

Київський національний університет ім. Тараса Шевченка

Кафедра мережевих та інтернет технологій

Лабораторна робота № 5

Дисципліна: Хмарні технології

Тема: Створення класифікатора документів із застосуванням Azure AI

Виконав: Студент групи МІТ-31

Полюхович Андрій

Хід роботи

- Створити консольний застосунок за прикладом.
 - Для початку роботи було створено новий Azure AI Language сервіс з необхідними налаштуваннями, після чого отримано його ключ доступу та ендпоінт.

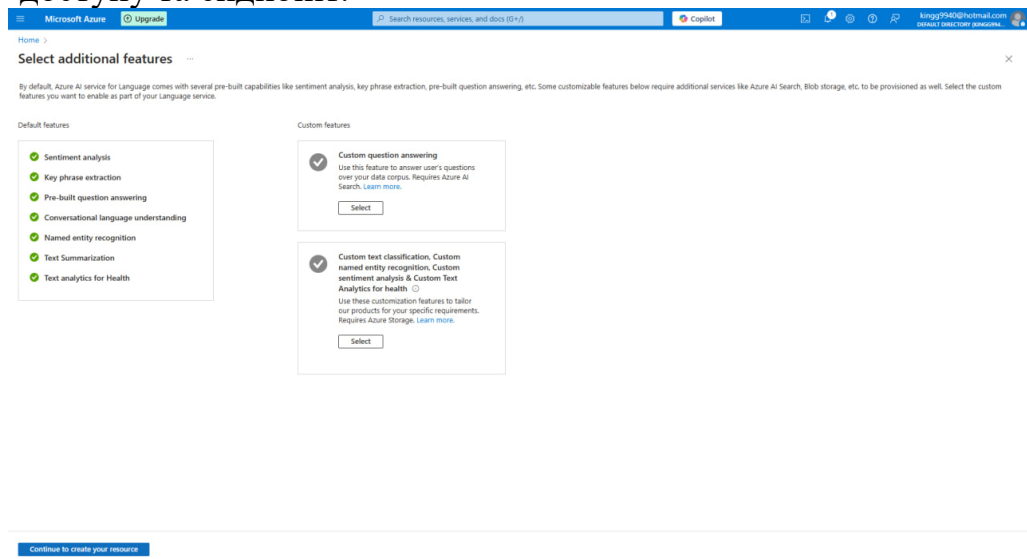


Рисунок 5.1- Перший етап створення Azure AI Language сервісу

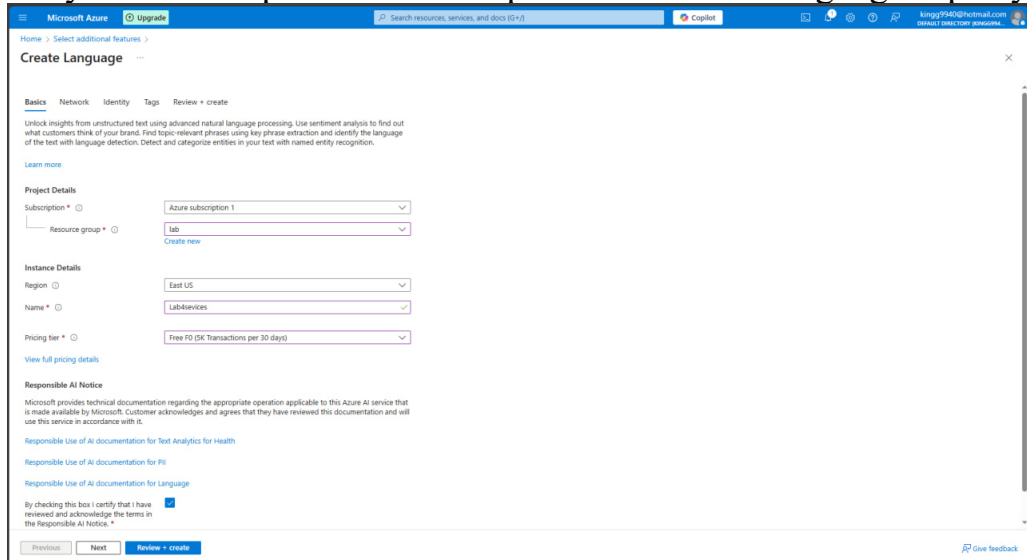


Рисунок 5.2 – Налаштування Azure AI Language сервісу

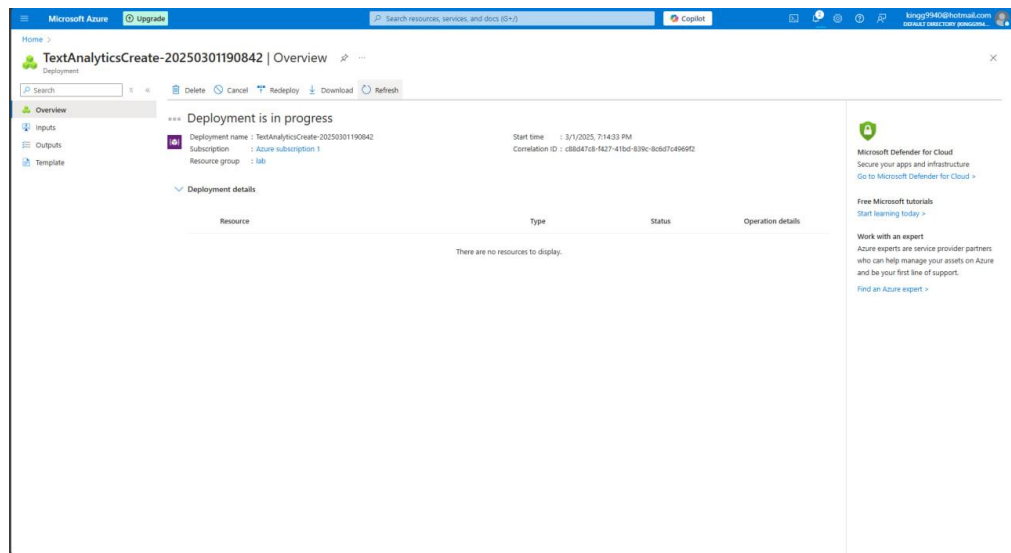


Рисунок 5.3 – Створений Azure AI Language сервіс

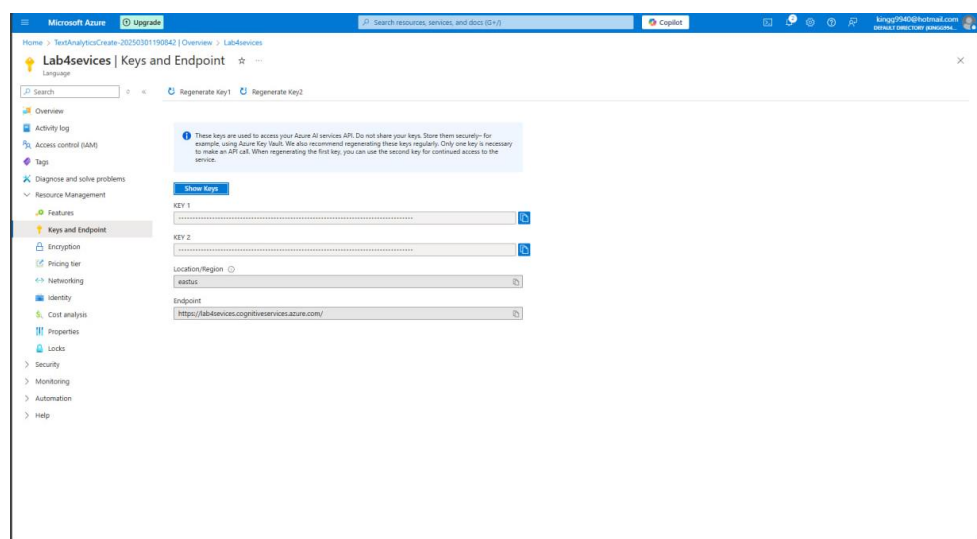


Рисунок 5.4 – Отримання ключа доступу та ендпоінту сервісу

- Далі потрібно створити консольний застосунок для зв'язування сутностей за прикладом як у методичці.

