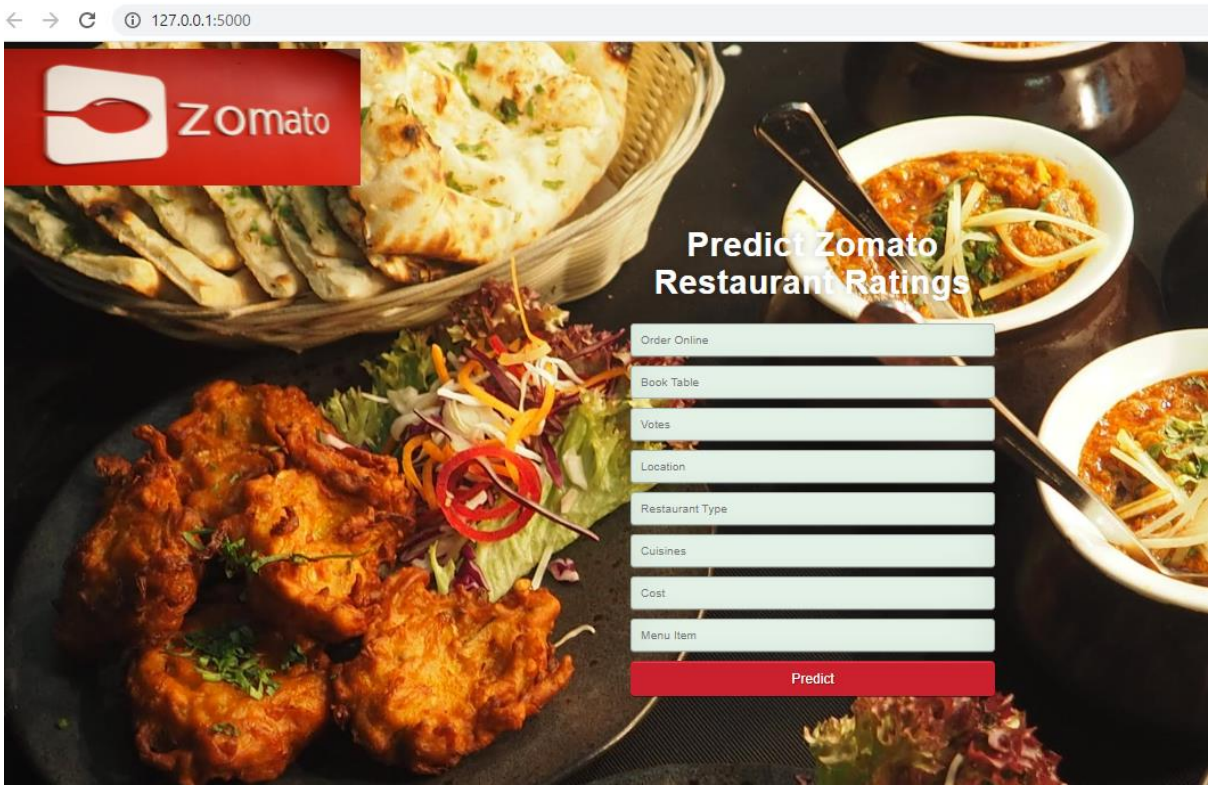
End-To-End Deployment of Zomato Restaurant Ratings



Main Objective:

The main agenda of this project is:

1. Perform Exploratory Data Analysis(EDA) and feature engineering on the Zomato Dataset.

2. Build an appropriate Machine Learning model to predict their respective Ratings based on certain features.

3. DEPLOY the Machine learning model via Flask that can be used to make live predictions of restaurants ratings.

STEPS:

