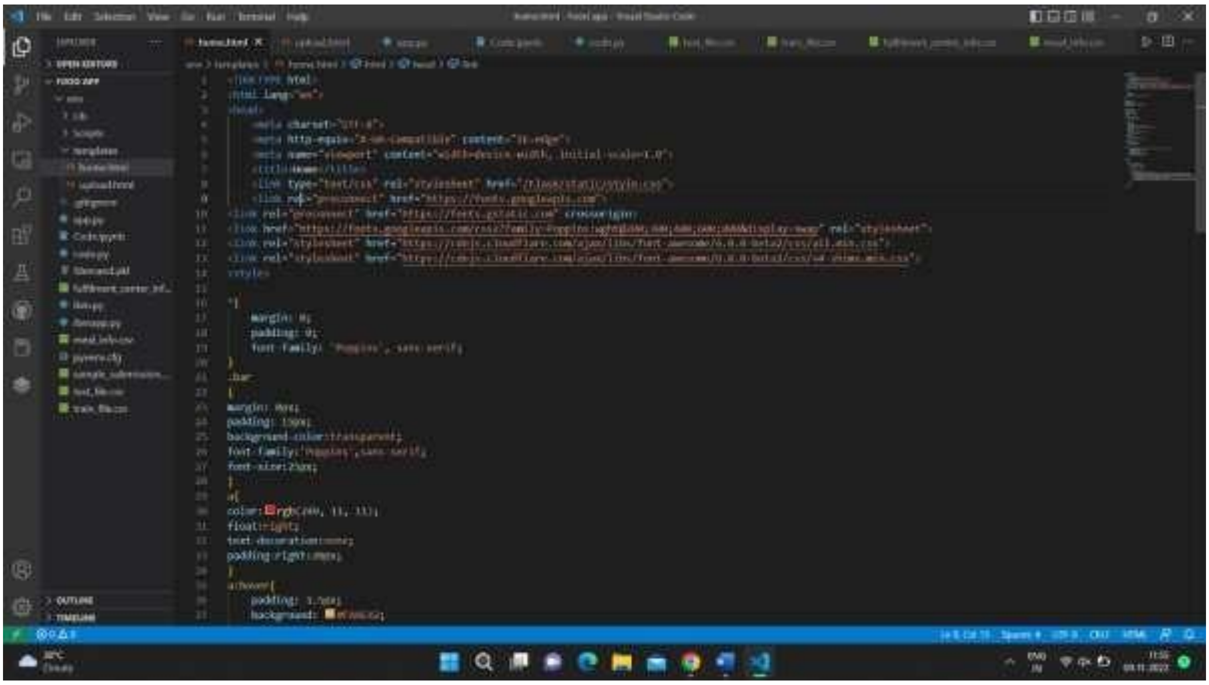
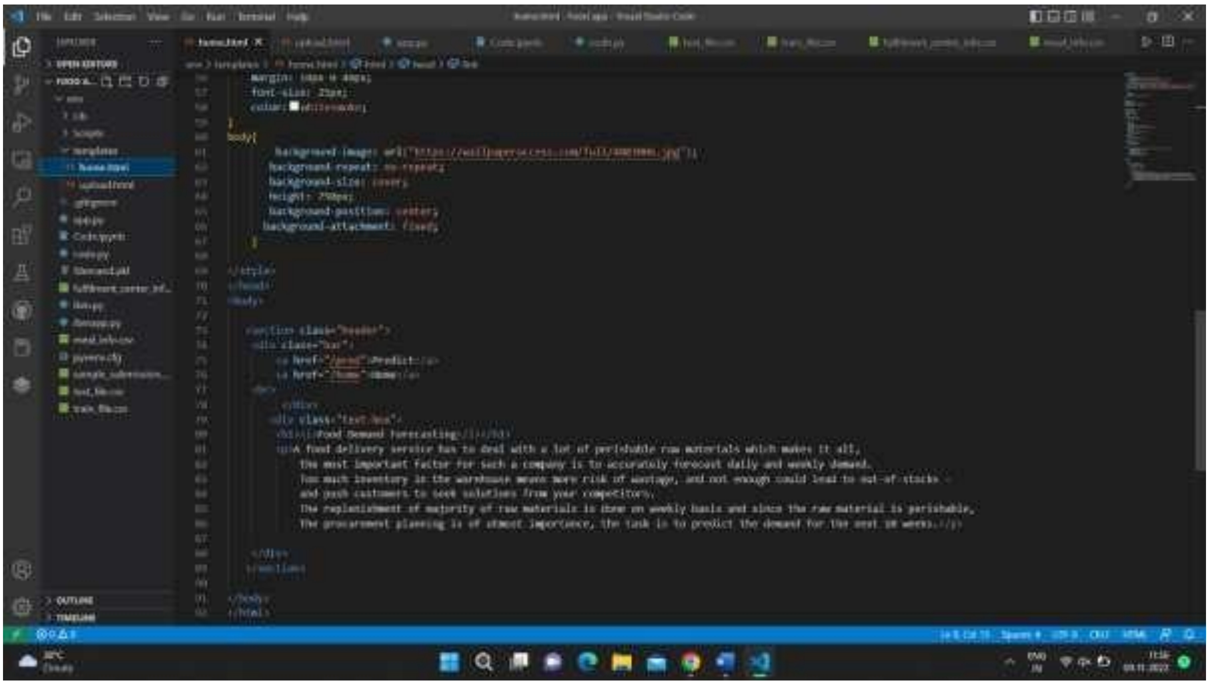
SPRINT 4
FLASK INTEGRATION

