# Customer Segmentation Project

## Table of Contents

## Project Overview

The Customer Segmentation project aims to categorize customers based on their purchasing behavior using K-Means clustering. The project provides a web interface that allows users to upload their CSV data, select the clustering model, and visualize the resulting customer segments through a scatter plot.

## Technologies Used

- **Flask**: A lightweight web framework for Python.
- **Pandas**: For data manipulation and analysis.
- **Matplotlib**: For creating static, animated, and interactive visualizations in Python.
- **Scikit-learn**: A library for machine learning in Python that provides simple and efficient tools for data mining and data analysis.

## 1. Ensure the directory structure includes:

- app.py: The main Flask application file.
- templates/: Folder containing HTML templates (e.g., index.html).
- static/: Folder to store static files like images.

## 2. Project Structure

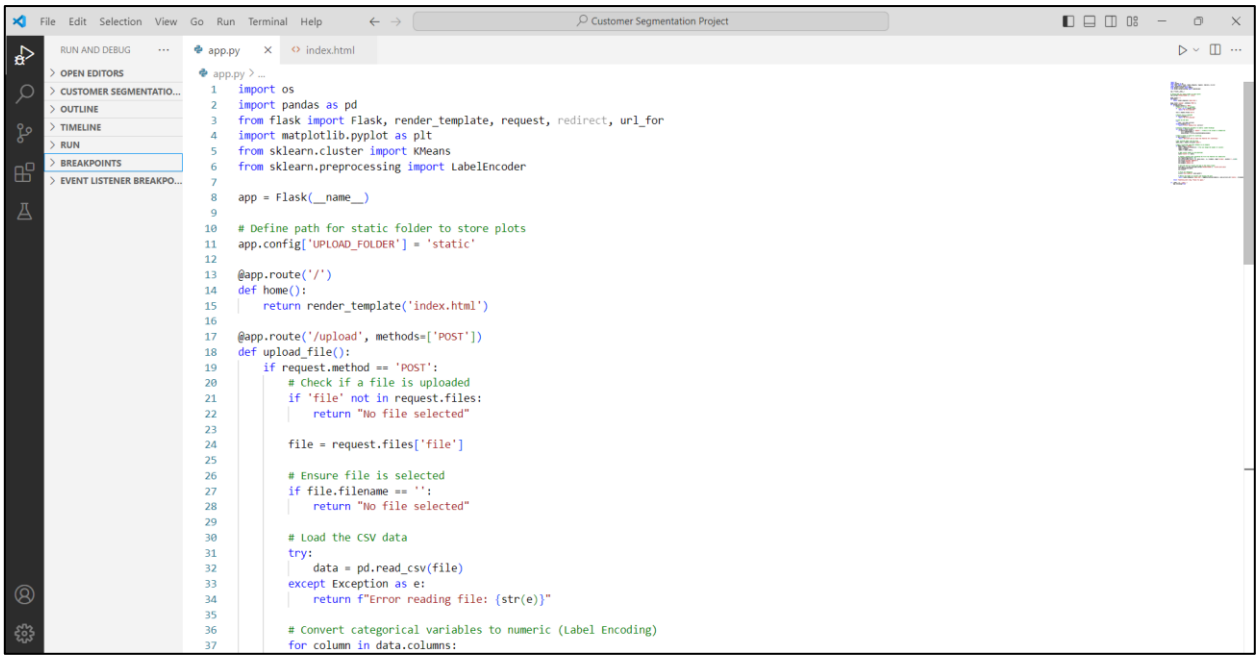
```
Customer Segmentation Project/
├── app.py
├── templates/
│   └── index.html
└── static/
```