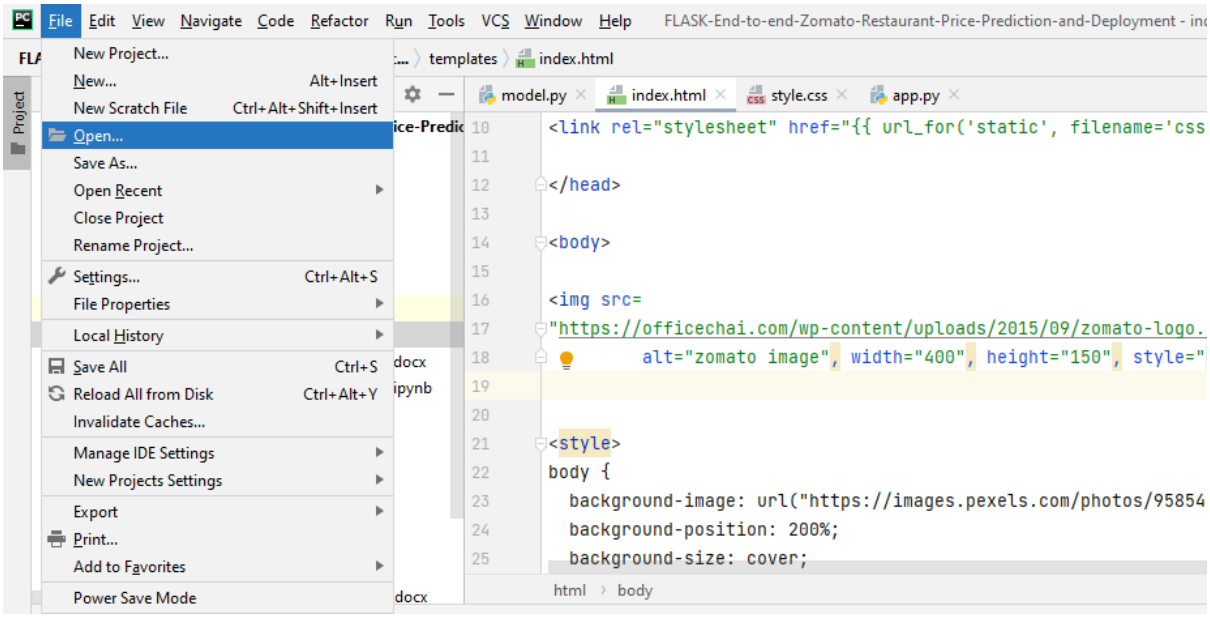
A. EDA , feature engineering and Model Building Part

1. Load the dataset and perform the necessary EDA in your Jupyter notebook.
2. 2. Build your Machine learning model and save the model using “pickle”

B. Deployment Part

1. In this project we will be using “pycharm.
2. Install your favourite IDE (e.g. pycharm) if not installed.

3. Setting up your Pycharm
    - Create a new folder in your system and give it a name (e.g. Zomato_review)
    - Open your pycharm IDE and click on file—>open—>{select the folder you created}

