Deployment Using Flask

Name: Judith Chepngetich

Batch Code: LISUM04

Submission Date: 29th November 2021

Submitted To: URL submitted on canvas portal under week 4 Assignment

Introduction

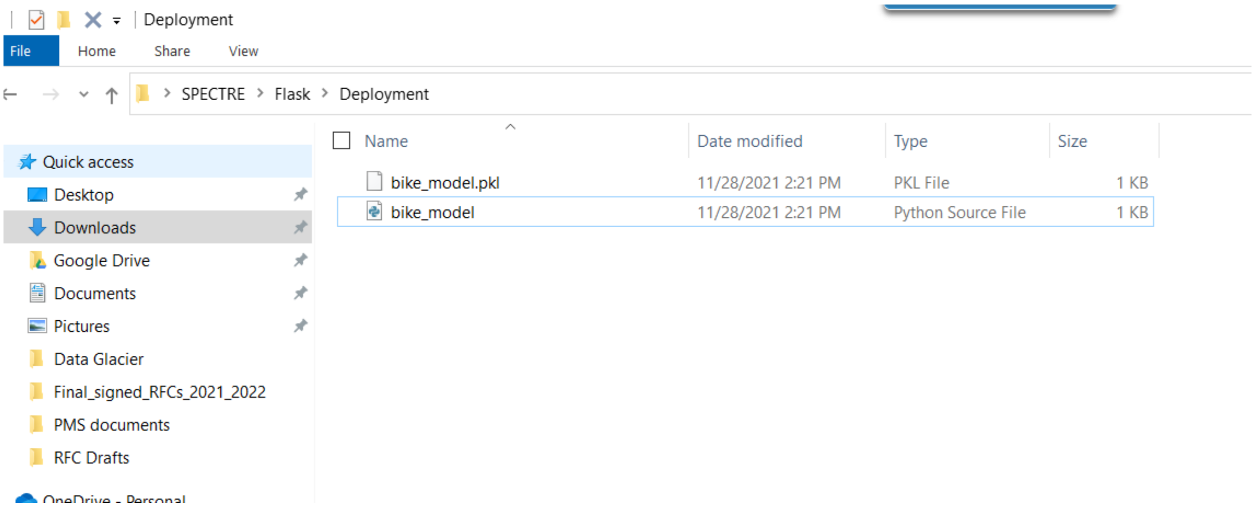
This document outlines the steps taken to deploy a machine learning model using flask. The dataset used for this model is the obtained from the link https://github.com/cambridgecoding/machinelearningregression/blob/master/data/bikes.csv.

The packages required are:

- Flask
- Pandas
- SKlearn
- Numpy
- Pickle

Step 1: Building the model

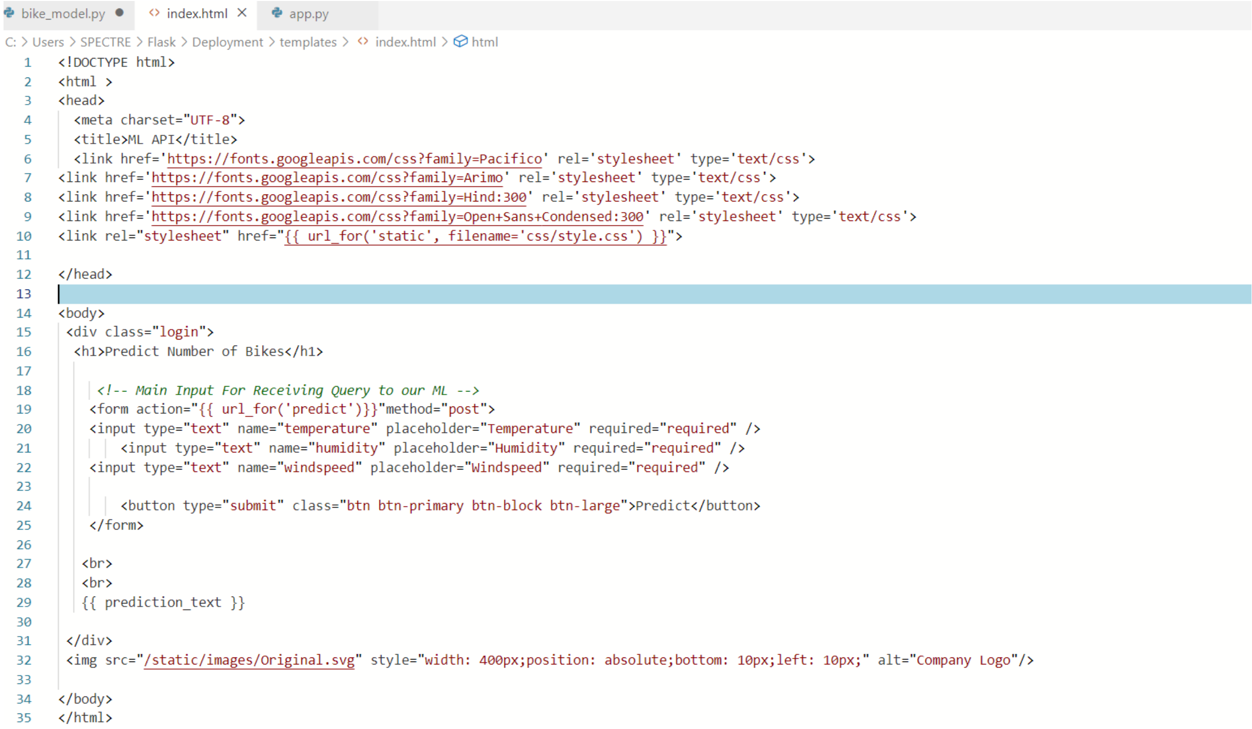
Three features i.e. temperature, humidity and wind speed are used to predict the number of bikes that will be rented out. The dataset is trained using regression and the resulting model saved as a pickle file as below:



Name	Date modified	Type	Size
bike_model.pkl	11/28/2021 2:21 PM	PKL File	1 KB
bike_model	11/28/2021 2:21 PM	Python Source File	1 KB

Step 2: Templates

This step involves creation of HTML templates that flask will render to take inputs from a user and also provide output after prediction. Below is the index.html file:

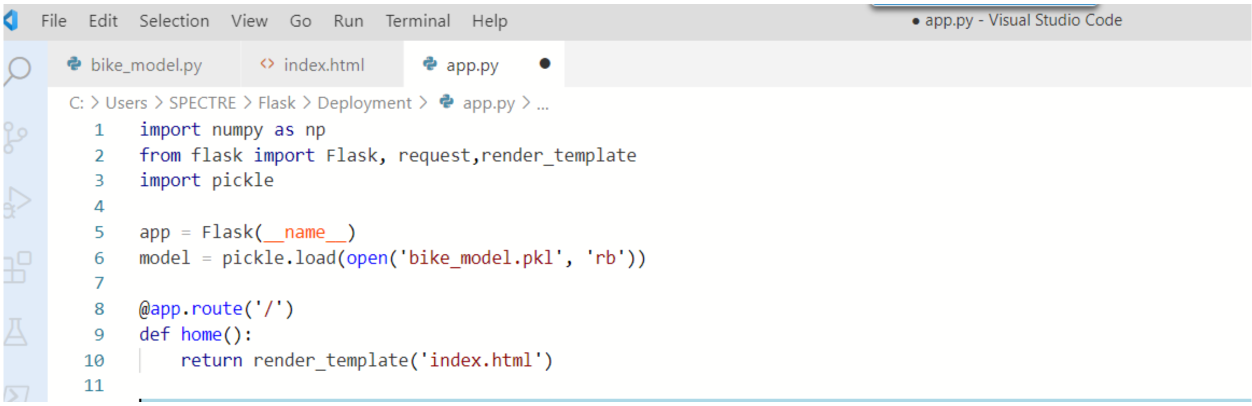


bike_model.py ●
index.html ×
app.py
C: > Users > SPECTRE > Flask > Deployment > templates > index.html > html

```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
  <h1>Predict Number of Bikes</h1>

     <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
    <input type="text" name="temperature" placeholder="Temperature" required="required" />
        <input type="text" name="humidity" placeholder="Humidity" required="required" />
    <input type="text" name="windspeed" placeholder="Windspeed" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>
 <img src="/static/images/Original.svg" style="width: 400px;position: absolute;bottom: 10px;left: 10px;" alt="Company Logo"/>

</body>
</html>
```