## Implementation Steps

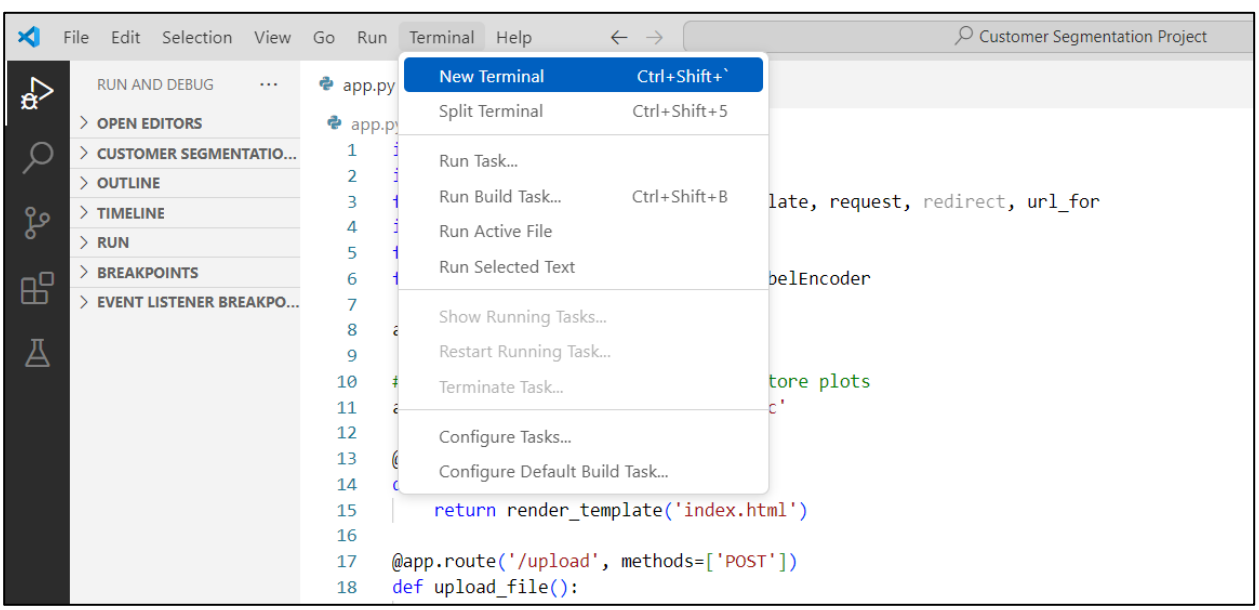
1. **Set up Flask App**:
    - Create a Flask application instance and configure the static folder for storing uploaded files and generated plots.
2. **Define Routes**:
    - **Home Route** (/): Renders the main page with the file upload form.
    - **Upload Route** (/upload): Handles file uploads, applies K-Means clustering, generates a plot, and displays the results.
3. **Handle CSV File Upload**:
    - Use Pandas to read the uploaded CSV file and apply K-Means clustering using Scikit-learn.
4. **Generate Cluster Plot**:
    - Use Matplotlib to create a scatter plot of the clustered data and save it in the static folder for later display.
5. **Render Results**:
    - Use Jinja2 templating to pass the number of clusters and the plot URL back to the HTML page.

# Running the Project

1. Open the app.py code in “visual code studio/ vs code” then in vs code open new terminal run **Run the Flask application:**




2. Open New Terminal:



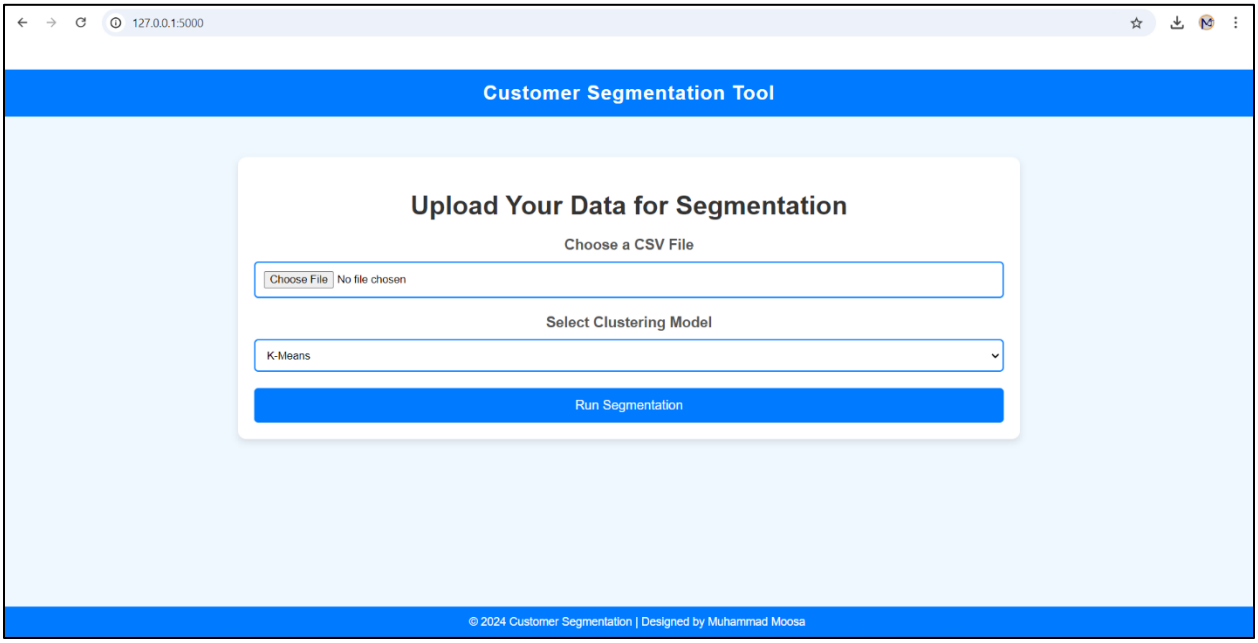
3. Run the Flask application: **app.py**

