

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти перший (бакалаврський)

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ МОНІТОРИНГУ ЗНАНЬ З МОВИ SQL З МОЖЛИВІСТЮ РОЗПІЗНАВАННЯ ОБЛИЧЧЯ СТУДЕНТА
(тема)

Виконав:
студент 4 курсу, групи ІТІНФ-19-1

Шелест В.А.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Яковлева О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Кобилін О.А.
(прізвище, ініціали)

2023 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту

(повна назва)

Кафедра Інформатики

(повна назва)

Рівень вищої освіти перший (бакалаврський)Спеціальність 122 Комп'ютерні науки

(код і повна назва)

Тип програми освітньо-професійнаОсвітня програма Інформатика

(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«_____» _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Шелесту Володимирі Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розробка застосунку для моніторингу знань з мови SQL з можливістю розпізнавання обличчя студента

затверджена наказом університету від 15 травня 2023 року № 474 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 22 травня 2023 р.

3. Вихідні дані до роботи науково-методична та науково-технічна література, матеріали конференцій, дані інтернет-мережі, бібліотека комп'ютерного зору з відкритим кодом DeepFace, платформа розробки .Net, мова програмування C#, фреймворк gRPC для платформи .Net, фреймворк Angular, мова програмування TypeScript, середовище розробки JetBrains Rider, середовище розробки Microsoft Visual Studio Code.

4. Перелік питань, що потрібно опрацювати в роботі _____

1. Огляд існуючих застосунків моніторингу знань студентів особливо в області знань мови SQL.

2. Вивчення підходів щодо ідентифікації особи.

3. Проектування архітектури застосунку для моніторингу знань студентів з мови SQL з можливістю ідентифікації особи студента.

4. Розробка алгоритму автоматичного оцінювання скрипту-відповіді студента.

5. Розробка застосунку для моніторингу знань студентів з мови SQL з можливістю ідентифікації особи студента.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Актуальність проблеми моніторингу знань студентів, постанова задачі, опис архітектури, демонстрація роботи із програмним застосунком, демонстрація роботи модулю верифікації, аналіз роботи алгоритму оцінювання відповіді студента.

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Консультант з дотримання діючих стандартів та норм	Доцент Творошенко І.С.		

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	10.04.2023	
2	Аналіз завдання, підбір літератури	11.04.23-12.04.23	
3	Аналіз літератури з досліджуваної проблеми	13.04.23-15.04.23	
4	Аналіз технічних засобів	16.04.23-18.04.23	
5	Розробка методу розпізнавання особи студента	19.04.23-21.04.23	
6	Програмна реалізація	22.04.23-23.05.23	
7	Оформлення пояснювальної записки	24.05.23-26.05.23	
8	Перевірка на плагіат	27.05.23	
9	Рецензування	28.05.23	
10	Підготовка презентації та доповіді	29.05.23-30.05.23	
11	Занесення роботи в електронний архів	31.05.23	
12	Попередній захист кваліфікаційної роботи	31.05.23	

Дата видачі завдання 10 квітня 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Яковлева О.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 60 с., 1 табл., 38 рис., 35 джерел.

РОЗПІЗНАВАННЯ ОБЛИЧЧЯ, ВЕРИФІКАЦІЯ, DEEPFACE, GRPC, C#, ANGULAR.

Об'єктом роботи є питання online тестування знань з можливістю ідентифікації особи студента.

Метою роботи є розробка застосунку для моніторингу знань з мови SQL з можливістю розпізнавання обличчя студента.

Було створено клієнт-серверний застосунок, де для реалізації клієнтської частини був використаний Angular, а серверна частина представляє собою кілька мікросервісів, написаних за допомогою платформи .Net, які комунікують між собою за допомогою gRPC технології. Для розпізнавання обличчя була використана бібліотека DeepFace. Середовищами розробки були Rider та Visual Studio Code. Для тестування серверної частини використовувався Postman.

У результаті роботи здійснена програмна реалізація застосунку для моніторингу знань з мови SQL з можливістю розпізнавання особи студента.

FACE RECOGNITION, VERIFICATION, DEEPDACE, GRPC, C#, ANGULAR.

The object of the work is the issue of online testing of knowledge with the possibility of identifying the identity of the student.

The aim of the work is to develop an application for monitoring knowledge of SQL with the ability to recognize the student's face.

A client-server application was created, where Angular was used to implement the client part, and the server part consists of several microservices written using the .Net platform that communicate with each other using gRPC technology. The DeepFace library was used for face recognition. The development environments were Rider and Visual Studio Code. Postman was used to test the server part.

As a result of the work, a software implementation of an application for monitoring knowledge of SQL language with the ability to recognize the student's face is carried out.

ЗМІСТ

Вступ.....	7
1 Сучасний стан питання тестування знань онлайн	8
1.1 Онлайн освіта та існуючі сервіси для онлайн навчання.....	8
1.2 Сучасні сервіси для тестування знань онлайн	9
1.3 Огляд наявних сервісів для тестування знань мови SQL	9
1.4 Методи та підходи для ідентифікації особистості на основі аналізу відеопотоку вебкамери.....	10
1.5 Сучасні підходи до проєктування вебсервісів	12
1.6 Необхідний інструментарій для проєктування та розробки вебсервісів	14
1.7 Інструментарій для налаштування робочого процесу у команді	15
1.8 Постанова задачі	16
2 Проєктування вебсервісу для тестування знань мови SQL	18
2.1 Специфікація вимог до застосунку	18
2.2 Проєктування архітектури застосунку	19
2.3 Модуль ідентифікації Студента	20
2.3.1 Метод розпізнавання обличчя на основі відеопотоку з вебкамери.....	20
2.3.2 Розробка алгоритму верифікації обличчя та поведінки застосунку за результатами верифікації.....	22
2.3.3 Проєктування модулю верифікації обличчя студента.....	23
2.4 Модуль керування базами даних	25
2.5 Модуль тестування.....	26
2.6 Розробка дизайну застосунку	28
3 Розробка вебсервісу для тестування знань мови SQL.....	36
3.1 Налаштування програмного середовища.....	36
3.2 Наповнення бази даних застосунку тестовими даними	36
3.2.1 Створення теми.....	36

	6
3.2.2 Створення тестової бази даних	37
3.2.3 Створення питань	40
3.2.4 Створення тесту	42
3.2.5 Створення групи	43
3.2.6 Створення Студентів	43
3.2.7 Створення тесту за розкладом.....	45
3.3 Перевірка модулю ідентифікації Студента.....	45
3.4 Перевірка роботи модулю автоматичного визначення оцінки студента за конкретне питання тесту	48
Висновки	55
Перелік джерел посилання	57

ВСТУП

В сучасному світі інформаційних технологій мова SQL (Structured Query Language) є невід'ємною складовою баз даних та систем управління базами даних. За допомогою SQL можна виконувати різноманітні операції з даними, такі як створення, редагування, видалення, а також вибірка необхідної інформації.

Водночас, знання мови SQL є досить складною та важкою задачею для багатьох студентів, особливо для тих, хто не має великого досвіду у програмуванні. Крім того, з метою забезпечення безпеки та ідентифікації користувачів у багатьох навчальних закладах вимагається ідентифікація особи студента перед початком занять.

З метою полегшення процесу вивчення мови SQL та ідентифікації студентів, метою цієї дипломної роботи є розробка застосунку, який буде здатен моніторити знання студентів з мови SQL та розпізнавати особу студента за допомогою відеокамери. Для цього будуть використані сучасні методи машинного навчання та комп'ютерного зору.

Застосунок буде включати в себе дві основні складові: перша складова буде відповідати за моніторинг знань студентів, а друга – за розпізнавання особи студента. Перша складова буде базуватися на інтерактивній системі тестування, яка дозволить студентам проходити тести з мови SQL та отримувати негайний результат тестування. Друга складова буде використовувати відеокамеру для розпізнавання особи студента.

Розробка такого застосунку має великий практичний і науковий інтерес, оскільки він може бути використаний в навчальних закладах для полегшення процесу навчання мови SQL та контролю рівня знань студентів. Крім того, розробка такого застосунку дає можливість дослідити та застосувати сучасні методи машинного навчання та комп'ютерного зору для вирішення практичних задач.

1 СУЧАСНИЙ СТАН ПИТАННЯ ТЕСТУВАННЯ ЗНАНЬ ОНЛАЙН

1.1 Онлайн освіта та існуючі сервіси для онлайн навчання

Онлайн освіта – це форма навчання, яка використовує інтернет-технології для забезпечення доступу до освітніх матеріалів, комунікації з викладачами та однолітками, оцінки знань та отримання сертифікатів. На сьогоднішній день існує багато сервісів онлайн освіти, які пропонують різноманітні курси для різних аудиторій. Ось декілька прикладів:

– EdEra – студія онлайн-освіти, яка створює онлайн-курси, спецпроекти, інтерактивні підручники та освітні блоги [1]. EdEra спеціалізується на курсах з гуманітарних та соціальних дисциплін, а також на курсах для вчителів та громадських активістів;

– Prometheus – найбільша платформа онлайн-курсів в Україні, яка співпрацює з провідними українськими та світовими університетами, експертами та організаціями [2]. Prometheus пропонує курси з різних сфер: від економіки та маркетингу до футболу та фандрайзингу;

– Coursera – міжнародна платформа онлайн-освіти, яка об'єднує понад 200 університетів та компаній з усього світу [3]. Coursera налає доступ до тисяч курсів, спеціалізацій, професійних сертифікатів та ступенів з різних галузей знань: від мистецтва та гуманітарних наук до інформатики та бізнесу.

Ці сервіси надають можливість студентам отримувати знання та сертифікати, спілкуватися з викладачами та однолітками з усього світу та отримувати доступ до матеріалів у будь-який час та з будь-якого місця з Інтернетом. Кожен із перерахованих сервісів має свої переваги та недоліки. Але найголовнішими упущеннями є відсутність ідентифікації студента за будь-якою ознакою та неможливість інтегрування курсу із існуючою системою дистанційного навчання.

1.2 Сучасні сервіси для тестування знань онлайн

Існує багато сервісів для тестування знань онлайн, які дозволяють створювати різні типи тестів, проводити опитування в режимі реального часу або в асинхронному форматі, аналізувати результати та отримувати зворотний зв'язок. Прикладами таких сервісів можуть слугувати Kahoot! [4], LearningApps [5]. Також можна дуже ефективно оцінювати знання студента на системі dl.nure.ua при правильному поводженні з нею. Але всі вони надають можливість створювати тести із декількома варіантами відповідей, якщо у тесті повинні бути питання з розгорненою відповіддю, то викладач змушений буде перевірити відповідь кожного студента з кожного розгорнутого питання. Перевірка таких тестів потребує великої кількості часу та зусиль.

1.3 Огляд наявних сервісів для тестування знань мови SQL

Більшість застосунків вже існуючих на даний момент використовують реляційну базу даних. Мова SQL створена для того, щоб дати можливість програмісту редагувати структуру та зміст бази даних. Наразі знання цієї мови є однією із найголовніших вимог при працевлаштуванні молодих спеціалістів.

Існує чимало різних систем управління базами даних, які використовують мову SQL (але кожна з них має свої відмінності у деяких деталях): MS SQL Server, MySQL, PostgreSQL та інші. Перевірити та поліпшити свої практичні знання мови SQL можна за допомогою наступних сервісів:

– CodeWars – це сервіс, який допомагає вивчити, тренувати та вдосконалювати свої навички програмування, розв'язуючи завдання різних типів та рівнів складності [6]. Сервіс надає завдання із створення запитів,

підзапитів, функцій, тощо. Найголовнішою перевагою CodeWars є можливість переглядати відповіді інших користувачів та оцінювати їх. Рішення з найбільшою оцінкою вважається взірцевим;

– LeetCode – це сервіс, який допомагає підготуватися до технічних інтерв'ю, розвивати свої навички та знання програмування, розв'язуючи понад 2600 завдань різних тем та рівнів складності [7]. Дозволяється обрати одну з 14 популярних мов програмування, включаючи мову SQL. На LeetCode також є можливість переглядати рішення інших користувачів, але також ваше рішення порівнюється по таким параметрам, як швидкість виконання, продуктивність пам'яті, тощо.

Існують і багато інших схожих сервісів, але на їх основі складно, а найчастіше неможливо, побудувати навчальний процес та налаштувати зв'язок із студентом.

1.4 Методи та підходи для ідентифікації особистості на основі аналізу відеопотоку вебкамери

Ідентифікація особистості за обличчям є одним з видів біометричної ідентифікації, який використовує фізичні або поведінкові характеристики людини для підтвердження її особи. Для ідентифікації особистості на основі аналізу відеопотоку вебкамери можна використовувати різні методи та підходи. Але у кожному з цих методів можна виділити загальні етапи:

– виявлення обличчя. На цьому етапі система автоматично визначає присутність та положення обличчя на зображенні, що надходить від вебкамери;

– вирівнювання обличчя [8]. На цьому етапі зображення обличчя перетворюється на позиції ознак обличчя, зберігаючи цілісну структуру форми обличчя;

– отримання ознак. На цьому етапі система перетворює зображення обличчя на набір числових значень, які характеризують його форму, текстуру, кольори та інші атрибути;

– співставлення ознак [9, 10]. На цьому етапі система порівнює набори ознак, які збережені в базі даних;

– результат ідентифікації. На цьому етапі система повертає результат ідентифікації особистості: ім'я та іншу інформацію про особу, яка співпадає з зображенням обличчя; ступінь достовірності результату; список кандидатів на ідентифікацію; повідомлення про відсутність збігів або помилки.

Серед методів ідентифікації особистості за обличчям можна виділити наступні методи:

– геометричний метод – метод, який базується на вимірюванні та порівнянні відстаней, кутів та інших параметрів між характерними рисами обличчя, такими як очі, ніс, рот, підборіддя. Переваги: простота реалізації, низькі витрати на обчислення. Недоліки: низька точність, чутливість до зміни освітлення, повороту та експресії обличчя;

– метод головних компонент – це метод, який базується на зменшенні розмірності та виділенні найбільш важливої інформації з зображення обличчя. Цей метод використовує алгоритми лінійної алгебри, що обчислюють головні компоненти – ортогональні вектори, які описують напрямки найбільшої варіації даних. Переваги: висока швидкість, ефективне використання пам'яті. Недоліки: чутливість до зміни освітлення та повороту обличчя, нестабільність при додаванні нових даних;

– метод гнучкого порівняння графів – це метод, який використовує графове представлення зображення обличчя, де вершини графа відповідають ключовим точкам обличчя (наприклад, очі, ніс, рот), а ребра – відстаням між ними. Цей метод показує високу точність розпізнавання та низьку чутливість до освітлення, міміки та повороту голови. Переваги: висока точність, низька чутливість до освітлення, міміки та повороту голови. Недоліки: складність реалізації, високий час обчислення;

– метод з використанням нейронних мереж – метод, який використовує штучні нейрони для аналізу та класифікації зображень обличчя. Нейронні мережі можуть вивчати особливості обличчя, такі як форма, колір, текстура, вираз, тощо, і порівнювати їх з базою даних. Переваги: висока точність, здатність враховувати складні та абстрактні ознаки обличчя, можливість роботи з різними форматами даних. Недоліки: високий час та обсяг навчання, потреба у великому наборі даних, складність інтерпретації результатів. Найбільш стійкими до завад є згорткові нейронні мережі [11, 12]. Згорткові нейронні мережі – це клас штучних нейронних мереж, які використовують математичну операцію, яка називається згортка, замість загального множення матриць принаймні в одному зі своїх шарів.

