Scraping, storing and processing v2

Part 1: Web Scraping and Database Storage

Url to scrap: https://subway.com.my/find-a-subway
First, filter the search by "**kuala lumpur**". Scrape the names, addresses, operating hours, waze link of outlets from a given webpage that has multiple pages of content. Ensure your script can handle pagination to navigate through all available pages.
Store the scraped data into a database, designing the schema in a way that you find suitable for this task.

Part 2: Geocoding

For each outlet, retrieve its geographical coordinates based on the stored address.

Part 3: API Development

Develop a backend API (FastAPI, Flask, Django) to serve the outlet data, including their geographical coordinates.

Part 4: Frontend Development and Visualization

Create a web application that interacts with your API to visualise the outlets on a map.
Implement functionality to display a 5KM radius catchment around each outlet on the map.
Highlight or mark the outlets that intersect with any other outlet's 5KM radius catchment.

Part 5: Documentation and Instructions

Add a search box to allow the user to enter a query. Examples of query that you would need to handle include:

1. Which are the outlets that closes the latest
2. How many outlets are located in Bangsar

You are free to use RAG, LLM or NLP to achieve this.

Part 6: Documentation and Instructions

Provide documentation with instructions on how to set up and run your application, and any other necessary information required to understand and use your solution.

Submission

* Submit all the source code via git.
* Provide the link to a hosted version of the solution.
* A detailed README.md file.
* Preferable python version 3.8-3.11
* Please omit all api keys/secret when pushing to git or when you share the code

Submit to jermaine@mindhive.asia (cc: johnson@mindhive.asia)