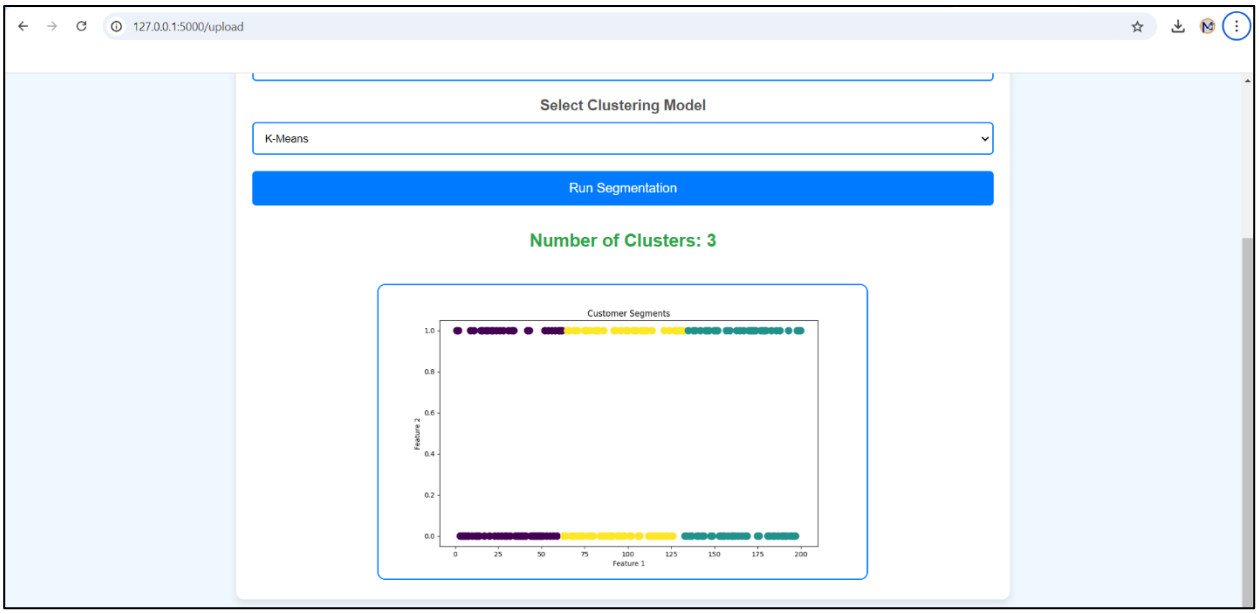
```
13  @app.route('/')
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Users\PMLS\Desktop\Customer Segmentation Project>
python app.py
```

4. Open your web browser and navigate to

```
* Running on http://127.0.0.1:5000
```

**On Web browser output:**






Choose the file of the your “dataset” file then click on button **On Segmentation**

# Usage Instructions

1. **Upload Data**:
    - Click on the "Choose a CSV File" button to select your data file. Ensure the file is in CSV format with at least two numeric features for clustering.
2. **Select Clustering Model**:
    - Choose the K-Means model from the dropdown menu.
3. **Run Segmentation**:
    - Click the "Run Segmentation" button to submit the form.
4. **View Results**:
    - After processing, the page will display the number of clusters identified and the corresponding cluster plot.

# Conclusion

This Customer Segmentation project serves as a practical implementation of K-Means clustering, providing a user-friendly interface for analyzing customer behavior. Users can easily visualize segmentation results, aiding in better decision-making for marketing and sales strategies.