

Київський національний університет ім. Тараса Шевченка

Кафедра мережесих та інтернет технологій

Лабораторна робота № 4

Дисципліна: Хмарні технології

Тема: Створення класифікатора документів із використанням Azure AI

Виконав: Студент групи МІТ-31

Пугач Назар

Мета: Створити класифікатор документів із використанням Azure AI.

Хід роботи

Завдання 4.1: Створити Azure AI Language сервіс.

Завдання 4.2: Створити Azure Storage Account.

Завдання 4.3: Створити новий проект із застосуванням шаблону Azure Functions.

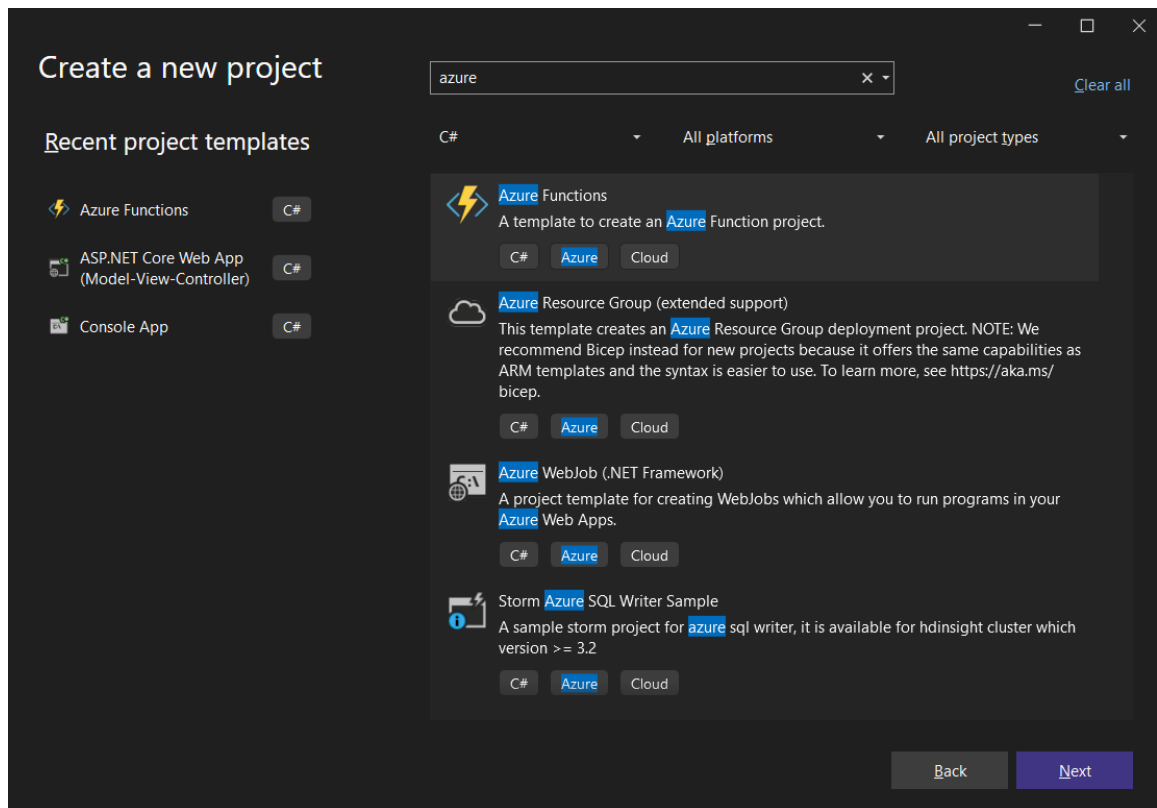


Рисунок 4.1 – Створення проекту.

Завдання 4.4: Встановити бібліотеки для роботи із хмарними сервісами.

1. Azure.AI.TextAnalytics
2. Azure.Storage.Blobs
3. Microsoft.Extensions.Azure

Завдання 4.5: Додати необхідні ключі та значення у файл local.settings.json.

```
{
  "IsEncrypted": false,
  "Values": {
    "AzureWebJobsStorage": "UseDevelopmentStorage=true",
    "FUNCTIONS_WORKER_RUNTIME": "dotnet-isolated",
    "sourceContainerName": "source",
    "targetContainerName": "destination",
    "blobConn": "conn-string",
    "textAnalyticsEndpoint": "endpoint",
    "textAnalyticsLocation": "location",
    "textAnalyticsKey": "key"
  }
}
```

Рисунок 4.2 – Значення у файл local.settings.json.

