

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:

- Розроблено веб-додаток на Flask, який дозволяє виконувати CRUD операції з об'єктами.
- Код включає ендпойнти для створення, читання, оновлення та видалення даних.

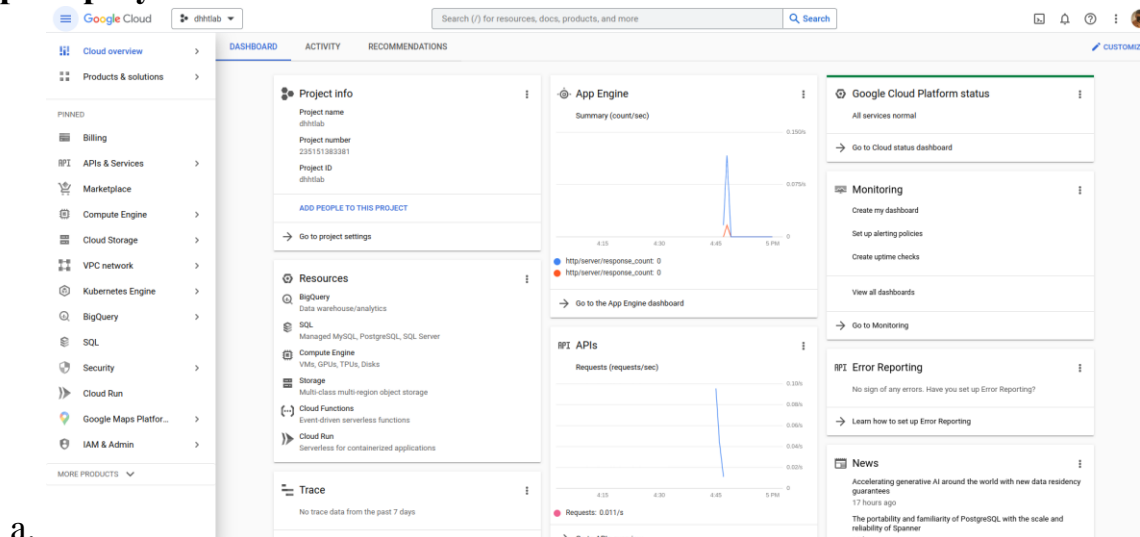
2. Розгортання проекту в хмарі Google:

- Проект було розгорнуто в середовищі Google Cloud Platform (GCP).
- Використано Google App Engine для розгортання Flask-додатку.
- Налаштовано необхідні конфігурації для забезпечення доступу до додатку через Інтернет.

3. Тестування функціоналу:

- Проведено тестування всіх ендпойнтів через Postman для перевірки CRUD операцій.
- Перевірено відповіді сервера на коректність та повноту даних.

4. Скріни результатів:



CRUD App

Create

ID	Name	Description	Actions
1	<input type="text" value="ADMIN"/>	<input type="text" value="ADMIN"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>
2	<input type="text" value="USER"/>	<input type="text" value="USER"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>

b.

CRUD App

Create

ID	Name	Description	Actions
1	<input type="text" value="ADMIN"/>	<input type="text" value="ADMIN"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>

c.

CRUD App

Create

ID	Name	Description	Actions
1	<input type="text" value="ADMIN2"/>	<input type="text" value="ADMIN2"/>	<input type="button" value="Update"/> <input type="button" value="Delete"/>

d.

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 виявилось 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.