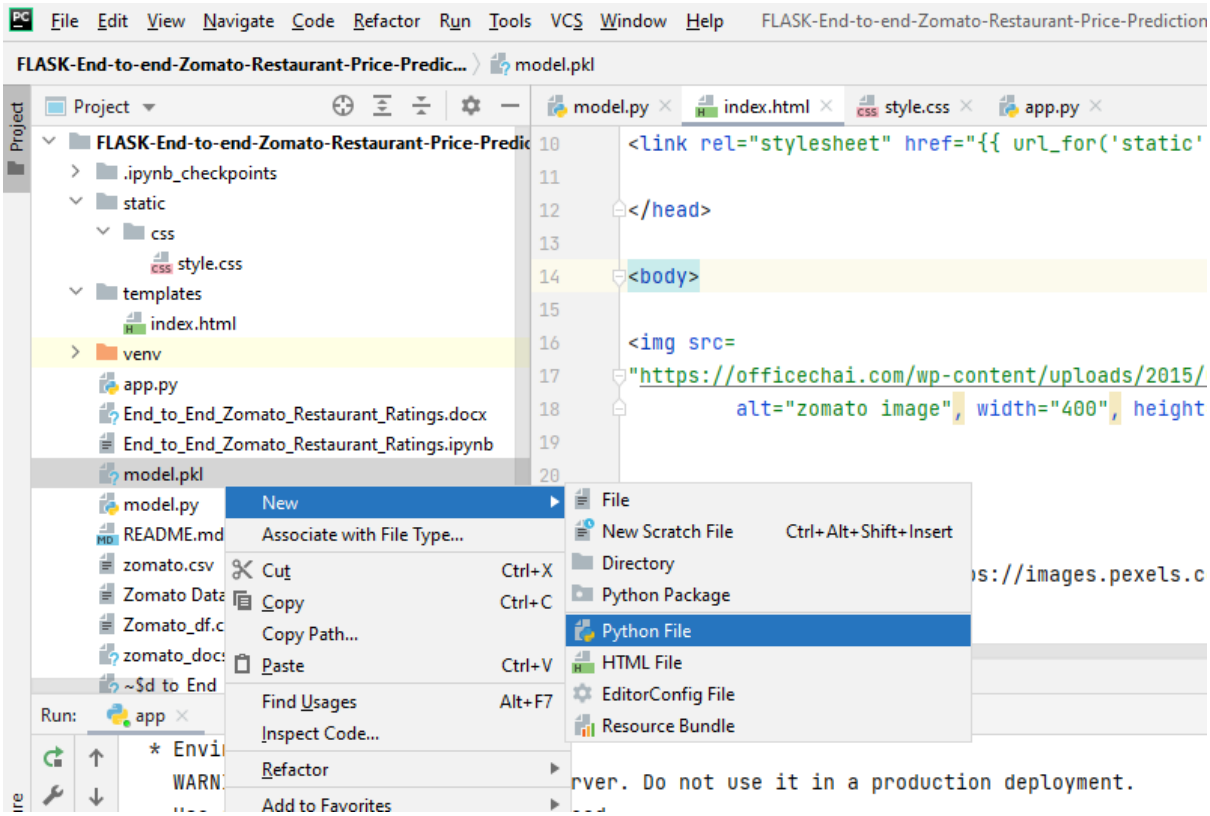


4. Optional (but recommended)
    - You can create a virtual environment to avoid any conflict in library dependencies (recommended) follow below links to create a virtual environment: https://bit.ly/2CwnTfo

