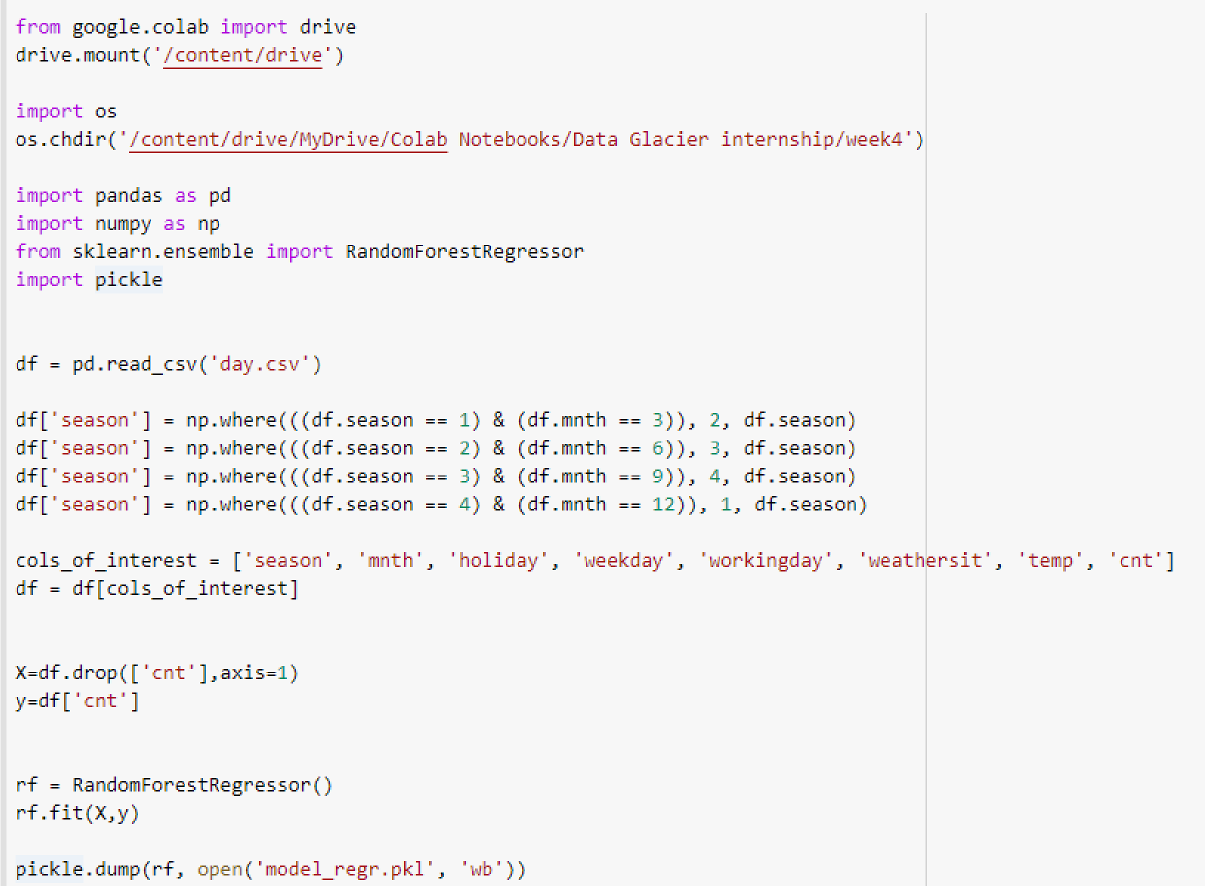
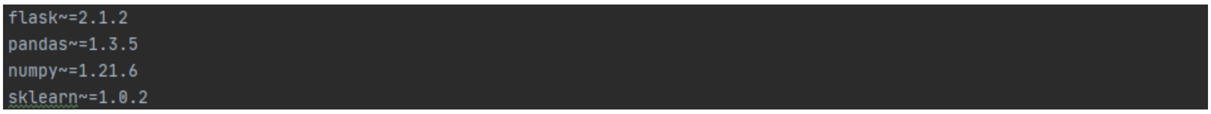
Name: Kseniia Nosenko
Submission date: 23.06.2022
Submitted to Github https://github.com/KseniyaLem/Internship/tree/main/Week4

1. The model was created and saved. Bike rental data was selected and slightly changed (some columns were removed so as not to overload the model and flask).