На етапі отримання ознак зараз більшість переваг на стороні нейромережевого підходу через те, що дозволяє отримувати ознаки високого рівня абстракції, які краще відображають сутність об'єктів та їх взаємозв'язки.

У відкритому доступі існує велика кількість вже навчених моделей як для вирішення кожного з етапів розпізнавання обличчя (FaceNet [13], RenitaFace [14]), так й бібліотеки для комплексного вирішення задачі, наприклад, DeepFace [15].

Отже, із наразі відомих методів ідентифікації особистості на основі аналізу відеопотоку вебкамери немає якогось одного ідеального. Кожен із способів має як свої переваги, так і недоліки. Кінцевий вибір методу ідентифікації особистості залежить від бізнес вимог до цього процесу в умовах якогось проєкту.

1.5 Сучасні підходи до проєктування вебсервісів

Вебсервіс – це програма, яка надає певну функціональність через Інтернет за допомогою стандартних протоколів, таких як HTTP, SOAP,

XML-RPC, REST, тощо. Вебсервіси можуть бути використані для різних цілей, таких як обмін даними, інтеграція систем, автоматизація бізнес-процесів, надання віддалених сервісів тощо. Для проектування вебсервісів існують різні підходи та методології, які залежать від типу та складності вебсервісу, його архітектури, технологій та інструментів розробки. Ось деякі з них:

– модельно-орієнтований підхід (Model-Driven Approach). Цей підхід передбачає використання моделей для опису та аналізу вимог, структури та поведінки вебсервісу на різних рівнях абстракції. Моделювання дозволяє спростити та узагальнити складну реальність, а також забезпечити зв'язок між бізнес-логікою та технологічною реалізацією вебсервісу. Для моделювання вебсервісів можна застосовувати різні мови та стандарти, такі як UML (Unified Modeling Language), BPMN (Business Process Model and Notation), WSDL (Web Services Description Language), BPEL (Business Process Execution Language) тощо;

– сервісно-орієнтований підхід (Service-Oriented Approach). Цей підхід передбачає організацію вебсервісу як набору слабо зв'язаних сервісів, які можуть бути повторно використані та компоновані для задоволення різних потреб користувачів. Сервіс – це самодостатня логічна одиниця функціональності, яка має чітко визначений інтерфейс та контракт для спілкування з іншими сервісами. Сервісно-орієнтований підхід сприяє гнучкості, масштабованості, інтегрованості та адаптивності вебсервісу.

Для реалізації сервісно-орієнтованого підходу можна використовувати різні стилі та парадигми вебсервісів, такі як SOAP (Simple Object Access Protocol), REST (Representational State Transfer), GraphQL (Graph Query Language) тощо;

– ресурсно-орієнтований підхід (Resource-Oriented Approach). Цей підхід передбачає організацію вебсервісу як набору ресурсів, які можуть бути представлені в різних форматах та доступні за допомогою єдиного ідентифікатора (URI). Ресурс – це будь-яка одиниця інформації, яка має

значення для користувача або додатку, наприклад, документ, зображення, відео, запис у базі даних тощо. Ресурсно-орієнтований підхід сприяє простоті, універсальності та ефективності вебсервісу. Для реалізації ресурсно-орієнтованого підходу можна використовувати стиль REST або його модифікації.

Усі підходи мають як свої переваги, так і недоліки. Тому вибір того чи іншого підходу залежить від вимог проєкту та досвіду команди розробників. Досвід команди важливий в цьому питанні через те, що програмісти зможуть швидше і якісніше написати додаток за підходом, по якому вони вже створювали інші програми.

1.6 Необхідний інструментарій для проєктування та розробки вебсервісів

Для проєктування та розробки вебсервісів потрібно використовувати різні інструменти та технології, які залежать від типу та складності вебсервісу, його архітектури, технологій та інструментів розробки. Ось деякі з них:

– інструменти для спілкування та співпраці. Це інструменти, які допомагають організувати зв'язок між розробниками, клієнтами та іншими зацікавленими сторонами проєкту. До таких інструментів належать Skype, Microsoft Teams, Slack, Google Drive, OneDrive, Dropbox, тощо;

– інструменти для моделювання та прототипування. За допомогою цих інструментів можна описати та проаналізувати вимоги, структуру та поведінку вебсервісу на різних рівнях абстракції. До таких інструментів можна віднести UML (Unified Modeling Language), BPMN (Business Process Model and Notation), WSDL (Web Services Description Language), BPEL (Business Process Execution Language), Azure, тощо;

– інструменти для розробки та програмування. Це інструменти, які допомагають створити код та логіку вебсервісу за допомогою певних мов програмування та фреймворків, наприклад C#, Java, PHP, Angular, React та інші. Розробляти, як клієнт частину, так і серверну частину застосунку, можна у Visual Studio Code;

– інструменти для тестування та налагодження. Це інструменти, які допомагають перевірити якість і функціональність вебсервісу, виявити та усунити помилки і дефекти. До таких інструментів належать Cypress, XUnit, Postman, Chrome DevTools та багато інших;

– інструменти для розгортання та підтримки. Це інструменти, які допомагають запустити і підтримувати вебсервіс на певному сервісі. До таких інструментів можна віднести AWS (Amazon Web Services), Heroku, Docker, Git, Azure DevOps та інші.

Вибір інструментів, за допомогою яких буде реалізований вебсервіс чи будь-який інший застосунок, частіше є одним із найважливіших аспектів на перших етапах розробки додатку. Також вибір одних інструментів може бути важливіший за вибір інших, наприклад для проєкту набагато важливіше обрати мову програмування, яка б закривала усі потреби та запити від проєкту, та була легкою в реалізації та підтримці, аніж вибір інструменту для моделювання та прототипування, який може навіть не знадобитись.

1.7 Інструментарій для налаштування робочого процесу у команді

Зараз Kanban дошки є найкращим інструментом для управління задачами у команді. Створення задачі на дошці, можливість присвоєння задачі окремій людині, перегляд статусу виконання та деяких приміток, перегляд історію змін – це усе є те, за що менеджери та працівники обирають Kanban для управління задачами [16].

Найпопулярнішими та найпотужнішими Kanban сервісами на даний момент є Jira, Azure DevOps, або, якщо обирати із безкоштовних, Trello [17], monday.com [18] або Asana [19]. Але для налаштування процесу роботи у рамках розробки програмного застосунку для моніторингу знань з мови SQL з можливістю розпізнавання особи студента був обраний Notion.

Насамперед Notion [20] – це програмний застосунок для нотаток, але він має свої переваги у організації командної роботи над проєктом. По перше Notion надає усі базові можливості роботи у Kanban: створення дошок задач, можливість присвоєння задачі користувачеві (але у Notion можливо присвоїти задачі декільком людям, якщо вони працюють у парі, наприклад), можливість надавати деталі до задачі та обговорювати задачу з декількома користувачами. Мінус Notion, наприклад, перед Azure DevOps в тому, що він не зберігає історію змін задачі, так як це відбувається у додатку від Microsoft, але в свою чергу Notion із невеликої перевагу, тому що усі зміни відображаються у режимі реального часу.

Головною перевагою Notion є можливість кастомізації як сторінок, задач, так і усього процесу роботи. При створенні задачі у Jira Вам надається великий перелік полів, які Ви можете залишити порожніми. Notion же дозволяє створити перелік полів, які потрібні саме під конкретний проєкт, саме під конкретні умови роботи.

1.8 Постанова задачі

Наразі мова SQL є важливою складовою сучасної інформаційної технології та має велике значення в інформаційному суспільстві, але не вистачає сервісу, на основі якого можна було б створити навчальний процес з вивчення цієї мови. Водночас, недостатньої уваги приділяється питанням контролю та моніторингу знань студентів у цій області.

Таким чином, питання розробки сервісу для моніторингу знань мови SQL з можливістю ідентифікації особистості студента є актуальною задачею, вирішення якої суттєво допоможе організувати навчальний процес під час вивчення дисциплін, пов'язаних з базами даних.

Об'єктом роботи є питання online тестування знань з можливістю ідентифікації особи студента.

Метою роботи є розробка застосунку для моніторингу знань з мови SQL з можливістю розпізнавання обличчя студента.

Для досягнення цієї мети були сформульовані наступні задачі:

- розглянути сучасний стан питання моніторингу знань студентів, особливо в області знань мови SQL;
- вивчити підходи щодо ідентифікації особи, особливо приділити увагу методам розпізнавання обличчя на основі аналізу відеопотоку;
- розглянути сучасні підходи до проєктування вебсервісів;
- спроектувати архітектуру системи моніторингу знань студентів з мови SQL з можливістю ідентифікації особи студента;
- спроектувати модуль тестування знань з мови SQL для таких типів користувачів: адміністратор, вчитель, студент та звичай користувач, який забезпечує можливість автоматичної перевірки відповідей та відображення результатів студента;
- спроектувати модуль ідентифікації обличчя на основі аналізу відеопотоку вебкамери;
- розробити алгоритм поведінки застосунку за результатами ідентифікації;
- розробити інтерфейс користувача, який буде зручним та простим у використанні;
- реалізувати спроектовані модулі тестування та ідентифікації особи;
- провести тестування розробленого застосунку й оцінити роботу кожного з модулів, та застосунку в цілому.

2 ПРОЄКТУВАННЯ ВЕБСЕРВІСУ ДЛЯ ТЕСТУВАННЯ ЗНАНЬ МОВИ SQL

2.1 Специфікація вимог до застосунку

Застосунок для тестування знань мови SQL – це система для моніторингу знань студентів з мови SQL та для введення навчального процесу.

Система має підтримувати 4 типи користувачів:

– Вчитель. Користувач з даною роллю повинен мати можливість створення питань, тестів, назначати час проходження тестів, створювати студентів у системі. Окрім того – переглядати результати тестів студента. Система має автоматично оцінювати відповідь на питання тесту, а користувач з роллю Вчитель повинен мати можливість поставити свою оцінку, якщо система зависила чи занизила оцінку;

– Адміністратор. Користувач з даною роллю повинен мати усі можливості, що надаються користувачу з роллю Вчитель, а також повинен мати можливість створювати/редагувати/видаляти не тільки користувачів-студентів, а і користувачів з іншими ролями, окрім користувачів з роллю Адміністратор;

– Студент. Користувач з цією роллю повинен мати можливість проходження тестів, за рахунок написання SQL-скриптів на конкретне запитання тесту. Також – переглядати результати написаних їм тестів та оцінки виставленні системою та Вчителем чи Адміністратором;

– Звичайний користувач. Користувач з цією роллю повинен мати можливість надавання відповідей, які виставленні Викладачем чи Адміністратором у вільний доступ (ці питання не можуть бути використанні у тестах для Студентів).

2.2 Проєктування архітектури застосунку

Вебсервіс – є програмою із клієнт-серверною архітектурою, клієнт якої написаний за допомогою фреймворку Angular, а сервер – розбитий на декілька мікросервісів, які комунікують між собою за допомогою gRPC. Усі мікросервіси написані з використанням мови C#, окрім модулю розпізнавання обличчя студента, який написаний на мові Python. Діаграму архітектури програмного застосунку показано на рисунку 2.1.

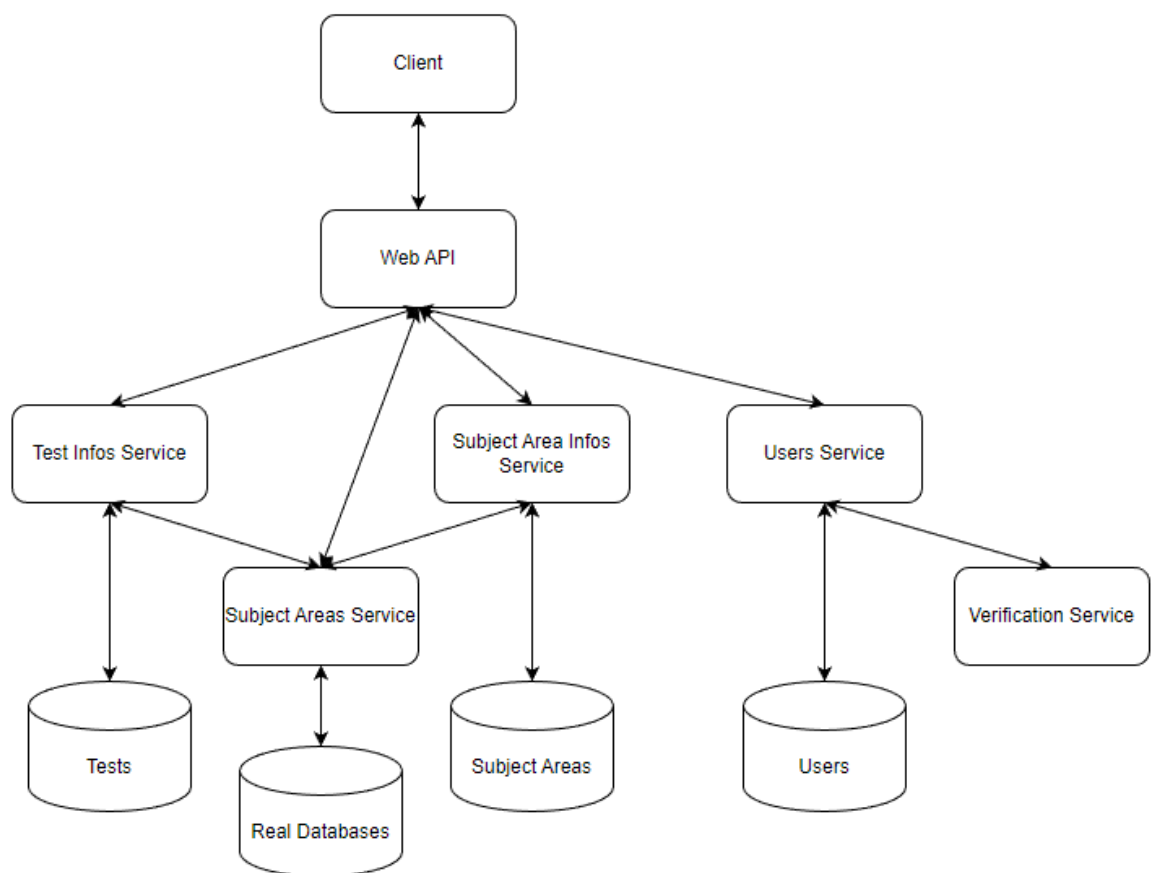


Рисунок 2.1 – Діаграма архітектури застосунку

Client – інтерфейс, який бачить користувач у своєму веббраузері.

Web API опрацьовує усі запити з Client сервісу, він виконує роль медіатора між клієнтом та сервером, також цей сервіс відповідальний за розподілення навантаження між іншими сервісами.

Test Infos Service оброблює запити створення/редагування/видалення тестів/питань, надання відповіді на питання Студентом.

Subject Area Infos Service оброблює запити на маніпулювання даними існуючих баз даних. Зберігає інформацію про усі бази даних, таблиці, які наявні в цій базі даних, повну інформацію про атрибути усіх таблиць.

Users Service маніпулює даними про усіх користувачів, окрім того допомагає авторизувати та зареєструвати користувача та верифікувати обличчя Студента.

