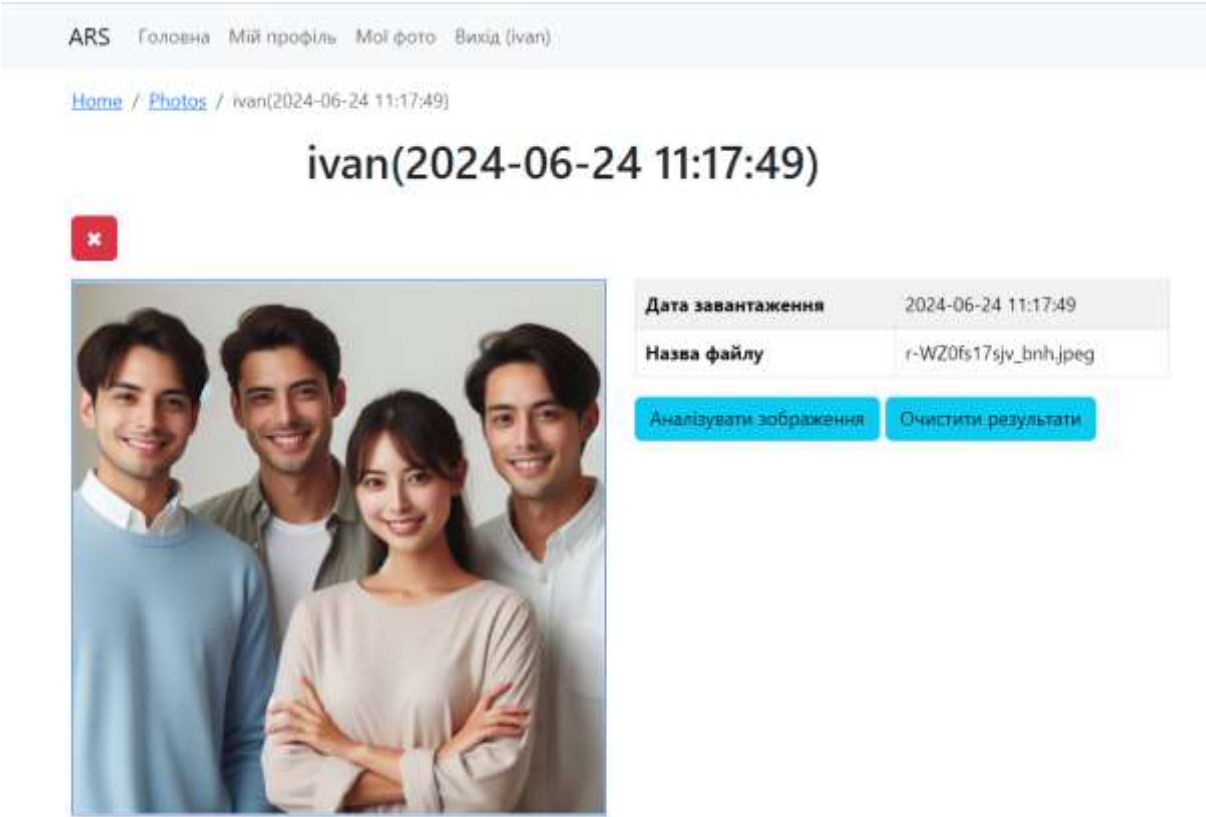


Рис. 3.14. Сторінка з деталями про завантажене фото

Якщо зображення вже має результати аналізу, то вони виводяться в таблиці та на самому зображенні (рис. 3.15). Таблиця демонструє вік та точність розпізнавання для кожного виявленого обличчя. На фото всі виявлені обличчя виділяються прямокутником зі значенням визначеного віку в лівому верхньому куті. При наведенні курсора на даний прямокутник з'являється напис з деталями розпізнавання: порядковий номер, вік і точність (рис. 3.16).