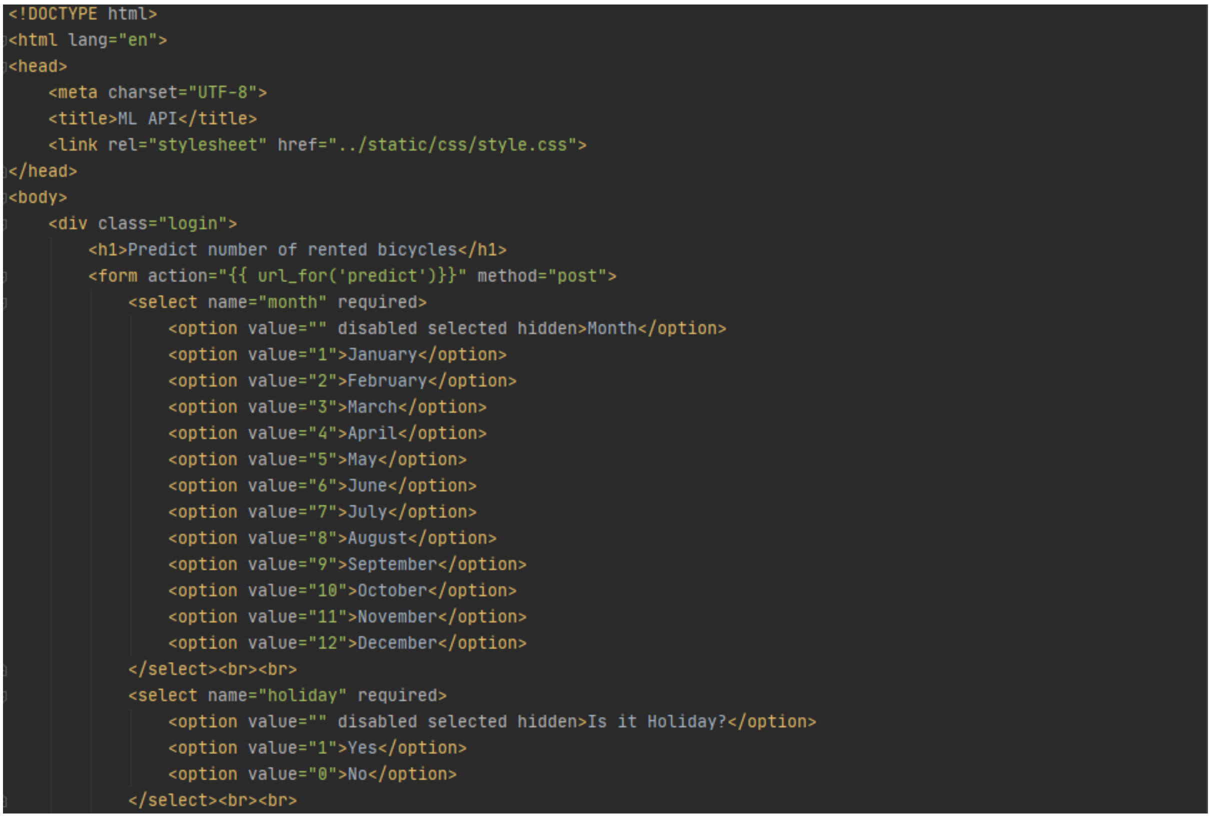
```
from google.colab import drive
drive.mount('/content/drive')

import os
os.chdir('/content/drive/MyDrive/Colab Notebooks/Data Glacier internship/week4')

import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
import pickle


df = pd.read_csv('day.csv')

df['season'] = np.where(((df.season == 1) & (df.mnth == 3)), 2, df.season)
df['season'] = np.where(((df.season == 2) & (df.mnth == 6)), 3, df.season)
df['season'] = np.where(((df.season == 3) & (df.mnth == 9)), 4, df.season)
df['season'] = np.where(((df.season == 4) & (df.mnth == 12)), 1, df.season)

cols_of_interest = ['season', 'mnth', 'holiday', 'weekday', 'workingday', 'weathersit', 'temp', 'cnt']
df = df[cols_of_interest]


X=df.drop(['cnt'],axis=1)
y=df['cnt']


rf = RandomForestRegressor()
rf.fit(X,y)

pickle.dump(rf, open('model_regr.pkl', 'wb'))
```