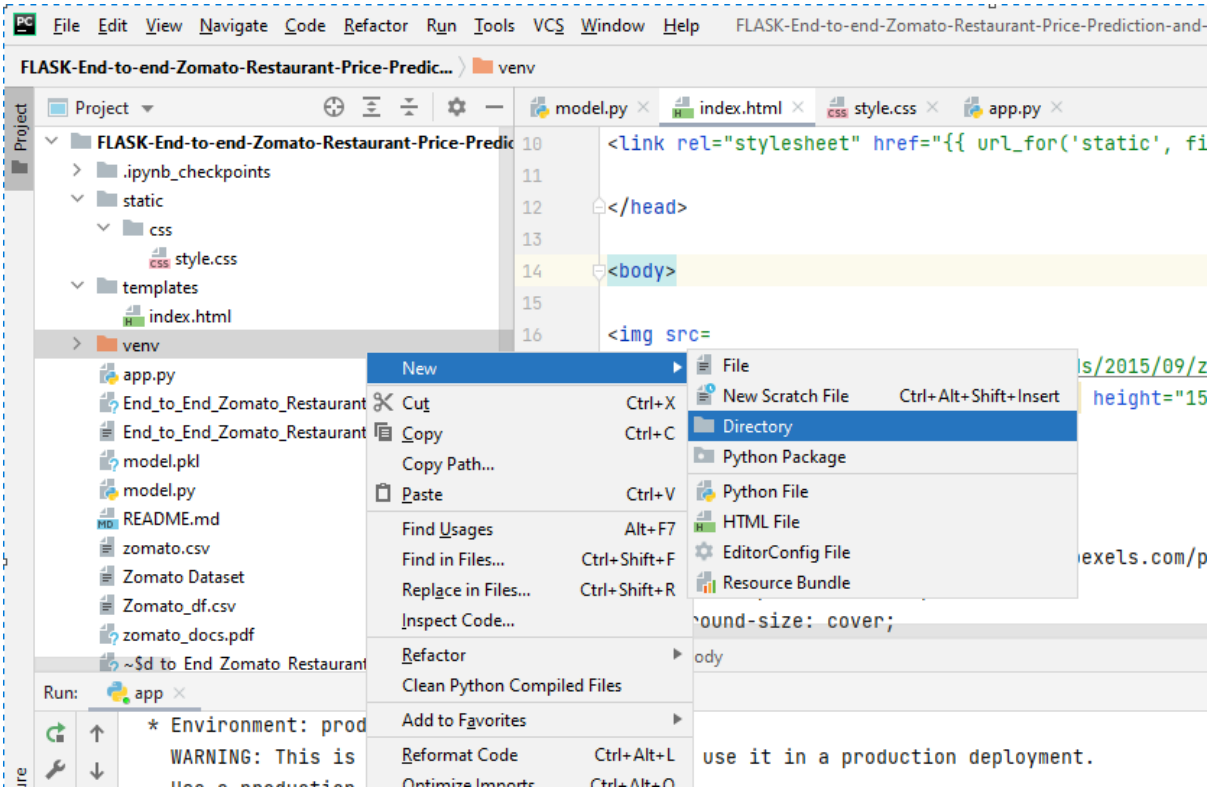
5. Files you will need:
    - **Model.py file** :- This file contains the code for building our model that is used predicting the restaurant ratings.
    - **.CSV file** :- This contains our data that we have already cleaned and saved(Zomato_df.csv).
    - **Template**: The template file contains the html and css documents used in building our web app.
        - **.html file**
        - **.css file**

    - **App.py file** :- This contains the Flask API's that receives restaurant details via a GUI/API calls, then make the prediction of restaurant ratings based on our model and returns the ratings.

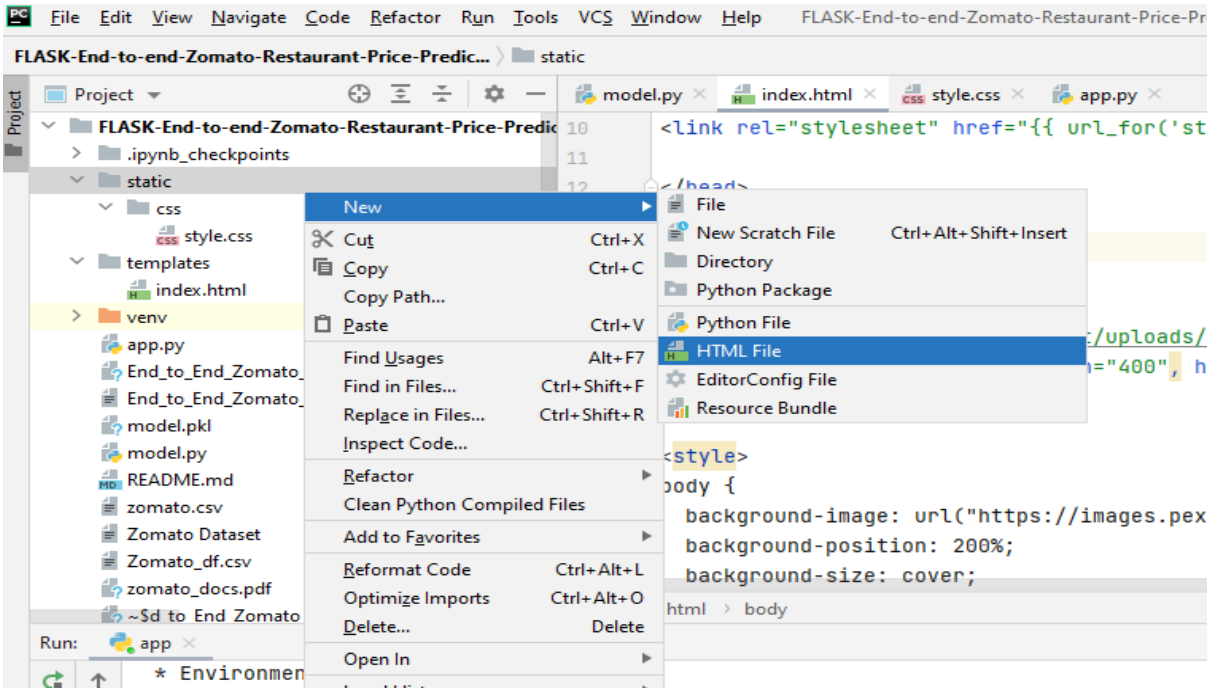
6. Creating your files
    - Create a new python file and name it app.y {your home folder}—>New—>Python file