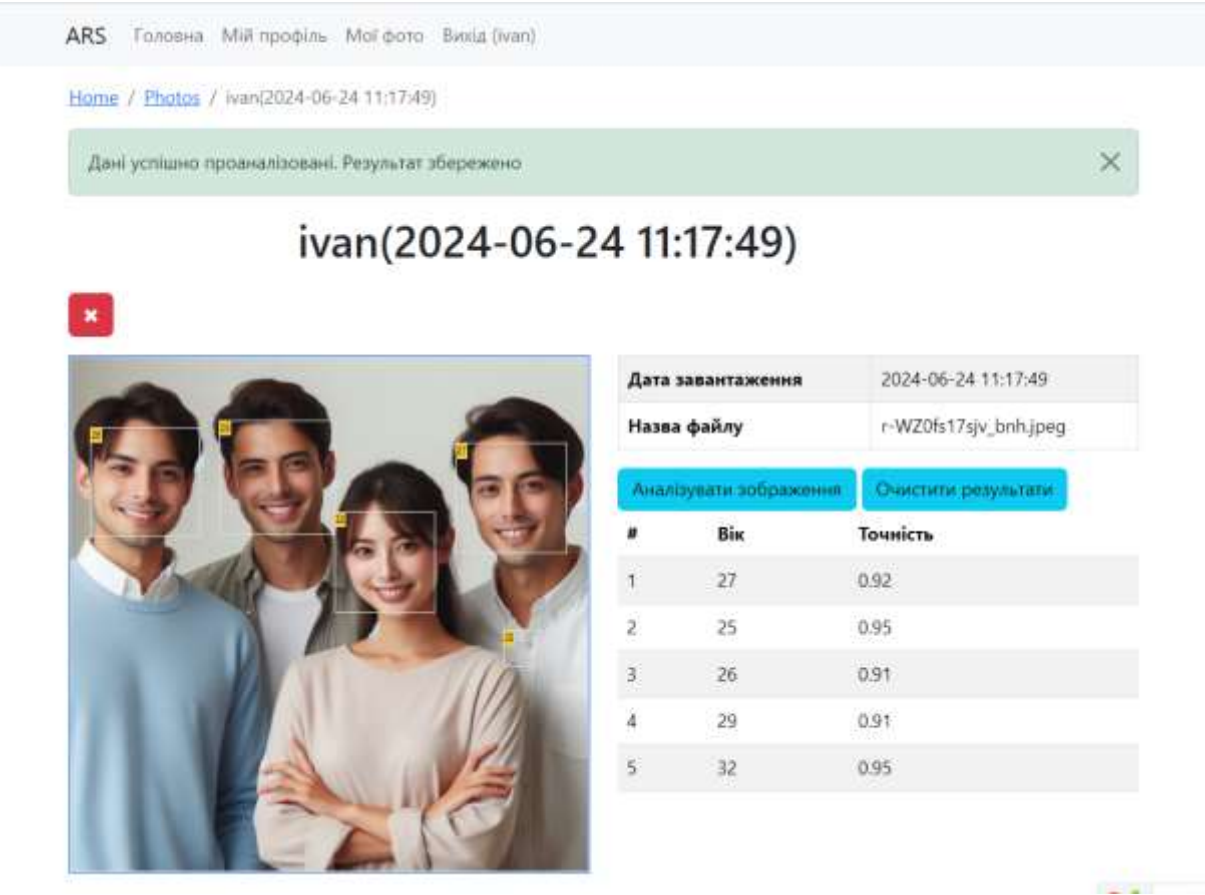


Рис. 3.15. Сторінка з деталями про завантажене фото та результатами розпізнавання



Рис. 3.16. Вивід результатів розпізнавання при наведенні курсора на фото

Отже, в даному підрозділі нами було здійснено опис основних веб-сторінок та функцій, а також інструкцію щодо використання веб-системи для розпізнавання віку по фото.