Subject Areas Service проводить запити створення чи видалення баз даних, проводить маніпуляції із існуючими базами даних. Виконує SQL скрипти Студентів та повертає результат скрипту у вигляді JSON.

2.3 Модуль ідентифікації Студента

Модуль ідентифікації студента – є модулем, в якому відбувається ідентифікація та верифікація студента за його зображенням, отриманим з відеопотоку вебкамери. Verification Service тісно пов'язаний із Users Service, тому потрібно також і розуміння за що відповідає другий.

Users Service відповідає за усі дії, що відносяться до інформації про користувачів. Users Service оброблює та зберігає дані у базі даних Users про ролі користувачів, інформацію про них та інформацію про групи Студентів.

Verification Service в свою ж чергу відповідальний тільки за ідентифікацію та верифікацію студента на основі двох зображень.

2.3.1 Метод розпізнавання обличчя на основі відеопотоку з вебкамери

Існує багато методів розпізнавання обличчя. Усі мають свої недоліки і переваги. Нижче описані лише деякі з них [21-25]:

– метод порівняння на графах – метод, який використовує графове представлення обличчя та його елементів, таких як очі, ніс, рот, тощо. Цей

метод порівнює структуру та відстань між елементами обличчя на зображенні з еталонними графами в базі даних. Цей метод ефективний для розпізнавання обличчя з різною орієнтацією та експресією;

– метод головних компонентів – метод, який використовує статистичний аналіз для знаходження найбільш суттєвих ознак на обличчях. Цей метод перетворює зображення обличчя на набір головних компонентів (або власних обличь), які описують найбільшу частину варіації між обличчями. Цей метод ефективний для розпізнавання обличчя за умови, що вони мають однаковий розмір, орієнтацію та освітлення;

– нейронні мережі – метод, який використовує штучний інтелект для навчання та узагальнення ознак з обличь. Цей метод використовує багатопшарову нейронну мережу, яка приймає на вхід зображення обличчя та виводить клас або марку. Цей метод потребує багато даних для навчання та може бути дуже точним для розпізнавання обличчя з різних ракурсів, масштабів та умов освітлення.

Алгоритми розпізнавання можна поділити на дві частини – ті, що включають в себе використання нейронних мереж, та ті, що працюють на алгоритмах без використання нейронних мереж [26]. Недоліками нейронних мереж є складність навчання, так як для кожного нового набору даних доводиться аналізувати інформацію та навчати систему знов, що є досить складним процесом. Для алгоритмів, які не використовують нейронні мережі такі етапи, як предобробка та сегментація є обов'язковими для того, щоб результати розпізнавання були коректні, а, відповідно, не вірно підібрані алгоритми сегментації можуть привести до того, що результати розпізнавання будуть не коректні [27].

Найбільш ефективними нейронними мережами для розпізнавання обличчя є згорткові нейронні мережі [28-30]. Це такий тип штучних нейронних мереж, які використовують операцію згортки для витягування ознак з вхідних даних. Згортка – це математичка операція, яка застосовує фільтр (або ядро) до даних, щоб отримати нове представлення даних.

Згорткові нейронні мережі складаються з різних шарів, таких як згортковий шар, агрегувальний шар, повноз'єднаний шар та шар втрат.

Недолік методу з використанням нейронних мереж (потреба в великому обсязі даних для навчання) перебивають його переваги у вигляді ефективності розпізнавання при різних умовах, як поганих, так і хороших. Існує безліч вже натренованих нейронних мереж із зрозумілим інтерфейсом користування у вільному доступі і з відкритим вихідним кодом. Саме тому для вирішення даної задачі була обрана бібліотека DeerFace, яка використовує вже натреновану на мільйонах зображень нейронну мережу. Автори даної бібліотеки стверджують, що точність результатів даної нейронної мережі становить більш ніж 97,35% [31]. Також перевага даній бібліотеці була надана спираючись на дослідження Мей Вонг та Вейхон Ден із порівняння між собою різних інструментів для розпізнавання обличчя [32]. За результати їх дослідження DeerFace мав точність розпізнавання обличчя 99,8%, в той час коли точність, із якою обличчя з таких самих зображень розпізнавала людина, склала лише 97,53%. Що говорить про те, що дана нейронна мережа справляється із вирішенням даної проблеми краще за людину [33].

2.3.2 Розробка алгоритму верифікації обличчя та поведінки застосунку за результатами верифікації

Алгоритм верифікації обличчя складається з наступних кроків:

Крок 1. Взяти із відеопотоку з вебкамери фрейм із зображенням.

Крок 2. Відправити зображення на Web API, який в свою чергу відправить це зображення на User Service.

Крок 3. Дістати із сховища вихідне зображення Студента, який проходить верифікацію.

Крок 4. Відправити обидва зображення до Verification Service.

Крок 5. Детектування обличчя. Аналізування зображення на наявність областей, які можуть містити обличчя.

Крок 6. Нормалізація. Перетворення зображення таким чином, щоб воно мало стандартний розмір, положення, освітлення і орієнтацію. Це дозволяє підвищити якість розпізнавання обличчя і зменшити вплив шуму і деформацій.

Крок 7. Видобуття ознак. Аналізування зображення на наявність характерних рис, які можуть використовуватися для розпізнавання особи, та перетворення їх на набори числових значень (ознак). Цей крок допомагає зменшити розмір даних та виділити суттєву інформацію для розпізнавання.

Крок 8. Порівняння видобутих ознак. Порівняння векторів, здобутих із двох зображень. Результатом порівняння є коефіцієнт подібності.

Крок 9. Повернення результату верифікації. Якщо коефіцієнт порівняння більший за порогове значення (порогове значення за замовчуванням у DeepFace дорівнює 0,4), то повертається результат, який говорить, що Студент був верифікований і може проходити тест. Якщо цей коефіцієнт менше порогового значення чи якщо на зображенні взятому з відеопотоку два обличчя, верифікація вважається непройденою і програма попросить Студента пройти верифікацію повторно.

2.3.3 Проєктування модулю верифікації обличчя студента

Коли користувач намагається отримати доступ до проходження тесту, він/вона повинен пройти верифікацію обличчя. Для цього програма вмикає вебкамеру користувача, користувач в свою чергу обирає момент, в який буде витягнутий фрейм з відеопотоку з вебкамери. Цей фрейм буде відправлений на Web API у вигляді масиву байтів, після чого Web API перенаправить цей запит до Users Service, який із бази даних Users витягне шлях до вихідного зображення на Users Storage. Users Storage – це якесь сховище (хмара чи

навіть пам'ять серверу, на якому працює Users Service). Після отримання шляху до вихідного зображення користувача, який намагається пройти верифікацію, зображення зчитується і перетворюється у масив байтів. Далі обидва зображення у вигляді масивів байтів передається до Verification Service, який тимчасово зберігає обидва зображення локально, тобто на Verification Storage, після чого використовуємо ці збережені зображення у методі верифікації, який надає бібліотека DeepFace. Отримаємо результати верифікації студента за двома зображеннями, після чого видаляємо створенні зображення для верифікації. Після чого повертаємо результат у вигляді JSON із єдиним булевим полем, яке зберігає в собі результат верифікації. Схема цього модулю показана на рисунку 2.2.

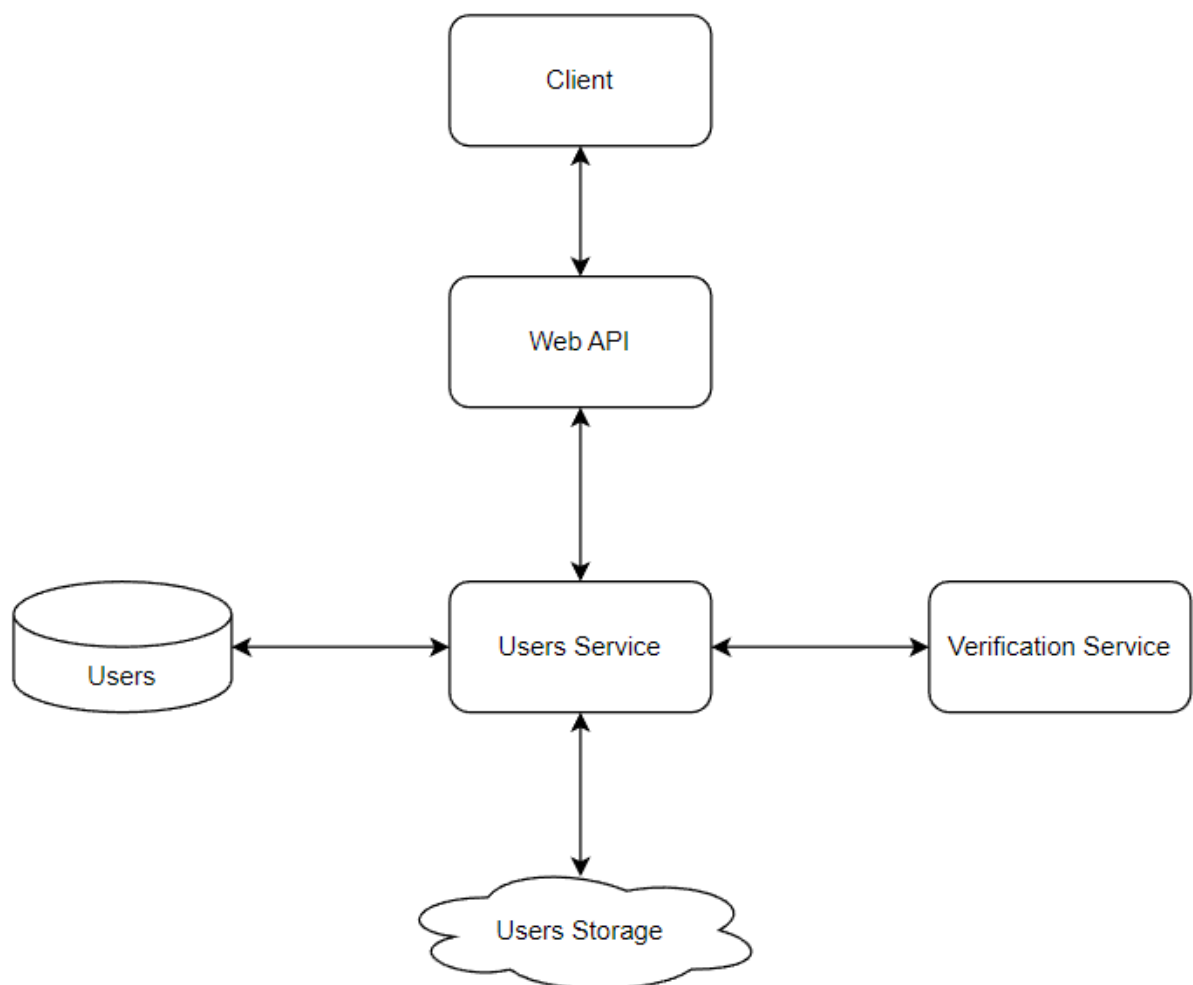


Рисунок 2.2 – Діаграма архітектури модулю верифікації студента

2.4 Модуль керування базами даних

Модуль керування базами даних є проміжною ланкою між клієнтом та реальними тестовими базами даних. База даних цього модуль має повну інформацію про усі тестові бази даних, інформацію про усі їх таблиці та атрибути, що допомагає Вчителю/Адміністратору легко налаштовувати усі бази даних прямо із застосунку.

Структуру бази даних цього модулю продемонстровано на рисунку 2.3.

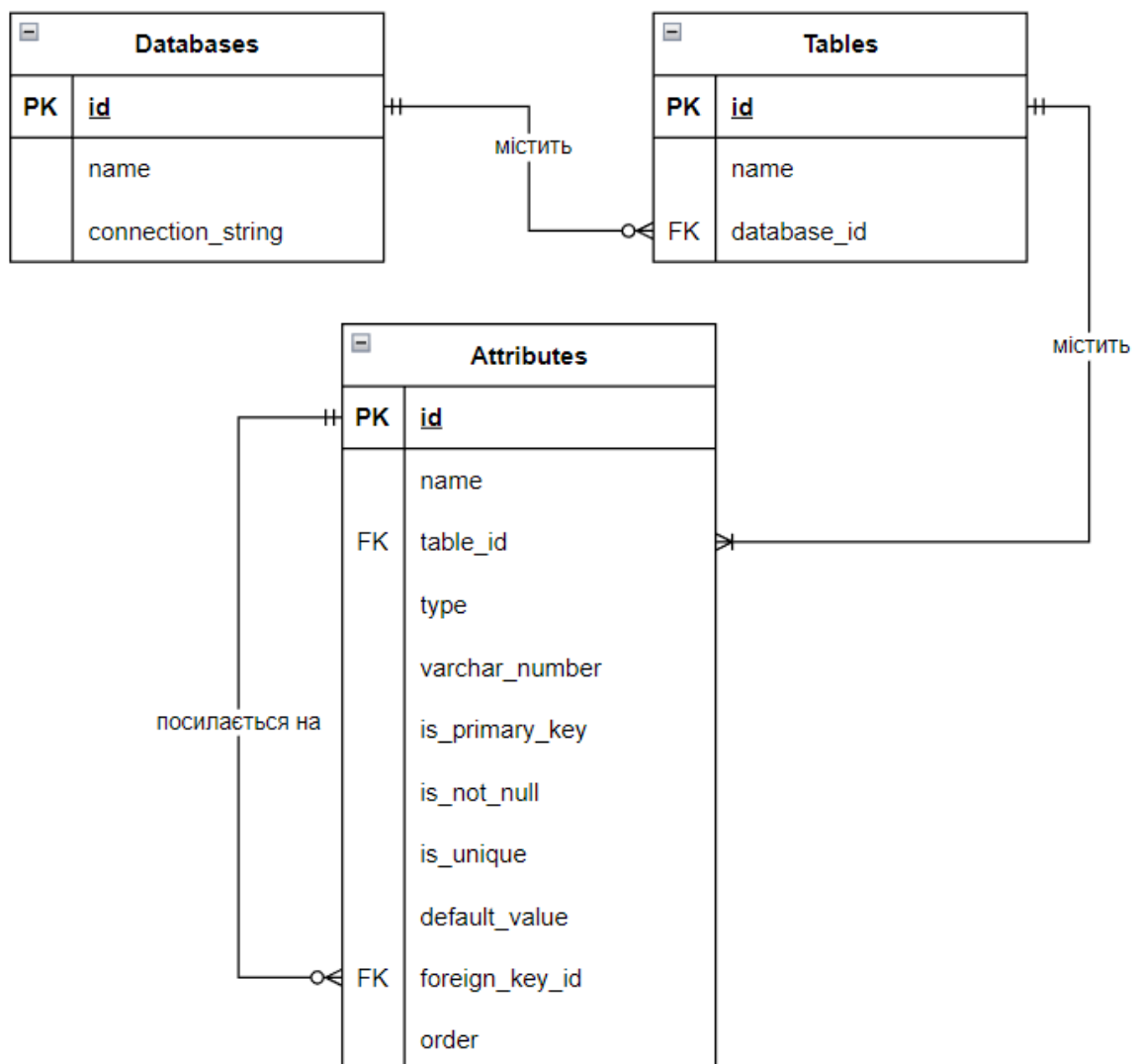


Рисунок 2.3 – Структура бази даних модулю керування базами даних

2.5 Модуль тестування

Модуль тестування є основним для застосунку, тому що більшість доменної логіки лежить саме в його основі. Створення тестів та питань для них, автоматична перевірка відповідей студентів та інше. Також цей модуль містить у собі слабкі відносини із іншими модулями: ідентифікатор користувача, який буде складати тест, чи ідентифікатор бази даних, з якою пов'язане питання із тесту. Але в середині модулю немає жодної інформації про користувачів чи бази даних, лише їх ідентифікатори.