requirements

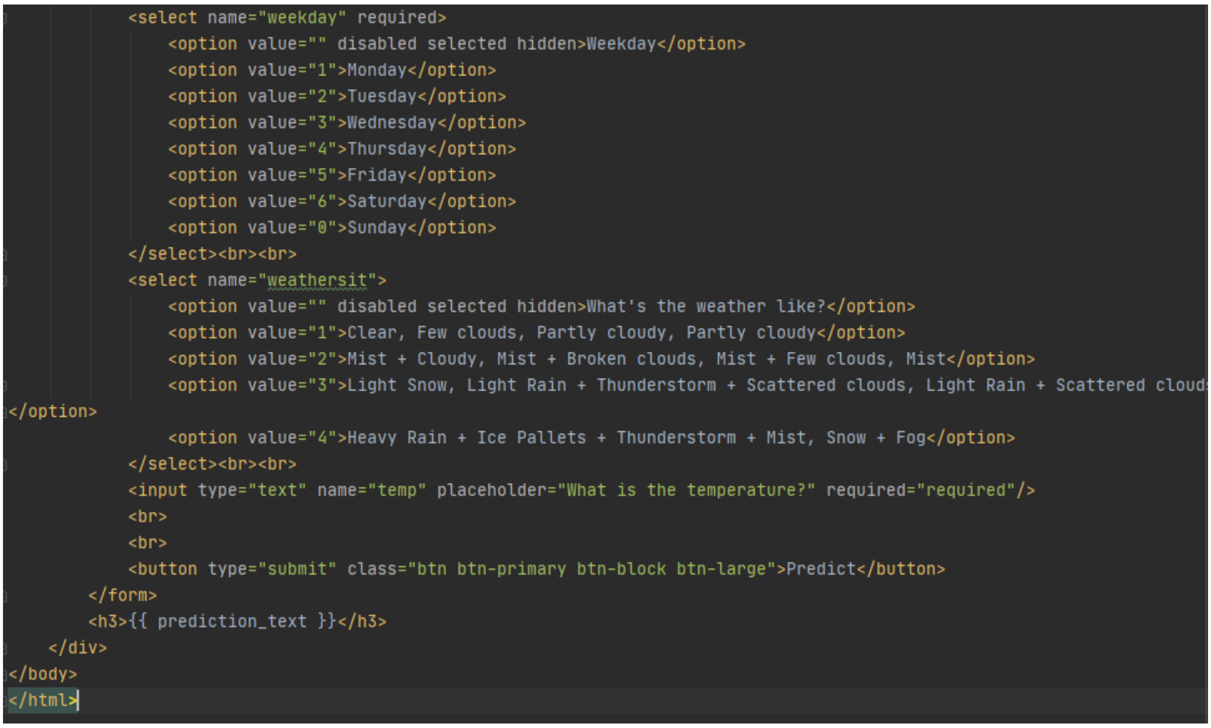


```
flask~=2.1.2
pandas~=1.3.5
numpy~=1.21.6
sklearn~=1.0.2
```