Step 3. The application

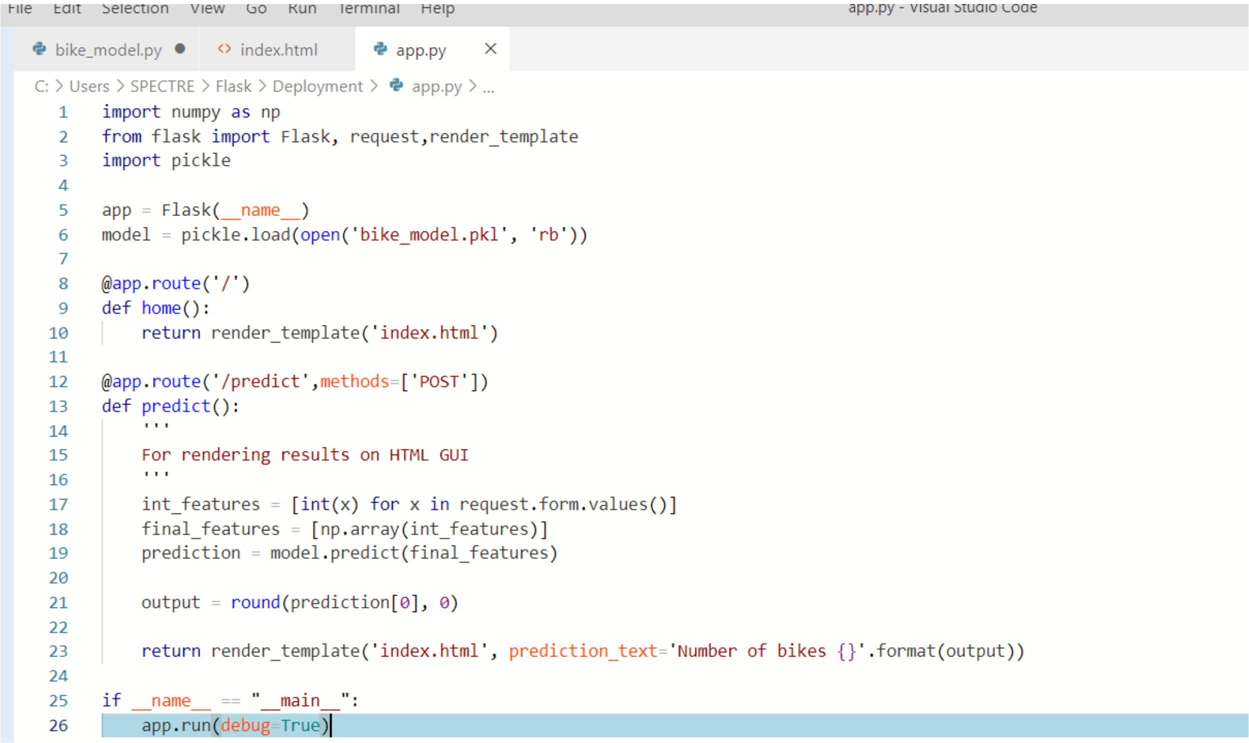
An application is created that will run and retrieve input from the forms, process the data using the model, and then provide feedback/output back to the user. Flask routes and renders a template as provided by index.html file. The home function that renders the main page is as shown below:



File
Edit
Selection
View
Go
Run
Terminal
Help
● app.py - Visual Studio Code
bike_model.py
index.html
app.py ●
C: > Users > SPECTRE > Flask > Deployment > app.py > ...

```python
import numpy as np
from flask import Flask, request,render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('bike_model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