3.3. Тестування роботи системи

В ході тестування локальної версії були виявлені такі помилки:

Таблиця 3.1

Помилки при тестуванні локальної версії

1. Під час встановлення модуля TensorFlow:	
<i>Windows doesn't accept long paths by default</i>	обмеження максимальної довжини шляху. Ця помилка пояснюється тим, що по замовчуванню операційна система не підтримує довгі шляхи [3]. Для виправлення даної помилки було змінено значення ключа реєстру <i>Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\LongPathsEnabled</i> на <i>dword:00000001</i> .
2. Під час запуску застосунку:	
<i>You have tensorflow 2.16.1 and this requires tf-keras package</i>	Дана помилка пояснюється тим, що поточна версія tensorflow 2.16.1 не підтримує пакет keras, встановлений по замовчуванню. Тому потрібно вручну встановити `pip install tf-keras` або зробити даунгрейд самої бібліотеки tensorflow.
3. При надсиланні POST-запиту на локальний сервер застосунку засобами Postman	
<i>When testing an API locally, you need to use the Postman Desktop Agent</i>	Дана помилка виникає при тестуванні застосунку, що працює на локальному сервері. Ця проблема вирішена за рахунок встановлення

	служби Desktop Agent для можливості надсилати запити з віддаленої служби Postman на localhost.
400 BAD REQUEST "No file part"	Дана помилка є відповіддю від сервера, який не отримав надісланий файл зображення. Дана помилка в цілому не впливає на роботу застосунку.
4. При надсиланні POST-запиту на локальний сервер застосунку засобами інструмента командного рядка curl, а саме: <pre>curl -X POST -F '@file=images/image01.jpg' http://localhost:5000/predict_age</pre>	
400 BAD REQUEST "No file part"	Вирішення даної помилки відбулось за рахунок додавання атрибуту -H "Content-Type: multipart/form-data" до команди curl: <pre>curl -i -X POST -H "Content-Type: multipart/form-data" -F "file=@images/image01.jpg" http://127.0.0.1:5000/predict_age</pre> Результат правильної роботи застосунку при надсиланні запиту через командний рядок наведено в підрозділі 3.1 на рис. 3.1
5. При надсиланні POST-запиту на локальний сервер застосунку засобами розробленої веб-сторінки через браузер за адресою http://127.0.0.1:5000 (рис. 3.2)	
Помилки немає	Очікуваний результат – отриманий JSON з даними передбачення (підрозділ 3.1, рис. 3.3)
6. В процесі контейнеризації застосунку	
ImportError: libGL.so.1: cannot open shared object file: No such file or directory	Проблема відсутності даної бібліотеки вирішується шляхом додавання в Dockerfile директиви <pre>RUN apt-get update && apt-get install ffmpeg libsm6 libxext6 -y</pre> Ці команди встановлюють залежності cv2, які зазвичай присутні на локальній машині, але можуть бути відсутні в контейнері Docker, що спричиняє проблему.

Таким чином нами було наведено приклад процесу тестування API-модуля для розпізнавання віку людини за її фото. На даний момент всі виявлені помилки виправлено, функції системи працюють коректно. Зауважимо, що система не обмежиться вже розробленими функціями, процес її розробки продовжується і планується реалізувати функції для роботи з аналітикою, статистикою і генерацією звітів, автоматизувати процес розгортання такої системи на хмарних сервісах з використанням технологій Docker та Jenkins.

Висновки до третього розділу

В третьому розділі роботи розроблено кроки щодо встановлення та налаштування API-модуля для розпізнавання віку людини за фото, зокрема розглянуто етапи налаштування локального середовища розробки, створення, запуск, тестування та контейнеризації застосунку; здійснено опис основних веб-сторінок та функцій, а також інструкцію щодо використання веб-системи для розпізнавання віку по фото; наведено приклад процесу тестування API-модуля для розпізнавання віку людини за її фото. На даний момент всі виявлені помилки виправлено, функції системи працюють коректно. На даний час, процес розробки системи не завершився і планується реалізувати функції для роботи з аналітикою, статистикою і генерацією звітів, автоматизувати процес розгортання такої системи на хмарних сервісах з використанням сучасних DevOps-технологій.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

В роботі було здійснено розробку веб-системи з двома компонентами: багатокористувацького веб-сайту та API-модуля з можливостями розпізнавання віку людини за її обличчям на фото. Результатом виконаної роботи є веб-система, що дозволяє користувачам завантажувати свої фотографії для розпізнавання віку з використанням DeerFace, переглядати результати та історію розпізнавань, з функціями реєстрації, аутентифікації, зручним інтерфейсом та адміністративним модулем. В ході роботи були поставлені поетапні завдання і нижче викладено відповідні висновки про їх виконання:

Визначено теоретичне і практичне значення роботи, здійснено постановку задачі та формулювання завдань до виконання проекту. Здійснено огляд сучасних бібліотек і фреймворків для машинного зору, зокрема для розв'язку задачі визначення віку людини за фото і визначено залежно від конкретних вимог які бібліотеки підходять для нашого проекту. Для розробки багатокористувацької системи для розпізнавання віку по фото людини здійснено вибір і обґрунтовано 2 стеки технологій відповідно для розробки багатокористувацького веб-сайту та API-застосунку для розпізнавання.

На етапі проектування здійснено розробку, ілюстрацію та опис UML-діаграм варіантів використання, активностей, послідовності та класів. Використання цих діаграм у процесі розробки веб-системи для розпізнавання віку по фото обличчя допомагає забезпечити структурований підхід до проектування, розробки та тестування системи. Спроектовано та створено реляційну базу даних засобами PHPMyAdmin. Така база даних призначена для забезпечення зберігання, організації та швидкого доступу до даних, включаючи користувачів, завантажені фото та результати розпізнавання. Наведено етапи розробки API-застосунку, як компоненту веб-системи для розпізнавання віку по фото.

Розроблено кроки щодо встановлення та налаштування API-модуля для розпізнавання віку людини за фото; здійснено опис основних веб-сторінок та

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

функцій, а також інструкцію щодо використання веб-системи для розпізнавання віку по фото; наведено приклад процесу тестування API-модуля для розпізнавання віку людини за її фото. На даний момент всі виявлені помилки виправлено, функції системи працюють коректно.

Слід зауважити, що процес розробки системи не завершився і планується реалізувати функції для роботи з аналітикою, статистикою і генерацією звітів, автоматизувати процес розгортання такої системи на хмарних сервісах з використанням сучасних DevOps-технологій.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Python download page for Windows. [Електронний ресурс]. Режим доступу: <https://www.python.org/downloads/windows/>
2. How to Install Python on Windows 10. [Електронний ресурс]. Режим доступу: <https://www.digitalocean.com/community/tutorials/install-python-windows-10>
3. Maximum Path Length Limitation [Електронний ресурс]. Режим доступу: <https://learn.microsoft.com/en-us/windows/win32/fileio/maximum-file-path-limitation>
4. OpenCV - Open Computer Vision Library. [Електронний ресурс] - Режим доступу: <http://dlib.net/>
5. TensorFlow - An end-to-end platform for machine learning. [Електронний ресурс] - Режим доступу: <https://www.tensorflow.org/>
6. PyTorch. [Електронний ресурс] - Режим доступу: <https://pytorch.org/>
7. Фреймворк Deepface. [Електронний ресурс]. Режим доступу: <https://github.com/serengil/deepface>
8. Бібліотека Dlib. [Електронний ресурс]. Режим доступу: <http://dlib.net/>
9. PHP: Hypertext Preprocessor. Docs [Електронний ресурс] - Режим доступу до ресурсу: <https://www.php.net/docs.php>
10. Yii2 PHP Framework. Guide [Електронний ресурс] - Режим доступу до ресурсу: <https://www.yiiframework.com/doc/guide/2.0/en>
11. Fabrizio Caldarelli. Yii2 By Example. Packt Publishing, 2015
12. Matt Zandstra. PHP Objects, Patterns, and Practice. Apress; 3rd ed. edition, 2010
13. Документація CSS-фреймворку Bootstrap [Електронний ресурс] - Режим доступу до ресурсу: <https://getbootstrap.com/>
14. I. Masi, Y. Wu, T. Hassner and P. Natarajan, "Deep Face Recognition: A Survey," 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		

(SIBGRAPI), Parana, Brazil, 2018, pp. 471-478, doi: 10.1109/SIBGRAPI.2018.00067.