2. HTML file was created.

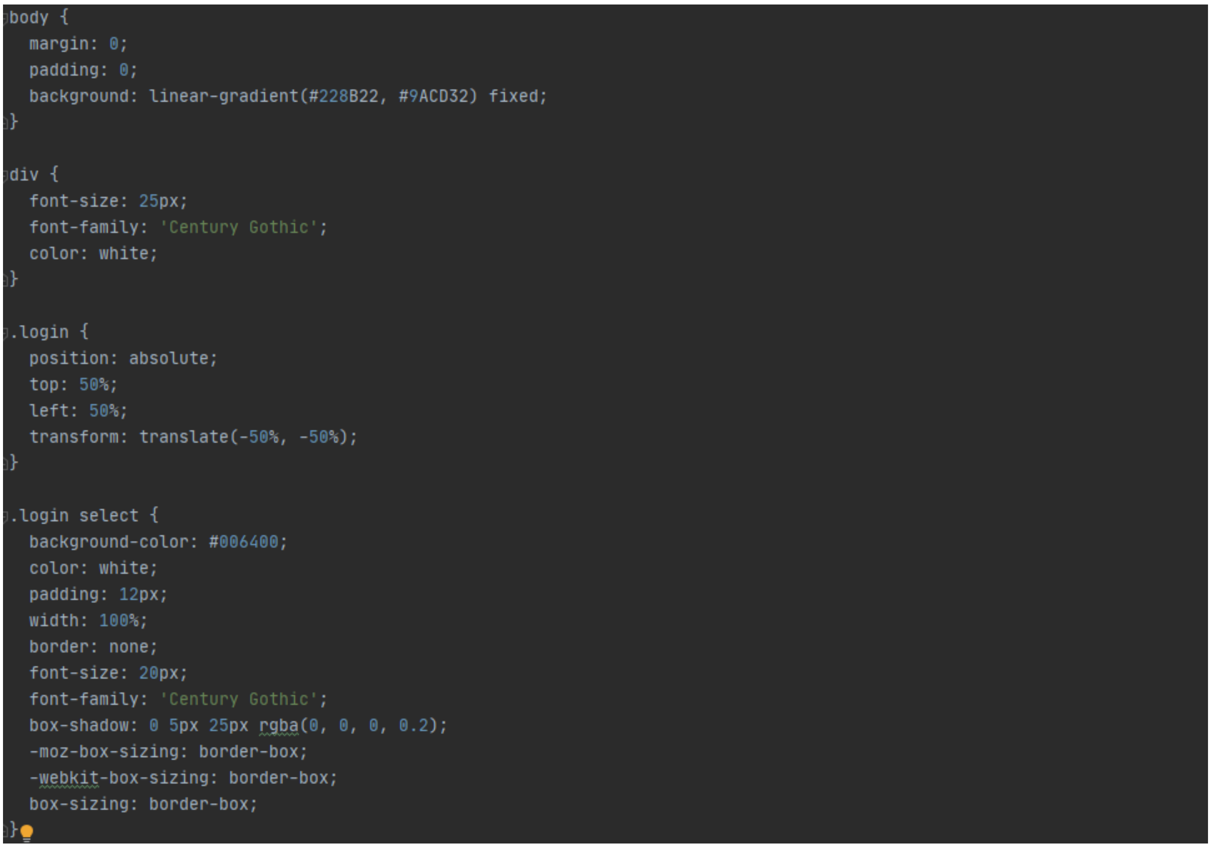


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
    <link rel="stylesheet" href="../static/css/style.css">
</head>
<body>
    <div class="login">
        <h1>Predict number of rented bicycles</h1>
        <form action="{{ url_for('predict')}}" method="post">
            <select name="month" required>
                <option value="" disabled selected hidden>Month</option>
                <option value="1">January</option>
                <option value="2">February</option>
                <option value="3">March</option>
                <option value="4">April</option>
                <option value="5">May</option>
                <option value="6">June</option>
                <option value="7">July</option>
                <option value="8">August</option>
                <option value="9">September</option>
                <option value="10">October</option>
                <option value="11">November</option>
                <option value="12">December</option>
            </select><br><br>
            <select name="holiday" required>
                <option value="" disabled selected hidden>Is it Holiday?</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select><br><br>
```

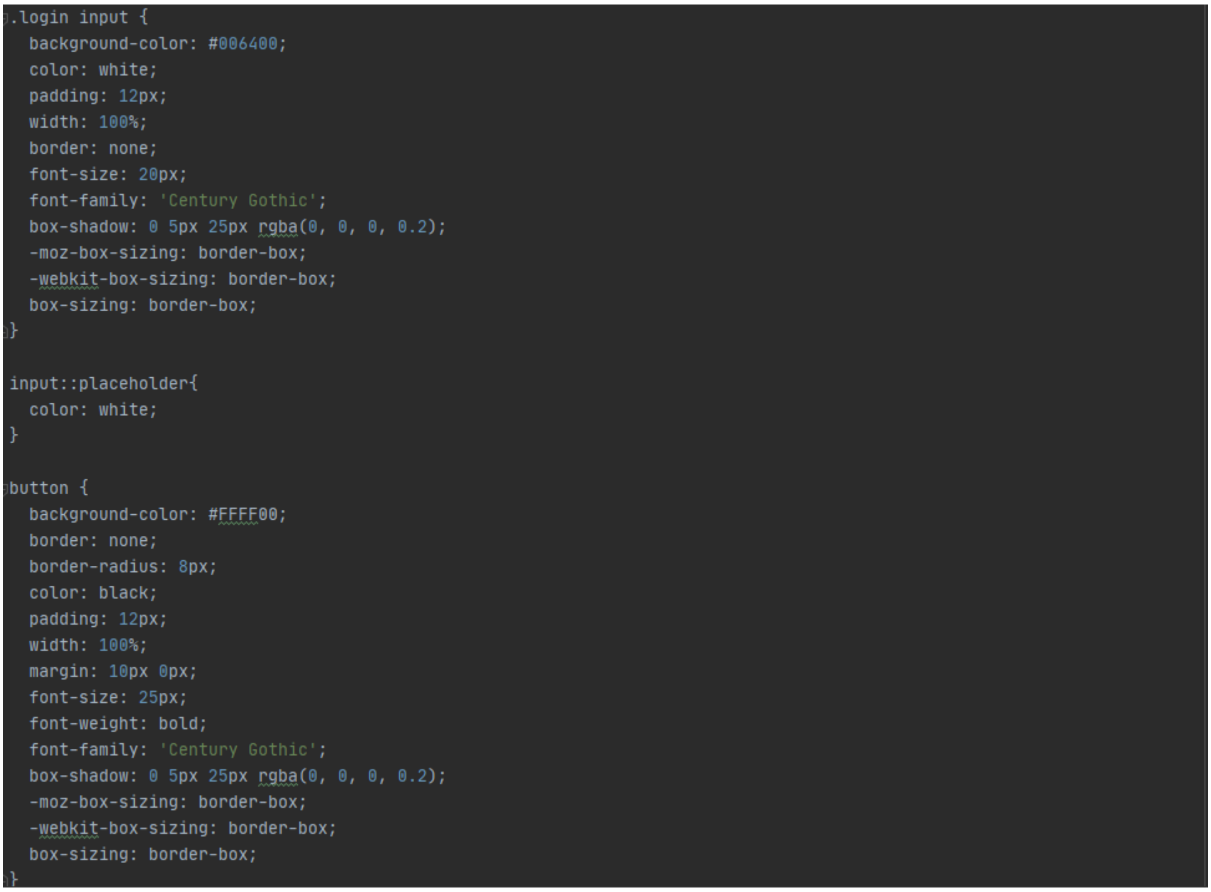