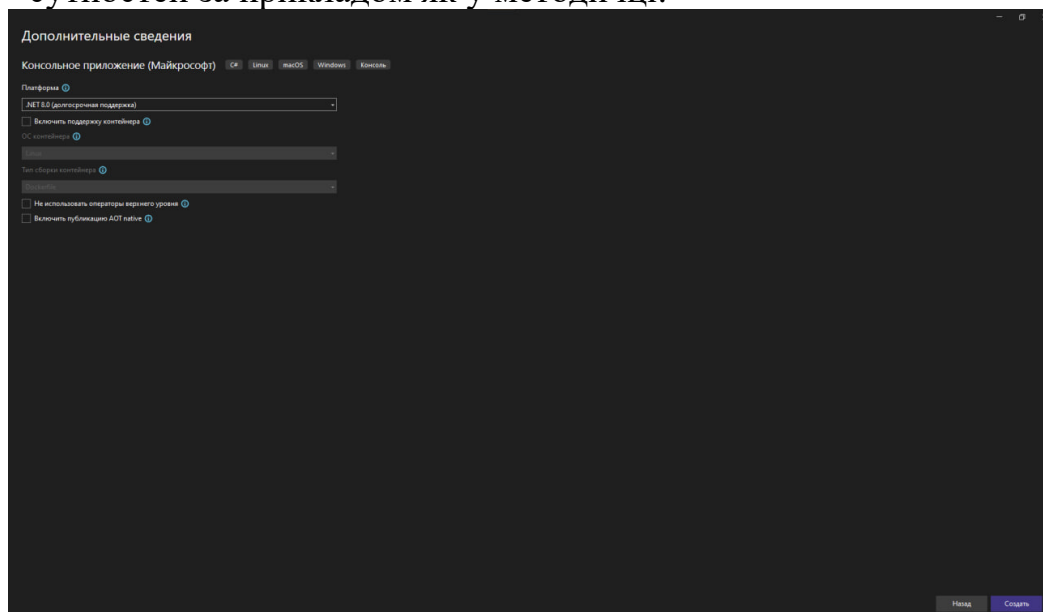


Рисунок 5.5 – створення консольного застосунку

- Далі потрібно додати бібліотеку `Azure.AI.TextAnalytics` у свій проєкт.

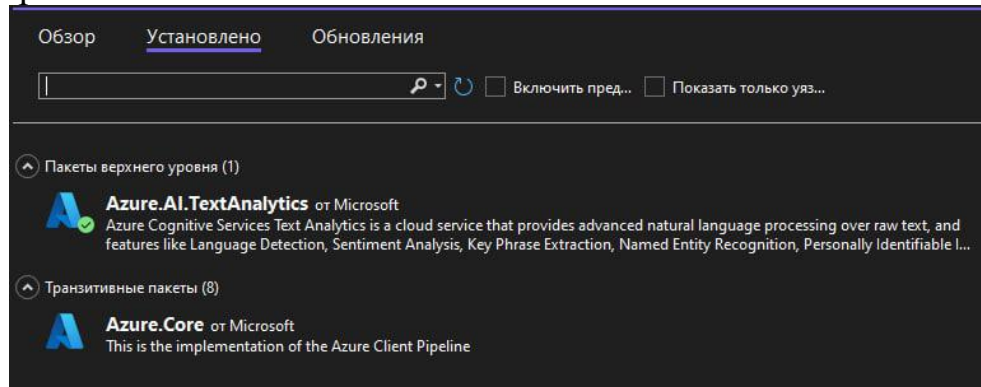


Рисунок 5.6 – встановленні пакети

- Далі потрібно написати код для виклику Entity Linking API, за приклад я взяв код з методички. Програма викличе `RecognizeLinkedEntities()` і проаналізує текст про Microsoft та Білла Гейтса.

