|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**  Дисципліна  **«**Хмарні технології**»**  **Індивідуальне завдання № 2**  «СТВОРЕННЯ ТА РОЗГОРТАННЯ ДОДАТКУ В ХМАРНОМУ СЕРЕДОВИЩІ GOOGLE» | | | |
| **Виконав:** | Гоша Давід | **Перевірив**: |  |
| Група | ІПЗ-43 | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

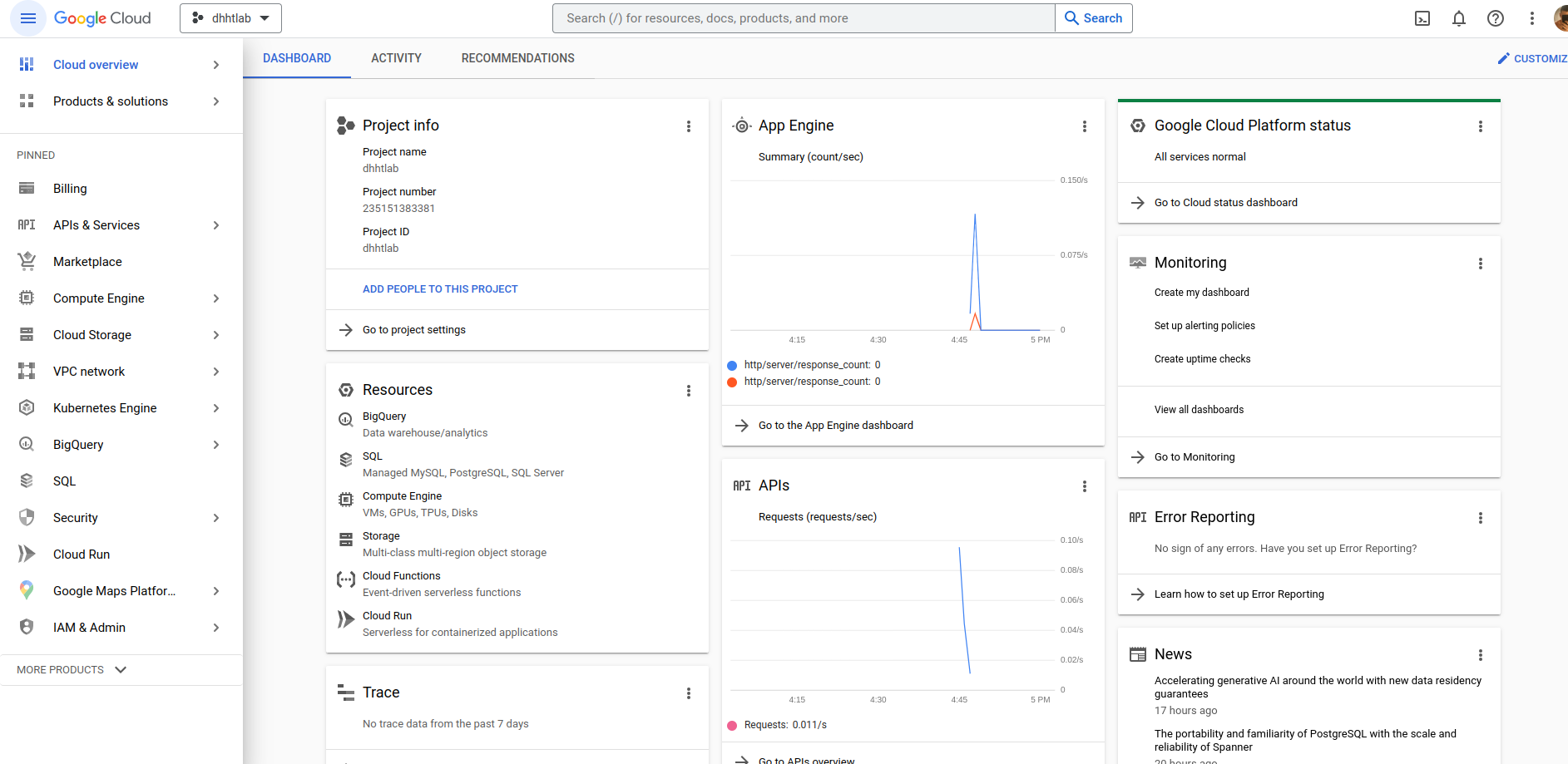
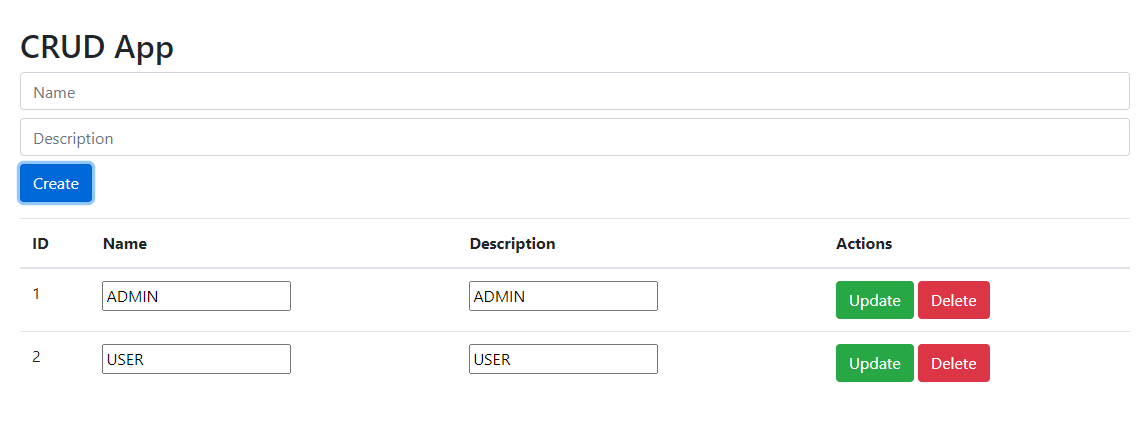
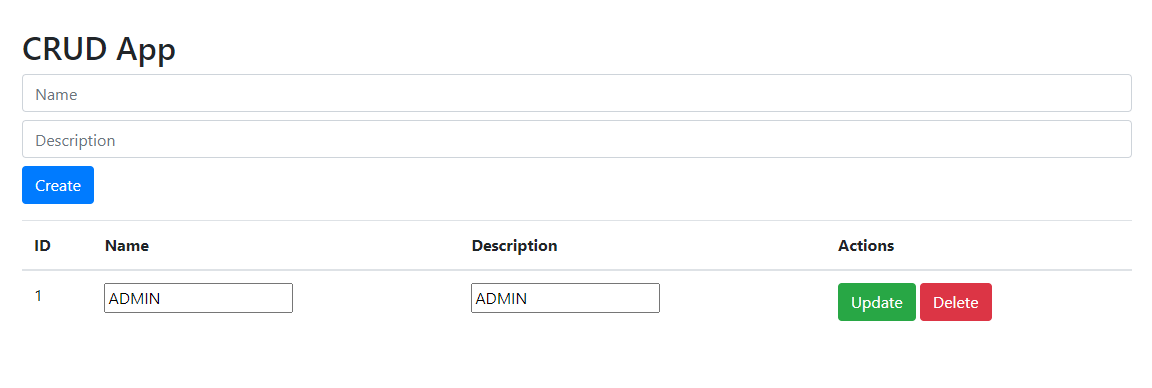
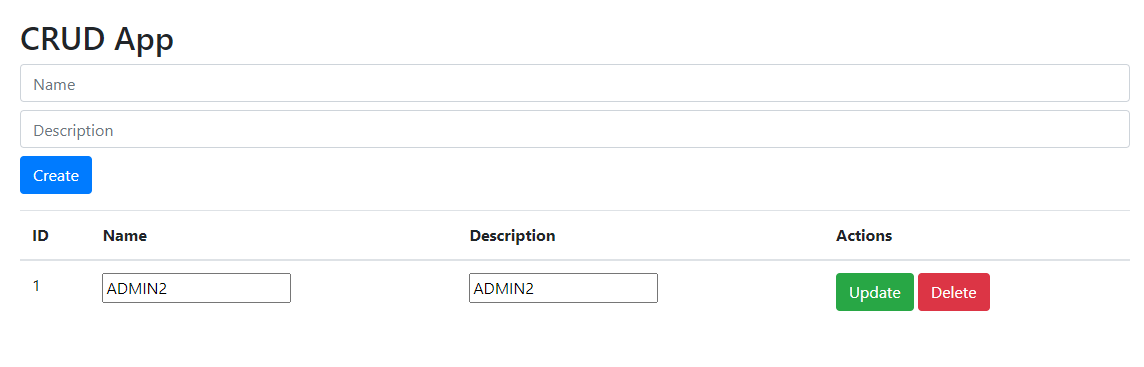
**Мета роботи:**

Дослідити створення та розгортання проекту засобами хмарного середовища GOOGLE.