```html
            <select name="weekday" required>
                <option value="" disabled selected hidden>Weekday</option>
                <option value="1">Monday</option>
                <option value="2">Tuesday</option>
                <option value="3">Wednesday</option>
                <option value="4">Thursday</option>
                <option value="5">Friday</option>
                <option value="6">Saturday</option>
                <option value="0">Sunday</option>
            </select><br><br>
            <select name="weathersit">
                <option value="" disabled selected hidden>What's the weather like?</option>
                <option value="1">Clear, Few clouds, Partly cloudy, Partly cloudy</option>
                <option value="2">Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist</option>
                <option value="3">Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered cloud
</option>
                <option value="4">Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog</option>
            </select><br><br>
            <input type="text" name="temp" placeholder="What is the temperature?" required="required"/>
            <br>
            <br>
            <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
        </form>
        <h3>{{ prediction_text }}</h3>
    </div>
</body>
</html>
```

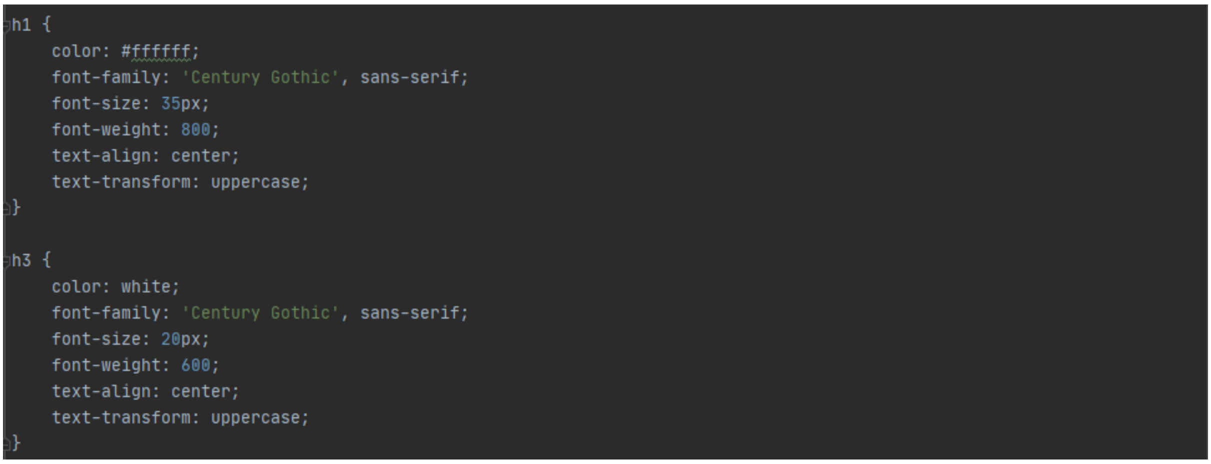
3. CSS file was created