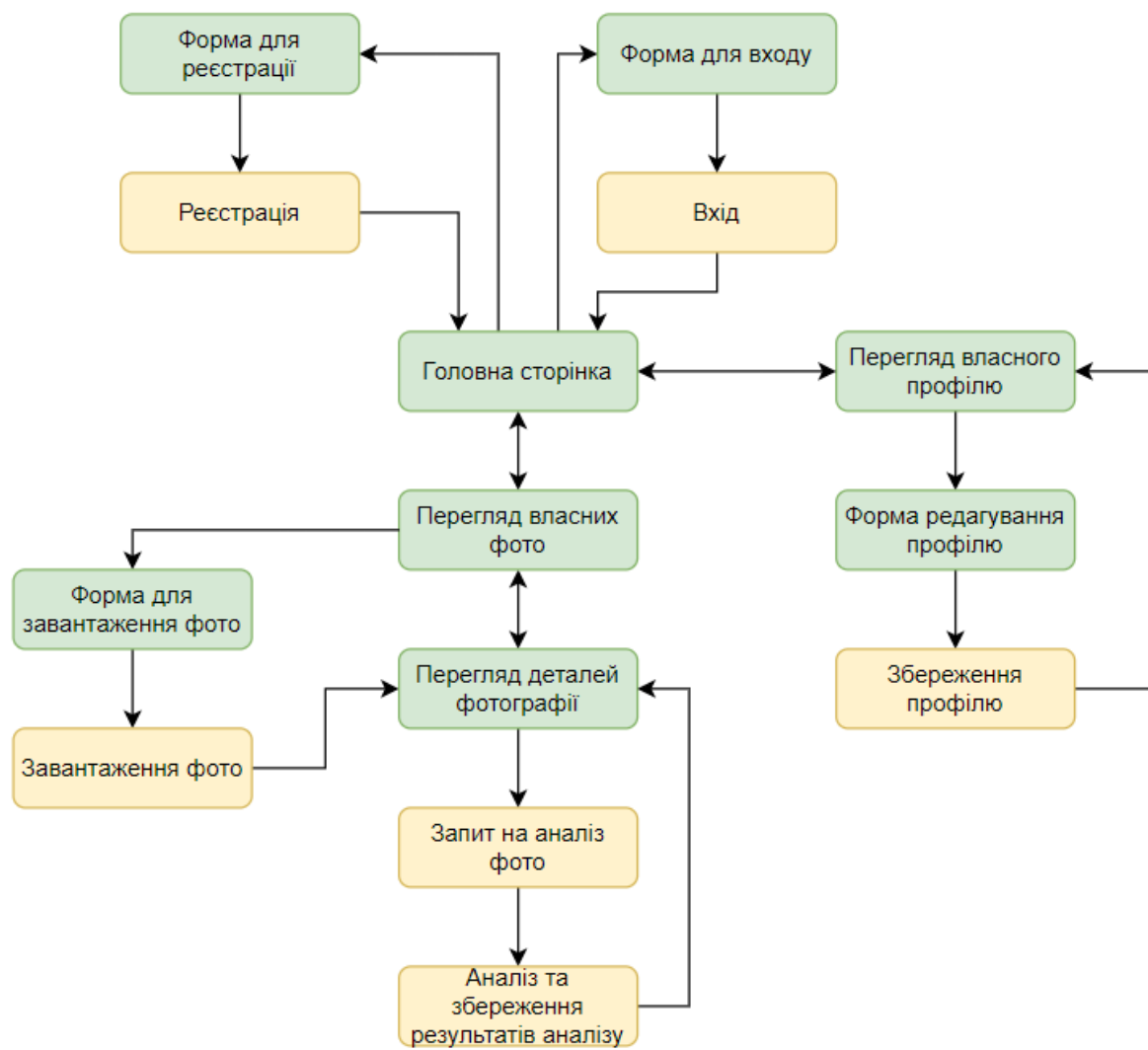
15. Mei, Wang & Deng, Weihong. (2018). Deep Face Recognition: A Survey. Neurocomputing. 429. 10.1016/j.neucom.2020.10.081.

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						59
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

Візуальне представлення мапи веб-сайту



Метод actionRecognize() контролера PhotoController.