HTML CODE

Home.html

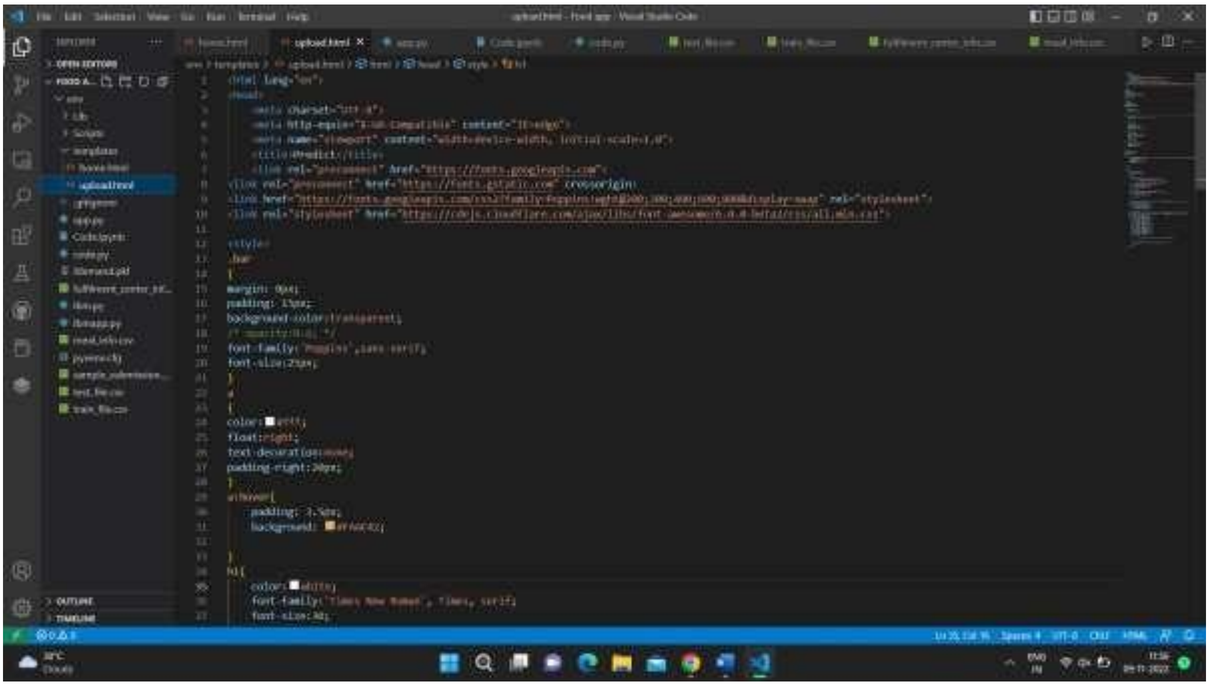


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home</title>
    <link type="text/css" rel="stylesheet" href="/flask/static/style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@[illegible]&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/all.min.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta2/css/v4-shims.min.css">
<style>