```
body {
  margin: 0;
  padding: 0;
  background: linear-gradient(#228B22, #9ACD32) fixed;
}

div {
  font-size: 25px;
  font-family: 'Century Gothic';
  color: white;
}

.login {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.login select {
  background-color: #006400;
  color: white;
  padding: 12px;
  width: 100%;
  border: none;
  font-size: 20px;
  font-family: 'Century Gothic';
  box-shadow: 0 5px 25px rgba(0, 0, 0, 0.2);
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}
```

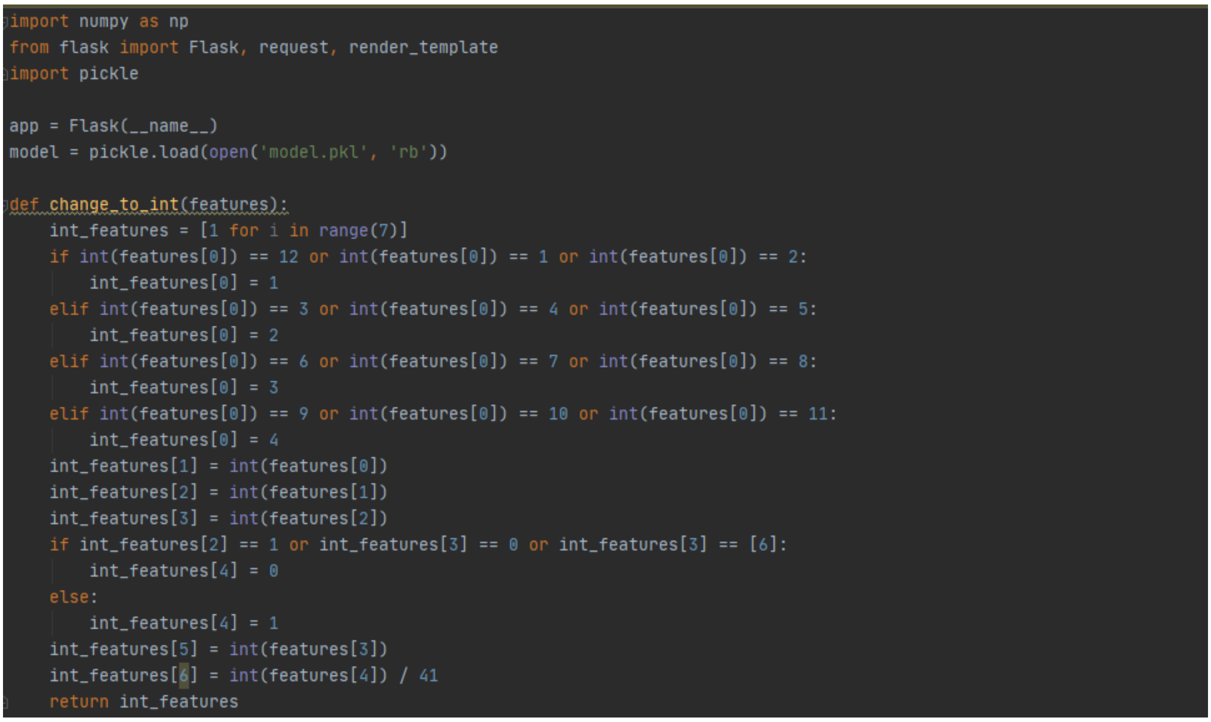


```
.login input {
  background-color: #006400;
  color: white;
  padding: 12px;
  width: 100%;
  border: none;
  font-size: 20px;
  font-family: 'Century Gothic';
  box-shadow: 0 5px 25px rgba(0, 0, 0, 0.2);
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}