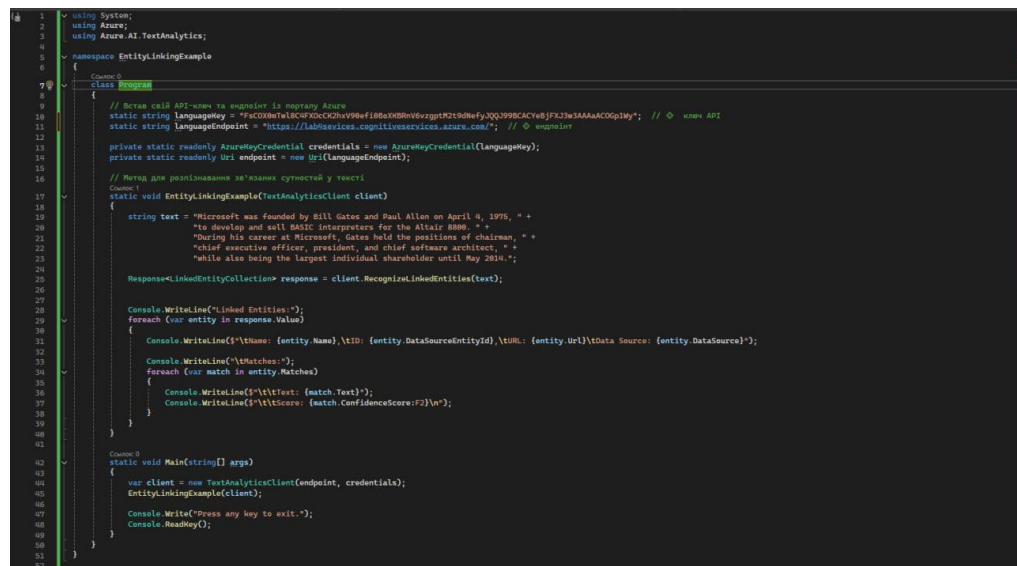


Рисунок 5.7 – Код в Program.cs

- Результат виконання.

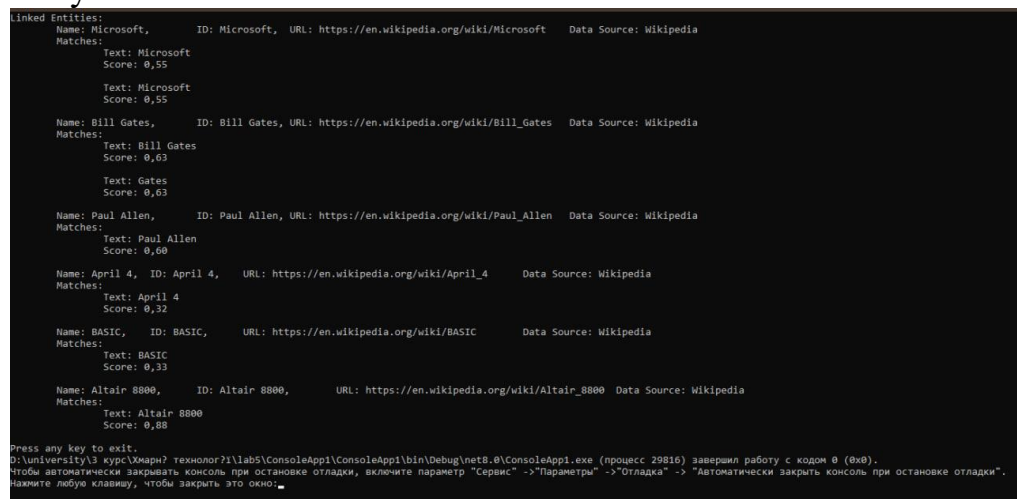


Рисунок 5.8 – виведення результату в консолі

- Наступне завдання це створити Web-застосунок за шаблоном ASP.NET MVC, що пропонує користувачу ввести текст та розпізнати сутності з нього й пов'язати їх із відповідними посиланнями на базу знань. Результати вивести у формі таблиці.
 - Клас EntityModel містить моделі даних для збереження інформації про введений текст та розпізнані сутності. Клас EntityModel має два основних поля: InputText, що зберігає текст, введений користувачем, і Entities, що містить список знайдених сутностей у цьому тексті. Клас EntityResult описує окрему сутність, яка має три властивості: Name (назва сутності), Url (посилання на відповідну сутність) і ConfidenceScore (впевненість у правильності розпізнавання сутності). Це дозволяє передавати дані між контролером та виглядом (view) у веб-застосунку.

```
1 using System.Collections.Generic;
2
3 namespace WebApplication1.Models
4 {
5     Ссылка 4
6     public class EntityModel
7     {
8         Ссылка 2
9         public string InputText { get; set; }
10
11         Ссылка 5
12         public List<EntityResult> Entities { get; set; }
13     }
14
15     Ссылка 3
16     public class EntityResult
17     {
18         Ссылка 2
19         public string Name { get; set; }
20
21         Ссылка 3
22         public string Url { get; set; }
23
24         Ссылка 2
25         public double ConfidenceScore { get; set; }
26     }
27 }
```