*{
    margin: 0;
    padding: 0;
    font-family: 'Poppins', sans-serif;
}
.bar
{
margin: 40px;
padding: 130px;
background-color:transparent;
font-family:'Poppins',sans-serif;
font-size:25px;
}
a{
color:rgb(240, 11, 11);
float:right;
text-decoration:none;
padding-right:20px;
}
a:hover{
    padding: 1.5px;
    background: #[illegible];
```

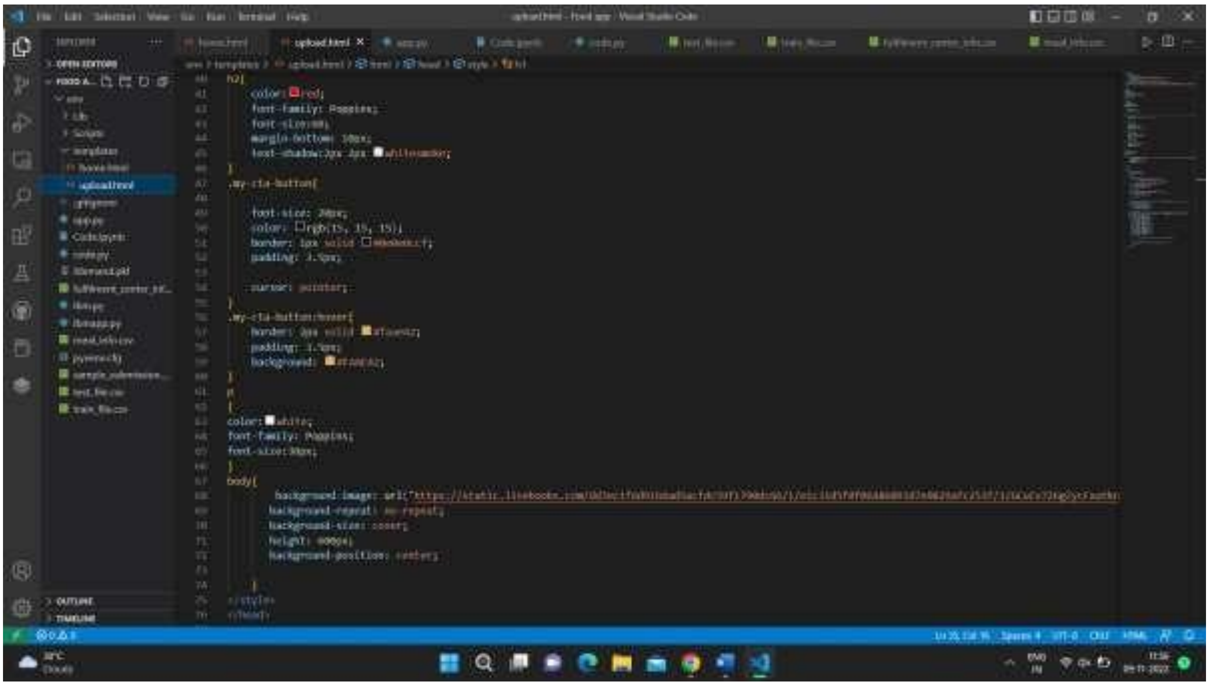


```html
    margin: 10px 0 40px;
    font-size: 25px;
    color: whitesmoke;
}
body{
      background-image: url("https://wallpaperaccess.com/full/[illegible].jpg");
     background-repeat: no-repeat;
     background-size: cover;
     height: 750px;
     background-position: center;
    background-attachment: fixed;
   }

</style>
</head>
<body>

  <section class="header">
   <div class="bar">
       <a href="/pred">Predict</a>
       <a href="/home">Home</a>
   <br>
       </div>
     <div class="text-box">
      <h1><i>Food Demand Forecasting</i></h1>
      <p>A food delivery service has to deal with a lot of perishable raw materials which makes it all,
        The most important factor for such a company is to accurately forecast daily and weekly demand.
        Too much inventory in the warehouse means more risk of wastage, and not enough could lead to out-of-stocks -
        and push customers to seek solutions from your competitors.
        The replenishment of majority of raw materials is done on weekly basis and since the raw material is perishable,
        The procurement planning is of utmost importance, the task is to predict the demand for the next 10 weeks.</p>