**Завдання роботи:**

Створити програмний код проекту в хмарному середовищі, розгорнути проект в хмарі, використовуючи навички, набуті в ході виконання 1 та 2 лабораторної роботи.

**Опис виконання та оформлення результатів роботи:**

1. **Створення програмного коду:**
   1. Розроблено веб-додаток на Flask, який дозволяє виконувати CRUD операції з об'єктами.
   2. Код включає ендпойнти для створення, читання, оновлення та видалення даних.
2. **Розгортання проекту в хмарі Google:**
   1. Проект було розгорнуто в середовищі Google Cloud Platform (GCP).
   2. Використано Google App Engine для розгортання Flask-додатку.
   3. Налаштовано необхідні конфігурації для забезпечення доступу до додатку через Інтернет.
3. **Тестування функціоналу:**
   1. Проведено тестування всіх ендпойнтів через Postman для перевірки CRUD операцій.
   2. Перевірено відповіді сервера на коректність та повноту даних.
4. **Скріни результатів:**
   1. 
   2. 
   3. 
   4. 

**Код:**

from flask import Flask, request, jsonify, render\_template

app = Flask(\_\_name\_\_)

items = []

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

def create\_item():

    data = request.json

    item\_id = len(items) + 1

    name = data.get('name')

    description = data.get('description')

    item = {'id': item\_id, 'name': name, 'description': description}

    items.append(item)

    return jsonify(item)

@app.route('/items/<int:item\_id>', methods=['GET'])

def get\_item(item\_id):

    item = next((item for item in items if item['id'] == item\_id), None)

    if not item:

        return jsonify({'error': 'Item not found'}), 404

    return jsonify(item)

@app.route('/items/<int:item\_id>', methods=['PUT'])

def update\_item(item\_id):

    item = next((item for item in items if item['id'] == item\_id), None)

    if not item:

        return jsonify({'error': 'Item not found'}), 404

    data = request.json

    item['name'] = data['name']

    item['description'] = data['description']

    return jsonify(item)

@app.route('/items/<int:item\_id>', methods=['DELETE'])

def delete\_item(item\_id):

    global items

    item = next((item for item in items if item['id'] == item\_id), None)

    if not item:

        return jsonify({'error': 'Item not found'}), 404

    items = [item for item in items if item['id'] != item\_id]

    return '', 204

@app.route('/items/', methods=['GET'])

def get\_items():

    return jsonify(items)

@app.route('/')

def index():

    return render\_template('index.html')

if \_\_name\_\_ == '\_\_main\_\_':

    app.run()

**Файл конфігурації:**

runtime: python39

entrypoint: gunicorn -b :$PORT app:app

instance\_class: F2

automatic\_scaling:

  target\_cpu\_utilization: 0.65

  min\_instances: 1

  max\_instances: 10

resources:

  cpu: 1

  memory\_gb: 0.5

  disk\_size\_gb: 10

**Висновки:**

В ході виконання проекту ми дослідили основні аспекти створення та управління базами даних в середовищі Azure. Було виявлено, що Azure надає гнучкі та масштабовані можливості для роботи з базами даних, що є важливим для сучасних бізнес-вимог. Особливо важливим виявилося використання еластичних пулів, які дозволяють ефективно масштабувати ресурси відповідно до потреб користувачів та оптимізувати витрати.

Розроблена в рамках проекту програма, яка використовує веб-інтерфейс для взаємодії з базою даних, значно спрощує процес роботи з даними. Це робить програму доступною для широкого кола користувачів, навіть тих, хто не має технічних навичок прямого з'єднання з базою даних. Такий підхід підвищує ефективність роботи з даними та розширює можливості їх використання.

Крім того, в проекті було встановлено, що використання хмарних технологій, зокрема Azure, сприяє підвищенню гнучкості та масштабованості інформаційних систем. Це дозволяє компаніям швидко адаптуватися до змінних умов ринку та потреб користувачів, що є ключовим фактором успіху в сучасному динамічному бізнес-середовищі.

Загалом, результати проекту підтверджують значні переваги використання облачних рішень для управління базами даних, забезпечуючи високу продуктивність, масштабованість та гнучкість в управлінні даними.