Рисунок 5.9 – EntityModel.cs

- HomeController використовується для даного додатку для аналізу тексту на наявність зв'язаних сутностей (персон, організацій, місць тощо). Користувач вводить текст, який відправляється до Azure API для розпізнавання сутностей. Результати виводяться на сторінці у вигляді таблиці з посиланнями та рівнями впевненості. Код виконує три основні етапи: отримання даних, обробка результатів і виведення. Це забезпечує зручну інтеграцію текстового аналізу у веб-застосунок.

```

1 using System.Diagnostics;
2 using Microsoft.AspNetCore.Mvc;
3 using WebApplication1.Models;
4 using Azure;
5 using Azure.AI.TextAnalytics;
6 using System.Collections.Generic;
7 using System.Linq; // Для використання FirstOrDefault()
8
9 namespace WebApplication1.Controllers
10 {
11     [Служок]
12     public class HomeController : Controller
13     {
14         private readonly ILogger<HomeController> _logger;
15
16         // API-ключ і кінцевий пункт
17         private static readonly string languageKey = "F5C0X0mTwL8C4F0ccCh2hxV90efi0BoXhBRnV6vzgptM2tdNefyJQQJ999CACy6BjFXJ3m3AAAAAC0Gp1My"; // API ключ
18         private static readonly string languageEndpoint = "https://lab4services.cognitiveservices.azure.com/"; // кінцевий пункт
19
20         // Ініціалізація клієнта
21         private readonly TextAnalyticsClient _client = new TextAnalyticsClient(
22             new Uri(languageEndpoint), new AzureKeyCredential(languageKey));
23
24         [Служок]
25         public HomeController(ILogger<HomeController> logger)
26         {
27             _logger = logger;
28         }
29
30         // Відображення початкової сторінки
31         [Служок]
32         public IActionResult Index()
33         {
34             return View(new EntityModel());
35         }
36
37         // Біл для аналізу введеного тексту
38         [HttpPost]
39         [Служок]
40         public IActionResult AnalyzeText(string inputText)
41         {
42             var model = new EntityModel { InputText = inputText, Entities = new List<EntityResult>() };
43
44             if (!string.IsNullOrEmpty(inputText))
45             {
46                 // Виконання розпізнавання сутностей
47                 var response = _client.RecognizeLinkedEntities(inputText);
48
49                 foreach (var entity in response.Value)
50                 {
51                     // Перевіряємо чи є елементи у Matches
52                     if (entity.Matches.Any()) // Перевірка на наявність хоча б одного матчу
53                     {
54                         var firstMatch = entity.Matches.First(); // Отримуємо перший матч
55                         model.Entities.Add(new EntityResult
56                         {
57                             Name = entity.Name,
58                             Url = firstMatch.Url,
59                             ConfidenceScore = firstMatch.ConfidenceScore.ToString("F2")
60                         });
61                     }
62                 }
63             }
64
65             return View(model);
66         }
67     }
68 }