   </div>
  </section>

</body>
</html>
```

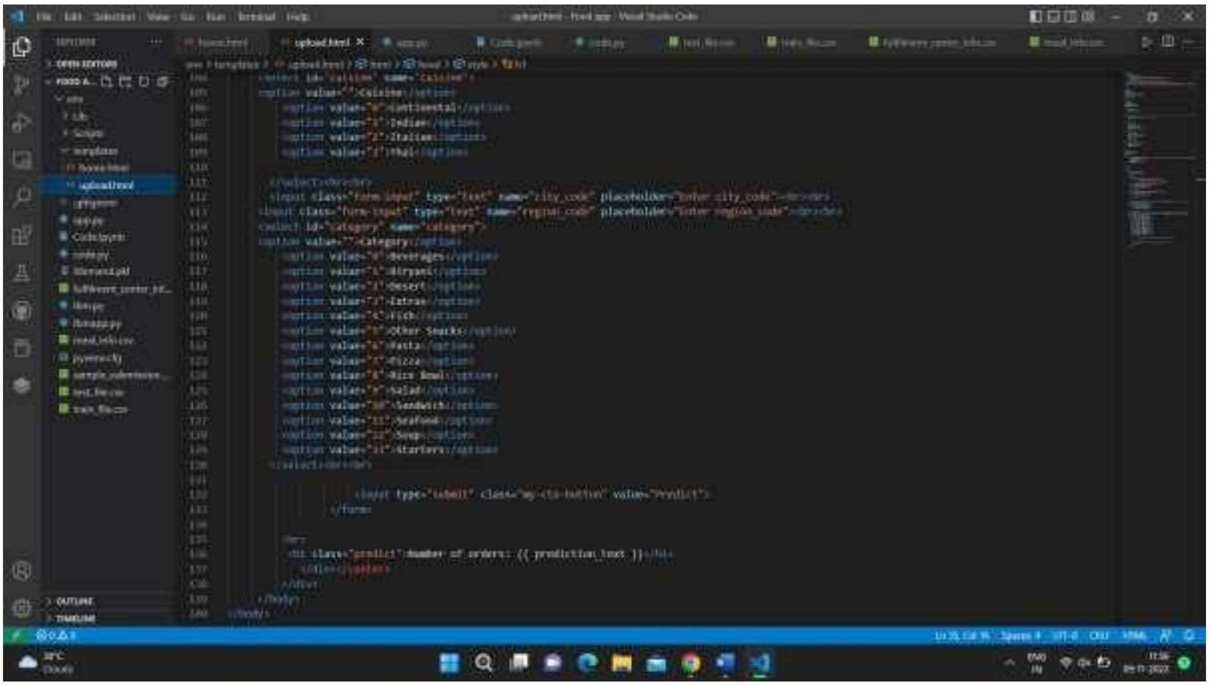
Upload.html



```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Predict</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@200;300;400;500;600&display=swap" rel="stylesheet">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css">

<style>
.bar
{
margin: 0px;
padding: 15px;
background-color:transparent;
/* opacity:0.6; */
font-family: 'Poppins',sans-serif;
font-size:25px;
}
a
{
color: #fff;
float:right;
text-decoration:none;
padding-right:20px;
}
a:hover{
    padding: 3.5px;
    background: #FAAC42;