```

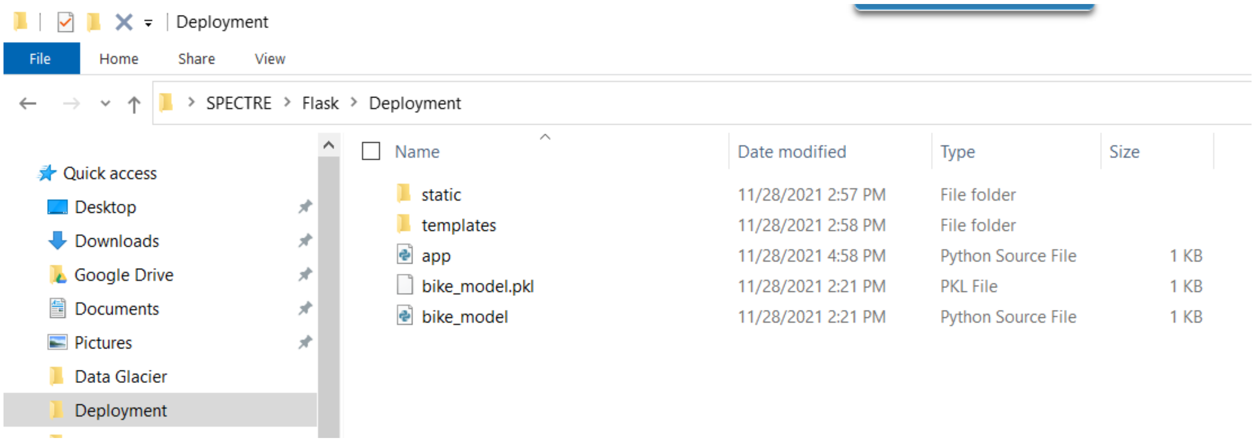
Flask also provides a routing for prediction, whereby it receives a post request from a user and uses the model to process the data, predict and render results on the URL. The predict function that handles this is as shown below:


File Edit Selection View Go Run Terminal Help
app.py - Visual Studio Code
bike_model.py ●
index.html
app.py ×
C: > Users > SPECTRE > Flask > Deployment > app.py > ...

```python
import numpy as np
from flask import Flask, request,render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('bike_model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    int_features = [int(x) for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = model.predict(final_features)

    output = round(prediction[0], 0)

    return render_template('index.html', prediction_text='Number of bikes {}'.format(output))

if __name__ == "__main__":
    app.run(debug=True)
```

Step 4: Checking files

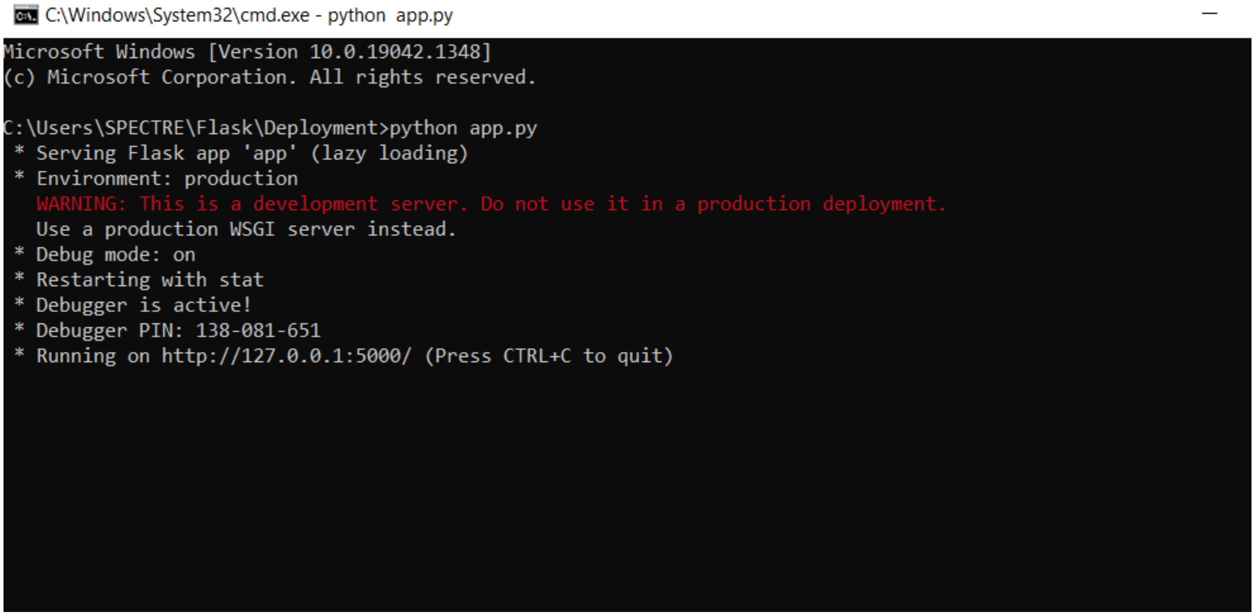
Confirming that all required files are in the folder