```

Рисунок 5.8 – HomeController.cs

- Index.cshtml дозволяє користувачеві ввести текст, відправити його на сервер для аналізу та відобразити результати розпізнавання сутностей. Користувач вводить текст у форму, натискає кнопку для відправлення запиту, і текст передається на сервер, де розпізнаються сутності за допомогою Azure Text Analytics API. Результати відображаються у вигляді таблиці з сутностями, їх посиланнями та рівнями впевненості. Основні елементи шаблону включають визначення моделі для передавання даних з контролера до представлення та цикл для відображення кожної розпізнаної сутності в таблиці.

```

1 @model WebApplication1.Models.EntityModel
2
3 @{
4     ViewBag.Title = "Entity Linking";
5 }
6
7 <h2>Entity Linking</h2>
8
9 <form method="post" action="@Url.Action("AnalyzeText", "Home")">
10     <label>Введіть текст:</label>
11     <textarea name="inputText" rows="4" class="form-control">@Model.InputText</textarea>
12     <br>
13     <button type="submit" class="btn btn-primary">Аналізувати</button>
14 </form>
15
16 @if (Model.Entities != null && Model.Entities.Count > 0)
17 {
18     <h3>Результати:</h3>
19     <table class="table table-bordered">
20     <tr>
21         <th>Сутність</th>
22         <th>Посилання</th>
23         <th>Впевненість</th>
24     </tr>
25     @foreach (var entity in Model.Entities)
26     {
27         <tr>
28             <td>@entity.Name</td>
29             <td><a href="@entity.Url" target="_blank">@entity.Url</a></td>
30             <td>@entity.ConfidenceScore.ToString("F2")</td>
31         </tr>
32     }
33     </table>
34 }

```

Рисунок 5.8 – Index.cshtml

- Робота веб-додатка

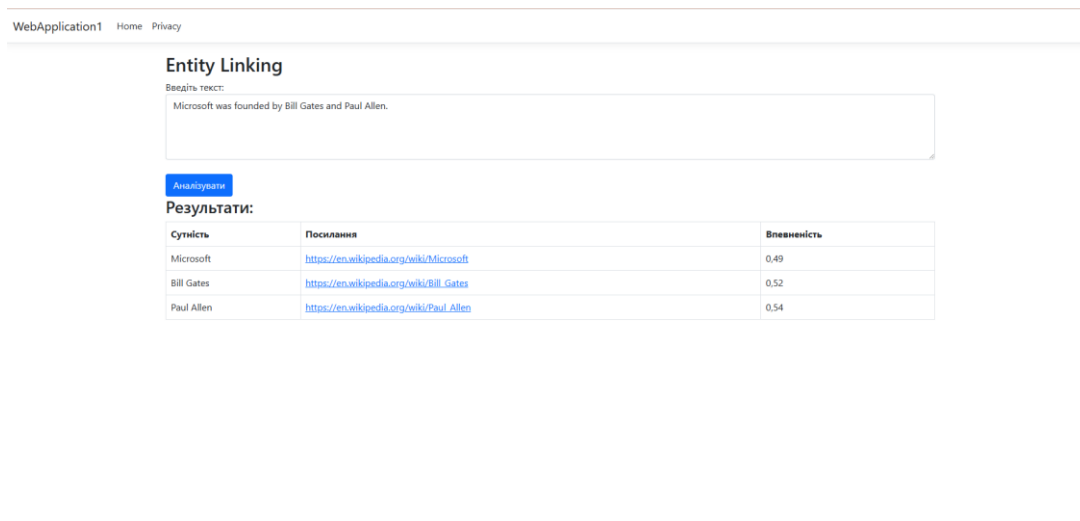


Рисунок 5.8 – як працює веб-додаток

- Перевірити точність розпізнавання і валідність гіперпосилань.
 - Точність розпізнавання тексту "Microsoft was founded by Bill Gates and Paul Allen" системою виявилася точною, правильно розпізнавши три сутності: Microsoft (посилання на сторінку у Вікіпедії: Microsoft), Bill Gates (посилання на сторінку у Вікіпедії: Bill Gates) і Paul Allen (посилання на сторінку у Вікіпедії: Paul Allen). Впевненість у розпізнаванні кожної сутності варіюється від 0,49 до 0,54, що вказує на помірну впевненість у розпізнаванні. Всі надані посилання ведуть до відповідних статей у Вікіпедії, що є достовірними та актуальними джерелами для цих сутностей, і працюють належним чином, ведучи до правильних сторінок.