Структуру бази даних цього модулю продемонстровано на рисунку 2.4.

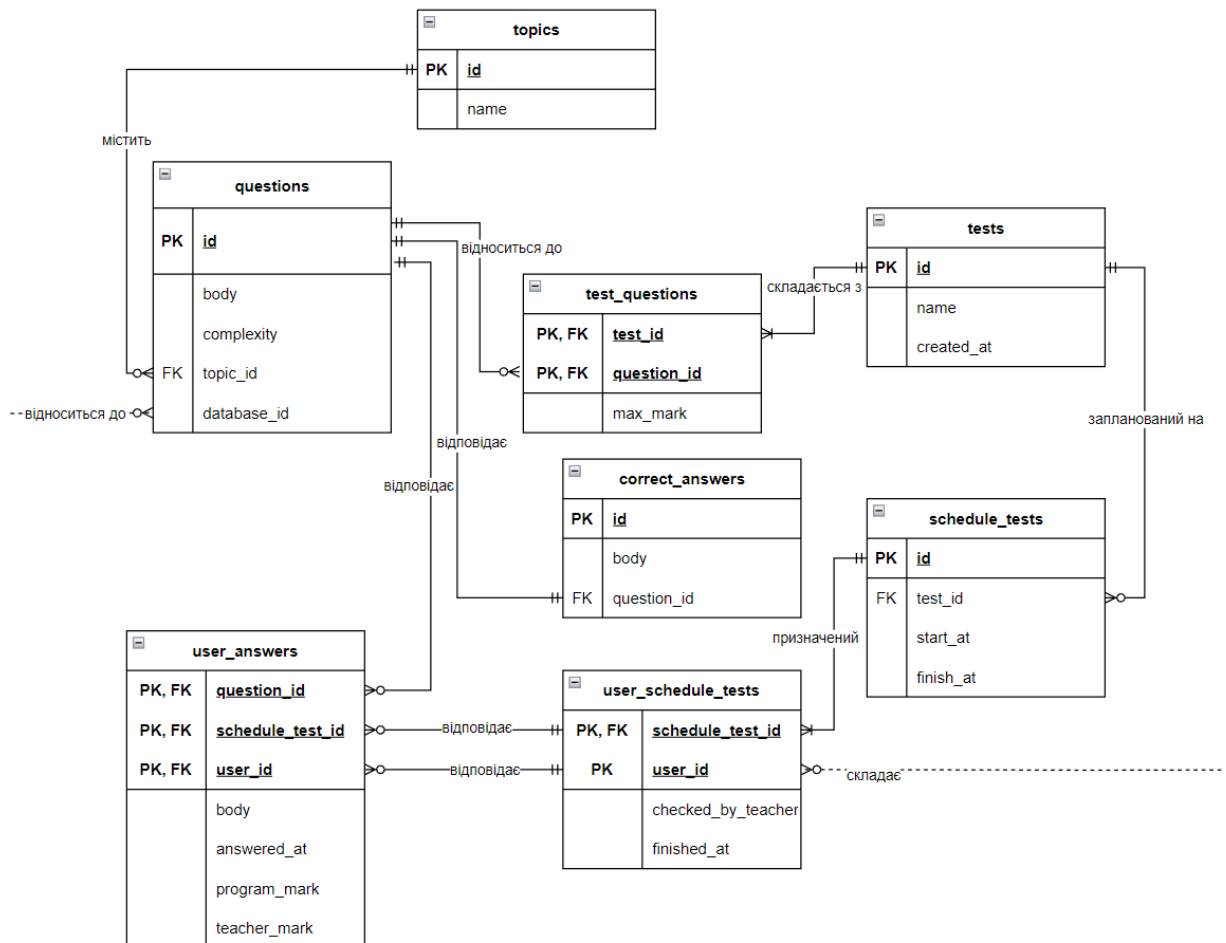


Рисунок 2.4 – Структура бази даних модулю тестування

Модуль тестування має у собі одну дуже важливу логіку: алгоритм оцінювання SQL-запиту.

Алгоритм перевіряє правильність виконання лише *select*-запитів та не має підтримки пошуку ключових слів.

Сам алгоритм складається з декількох кроків:

Крок 1. Отримання результатів виконання двох скриптів: скрипт-відповіді Студента і скрипт правильної відповіді. Результати повертаються у вигляді json.

Крок 2. Перетворювання результатів виконання із формату json у масив словників (ключ – назва атрибуту, значення – значення конкретного атрибуту для конкретного рядка).

Крок 3. Порівняння кількості рядків результатів виконання запитів. Якщо результати мають різну кількість рядків, оцінка зменшується вдвоє від максимально-можливої.

Крок 4. Порівняння кількості атрибутів у результаті. Якщо кількість атрибутів відрізняється, оцінка зменшується на 10% від тієї, що була отримана на Кроці 3.

Крок 5. Порівняння назв атрибутів у результатах виконання обох скриптів. Якщо кількість атрибутів, яким не знайшли відповідної назви більша за різницю кількості атрибутів результатів виконання скриптів, оцінка зменшується на 10% від тієї, яка була отримана на Кроці 4.

Крок 6. Порівняння даних. Рахуємо усі значення, які співпадають. Також розраховуємо максимальну кількість схожих значень: максимальна кількість атрибутів серед двох результатів помножена на максимальну кількість строк серед двох результатів. Кількість співпалих значень, потрібно поділити на максимальну кількість схожих значень. Отриманий коефіцієнт помножається на оцінку отриману на Кроці 5 та округлюється до другого знаку після коми.

2.6 Розробка дизайну застосунку

Усі користувачі мають однакову головну панель із кнопками авторизації та реєстрації та панель навігації (відрізняється тільки зміст даної панелі). Виходячи із вимог до застосунку деякі типи користувачів мають деякі обмеження функціоналу. Усі ці обмеження впливають на те, що буде бачити перед собою користувач.

Користувач із роллю Адміністратор чи Вчитель має 7 меню на панелі навігації (рис. 2.5): Tests, Themes, Questions, Subjects, Users, Groups та Schedule Tests.

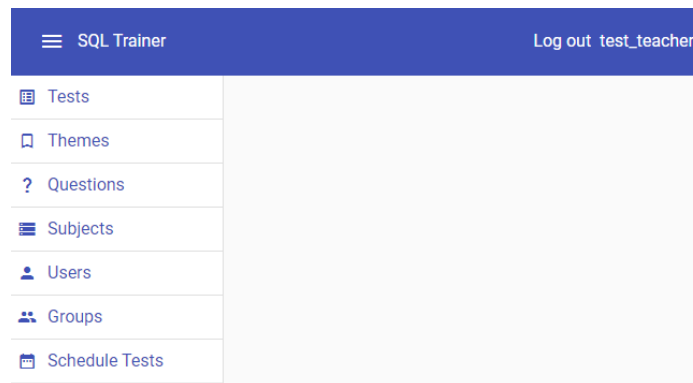
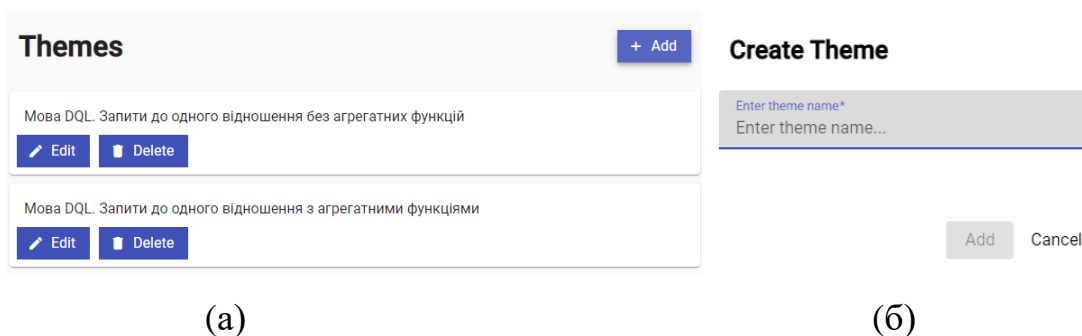


Рисунок 2.5 – Головне меню Вчителя/Адміністратора

У меню Themes відображаються існуючі у системі теми (рис. 2.6 (а)). Користувач може створювати, редагувати чи видаляти теми. У меню створення/редагування (рис. 2.6 (б)) користувачу надається лише поле для назви теми.

У меню Questions користувач бачить усі питання наявні у базі даних застосунку (рис. 2.7 (а)). Користувач має можливість додавати чи видаляти питання, редагувати зміст питання, його складність, тощо.

У меню створення/редагування питання (рис. 2.7 (б)) користувач повинен обрати тему із списку, обрати базу даних, до якої буде робитись запит, ввести саме питання, ввести складні запитання та написати правильну відповідь до цього питання.



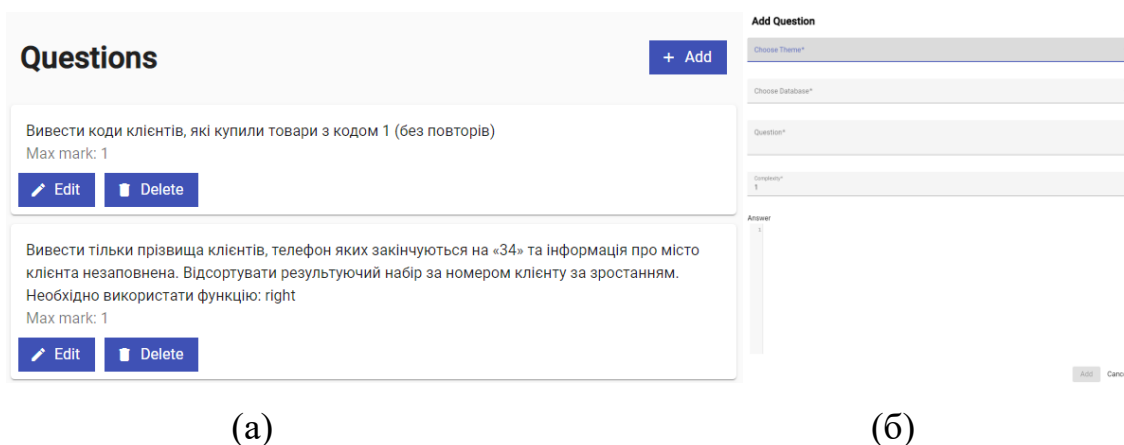
(a)

(б)

Рисунок 2.6 – Меню Themes:

(a) вигляд меню Themes;

(б) вигляд модального меню створення/редагування теми



(a)

(б)

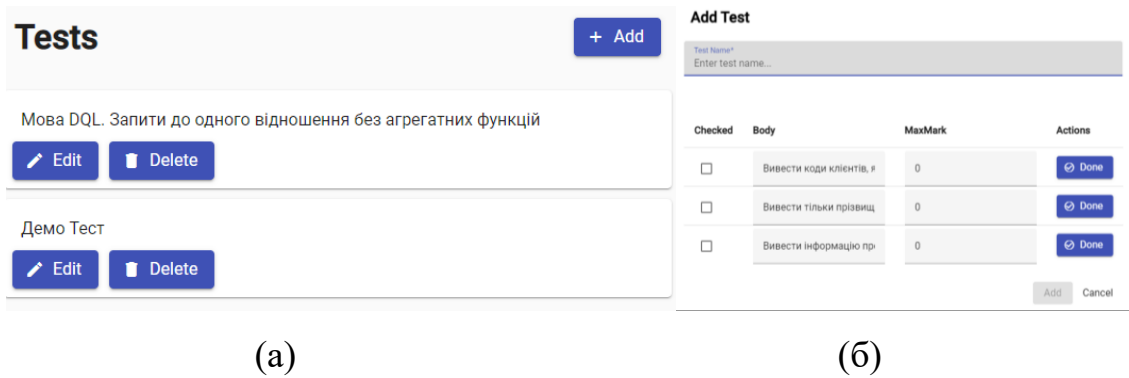
Рисунок 2.7 – Меню Questions:

(a) вигляд меню Questions;

(б) вигляд модального меню створення/редагування питання

Перейшовши до меню Tests (рис. 2.8 (a)) користувач бачить усі існуючі тести. Користувач має можливість створювати/редагувати/видаляти ці тести. У меню створення/редагування тесту (рис. 2.8 (б)) є поле для назви тесту та таблиця із можливістю вибору запитань, які будуть складати цей тест.

У меню Groups (рис. 2.9 (a)) користувачеві відображають групи студентів. Користувач також має можливість додавати/редагувати/видаляти групи. При створенні чи редагуванні групи (рис. 2.9 (б)) користувачеві потрібно лише ввести назву цієї групи.



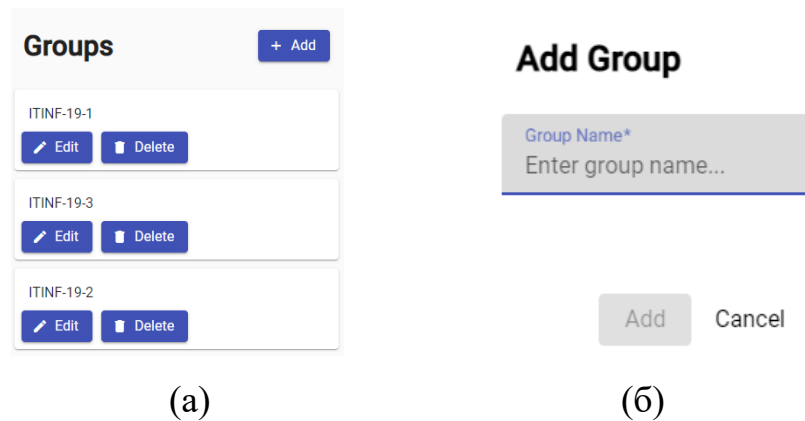
(a)

(б)

Рисунок 2.8 – Меню Tests:

(a) вигляд меню Tests;

(б) вигляд модального меню створення/редагування тесту



(a)

(б)

Рисунок 2.9 – Меню Groups:

(a) вигляд меню Groups;

(б) вигляд модального меню створення/редагування групи

У меню Users (рис. 2.10 (a)) користувач із роллю Адміністратор бачить усіх користувачів системи окрім інших користувачів із роллю Адміністратор, Вчителю відображаються лише Студенти. Користувач і з роллю Вчитель, і з роллю Адміністратор може додавати, редагувати та видаляти інших користувачів, але з деякими обмеженнями. Адміністратор може робити усі вище перелічені дії з усіма записами про користувачів окрім тих, у яких роль так само Адміністратор. Вчитель може створювати/редагувати/видаляти записи лише про користувачів із роллю Студент.

У меню створення/редагування користувача (рис. 2.10 (б, в)) вимагається ввести ім'я користувача його логін чи email та пароль для цього користувача. При створенні користувача з роллю Студент обов'язково потрібно обрати групу із наявних у системі та завантажити фотографію Студента для можливості верифікувати його обличчя перед проходженням тесту.