}
h1{
    color: white;
    font-family:'Times New Roman', Times, serif;
    font-size:30;
```

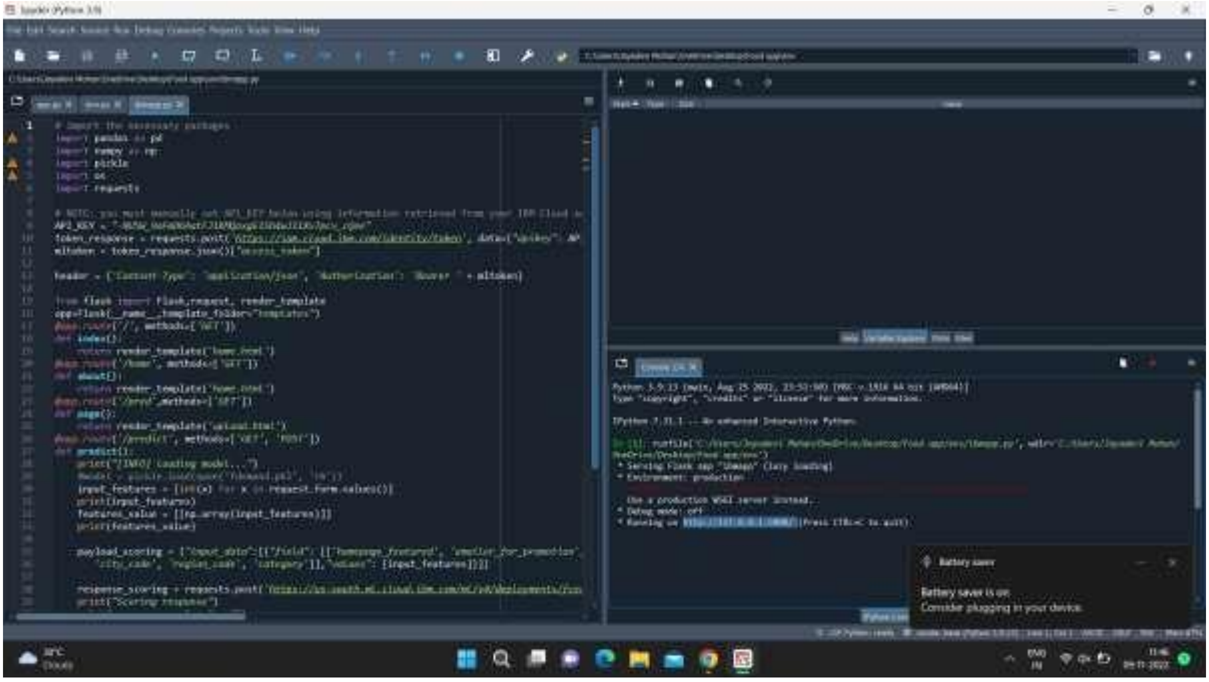


```
h2{
    color: red;
    font-family: Poppins;
    font-size:60;
    margin-bottom: 10px;
    text-shadow:3px 3px whitesmoke;
}
.my-cta-button{

    font-size: 20px;
    color: rgb(15, 15, 15);
    border: 1px solid #[illegible];
    padding: 3.5px;

    cursor: pointer;
}
.my-cta-button:hover{
    border: 2px solid #f1ae42;
    padding: 3.5px;
    background: #FAAC42;
}
p
{
color: white;
font-family: Poppins;
font-size:30px;
}
body{
        background-image: url("https://static.livebooks.com/[illegible]
        background-repeat: no-repeat;
        background-size: cover;
        height: 600px;
        background-position: center;

    }
</style>
</head>
```



```
<select id="cuisine" name="cuisine">
<option value="">Cuisine</option>
    <option value="0">Continental</option>
    <option value="1">Indian</option>
    <option value="2">Italian</option>
    <option value="3">Thai</option>

  </select><br><br>
  <input class="form-input" type="text" name="city_code" placeholder="Enter city_code"><br><br>
<input class="form-input" type="text" name="region_code" placeholder="Enter region_code"><br><br>
<select id="category" name="category">
<option value="">Category</option>
    <option value="0">Beverages</option>
    <option value="1">Biryani</option>
    <option value="2">Desert</option>
    <option value="3">Extras</option>
    <option value="4">Fish</option>
    <option value="5">Other Snacks</option>
    <option value="6">Pasta</option>
    <option value="7">Pizza</option>
    <option value="8">Rice Bowl</option>
    <option value="9">Salad</option>
    <option value="10">Sandwich</option>
    <option value="11">Seafood</option>
    <option value="12">Soup</option>
    <option value="13">Starters</option>
  </select><br><br>