Висновок

Під час виконання лабораторної роботи було розглянуто процес створення класифікатора документів із застосуванням Azure AI. Було налаштовано Azure AI Language сервіс, отримано його ключ доступу та ендпоінт, створено консольний застосунок та інтегровано бібліотеку Azure.AI.TextAnalytics. Додатково було розроблено веб-застосунок на основі ASP.NET MVC, який дозволяє аналізувати введений текст, розпізнавати сутності та зв'язувати їх із відповідними посиланнями на базу знань. Отримані результати продемонстрували ефективність розпізнавання сутностей, хоча рівень впевненості в деяких випадках був помірним. Всі отримані посилання виявилися валідними, що підтвердило коректність роботи сервісу.

Творче завдання

Тема :Автоматизований аналіз текстів для навчання

Для реалізації автоматизованого аналізу текстів у навчальному процесі, можна створити веб-застосунок, який допоможе студентам або учням працювати з текстами, аналізуючи сутності (персоналії, місця, події) і надаючи посилання на додаткові ресурси для поглибленого вивчення.

- Кроки для реалізації:

- Вибір інструментів і технологій:
 - Використовувати Microsoft Azure Cognitive Services або інші API для розпізнавання сутностей в текстах. Ці сервіси дозволяють ідентифікувати і класифікувати сутності, такі як особи, місця, організації та інші важливі елементи.
 - Створити веб-застосунок на основі ASP.NET MVC або ASP.NET Core, який дозволяє користувачам вводити тексти і отримувати результати аналізу.
- Процес введення тексту:
 - Студенти вводять або завантажують текст в спеціальну форму веб-застосунку. Це можуть бути фрагменти з історичних статей, літературних творів або наукових досліджень.
- Аналіз тексту:
 - Після відправки тексту на сервер, застосунок використовує API для аналізу тексту та розпізнавання сутностей. Це може бути:
 - Персоналії (історичні постаті, персонажі книг)
 - Місця (міста, країни, географічні об'єкти)
 - Події (історичні події, культурні заходи)
 - API повертає інформацію про знайдені сутності, їх посилання на додаткові ресурси (наприклад, Вікіпедія) і оцінку точності розпізнавання.
- Виведення результатів:
 - Після аналізу, студент отримує список знайдених сутностей разом з посиланнями на ресурси для поглибленого вивчення. Це може бути таблиця на веб-сторінці, де кожен елемент містить:
 - Назву сутності
 - URL на зовнішнє джерело (наприклад, Вікіпедію)
 - Оцінку впевненості в розпізнаванні сутності.
- Додаткові функції:
 - Можна додати можливість для студентів коментувати або робити замітки щодо розпізнаних сутностей.
 - Студенти можуть зберігати свої результати для подальшого аналізу або підготовки доповідей.
- Інтеграція з навчальними матеріалами:
 - Застосунок може бути інтегрований з іншими навчальними ресурсами, такими як електронні підручники або онлайн-курси.
 - За допомогою інтеграції з іншими системами (наприклад, LMS — Learning Management System), результати аналізу можна автоматично завантажувати на платформу для подальшого використання викладачами.
- Приклад використання:
 - Літературні твори: Студент завантажує текст роману, скажімо, "Майстер і Маргарита". Застосунок автоматично виділяє сутності

(Майстер, Маргарита, Москва) і надає посилання на Вікіпедію для глибшого розуміння персонажів і контексту твору.

- Історія: Студент вводить текст, пов'язаний з важливою історичною подією, наприклад, опис битви при Waterloo. Застосунок виділяє ключові особи (Наполеон Бонапарт, Велика Британія) і дає можливість поглиблено вивчити історичний контекст через посилання на статті.
- Переваги для навчання:
 - Швидкий доступ до додаткових ресурсів: Студенти можуть швидко отримати більше інформації про сутності без необхідності шукати їх вручну.
 - Поглиблене розуміння контексту: Інструмент дозволяє вивчати текст не тільки в рамках його змісту, а й через призму зовнішніх посилань на додаткові матеріали.
 - Покращення навичок пошуку інформації: Студенти навчаються працювати з сучасними інструментами для аналізу текстів і пошуку знань.

Цей підхід може стати потужним інструментом для студентів гуманітарних дисциплін, допомагаючи їм краще орієнтуватися в матеріалі та глибше розуміти його контекст.