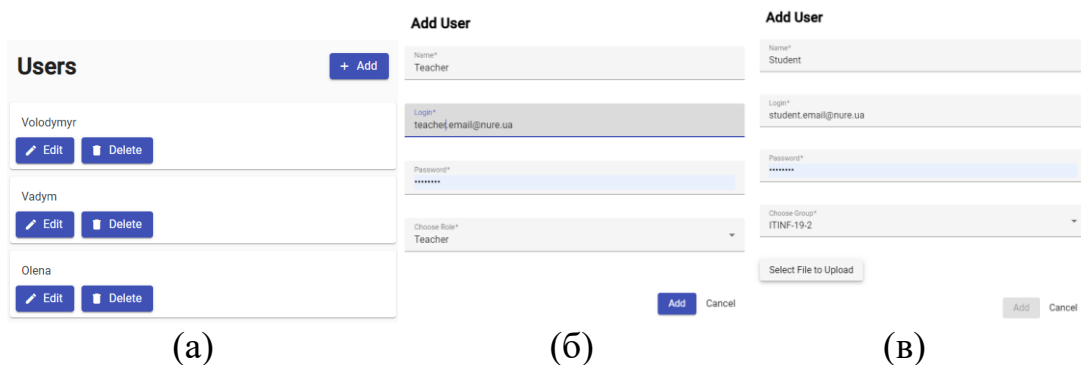


Рисунок 2.10 – Меню Users:

(а) вигляд меню Users;

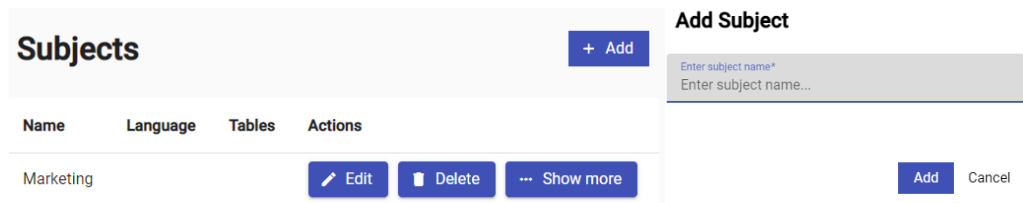
(б) вигляд модального меню створення/редагування для Адміністратора;

(в) вигляд модального меню створення/редагування для Вчителя

Subjects (рис. 2.11 (а)) – меню, у якому користувач може редагувати існуючі для тестування бази даних. Користувач може додати (рис. 2.11 (б)), видалити чи перейменувати базу даних.

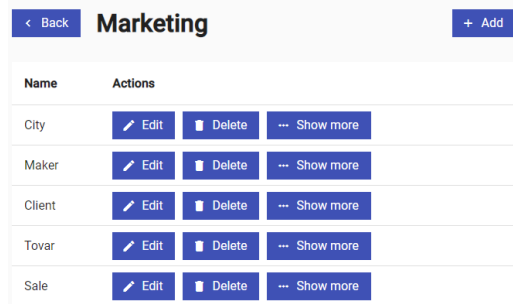
Також має можливість редагувати таблиці (рис. 2.11 (в)), наявні у конкретній базі. Редагування/створення таблиці (рис. 2.11 (г)) вимагає від користувача надати назву таблиці та додати щонайменше один атрибут. Атрибути повинні мати назву, тип обов'язково і якщо необхідно кількість символів у varchar, відмітки біля primary key, not null, unique, проставити default значення, обрати ключ на який має посилатись дане поле як зовнішній ключ у полі foreign key, встановити порядок, в якому атрибут буде відображатись у полі order. Також користувачеві надається можливість

наповнювати таблиці тестовими даними. Переглядання даних таблиці показано на рисунку 2.11 (г), а їх додавання на рисунку 2.11 (д).

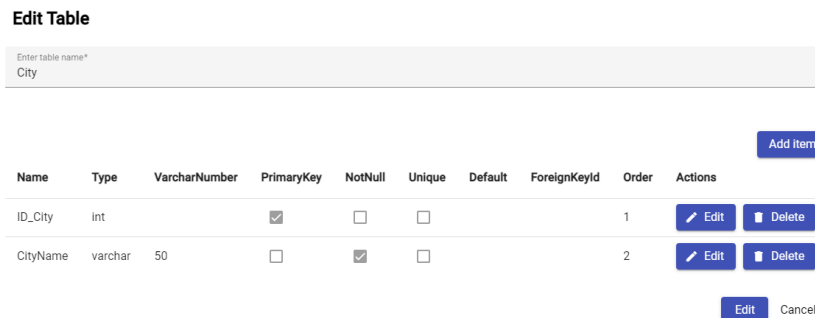


(a)

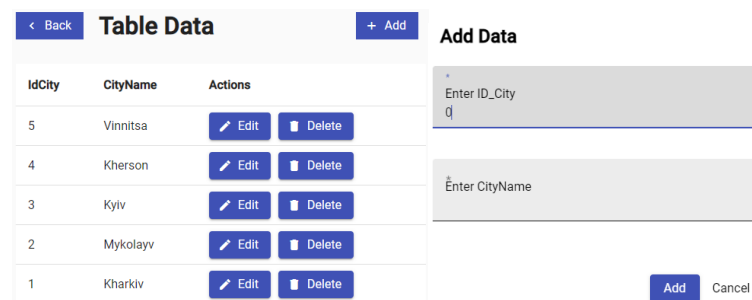
(б)



(B)



(Г)



(г)

(Д)

Рисунок 2.11 – Меню Subjects:

- (а) ВИГЛЯД МЕНЮ БАЗ ДАНИХ; (б) ВИГЛЯД МЕНЮ ДОДАВАННЯ БАЗИ ДАНИХ;
 (в) ВИГЛЯД МЕНЮ ТАБЛИЦЬ; (г) ВИГЛЯД МЕНЮ ДОДАВАННЯ/РЕДАГУВАННЯ ТАБЛИЦІ;
 (д) ВИГЛЯД МЕНЮ ДАНИХ ТАБЛИЦІ; (д) ВИГЛЯД МЕНЮ ДОДАВАННЯ ДАНИХ У ТАБЛИЦЮ

У меню Schedule Tests (рис. 2.12 (а)) користувач бачить розклад усіх назначених їм тестів. При створенні чи редагуванні тесту (рис. 2.12 (б)) по розкладу потрібно обрати тест, обрати час його проведення (початковий та кінцевий час) та обрати користувачів, які повинні пройти даний тест. Також для зручності вводу користувачеві надається список груп, коли користувач обирає групу, автоматично обираються усі Студенти, які прив'язані до даної групи.

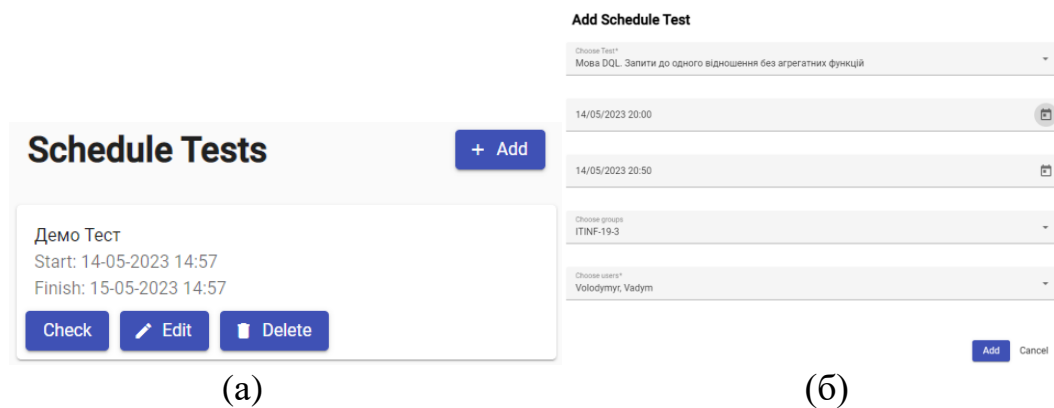


Рисунок 2.12 – Меню Schedule Tests:

(а) вигляд меню Schedule Tests;

(б) вигляд меню створення/редагування тесту за розкладом

Також користувач може видалити тест по розкладу за потреби. Окрім цього користувач з роллю Адміністратор чи Вчитель може переглядати відповіді, наданні кожним Студентом. Користувач не може переглянути результат складання тесту Студента, якщо той ще не закінчив його проходження. При перегляданні відповідей, наданих Студентом, користувач бачить питання, саму відповідь Студента та під нею правильну відповідь на це питання, нижче відображається максимальна оцінка, яку можна отримати за це питання, оцінку, яку автоматично проставила програма та поле для вводу оцінки, яка буде правильною на розсуд користувача.

Також для зручності користувача поруч із запитом є дерево-структура бази даних. При перевірці відповідей користувач може виконати запити із правильною відповіддю та відповіддю Студента.

Користувач із роллю Студент має лише меню Schedule Tests. На цьому меню так само як і для користувачів з роллю Адміністратор чи Вчитель відображаються тести по розкладу, але лише ті, в яких зазначено, що цей Студент повинен пройти даний тест. Тест має назву, час, коли його потрібно пройти та кнопку Start/Check. Кнопка Start відображається як неактивна, якщо тест ще не пройдений або вже минув, але Студент за якихось причин його не склав. Якщо Студент ще не пройшов даний тест та час складання цього тесту ще не сплив, Студент буде бачити активну кнопку Start. Якщо Студент склав тест у зазначений Вчителем/Адміністратором час, кнопка Start не буде відображатися, замість неї з'явиться кнопка Check із однією галочкою. Друга галочка з'явиться після того, як Вчитель/Адміністратор виставить власну оцінку за проходження тесту.

Якщо кнопка Start відображається і відображається як активна, що означає, що Студент може в даний скласти тест. Натиснувши на цю кнопку, Студентові відобразиться меню верифікації. Підібравши влучний ракурс та налаштувавши освітлення, Студент має натиснути кнопку Take Snapshot, після чого відбудеться процес верифікації. За умови невдалої верифікації, застосунок відобразить відповідне повідомлення та попросить Студента спробувати пройти верифікацію ще раз. За умови успішної верифікації Студента перекине у меню проходження тесту.

Меню проходження тесту (рис. 2.13) – це меню, яке складається з декількох частин: заголовок, структура бази даних, питання та поле для відповіді і панель навігації між питаннями. У заголовці відображаються кнопка виходу із тесту, назва самого тесту та таймер із часом, який залишився. Структура бази даних – це дерево, перший рівень якого назва бази даних, другий – таблиці, третій – атрибути. Панель навігації між питаннями – це панель, на якій списком відображенні усі питання та є дві кнопки Check All та Submit, але в один момент часу відображається тільки одна із них. Спочатку відображається кнопка Check All, натиснувши на цю кнопку перевіряться відповіді по усім питанням і якщо усі запити можуть

бути виконані без помилок, то кнопка Check All зникне і з'явиться Submit. Натиск кнопки Submit спровокує збереження відповідей на запитання та вихід студента із меню проходження тесту. Основна частина цього меню – це частина, на якій зображено тіло запитання, кнопки Execute, Next та Previous і поле для вводу скрипу з підтримкою підсвітки синтаксису та автоматичним закінченням операцій чи назв таблиць, атрибутів, тощо. Кнопки Next та Previous використовуються для навігації між питаннями, де Next перенесе користувача на питання вперед (якщо Студент не на останньому питанні, в такому випадку нічого не відбудеться, тому що кнопка буде неактивна), а кнопка Previous – на питання назад (якщо Студент не на першому питанні, в такому випадку нічого не відбудеться, тому що кнопка буде неактивна). При натисканні на кнопку Execute програма виконає запит на Web API із перевіркою скрипту, написаного Студентом. У разі наявності помилки у скрипті, Студентові під головним компонентом відобразиться текст помилки, яку видала система управління базами даних. У разі відсутності помилок, під головним компонентом користувач побачить таблицю-результат, тобто дані, які витягнув його/її скрипт із бази даних.

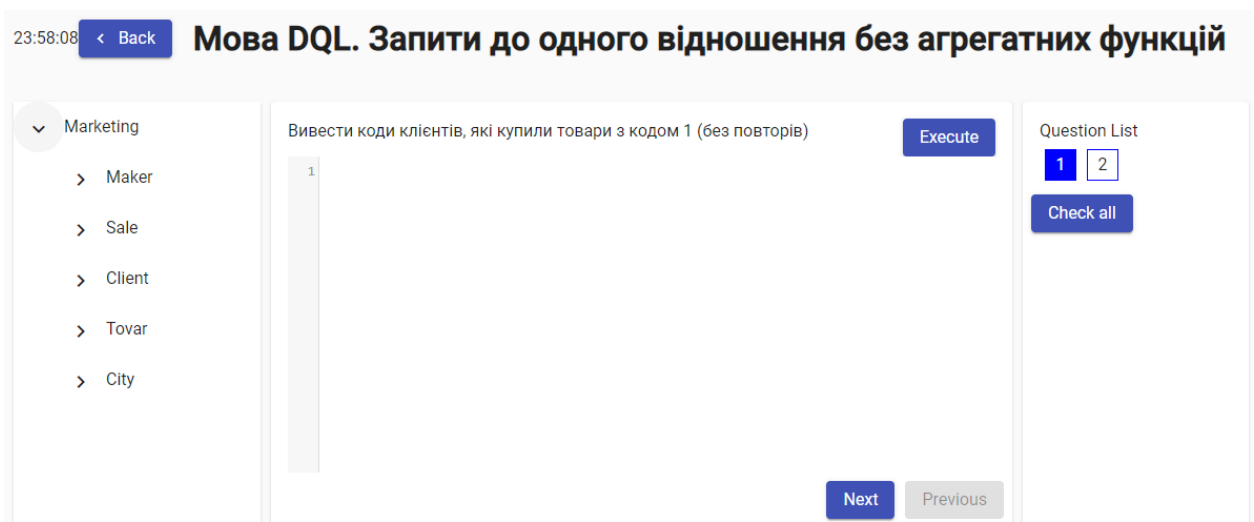


Рисунок 2.13 – Меню проходження тесту

3 РОЗРОБКА ВЕБСЕРВІСУ ДЛЯ ТЕСТУВАННЯ ЗНАНЬ МОВИ SQL

3.1 Налаштування програмного середовища

Через те, що програмний застосунок реалізований за принципом мікросервісів, то краще розгортати кожний сервіс на окремій машині і налаштувати їх зв'язок між собою за засобами локальних мереж. Кожен із сервісів, окрім Verification Service, потребує встановленій на машині .NET 7 та Postgres. Verification Service в свою чергу потребує встановленого Python 3, Flask [34], DeepFace. Розгортання Client потребує встановлених на машині npm, node JS та Angular CLI 15.

3.2 Наповнення бази даних застосунку тестовими даними

Для проведення більш точного аналізування результатів роботи програмного застосунку та результатів верифікації Студента за зображенням з відеопотоку з вебкамери.

3.2.1 Створення теми

Для демонстрації роботи застосунку було створено тему «Мова DQL» (рис. 3.1).