            <input type="submit" class="my-cta-button" value="Predict">
          </form>

    <br>
    <h1 class="predict">Number of orders: {{ prediction_text }}</h1>
      <!--</center>
    </div>
  </body>
</body>
```

ibmapp.py



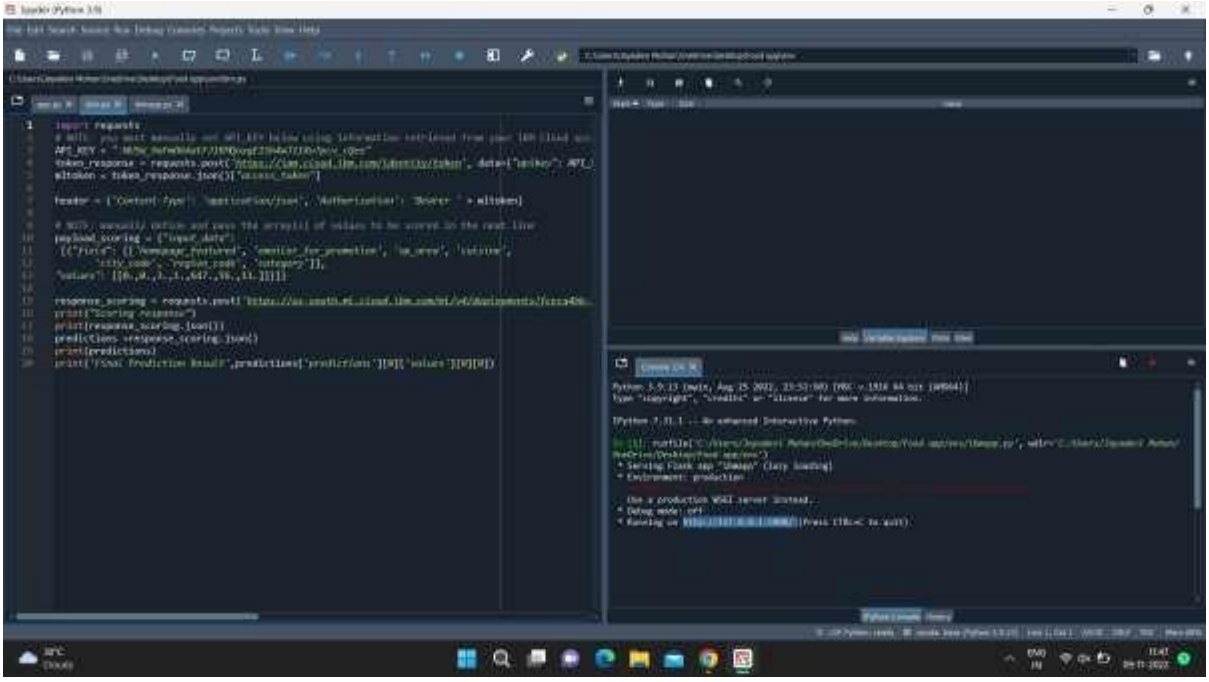
```
# import the necessary packages
import pandas as pd
import numpy as np
import pickle
import os
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud a
API_KEY = "[illegible]"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": AP
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

from flask import Flask,request, render_template
app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] Loading model...")
    #model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [int(x) for x in request.form.values()]
    print(input_features)
    features_value = [[np.array(input_features)]]
    print(features_value)