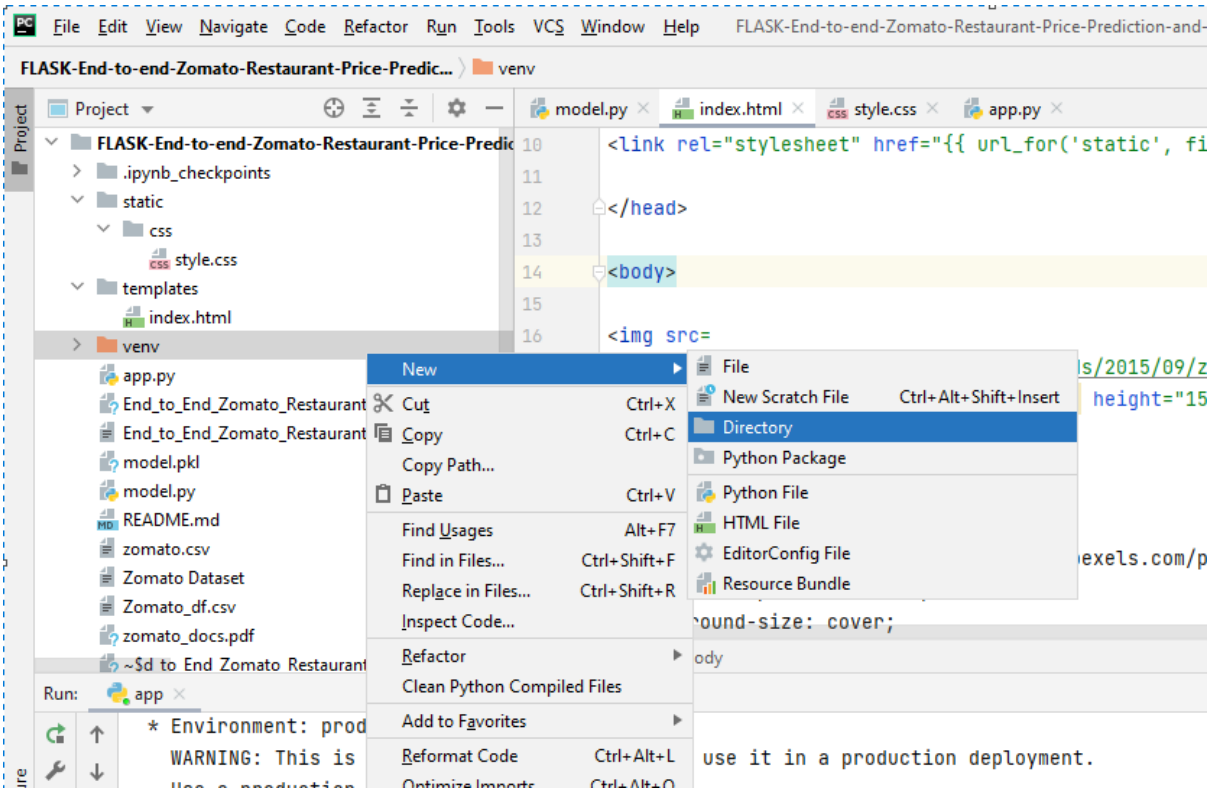
7. Create a template folder(for html file)
    - {your home folder}—>New—>Directory(name it as templates)