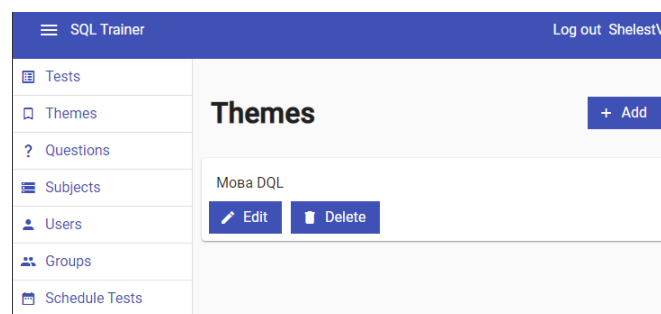


Рисунок 3.1 – Створена тема «Мова DQL»

3.2.2 Створення тестової бази даних

Для демонстрації роботи застосунку було створено базу даних Marketing (рис. 3.2).

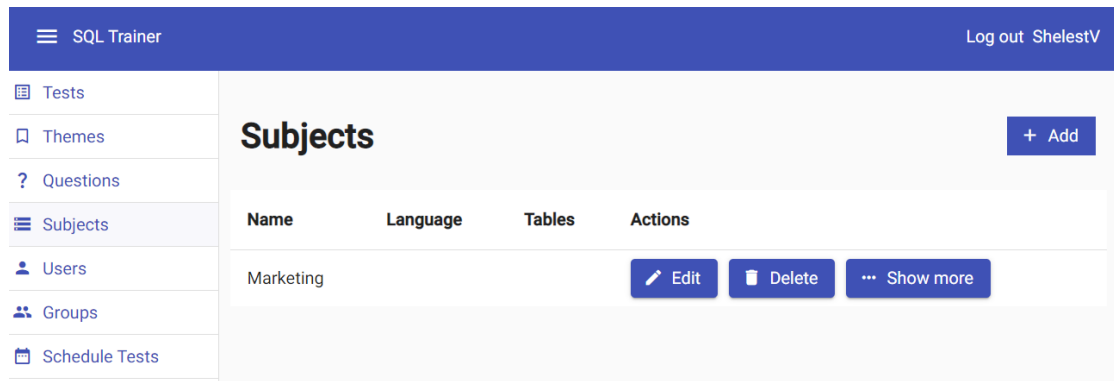


Рисунок 3.2 – Створена база даних Marketing

База даних Marketing має таблиці (рис. 3.3): City, Maker, Client, Tovar, Sale.

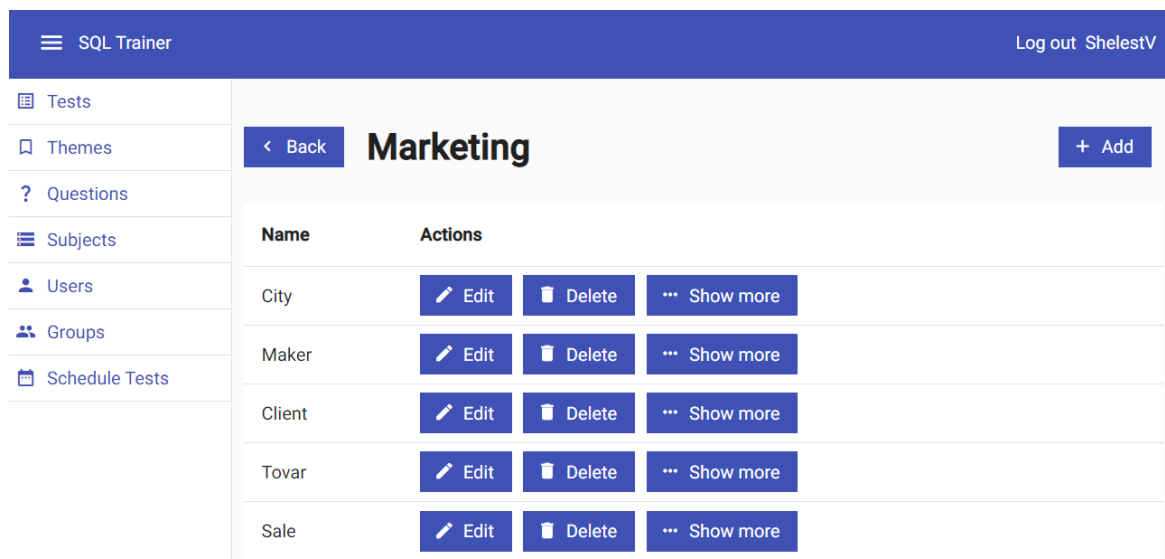


Рисунок 3.3 – Створені таблиці бази даних Marketing

Таблиця City (рис. 3.4) зберігає в собі інформацію про міста, має лише поля ID_City та CityName. ID_City це первинний ключ таблиці, ідентифікатор запису. CityName назва міста.

IdCity	CityName	Actions
5	Vinnitsa	Edit Delete
4	Kherson	Edit Delete
3	Kyiv	Edit Delete
2	Mykolaiv	Edit Delete
1	Kharkiv	Edit Delete

Рисунок 3.4 – Наповнення таблиці City

Таблиця Maker (рис. 3.5) зберігає дані про виробників. Має наступні атрибути: ID_Maker, NameMaker, ID_CityMaker, Phone та Address_in_city, де ID_Maker – це первинний ключ, NameMaker – назва/ім'я виробника, ID_CityMaker – ідентифікатор міста (зовнішній ключ, який посилається на таблицю City), Phone – номер телефону виробника, Address_in_city – адреса виробника.

IdMaker	NameMaker	IdCityMaker	Phone	AddressInCity	Actions
2	Mykola	2			Edit Delete
1	KNURE	1	+380634475396	Nauky Ave. 14	Edit Delete

Рисунок 3.5 – Наповнення таблиці Maker

У таблиці Client (рис. 3.6) зберігаються дані про клієнтів. Таблиця має поля: ID_Client (ідентифікатор), NameClient (ім'я клієнта), ID_CityClient (ідентифікатор міста клієнта), Phone (номер телефону клієнта).

IdClient	NameClient	IdCityClient	Phone	Actions
4	Dmytro	1		Edit Delete
3	Andriy	3		Edit Delete
2	Iryna	4		Edit Delete
1	Olena	3	+380634475396	Edit Delete

Рисунок 3.6 – Наповнення таблиці Client

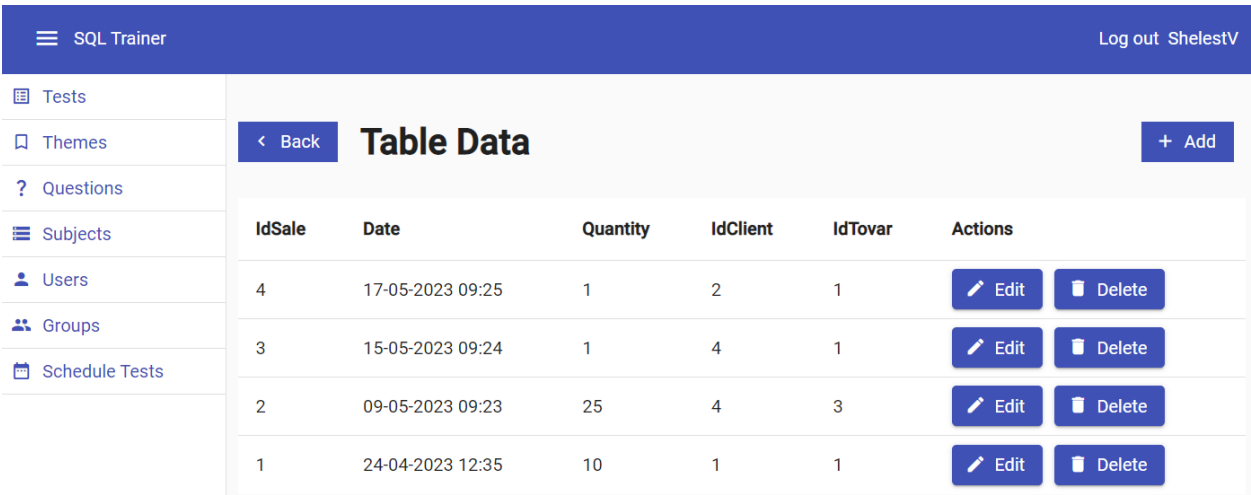
Таблиця Товар (рис. 3.7) зберігає в собі інформацію про товари виробника. Ідентифікатор товару зберігається у полі ID_Tovar, назва товару у NameTovar, ідентифікатор виробника у ID_Maker, матеріал, з якого був виготовлений товар, у Material, сорт товару (перший, другий, найвищий) у Sort, ціна товару у Price.

IdTovar	NameTovar	IdMaker	Material	Sort	Price	Actions
3	Ball	2	skin	second	200	Edit Delete
2	Laptop	1	plastic	The best	50	Edit Delete
1	Headphones	1	plastic	First	100	Edit Delete

Рисунок 3.7 – Наповнення таблиці Товар

Найголовнішою таблицею в цій базі даних є Sale (рис. 3.8). Дана таблиця пов'язує між собою Товар та Client, а пов'язавши інформацію про товари і клієнтів, можливо пов'язати Client та Maker. Sale зберігає у собі інформацію про те, коли конкретний клієнт купував конкретний товар в

атрибутах: `ID_Sale` (ідентифікатор запису, первинний ключ), `Date` (дата здійснення покупки), `Quantity` (кількість одиниць конкретного товару, які придбав конкретний клієнт), `ID_Client` (ідентифікатор клієнта, зовнішній ключ, який посилається на таблицю `Client`) та `ID_Tovar` (ідентифікатор товару, зовнішній ключ, який посилається на таблицю `Tovar`).



IdSale	Date	Quantity	IdClient	IdTovar	Actions
4	17-05-2023 09:25	1	2	1	Edit Delete
3	15-05-2023 09:24	1	4	1	Edit Delete
2	09-05-2023 09:23	25	4	3	Edit Delete
1	24-04-2023 12:35	10	1	1	Edit Delete

Рисунок 3.8 – Наповнення таблиці `Sale`

Для тестування системи питання для тесту будуть створені на основі саме цієї бази даних, саме з такими даними.

3.2.3 Створення питань

Для демонстрації проходження тесту буде достатньо таких запитань:

а) вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті;

б) вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку;

в) вивести кількість товарів (будь-яких), які купував клієнт за увесь час;

г) вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE);

г) вивести ім'я виробника та суму, на яку він продав товарів.

Правильні відповіді на запитання перелічені вище представлені у лістингах з 3.1 по 3.5.

Лістинг 3.1 Правильна відповідь на питання «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті»:

```
select distinct c.CityName, cl.NameClient, m.NameMaker
from City c
      inner join Client cl on c.ID_City = cl.ID_CityClient
      inner join Maker m on c.ID_City = m.ID_CityMaker
```

Лістинг 3.2 Правильна відповідь на питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку»:

```
select cl.NameClient, max(m.NameMaker) 'NameMaker', t.NameTovar,
      s.[Date], sum(t.Price * s.Quantity) 'Sum'
from Sale s
      inner join Client cl on s.ID_Client = cl.ID_Client
      inner join Tovar t on s.ID_Tovar = t.ID_Tovar
      inner join Maker m on t.ID_Maker = m.ID_Maker
group by cl.NameClient, t.NameTovar, s.[Date]
```

Лістинг 3.3 Правильна відповідь на питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку»:

```
select cl.NameClient, count(*) 'Count'
from Sale s
      inner join Client cl on s.ID_Client = cl.ID_Client
group by cl.NameClient
```

Лістинг 3.4 Правильна відповідь на питання «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)»:

```
select cl.*
from Client cl
      inner join Sale s on cl.ID_Client = s.ID_Client
where s.ID_Tovar = 1
```

Лістинг 3.5 Правильна відповідь на питання «Вивести ім'я виробника та суму, на яку він продав товарів»:

```
select m.NameMaker, sum(t.Price * s.Quantity) 'Sum'
from Maker m
      inner join Tovar t on m.ID_Maker = t.ID_Maker
      inner join Sale s on s.ID_Tovar = t.ID_Tovar
group by m.NameMaker
```

3.2.4 Створення тесту

Для створення тесту за розкладом потрібно мати тест, який можна перевикористовувати для проходження різними студентами/групами у різний час. Для демонстрації проходження тесту, він повинен складатися з усіх питань, створених у попередньому пункті.

Для кожного питання тесту також необхідно проставити і максимально можливий бал, який може отримати Студент при наданні відповіді на запитання. В один бал будуть оцінюватися питання: «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті» та «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)». В два бали будуть оцінюватись питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку». В три бали будуть оцінюватися питання «Вивести ім'я клієнта, ім'я виробника, назву товару,

дату купівлі/продажу та суму, яку сплатив клієнт за покупку» та «Вивести ім'я виробника та суму, на яку він продав товарів». Тож в сумі за цей тест Студент може отримати максимум 10 балів.

3.2.5 Створення групи

Для створення тесту за розкладом необхідно присвоїти тест хоча б одному Студентові, але окрім проходження тесту потрібно також продемонструвати і верифікацію Студента за зображенням витягнутим з відеопотоку з вебкамери. Для цього буде краще згрупувати Студентів. Тому необхідно створити групу ITINF-19-1 (рис. 3.9).

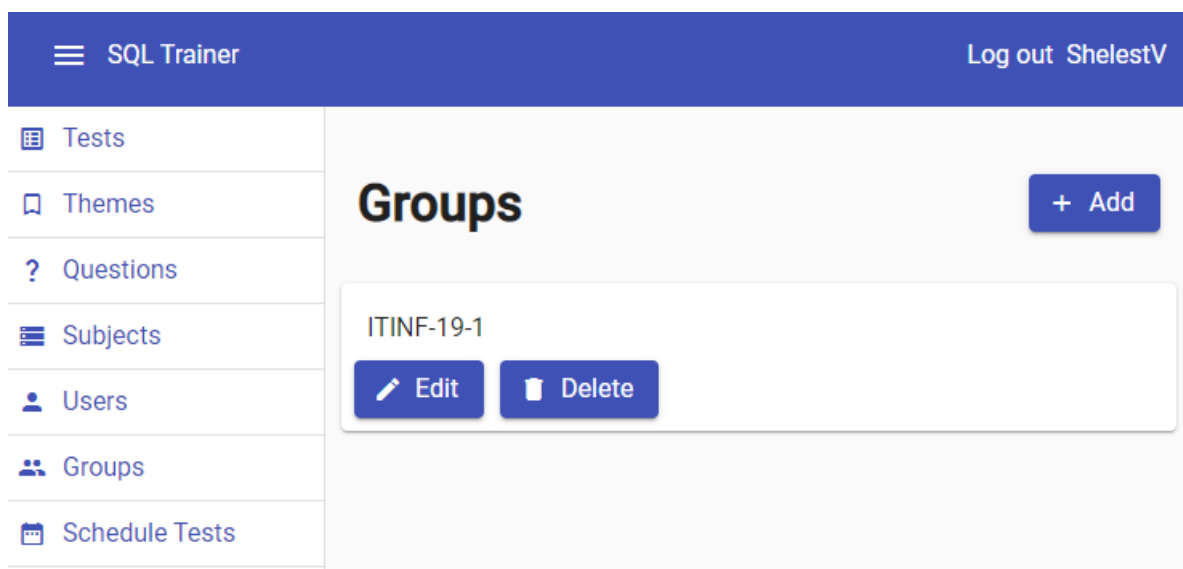


Рисунок 3.9 – Створена група ITINF-19-1

3.2.6 Створення Студентів

Необхідно створити двох студентів: один із зображенням, на якому буде обличчя, яке можна буде верифікувати, та зображення з іншим обличчям, де верифікація буде провалюватися.