Використовується для відправки запиту для аналізу фото, обробки та збереження результатів аналізу

```
public function actionRecognize($id)
{
    $model = $this->findModel($id);
    $imageFile = Yii::getAlias("@webroot") . '/images/photos/' . $model->image;

    if (file_exists($imageFile)) {
        $mimeType = mime_content_type($imageFile);
        $fp = fopen($imageFile, 'r');

        $cfile = curl_file_create($imageFile, $mimeType, 'testpic');
        $imgdata = ['file' => $cfile];
        $curl = curl_init();
        curl_setopt_array($curl, [
            CURLOPT_URL => "http://127.0.0.1:5000/predict_age",
            CURLOPT_POST => 1,
            CURLOPT_RETURNTRANSFER => 1,
            CURLOPT_CUSTOMREQUEST => "POST",
            CURLOPT_POSTFIELDS => $imgdata,
            CURLOPT_HTTPHEADER => [
                "Content-Type: multipart/form-data",
            ],
        ]);

        $response = curl_exec($curl);
        $httpcode = curl_getinfo($curl, CURLINFO_HTTP_CODE);
        curl_close($curl);
        if ($httpcode == '200') {
            $model->deleteResults();
            $model->date_recognition = date('Y-m-d h-i-s');
            $recognitionData = json_decode($response, true);
            $faces = $recognitionData['data'];
            foreach ($faces as $faceItem) {
                $faceRecognition = new FaceRecognition();
                $faceRecognition->photo_id = $model->id;
                $faceRecognition->recognition_data =
                    json_encode($faceItem);
                $faceRecognition->recognized_age = (int)$faceItem['age'];
                $faceRecognition->confidence =
                    floatval($faceItem['face_confidence']);
                if (!$faceRecognition->save()) {
                    print_r($faceRecognition->errors);
                    die;
                }
            }
            $model->save();
            Yii::$app->session->setFlash('success', 'Дані успішно
проаналізовані. Результат збережено');
```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->redirect(['view', 'id' => $model->id]);
    }
}
}

```

Лістинг моделі PhotoCreateForm.

Використовується для створення форми для завантаження фото, а також містить валідацію, код для завантаження і збереження файлу фото та інформації про нього в БД.

```

<?php

namespace app\models;

use Yii;
use yii\base\Model;
use yii\web\UploadedFile;

/**
 * This is the model class for table "user".
 */

/**
 * @property string|null $image
 */
class PhotoCreateForm extends Model
{
    public $image;

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['image'], 'file', 'skipOnEmpty' => true, 'extensions' =>
                'png, jpg, webp, jpeg'],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'image' => 'Фото',
        ];
    }
}

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						63
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public function uploadAndCreate()
{
    $this->image = UploadedFile::getInstance($this, 'image');
    if (!$this->validate()) {
        return null;
    }
    if ($this->image) {
        $path = Yii::getAlias('@webroot') . '/images/photos';
        $upload = new Upload();
        $upload->loadData(['file' => $this->image, 'path' => $path]);
        if ($upload->saveFile()) {
            $photo = new Photo();
            $photo->image = $upload->fileName;
            $photo->user_id = Yii::$app->user->getId();
            $photo->date_upload = date("Y-m-d h:i:s");
            $photo->save();
            return $photo;
        }
    }
    return false;
}
}

```

Лістинг моделі – класу Upload.

Даний клас призначений для збереження файлів зображень на сервері з новим іменем та тим самим розширенням

```

<?php

namespace app\models;

use Yii;
use yii\base\Model;
use yii\web\UploadedFile;

class Upload extends Model
{
    /**
     * @var UploadedFile
     */
    public $file;
    public $fileName;
    public $path;

    public function loadData($data)
    {
        $this->file = $data["file"];
        $this->path = $data["path"];
    }

    public function setName()
    {

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $this->fileName = Yii::$app->security->generateRandomString(16) .
        '.' . $this->file->extension;

    }

    public function saveFile()
    {
        $this->setName();
        if ($this->file->saveAs($this->path . '/' . $this->fileName)) {
            return true;
        }
        return false;
    }
}

```

					КН.КР.Б – 122 – 24 – ПЗ	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		