**A picture containing light, night

Description automatically generatedProject 3**

**Visualising Data**

**UK Leading Restaurants**

**Contents**

Project outline

* Data sources
* Findings

Project Report

* Web Scraping
* Visualisation
  + Flask
  + Leaflet

Team Roles

**Project outline**

Restaurants are open again, so here’s a great place to start if you need ideas

From top tapas bars to fancy fine dining rooms, barnstorming bistros to great gastropubs, our project brings you the top 100 places to eat across the UK as voted for by an elite academy of chefs, restaurateurs, and food writers nationwide.

**Data Sources**

Hardens annual poll of diners surveys thousands of participants which, as curated by the editors, form the basis for the inclusions and ratings in the guide.

We used one data source provided by their website <https://www.leadingrestaurants.co.uk/data/full-lists/hardens/>

The resulting data set contains 100 records,

For the final data we used MongoDB, a Non-Relational database.

**Findings**

The resulting csv was used for visualising data using a Python Flask- powered API, HTML/CSS, JavaScript.

This project is a combination of web scraping and Leaflet and Plotly.

**Project Report**

**Graphical user interface, text, application, chat or text message

Description automatically generatedDependencies**

* **Pandas**
* **Beautiful Soup**
* **Browser**
* **ChromeDriverManager**
* **Geocoder**
* **numpy**

**Graphical user interface, text, application, email

Description automatically generated**

**Web Scraping**

We used BeautifulSoup for web scraping purposes to pull the data out of HTML files. It allowed us to create a parse tree from the page source code and extract data in a hierarchical and more readable manner.

Graphical user interface, text

Description automatically generated

We have inspected the webpage and printed all the paragraphs contained in <div>.

Then, we created columns from the data scraped, found all div and created a list containing the following class names: ‘res\_names’ , ‘res\_location’ and ‘top\_pos’. These were later renamed for readability.

Graphical user interface, text, application, email

Description automatically generated

We extracted the links containing the restaurants name and their respective postcode.

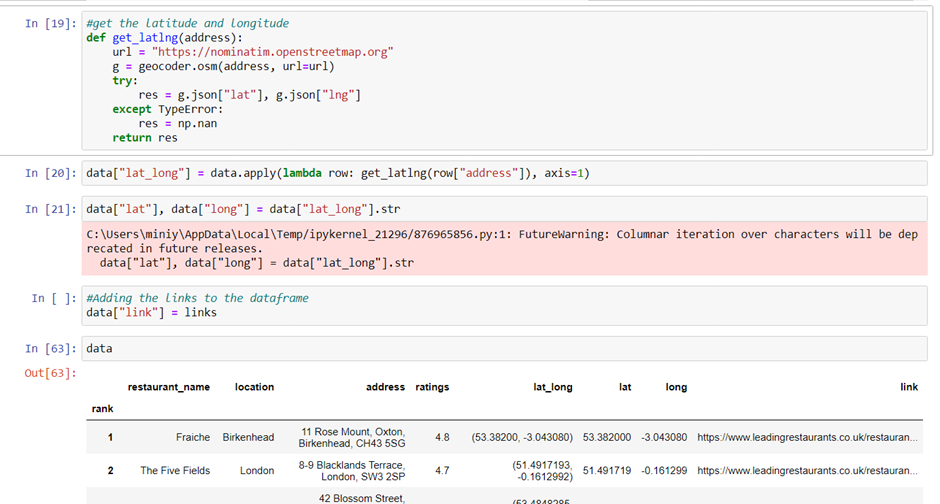
Text

Description automatically generated

We extracted the addresses and the ratings, then we added them to the dataframe.

Graphical user interface, text, application, email

Description automatically generated



We have also extracted the latitude and longitude.

These coordinates will have helped us to determine the location on the map.

Graphical user interface, text, application, email

Description automatically generated

The resulting data was saved as a csv in the Resources folder.

**Visualisation**

**WEB APPLICATION (FLASK)**

For this project, we included as part of the visualisation a framework based on Flask to carry out a web application. The choice of Flask is to take advantage of its flexibility and customization capabilities resulting in clean and concise coding.

It also gives as the room to populate our database easily. A very useful tool for back-end development.

Framework:

Text

Description automatically generatedIn this work we used only one script file called app.py

Steps:

* Importing modules: From flask import Flask
* Create a Flask object: app = Flask(\_\_name\_\_)
* Run the application using the run( ) function:
* Create a view function with the output function shown in the browser

Text

Description automatically generatedTable

Description automatically generated

Text

Description automatically generatedMap

Description automatically generatedVisualisation using Leaflet

Map

Description automatically generated

Text

Description automatically generated

Map

Description automatically generated

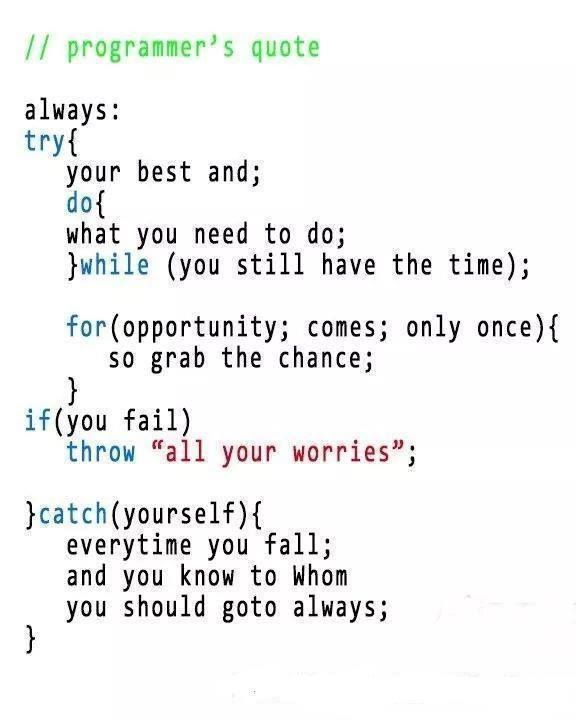
Text

Description automatically generatedText

Description automatically generatedVisulisation using Plotly

A picture containing text, measuring stick, antenna

Description automatically generated

**Team roles**

Although each person worked primarily on their allocated task, all steps were reviewed and completed as a team effort over more than 50 hours of collaborating.

**Monkagisi Mafavuke**

Primary task: Visualising data using Plotly, HTML/CSS, JavaScript &Leaflet

Secondary task: Scrape, Transform & Load Process,

**Maxwell Ansah**

Primary task: Visualising data using Flask, HTML/CSS, JavaScript

Secondary task: Scrape, Transform & Load Process,

**Mini Yadav**

Primary task: Scrape, Transform & Load Process, Finding Datasets

Secondary task: Project report

**Elena Dragomir**

Primary task: PowerPoint presentation, Project report

Secondary task: Scrape, Transform & Load Process,