Name	Date modified	Type	Size
static	11/28/2021 2:57 PM	File folder	
templates	11/28/2021 2:58 PM	File folder	
app	11/28/2021 4:58 PM	Python Source File	1 KB
bike_model.pkl	11/28/2021 2:21 PM	PKL File	1 KB
bike_model	11/28/2021 2:21 PM	Python Source File	1 KB

Step 5: Running the application

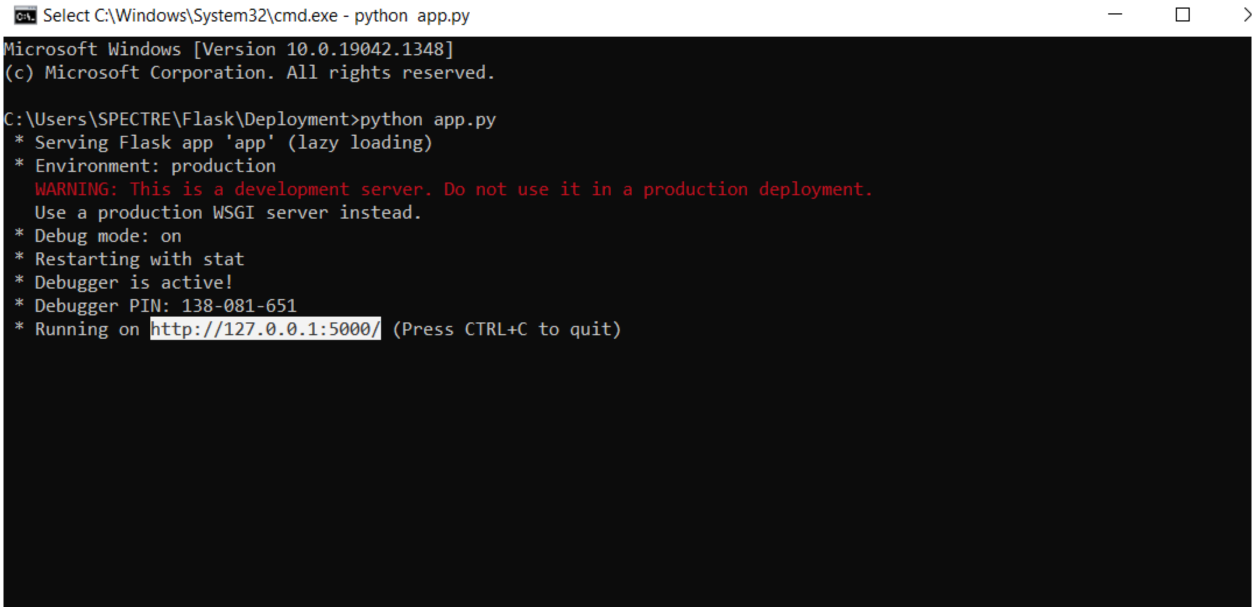
Running the app.py file from the command prompt gives the below output. This shows that the application is running.



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SPECTRE\Flask\Deployment>python app.py
 * Serving Flask app 'app' (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: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 138-081-651
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