Завдання 4.6: додати підтримку клієнта Azure та сервісу зберігання BLOB-об'єктів у наш додаток, оновивши файл Program.cs.

```
using Azure;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Extensions.Azure;
using Microsoft.Extensions.DependencyInjection;
using Microsoft.Extensions.Hosting;

var host = new HostBuilder()
    .ConfigureFunctionsWebApplication()
    .ConfigureServices(services =>
    {
        services.AddApplicationInsightsTelemetryWorkerService();
        services.ConfigureFunctionsApplicationInsights();
        services.AddAzureClients(b =>
        {
            b.AddBlobServiceClient(Environment.GetEnvironmentVariable("blobConn"));
            var endpoint = new
            Uri(Environment.GetEnvironmentVariable("textAnalyticsEndpoint"));
            var credential = new
            AzureKeyCredential(Environment.GetEnvironmentVariable("textAnalyticsKey"));
            b.AddTextAnalyticsClient(endpoint, credential);
        });
    })
    .Build();

host.Run();
```

Рисунок 4.3 – Оновлений файл Program.cs.

Завдання 4.7: імплементувати застосування Azure AI Language у Azure Function.

```
using System.Text;
using Azure.AI.TextAnalytics;
using Azure.Storage.Blobs;
using Microsoft.Azure.Functions.Worker;
using Microsoft.Extensions.Logging;

namespace lab4
{
    public class Function1
    {
        private readonly ILogger<Function1> _logger;
        private readonly TextAnalyticsClient _textAnalyticsClient;
        private readonly BlobServiceClient _blobServiceClient;
        private readonly BlobContainerClient _blobSourceContainerClient;
        private readonly BlobContainerClient _blobDestinationContainerClient;

        public Function1(ILogger<Function1> logger, TextAnalyticsClient
        textAnalyticsClient, BlobServiceClient blobServiceClient)
        {
            _logger = logger;
            _textAnalyticsClient = textAnalyticsClient;
            _blobServiceClient = blobServiceClient;

            _blobSourceContainerClient =
            _blobServiceClient.GetBlobContainerClient(Environment.GetEnvironmentVariable("so
            urceContainerName"));
            _blobDestinationContainerClient =
            _blobServiceClient.GetBlobContainerClient(Environment.GetEnvironmentVariable("ta
            rgetContainerName"));
        }
    }
}
```

```

[Function(nameof(Function1))]
public async Task Run([BlobTrigger("source/{name}", Source =
BlobTriggerSource.LogsAndContainerScan, Connection = "blobConn")] Stream stream,
string name)
{
    using var blobStreamReader = new StreamReader(stream);
    var content = await blobStreamReader.ReadToEndAsync();
    _logger.LogInformation($"C# Blob Trigger processed blob\n Name:
{name} \n Data: {content}");

    var detectedLanguage = await
_textAnalyticsClient.DetectLanguageAsync(content);
    var languageName = detectedLanguage.Value.Name;
    _logger.LogInformation($"Detected language: {languageName}");

    string targetBlobName = $"{languageName}/{name}";
    BlobClient blobClient =
_blobDestinationContainerClient.GetBlobClient(targetBlobName);
    byte[] byteArray = Encoding.UTF8.GetBytes(content);

    await blobClient.UploadAsync(new MemoryStream(byteArray));
    _logger.LogInformation($"Uploaded blob - {targetBlobName} to
{Environment.GetEnvironmentVariable("targetContainerName")} container.");

    await _blobSourceContainerClient.DeleteBlobIfExistsAsync(name);
    _logger.LogInformation($"Deleted blob - {name} from
{Environment.GetEnvironmentVariable("sourceContainerName")} container.");
}
}
}

```

Рисунок 4.4 – Оновлена Azure Function.

Завдання 4.8: Перевірити працездатність.

+ Add filter








Name	Modified
<input type="checkbox"/>  Arabic	
<input type="checkbox"/>  Chinese_Traditional	
<input type="checkbox"/>  English	
<input type="checkbox"/>  French	
<input type="checkbox"/>  Japanese	
<input type="checkbox"/>  Korean	
<input type="checkbox"/>  Ukrainian	

Рисунок 4.5 – Класифікатор визначив всі мови.

Висновок: Під час виконання лабораторної роботи я створив класифікатор документів із використанням Azure AI.