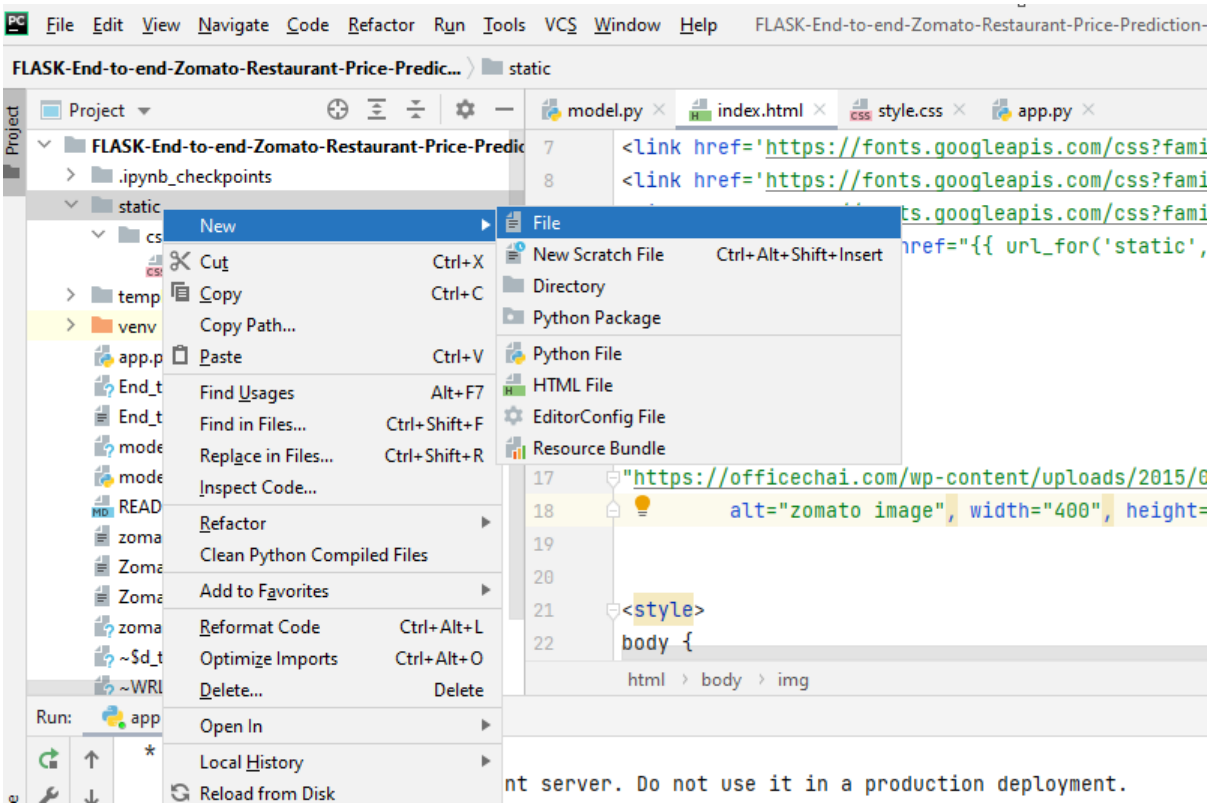
8. Create a HTML file:
    - templates—>New—>HTML file



9. Create a static folder (for css)
    - {your home folder}—>New—>Directory(name it as static)



10. Create a css file:
    - static—>New—>css file



Note: For reference I have attached all the file to folder, So you use it or create it by taking the reference of existing file.