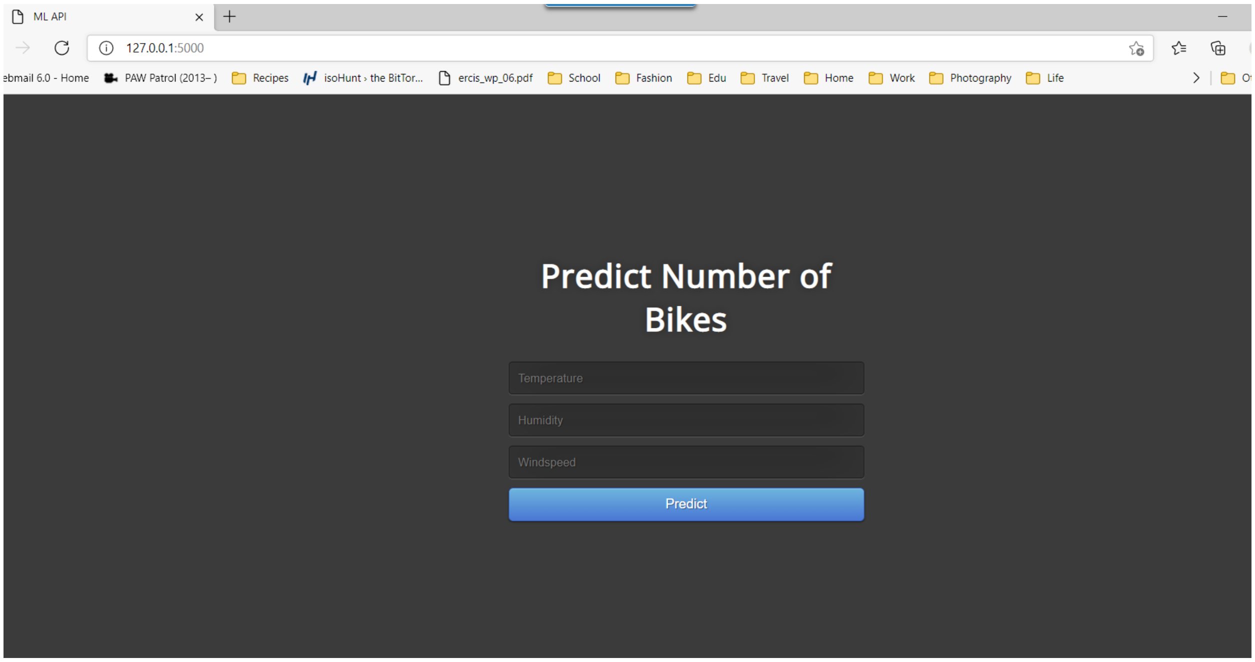
Step 6: Browser

Copy the URL provided and access it through a browser



```
Microsoft Windows [Version 10.0.19042.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SPECTRE\Flask\Deployment>python app.py
 * Serving Flask app 'app' (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: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 138-081-651
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

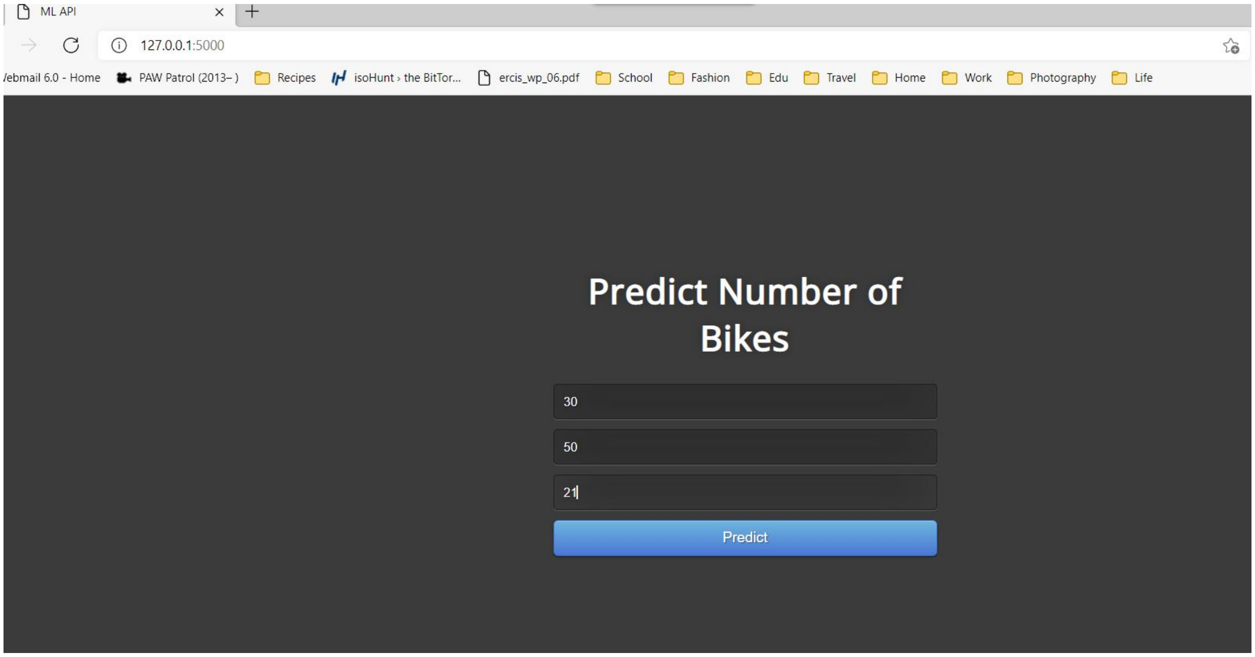
The below screen is displayed, which takes in input of temperature, humidity and windspeed.