Перший буде Volodymyr (рис. 3.10) із логіном volodymyr.shelest@nure.ua. Цей користувач буде мати зображення з обличчям, яке повинно буде пройти верифікацію успішно.

Edit User

Name*
Volodymyr

Login*
volodymyr.shelest@nure.ua

Choose Role*
Student

Choose Group*
ITINF-19-1

Select File to Upload




Рисунок 3.10 – Користувач Volodymyr

Другий – Vadym (рис. 3.11) із логіном vadym.naumenko@nure.ua. Цей користувач буде мати зображення з обличчям, яке не повинно пройти верифікацію.

Edit User

Name*
Vadym

Login*
vadym.naumenko@nure.ua

Choose Role*
Student

Choose Group*
ITINF-19-1

Select File to Upload

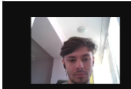


Рисунок 3.11 – Користувач Vadym

3.2.7 Створення тесту за розкладом

Створивши тему, питання, базу даних, тест, групу та студентів, можна перейти до створення найголовнішого елемента для демонстрування роботи верифікації Студента та перевірки роботи функції автоматичного проставлення оцінки – створення тесту за розкладом (рис. 3.12).

Тест за розкладом буде відноситись до того, який був створений у підпункті 3.2.4. Цей тест повинен бути присвоєний до групи створеної у підпункті 3.2.5.



Add Schedule Test

Choose Test*
Мова DQL

17/05/2023 23:30

21/05/2023 23:59

Choose groups
ITINF-19-1

Choose users*
Volodymyr, Vadym

Add Cancel

Рисунок 3.12 – Характеристики тесту за розкладом

3.3 Перевірка модулю ідентифікації Студента

Для перевірки коректності роботи модуля верифікації потрібно перевірити, як програмний застосунок поведе себе при різних умовах: повернутому від камери обличчю та при різних рівнях освітлення. Тому верифікацію особи було перевірено при трьох рівнях освітлення: із увімкненим верхнім світлом, із вимкненим верхнім світлом, де джерелами світла є лише монітор ноутбуку та другий монітор, який під'єднаний до

ноутбуку, та із вимкненим верхнім світлом, де єдиним джерелом світла є монітор ноутбуку на мінімальних налаштуваннях яскравості. Усі тести було проведено для двох користувачів із зображеннями обличчя різних людей (рис. 3.13).

При проходженні верифікації за зображенням показаним на рисунку 3.13 (а) результати були правильними.

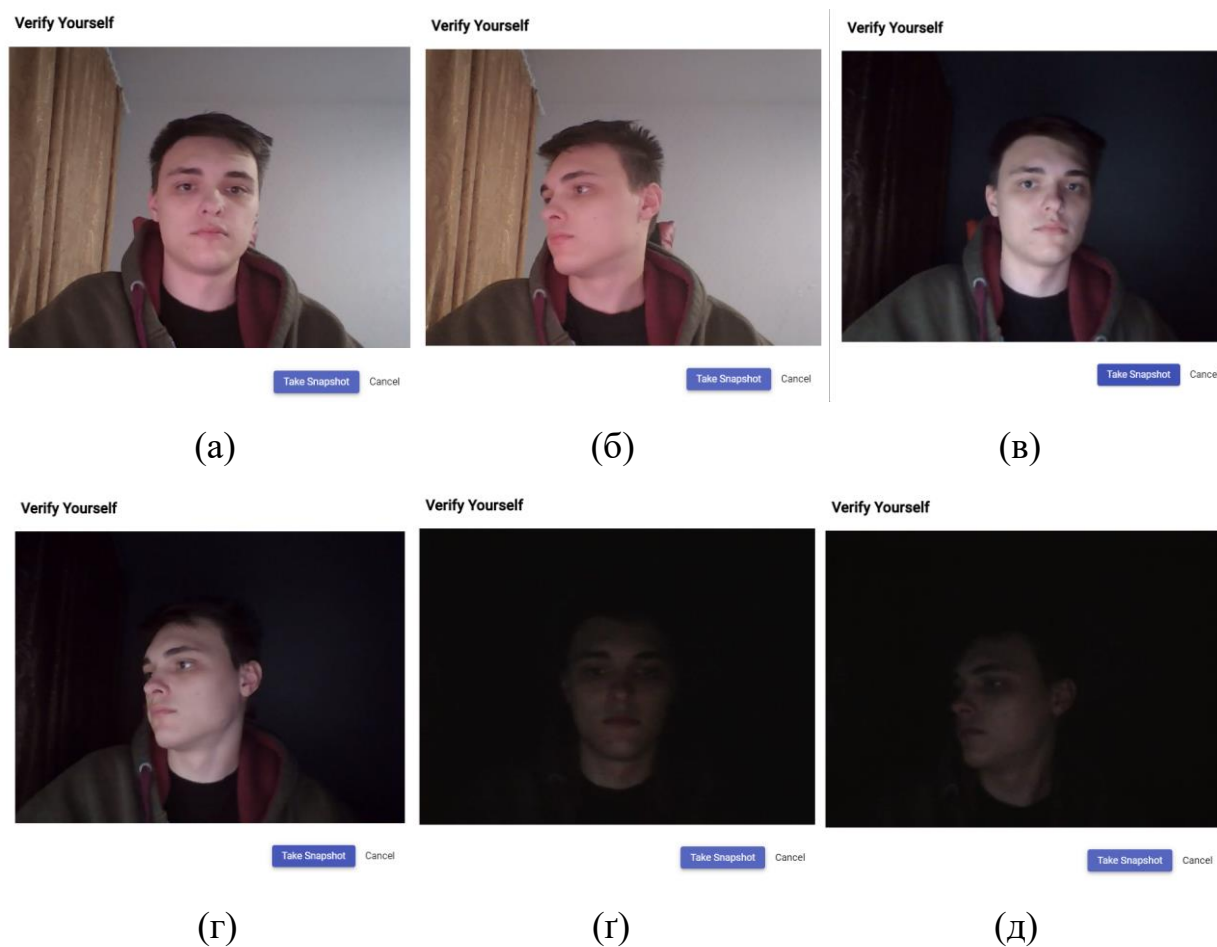


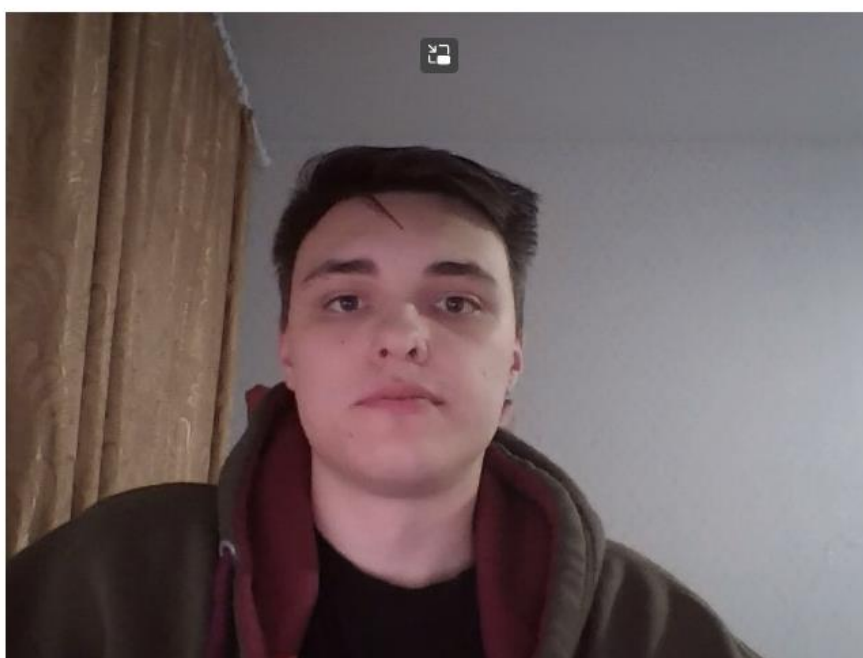
Рисунок 3.13 – Зображення, які відправлялись на верифікацію:
 (а) верхнє освітлення без повороту голови; (б) верхнє освітлення з поворотом голови; (в) освітлення від двох моніторів на максимальній яскравості без повороту голови; (г) освітлення від двох моніторів на максимальній яскравості з поворотом голови; (Г) освітлення тільки від монітору ноутбуку на мінімальних налаштуваннях яскравості без повороту голови; (д) освітлення тільки від монітору ноутбуку на мінімальних налаштуваннях яскравості з поворотом голови

Для Volodymyr верифікація проходить успішно і виконується передання Студента на сторінку складання тесту. Для Vadym верифікація не проходить, через що Студентові відображається наступне повідомлення (рис. 3.14).

При проходженні верифікації за зображеннями показаними на рисунку 3.13 результати незміні. Для Volodymyr верифікація успішна, для Vadym – ні. Це свідчить про те, що інша людина не зможе отримати доступ до проходження тесту замість Студенту.

Під час проходження тесту кожні 10 секунд відбувається верифікація обличчя студента, якщо правила верифікації порушено тричі, відповіді студента зберігаються та тест помічається як провалений за кожне питання оцінка буде 0.

Verify Yourself



You didn't pass verification please try again

Take Snapshot

Cancel

Рисунок 3.14 – Повідомлення після неуспішної верифікації, коли користувач зайшов не під своїм логіном

3.4 Перевірка роботи модулю автоматичного визначення оцінки студента за конкретне питання тесту

В тесті наявно 5 запитань. Відповідати на кожне з них правильно не має жодного сенсу, тому потрібно навмисно допустити помилки в деяких із питань, щоб перевірити, яку оцінку поставить програма у різних ситуаціях.

У разі, якщо скрипт, який Студент хоче перевірити не може бути виконаним через наявність будь-якого роду помилок, користувачеві виведеться повідомлення саме з системи управління базами даних про те, що саме не так в скрипті, який Студент намагається виконати (рис. 3.15).



Рисунок 3.15 – Повідомлення про помилку у скрипті

Перше питання «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті». В цьому питанні просять вивести лише назву міста, ім'я клієнта та ім'я виробника. Відповідь має неправильну кількість атрибутів, яку поверне запит Студента (рис. 3.16).

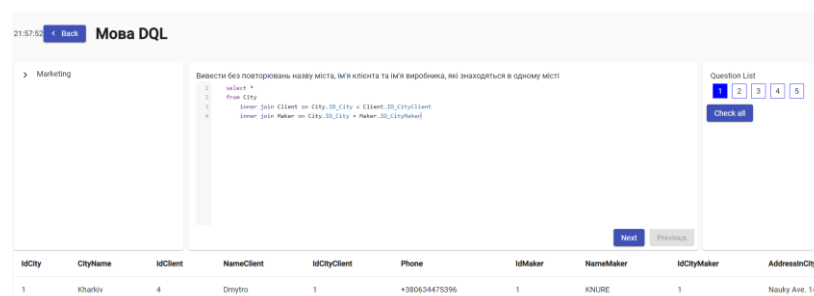


Рисунок 3.16 – Відповідь на питання «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті»

Друге питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку». В цьому питанні просять вивести суму, яку сплатив клієнт на покупку, маючи на увазі перемножити ціну товару (Price) та кількість одиниць товару (Quantity). Типова помилка для такого питання є виведення замість справжньої суми покупки суми цін товару (Price) без урахування кількості (Quantity). Саме тому для перевірки правильності виставлення оцінки була зроблена саме ця помилка (рис. 3.17).

NameClient	NameMaker	NameTovar	Date	Sum
Dmytro	KNJURE	Headphones	Mon 15 05 2023 6:24:42 AM	100
Dmytro	Mjykola	Ball	Tue 09 05 2023 6:23:36 AM	200
Iryna	KNJURE	Headphones	Wed 17 05 2023 6:25:11 AM	100
Olena	KNJURE	Headphones	Mon 24 04 2023 9:35:00 AM	100

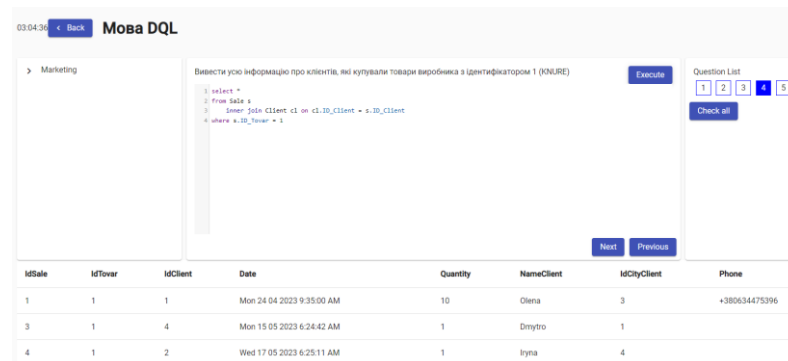
Рисунок 3.17 – Відповідь на питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку»

Третє питання «Вивести кількість товарів (будь-яких) які купував клієнт за увесь час». Найпоширенішою помилкою даного питання є відсутність групування по клієнтах, тобто виведення загальної кількості записів таблиці-результату. Тому саме таку помилку буде мати відповідь (рис. 3.18) на дане питання.

Count
4

Рисунок 3.18 – Відповідь на питання «Вивести кількість товарів (будь-яких) які купував клієнт за увесь час»

Четверте питання «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)». Найчастіше в таких запитах виводять інформацію з усіх таблиць, які поєднувались, а не лише з таблиці Client. Саме така відповідь була надана на це питання (рис. 3.19).

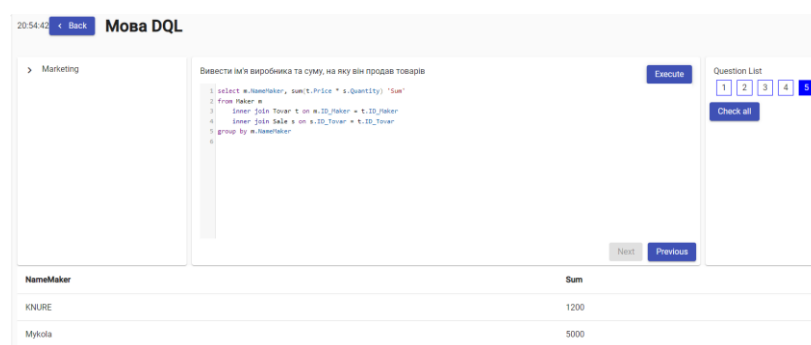


The screenshot shows a SQL query execution interface. The query is: `1 select *
2 from Sale s
3 inner join Client cl on s.ID_Client = s.ID_Client
4 where s.ID_Maker = 1`. The results table has columns: IdSale, IdTovar, IdClient, Date, Quantity, NameClient, IdCityClient, and Phone. The results are:

IdSale	IdTovar	IdClient	Date	Quantity	NameClient	IdCityClient	Phone
1	1	1	Mon 24 04 2023 9:35:00 AM	10	Olena	3	+380634475396
3	1	4	Mon 15 05 2023 6:24:42 AM	1	Dmytro	1	
4	1	2	Wed 17 05 2023 6:25:11 AM	1	Iryna	4	

Рисунок 3.19 – Відповідь на питання «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)»

П'яте питання «Вивести ім'я виробника та суму, на яку він продав товарів». На дане питання була надана абсолютно правильна відповідь. Це дозволяє перевірити, як програма себе буде вести із правильними відповідями (рис. 3.20).



