**Architecture Documentation**

**Sheffield FOI Insight Tool**

**Team Members**

**FOI AI Team**

|  |
| --- |
| Abokor Ahmed |
| Lasse |
| Peter Karakostas |
| Richard |

Submitted for: Sheffield AI Hackathon

Date: **14/05/2025**

Version: 1.0

**Purpose**

This document outlines the system architecture for the Sheffield FOI Insight Tool, which aims to improve public access to Freedom of Information (FOI) request data by leveraging web scraping, AI-powered summarisation, and interactive web publishing. The tool enhances transparency, reduces workload on council staff, and promotes data-driven engagement.

**High-Level Architecture Overview**

The system consists of five core components:

1. **Data Scraper (Python Script)**  
   Scrapes FOI data from [WhatDoTheyKnow.com](https://www.whatdotheyknow.com) for Sheffield City Council. Extracts request metadata, message history, and attachments, storing them as JSON files.
2. **Automated AI Analysis (Power Automate & Azure OpenAI)**  
   A Power Automate flow monitors the scraped data and pushes it to Azure’s OpenAI service, applying two custom prompts:
   * **Metadata Extraction Prompt**: Extracts structured metadata such as request themes and tags.
   * **Response Summarisation Prompt**: Generates concise, plain-language summaries of requests and responses.
3. **Structured Data Storage (Excel Template & JSON)**  
   The AI output is stored in a structured **Excel template** for validation. This data is later exported or transformed into **JSON format** for use in the web app.
4. **Web Publishing Layer (GitHub Pages)**  
   A static web interface loads the structured JSON data and presents:
   * Searchable, filterable FOI request summaries.
   * Interactive data visualisations (e.g., charts, filters, trend analysis).
   * Public access via a GitHub Pages URL.
5. **Data Transformation & Visualisation (Python Script)**A planned Python script will:
   * Convert structured Excel/JSON into web-ready formats.
   * Generate charts and tables to be embedded in the web interface.

|  |
| --- |
|  |
| Figure 1: [High-Level Architecture Diagram](https://www.mermaidchart.com/play#pako:eNp1UsuO2jAU_ZWrLCpYUPYsKqUBpHSYBuUBi2Y0MokhlhKb8aNMpuq_99oOLVDwIrk-93nO9a-gEjUNZsFkMil5JfieHWYlB2hJL4yewZ6907rkzr1vxalqiNSwSjFGmd1BkmMDhaLyRxlMnQEr0uN3WgYvto4_RTzCAOueQcQ0-6AcphBy0vZKo5VSRYmsGioxbWzzKK8vW2TCyIq6JnOiyXC_6ZKlkW2zbYiei7yh_RMXJ1gmMdZ_M1Rp9ah6hb-BgysfVm-GKRxU8Lt8sii1nda9bjBiSJ9BRxj_fOwfddFCkoMnEXNNJevO2E35KNuMRqMySIkfH--wZC3F-cfje6XXUlRUKcYPrvq_693pl6tk68YXJ_SFRouOaAqJXYDSkljaZw7-hG5_4YeRGHakPIzhmWpSW60-QWa6jsgeFu-YXF1m34yZGH002o2YaWkqjfVqr_hdlfO0iPIiXcxfz4oMzDDrf038-ZYl31-TIl8XuZ05l4SrvZAdplgXoA1bunswIXrcePiHtdm1TDWPVNzEWRGuMtvEbdMy_0lhw5QhmOdUVNcy5uHX1cJlZO65k11LPf3cmjfP079omEy-2Pc2AFHqAGTvASuDBexOPWItB4WxB3BZrsaVmkO5K8yFXejnYy4AFzDwxr17Oj7qjNqIYmjs_X-h4PcfqudcGA) |

|  |
| --- |
| **System Instructions to Azure OpenAi model:**  Summarise instruction:  “You are a summarisation assistant. Read the input text and return exactly one, plain-English sentence that captures the main point. Omit quotations, disclaimers, and unnecessary detail—output only the single sentence.”  Theme instruction:  “You are a classification assistant that analyses text and returns a short, descriptive theme that captures the main topic. The theme must be a general category like Education, Environment, Public Safety, Housing, Waste Management, or similar. Return only the theme with no explanation.”  Tag instruction:  “You are a tagging assistant for FOI requests. Your task is to return 3 to 5 relevant tags that describe the key subjects, entities, or data points mentioned in the request. Use lowercase, comma-separated keywords only. Do not repeat similar tags. Do not include explanations or full sentences — return only the tags.” |
| Figure 2: System Instructions to Azure OpenAi model |

|  |
| --- |
|  |
| Figure 3: Power Automate Flow |

|  |
| --- |
|  |
| Figure 4: Azure OpenAI (gpt-4o-mini) Metrics Screenshot |

|  |
| --- |
|  |
| Figure 5: CSV Output Extract |

1. **Deployment Considerations**

* **Short-Term (Hackathon)**
  + Static web app using pre-generated JSON data.
  + Manual trigger for AI processing via Power Automate.
* **Long-Term (Post-Hackathon)**
  + Fully automated data pipeline with live scraping and AI processing.
  + Dynamic data loading in web interface.

**Future Enhancements**

* Live data refresh and automated publishing on incremental refresh schedule.
* Multi-council or national FOI data integration.
* Richer, dashboard-style visualisations.
* Advanced filtering by tags, departments, and dates.
* Advanced analytics by refining NLP approach.