Predict Number of Bikes

Step 7: Input

Filling in a sample of the required features as 30, 50 and 21 for temperature, humidity and windspeed respectively,



Predict Number of Bikes

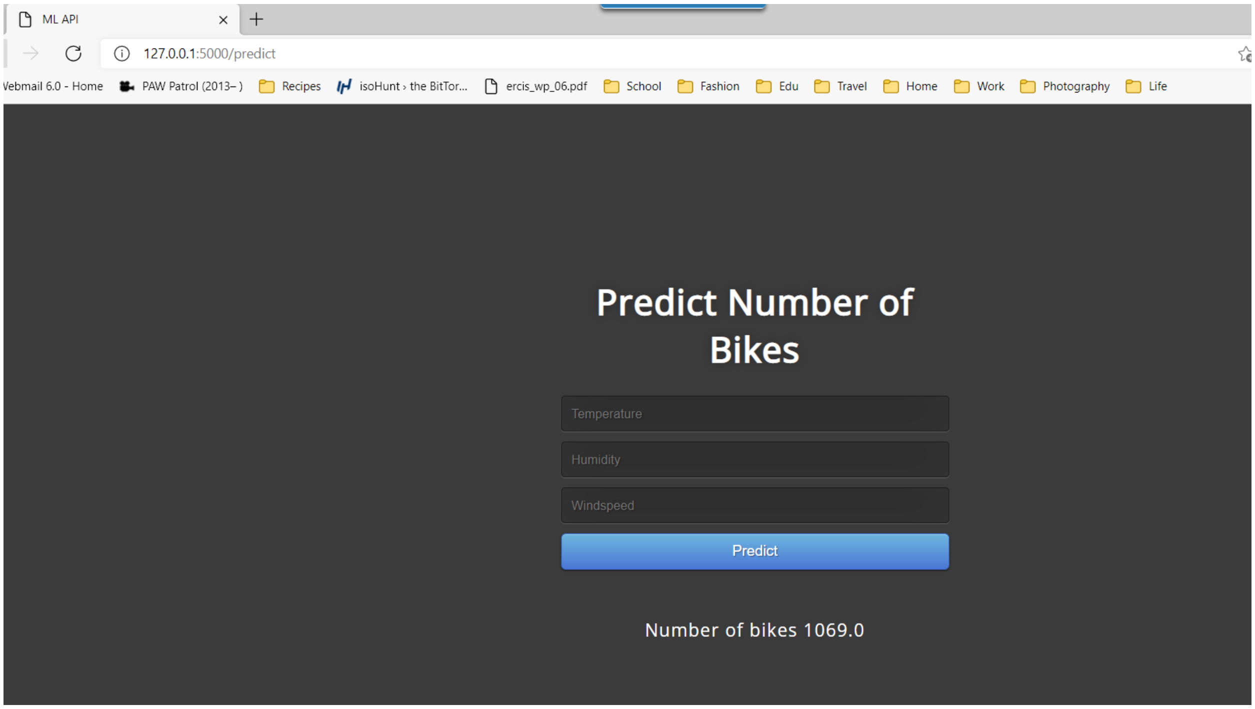
30

50

21

Step 8: Prediction

On clicking predict, the number of bikes is predicted at 1069 as shown in the below figure:



Predict Number of Bikes

Number of bikes 1069.0

This shows that the model was deployed successfully.

--------------------------------------End---