The screenshot shows a SQL query execution interface. The query is: `1 select m.NameMaker, sum(t.Price * s.Quantity) 'Sum'
2 from Maker m
3 inner join Tovar t on m.ID_Maker = t.ID_Maker
4 inner join Sale s on s.ID_Tovar = t.ID_Tovar
5 group by m.NameMaker`. The results table has columns: NameMaker and Sum. The results are:

NameMaker	Sum
KNURE	1200
Mykola	5000

Рисунок 3.20 – Відповідь на питання «Вивести ім'я виробника та суму, на яку він продав товарів»

Після збереження результатів (збереження відбувається після того, як натиснути кнопку Submit, яка з'являється замість кнопки Check All після

перевірки усіх запитів на те, що вони виконуються без викидання помилок) буде доступне меню перегляду результатів тесту, де буде відобразитись оцінка, яку виставила програма. Після чого Студент може перейти в меню перегляду результатів і побачити оцінки, які проставила програма за кожне питання.

Перше питання «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті». Програма автоматично проставила 0,25 бали із 1 можливого (рис. 3.21) через те, що замість 3 атрибутів, які просять вивести, Студент виводить значення по усім атрибутам (11 атрибутів). Це не вважається грубою помилкою, але це сповільнює запит і на деяких проєктах це може бути місцем, яке треба оптимізувати. Але Вчитель даний запит оцінив би у 0,9 балів.



Рисунок 3.21 – Оцінка за неправильну відповідь на питання «Вивести без повторювань назву міста, ім'я клієнта та ім'я виробника, які знаходяться в одному місті»

Друге питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку». Дана відповідь має грубу помилку, скоєну через неухважність Студента, яка на проєкті може призвести до великої кількості проблем. Рахування суми цін товарів, замість суми, на яку було продано товарів є критичною помилкою. Програма оцінила даний запит у 2,6 бали (рис. 3.22) через те, що дані майже повністю збігаються. Вчитель оцінив би такий скрипт у 1 чи навіть 0 балів.

Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку

User answer

Execute

```

1 select c1.NameClient, m.NameMaker, t.NameTovar, s.[Date], sum(t.Price) 'Sum'
2 from Sale s
3 inner join Client c1 on c1.ID_Client = s.ID_Client
4 inner join Tovar t on t.ID_Tovar = s.ID_Tovar
5 inner join Maker m on m.ID_Maker = t.ID_Maker
6 group by c1.NameClient, m.NameMaker, t.NameTovar, s.[Date]

```

NameClient	NameMaker	NameTovar	Date	Sum
Dmytro	KNURE	Headphones	Mon 15 05 2023 6:24:42 AM	100
Dmytro	Mykola	Ball	Tue 09 05 2023 6:23:36 AM	200
Iryna	KNURE	Headphones	Wed 17 05 2023 6:25:11 AM	100
Olena	KNURE	Headphones	Mon 24 04 2023 9:35:00 AM	100

Max question mark: 3
Automated mark: 2.6

Рисунок 3.22 – Оцінка за неправильну відповідь на питання «Вивести ім'я клієнта, ім'я виробника, назву товару, дату купівлі/продажу та суму, яку сплатив клієнт за покупку»

Третє питання «Вивести кількість товарів (будь-яких) які купував клієнт за увесь час». Дана відповідь абсолютно неправильна, тому що необхідно було вивести кількість товарів, які купував кожен клієнт, в той час як Студент виводить кількість куплених товарів для усіх клієнтів. Ця відповідь абсолютна неправильна і програма правильно оцінила цю відповідь у 0 балів (рис. 3.23).

Вивести кількість товарів (будь-яких) які купував клієнт за увесь час

User answer

Execute

```

1 select count(*) 'Count'
2 from Sale

```

Count

4

Max question mark: 2
Automated mark: 0

Рисунок 3.23 – Оцінка за неправильну відповідь на питання «Вивести кількість товарів (будь-яких) які купував клієнт за увесь час»

Четверте питання «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)». В даному питанні Студент повторив ту саму помилку, що і у першому питанні, вивів збиткову кількість атрибутів. Так само це не вважається грубою помилкою, але так само даний скрипт не може бути оціненим у максимальний бал. Тому оцінка 0,47 виставлена програмою (рис. 3.24) вважається заниженою за дану помилку і Вчитель міг би сміливо поставити 0,9 балів за відповідь на дане питання.

Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)

User answer

Execute

```

1 select *
2 from Sale s
3 inner join client c1 on c1.Id_Client = s.Id_Client
4 where s.Id_Tovar = 1

```

IdSale	IdTovar	IdClient	Date	Quantity	IdClient1	NameClient	IdCityClient	Phone
1	1	1	Mon 24 04 2023 9:35:00 AM	10	1	Olena	3	+380634475396
3	1	4	Mon 15 05 2023 6:24:42 AM	1	4	Dmytro	1	
4	1	2	Wed 17 05 2023 6:25:11 AM	1	2	Iryna	4	

Max question mark: 1
Automated mark: 0.47

Рисунок 3.24 – Оцінка за неправильну відповідь на питання «Вивести усю інформацію про клієнтів, які купували товари виробника з ідентифікатором 1 (KNURE)»

П'яте питання «Вивести ім'я виробника та суму, на яку він продав товарів». Відповідь на це питання була надана правильна і програма оцінила цю відповідь у максимальну можливу оцінку. Вчитель так само мусив би поставити максимальний бал за цю відповідь, а саме 3 бали (рис. 3.25). Тож в даному випадку, як і із третім питанням, програма правильно оцінила відповідь.

Спираючись на результати оцінювання, які наведені у таблиці 3.1, програмний застосунок з абсолютною точністю виставляє коректну оцінку за запит написаний правильно. Також оцінка виглядає правильною за умови, що дані отриманні із скрипту дуже сильно відрізняються від того, що повинно бути.

Вивести ім'я виробника та суму, на яку він продав товарів

User answer

Execute

```

1 select m.NameMaker, sum(t.Price * s.Quantity) 'Sum'
2 from Maker m
3     inner join Tovar t on m.ID_Maker = t.ID_Maker
4     inner join Sale s on s.ID_Tovar = t.ID_Tovar
5 group by m.NameMaker

```

NameMaker	Sum
KNURE	1200
Mykola	5000

Max question mark: 3
Automated mark: 3

Рисунок 3.25 – Оцінка за правильну відповідь на питання «Вивести ім'я виробника та суму, на яку він продав товарів»

Також даний алгоритм погано себе продемонстрував на перевірці відповідей з помилкою пов'язаною із розумінням питання чи з некритичними помилками, як неправильна кількість атрибутів.

Таблиця 3.1 – Результати відповідей на питання тесту

Номер питання	Програмна оцінка	Оцінка Вчителя	Максимальна оцінка
1	0,25	0,9	1
2	2,6	0	3
3	0	0	2
4	0,47	0,9	1
5	3	3	3
Сума	6,32	4,8	10

ВИСНОВКИ

У рамках даної кваліфікаційної роботи був розроблений клієнт-серверний програмний застосунок для моніторингу знань з мови SQL з можливістю розпізнавання обличчя студента. Для досягнення цієї мети були розглянуті різні методи розпізнавання обличчя та за результатами цього аналізу було обрано метод розпізнавання обличчя за допомогою нейронних мереж і серед безлічі різних нейронних мереж, які вирішують задачу розпізнавання обличчя, була обрана нейронна мережа, яка використовує бібліотека DeepFace. Такий вибір обумовлений високою точністю розпізнавання при різних умовах (погане освітлення, повороти голови, тощо) та відносно швидкого процесу розпізнавання.

Виконано всі поставлені задачі, а саме:

- розглянуто сучасний стан питання ідентифікації моніторингу знань студентів, були надані сервіси, які допомагають організувати учбовий процес, та сервіси, які надають задачі, які треба вирішити за допомогою тієї чи іншої мови програмування;

- розглянуто підходи щодо ідентифікації особи, були наведені переваги та недоліки кожного із підходів, та був обраний, як найбільш якісний підхід із використанням нейронних мереж;

- розглянуто сучасні підходи до проектування вебсервісів та обрано клієнт-серверну архітектуру через ряд своїх переваг;

- виконано усі етапи розробки програмного застосунку для моніторингу знань з мови SQL з можливістю розпізнавання обличчя студента;

- проведено тестування розробленого застосунку, за результатами якого були визначені переваги і недоліки даного програмного застосунку.

Даний програмний застосунок потрібно використовувати у навчальному процесі при викладанні дисциплін пов'язаних із мовою SQL. Алгоритм, який автоматично розраховує оцінку Студента за відповідь на

питання, допомагає скоріше допомагаю одразу знайти правильні відповіді і зосередитися на оцінюванні неправильних. Через те, що усі відповіді Студентів та оцінки Вчителів зберігаються у базі даних застосунку, у майбутньому планується натренувати нейронну мережу, яка буде розраховувати правильність відповіді.

Враховуючи, що студенти можуть спробувати обійти систему верифікації, у подальшому було б доцільно провести дослідження в області антиспуфінгу, наприклад, додати перевірку на рухливість обличчя.

Результати роботи апробовано у вигляді 2 тез доповідей під час 27-го Міжнародного молодіжного форуму «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ В XXI СТОЛІТТІ» [35] та 14-ої Міжнародної науково-практичної конференції «FREE AND OPEN SOURCE SOFTWARE» [16].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. EdEra – студія онлайн освіти. URL: <https://www.ed-era.com/> (дата звернення 13.04.2023).
2. Prometheus – Найбільша платформа онлайн-курсів в Україні. URL: <https://prometheus.org.ua/> (дата звернення 13.04.2023).
3. Coursera | Degrees, Certificates, & Free Online Courses. URL: <https://www.coursera.org/> (дата звернення 13.04.2023).
4. Kahoot! | Learning games | Make learning awesome! URL: <https://kahoot.com/> (дата звернення 13.04.2023).
5. LearningApps.org – interactive und multimediale Lernbausteine. URL: <https://learningapps.org/> (дата звернення 13.04.2023).
6. Codewars – Achieve mastery through coding practice and developer mentorship. URL: <https://www.codewars.com/> (дата звернення 13.04.2023).
7. LeetCode – The World’s Leading Online Programming Learning Platform. URL: <https://leetcode.com/> (дата звернення 13.04.2023).
8. O. Yakovleva, & K. Nikolaieva (2020). Research of descriptor based image normalization and comparative analysis of SURF, SIFT, BRISK, ORB, KAZE, AKAZE descriptors. *Advanced Information Systems*, 4(4), 89-101.
9. V. Gorokhovatskyi, & I. Tvoroshenko (2020). Image classification based on the Kohonen network and the data space modification.
10. Y. Daradkeh, V. Gorokhovatskyi, I. Tvoroshenko, S. Gadetska, M. Al-Dhaifallah, (2021). Methods of classification of images on the basis of the values of statistical distributions for the composition of structural description components, *IEEE Access* 9 92964–92973.
11. H. Li, Z. Lin, X. Shen, J Brandt, G. Hua (2015). A convolutional neural network Cascade for face detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5325–5334.

12. K. Zhang, Z. Zhang, Z. Li, Y. Qiao (2016). Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Signal Processing Letters* 23(10) 1499–1503.
13. Face recognition using Tensorflow. Davidsandberg/facenet. GitHub. URL: <https://github.com/davidsandberg/facenet> (дата звернення 17.04.2023).
14. ICT DeepFake Detection, CVPR2022. LightDXY/ICT_DeepFake. GitHub. URL: https://github.com/LightDXY/ICT_DeepFake (дата звернення 17.04.2023).
15. A Lightweight Face Recognition and Facial Attribute Analysis (Age, Gender, Emotion and Race) Library for Python. serengil/deepface. GutGub. URL: <https://github.com/serengil/deepface> (дата звернення 17.04.2023).
16. Шелест В.А., Яковлева О.В. (2023). ВИКОРИСТАННЯ ПРОГРАМНОГО ДОДАТКУ NOTION ДЛЯ ОРГАНІЗАЦІЇ КОМАНДНОЇ РОБОТИ НАД ПРОЄКТОМ. Матеріали XIV-ої Міжнародної науковопрактичної конференції «Free and Open Source Software», Харків, 14-16 лютого 2023 р.–Харків: Харківський національний економічний університет імені Семена Кузнеця, с. 110.
17. Trello – Manage Your Team’s Projects From Anywhere | Trello. URL: <https://trello.com/> (дата звернення 15.04.2023).
18. monday.com – monday.com | A new way of working. URL: <https://monday.com/> (дата звернення 15.04.2023).
19. Asana – Manage your team’s work, projects & tasks online. URL: <https://asana.com/> (дата звернення 15.04.2023).
20. Notion – Your wiki, docs & projects. Together. URL: <https://www.notion.so/> (дата звернення 15.04.2023).
21. Ткаченко, О., & Бойко, М. (2021). Деякі аспекти розпізнавання облич: моделі, алгоритми, методи, системи, застосування. *Цифрова платформа: інформаційні технології в соціокультурній сфері*, 4(1), 79-95.

22. Daradkeh, Y. I., Gorokhovatskyi, V., Tvoroshenko, I., & Zeghid, M. (2022). Tools for Fast Metric Data Search in Structural Methods for Image Classification. *IEEE Access*, *10*, 124738-124746.
23. Gorokhovatskyi, V., Tvoroshenko, I., Kobylin, O., & Vlasenko, N. (2023). Search for Visual Objects by Request in the Form of a Cluster Representation for the Structural Image Description. *Advances in Electrical and Electronic Engineering*, *21*(1), 19-27.
24. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. *Telecommunications and Radio Engineering*, *79*(10).
25. Yakovleva, O., Kovtunencko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System (COLINS-2023). In *CEUR Workshop Proceedings (Vol. 3403)*. pp. 69-86.
26. Ковтуненко А.Р., Яковлева О.В., Любченко В.А., & Янголенко О.В. (2020). Дослідження сумісного використання математичної морфології та згорткових нейронних мереж для вирішення задачі розпізнавання цінників.
27. Svyrydov, A. S. (2018). Метод підбору алгоритмів для розпізнавання зображень. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *1*(47), 137-140.
28. Потьомкіна К.О. (2020). Згорткові нейронні мережі як основа машинного навчання.
29. Невмержицький Р.В., Рудюк Л.В. (2021). ЗГОРТКОВІ НЕЙРОННІ МЕРЕЖІ.
30. Мазур, М. В. (2019). *Задача виявлення і класифікації об'єктів за допомогою згорткових нейронних мереж* (Doctoral dissertation, ВНТУ).
31. Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification.

In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).

32. Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, 429, 215-244.

33. Fine Tuning The Threshold in Face Recognition, (2020). URL: <https://sefiks.com/2020/05/22/fine-tuning-the-threshold-in-face-recognition/> (дата звернення 18.04.2023).

34. Flask – Welcome to Flask – Flask Documentation. URL: <https://flask.palletsprojects.com/en/2.3.x/> (дата звернення 06.05.2023).

35. Шелест В.А., Яковлева О.В. (2023). РОЗРОБКА СИСТЕМИ ТЕСТУВАННЯ ЗНАНЬ МОВИ SQL З ВИКОРИСТАННЯМ ВЕБКАМЕРИ ДЛЯ ІДЕНТИФІКАЦІЇ ОСОБИ СТУДЕНТА. 27-ий міжнародний молодіжний форум «РАДІОЕЛЕКТРОНІКА І МОЛОДЬ У ХХІ СТОЛІТТІ», с. 34-35.