    payload_scoring = {"input_data":[{"field": [['homepage_featured', 'emailer_for_promotion',
        'city_code', 'region_code', 'category']],"values": [input_features]}]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/foo
    print("Scoring response")
```

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Jayadevi Mohan/OneDrive/Desktop/food app/env/ibmapp.py', wdir='C:/Users/Jayadevi Mohan/OneDrive/Desktop/food app/env')
 * Serving Flask app "ibmapp" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Battery saver

Battery saver is on.
Consider plugging in your device.

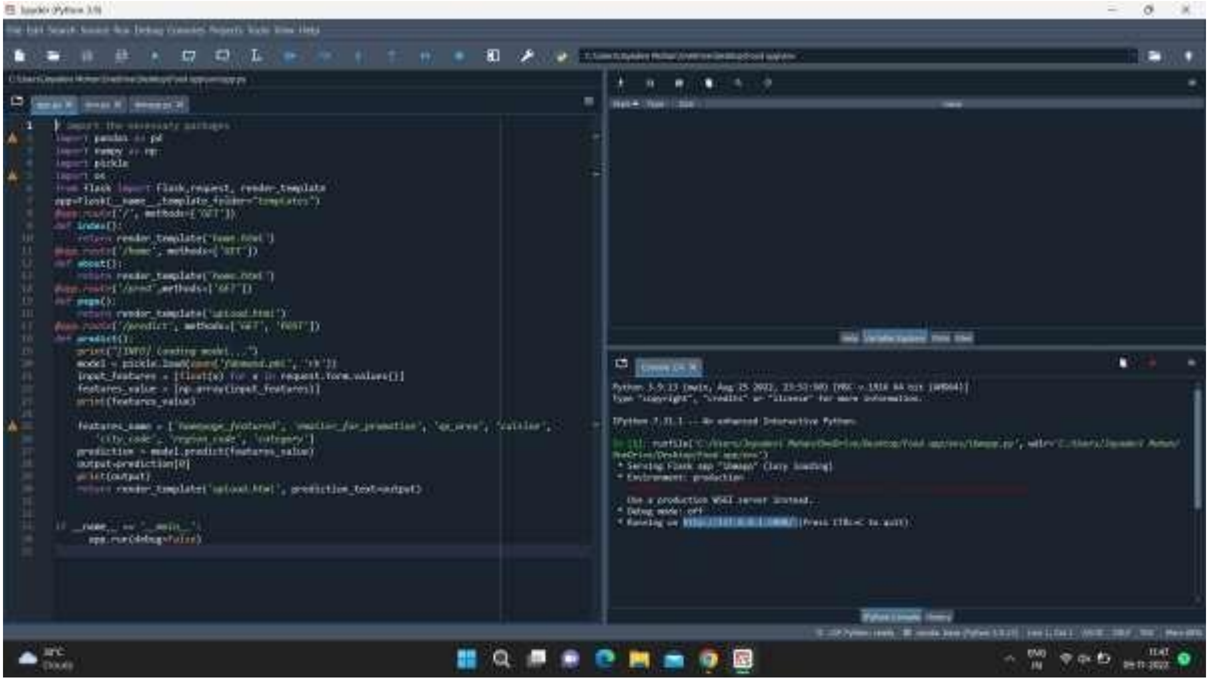
ibm.py



```
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud acc
API_KEY = "[illegible]"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data":
 [{"field": [['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
        'city_code', 'region_code', 'category']],
 "values": [[0.,0.,1.,1.,647.,56.,11.]]}]}

response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/[illegible]
print("Scoring response")
print(response_scoring.json())
predictions =response_scoring.json()
print(predictions)
print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
```

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Jayadev Mohan/OneDrive/Desktop/Food app/env/ibmapp.py', wdir='C:/Users/Jayadev Mohan/
OneDrive/Desktop/Food app/env')
 * Serving Flask app "ibmapp" (lazy loading)
 * Environment: production
   [illegible]
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

app.py



```
# import the necessary packages
import pandas as pd
import numpy as np
import pickle
import os
from flask import Flask,request, render_template
app=Flask(__name__,template_folder="templates")
@app.route('/', methods=['GET'])
def index():
    return render_template('home.html')
@app.route('/home', methods=['GET'])
def about():
    return render_template('home.html')
@app.route('/pred',methods=['GET'])
def page():
    return render_template('upload.html')
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    print("[INFO] loading model...")
    model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)

    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
       'city_code', 'region_code', 'category']
    prediction = model.predict(features_value)
    output=prediction[0]
    print(output)
    return render_template('upload.html', prediction_text=output)


if __name__ == '__main__':
    app.run(debug=False)
```

```
Python 3.9.13 (main, Aug 25 2022, 23:51:50) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.31.1 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Jayadev Mohan/OneDrive/Desktop/Food app/env/ibmapp.py', wdir='C:/Users/Jayadev Mohan/
OneDrive/Desktop/Food app/env')
 * Serving Flask app "ibmapp" (lazy loading)
 * Environment: production
   [illegible]
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```