Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра «Програмної інженерії»

Лабораторна робота № 5  
«РОЗРОБКА IoT КЛІЄНТА.»

Роботу виконав:

студент 3 курсу

факультету

«Комп’ютерні науки»

групи ПЗПІ-22-6

Іванов Д.С.

Перевірив:  
ст. викладач кат. ПІ  
Сокорчук І.П.

Харків 2025

| № | Дата | Версія звіту | Опис змін та виправлень |
| --- | --- | --- | --- |
| 1 | 25.08.2025 | 0.1 | Створено початковий варіант звіту |
| 2 | 26.08.2025 | 0.2 | Розширено опис процесу розгортання, додано згадку про CI/CD та Docker |
| 3 | 26.08.2025 | 1.0 | Фіналізація та підготовка до завантаження |

ЗАВДАННЯ

Метою лабораторної роботи є повноцінне розгортання програмної системи FoodCare Monitor System у хмарному середовищі, а також верифікація її функціональності.

Для цього необхідно:

1. Підготувати хмарну інфраструктуру для розміщення бекенду та бази даних.
2. Розгорнути серверний додаток (API) та базу даних у хмарному середовищі.
3. Налаштувати зв'язок між компонентами системи.
4. Виконати функціональне тестування розгорнутої системи та підготувати відеозапис демонстрації.
5. Оформити звітну документацію з детальним описом виконаної роботи.
6. Завантажити звіт у PDF-форматі та відео на YouTube.
7. Завантажити оновлений код та звіт у GitHub-репозиторій.

3. ОПИС ВИКОНАНОЇ РОБОТИ

3.1. Вибір та підготовка інфраструктури Для розгортання програмної системи FoodCare Monitor System була обрана хмарна платформа Microsoft Azure, що забезпечує надійність, масштабованість та широкий спектр інструментів для розробників. Усі компоненти були розміщені в єдиній Resource Group під назвою FoodCare-Deploy-RG для зручного управління та моніторингу.

3.2. Розгортання бази даних Для зберігання даних було розгорнуто гнучкий сервер Azure Database for PostgreSQL - Flexible Server. Цей сервіс надає повністю керовану базу даних, що дозволило уникнути ручного налаштування та адміністрування. Після створення сервера з іменем foodcare-postgres-server, було налаштовано мережеві правила для доступу ззовні та створено базу даних foodcare\_db. Дані для міграції були завантажені за допомогою інструменту Azure Data Studio, що підтримує підключення до PostgreSQL.

3.3. Розгортання серверної частини (API) Серверна частина, розроблена на ASP.NET Core, була розгорнута як Azure App Service. Цей сервіс дозволяє легко публікувати веб-додатки без необхідності налаштування віртуальних машин. Процес розгортання включав наступні кроки:

1. Клонування проєкту з GitHub-репозиторію.
2. Внесення змін до конфігураційного файлу appsettings.json, де було вказано рядок підключення до розгорнутої бази даних PostgreSQL в Azure.
3. Публікація проєкту безпосередньо з Visual Studio до Azure App Service.
4. Включення підтримки Docker та автоматизоване складання образу (див. Додаток В).

3.4. Верифікація функціоналу Після успішного розгортання було проведено комплексне тестування, що охоплювало взаємодію між усіма частинами системи: бекендом, веб-клієнтом та мобільним клієнтом. Тестування підтвердило, що API коректно обробляє запити від клієнтів, а дані, що надходять від симулятора IoT-сенсорів, відображаються у веб- та мобільному застосунках у реальному часі.

4. ВИСНОВКИ

Під час виконання лабораторної роботи було успішно здійснено повний цикл розгортання програмної системи FoodCare Monitor System у хмарному середовищі Microsoft Azure.

Основні результати:

* Освоєно процес підготовки хмарної інфраструктури для розміщення багатокомпонентної системи.
* Успішно розгорнуто серверний додаток (ASP.NET Core API) та базу даних (PostgreSQL) як PaaS-сервіси в Azure.
* Встановлено та верифіковано коректну взаємодію між усіма розгорнутими частинами системи.
* Підготовлено повний звіт та супровідну документацію, що детально описує процес розгортання.

ДОДАТОК А

Відеозапис та хронологічний опис

Хронологічний опис відеозапису:

* 0:00 - Вступ: короткий огляд цілей лабораторної роботи та архітектури розгортання.
* 0:30 - Демонстрація інтерфейсу адміністратора: управління користувачами та їх ролями.
* 1:15 - Створення нового складського приміщення через веб-клієнт.
* 2:00 - Створення та додавання продукту до складу.
* 2:45 - Демонстрація підключення та роботи IoT-клієнта: надходження даних від сенсорів.
* 4:00 - Верифікація синхронізації: перегляд показників сенсорів у реальному часі у веб-клієнті.
* 5:15 - Перехід до мобільного застосунку: авторизація та перегляд списку складів.
* 6:00 - Демонстрація динамічного оновлення показників сенсорів у мобільному застосунку.
* 7:00 - Тестування адміністративних функцій: створення резервної копії бази даних через API.
* 7:30 - Короткий підсумок та висновки.

ДОДАТОК Б

Програмний код

В.1 Dockerfile для контейнеризації ASP.NET Core API

Dockerfile

# Use a Microsoft .NET SDK image to build the application

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build

WORKDIR /app

# Copy the .csproj file and restore dependencies

COPY \*.csproj ./

RUN dotnet restore

# Copy the entire project and publish

COPY . ./

RUN dotnet publish -c Release -o out

# Use a minimal ASP.NET Core runtime image to run the application

FROM mcr.microsoft.com/dotnet/aspnet:8.0

WORKDIR /app

COPY --from=build /app/out .

ENTRYPOINT ["dotnet", "FoodCare.Api.dll"]