input::placeholder{
  color: white;
}

button {
  background-color: #FFFF00;
  border: none;
  border-radius: 8px;
  color: black;
  padding: 12px;
  width: 100%;
  margin: 10px 0px;
  font-size: 25px;
  font-weight: bold;
  font-family: 'Century Gothic';
  box-shadow: 0 5px 25px rgba(0, 0, 0, 0.2);
  -moz-box-sizing: border-box;
  -webkit-box-sizing: border-box;
  box-sizing: border-box;
}
```

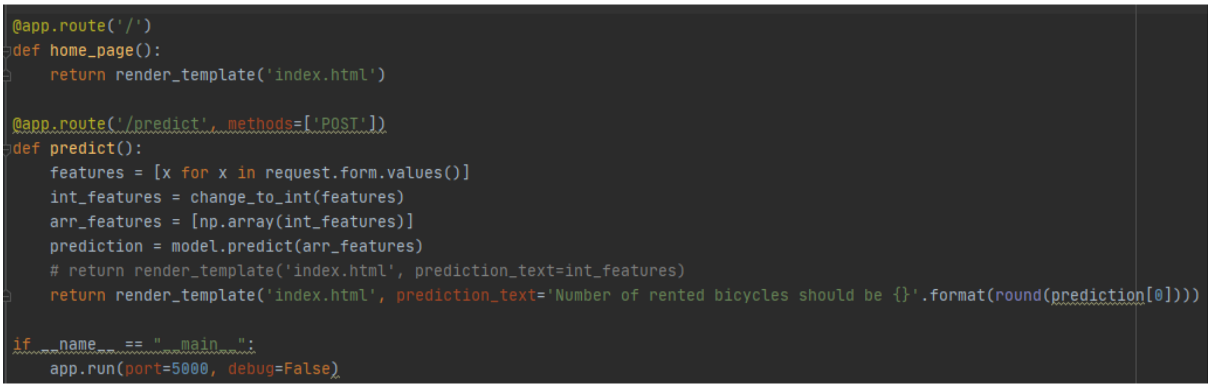


```
h1 {
    color: #ffffff;
    font-family: 'Century Gothic', sans-serif;
    font-size: 35px;
    font-weight: 800;
    text-align: center;
    text-transform: uppercase;
}

h3 {
    color: white;
    font-family: 'Century Gothic', sans-serif;
    font-size: 20px;
    font-weight: 600;
    text-align: center;
    text-transform: uppercase;
}
```

4. APP.py file was created (flask)

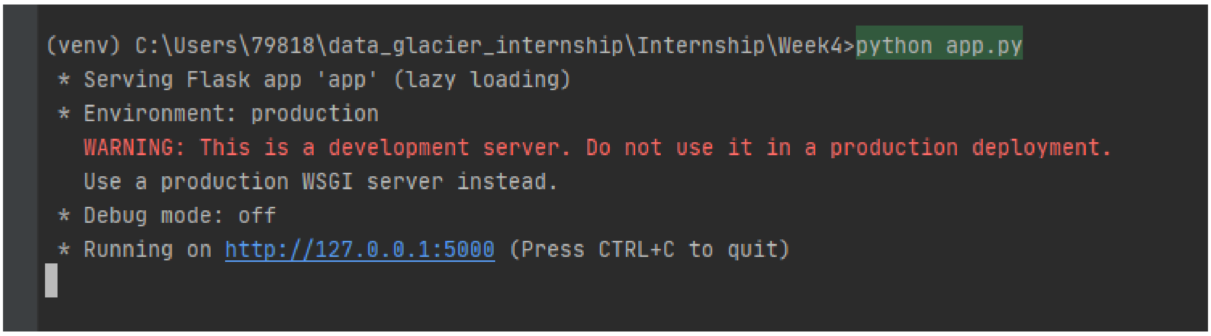


```
import numpy as np
from flask import Flask, request, render_template
import pickle

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

def change_to_int(features):
    int_features = [1 for i in range(7)]
    if int(features[0]) == 12 or int(features[0]) == 1 or int(features[0]) == 2:
        int_features[0] = 1
    elif int(features[0]) == 3 or int(features[0]) == 4 or int(features[0]) == 5:
        int_features[0] = 2
    elif int(features[0]) == 6 or int(features[0]) == 7 or int(features[0]) == 8:
        int_features[0] = 3
    elif int(features[0]) == 9 or int(features[0]) == 10 or int(features[0]) == 11:
        int_features[0] = 4
    int_features[1] = int(features[0])
    int_features[2] = int(features[1])
    int_features[3] = int(features[2])
    if int_features[2] == 1 or int_features[3] == 0 or int_features[3] == [6]:
        int_features[4] = 0
    else:
        int_features[4] = 1
    int_features[5] = int(features[3])
    int_features[6] = int(features[4]) / 41
    return int_features
```

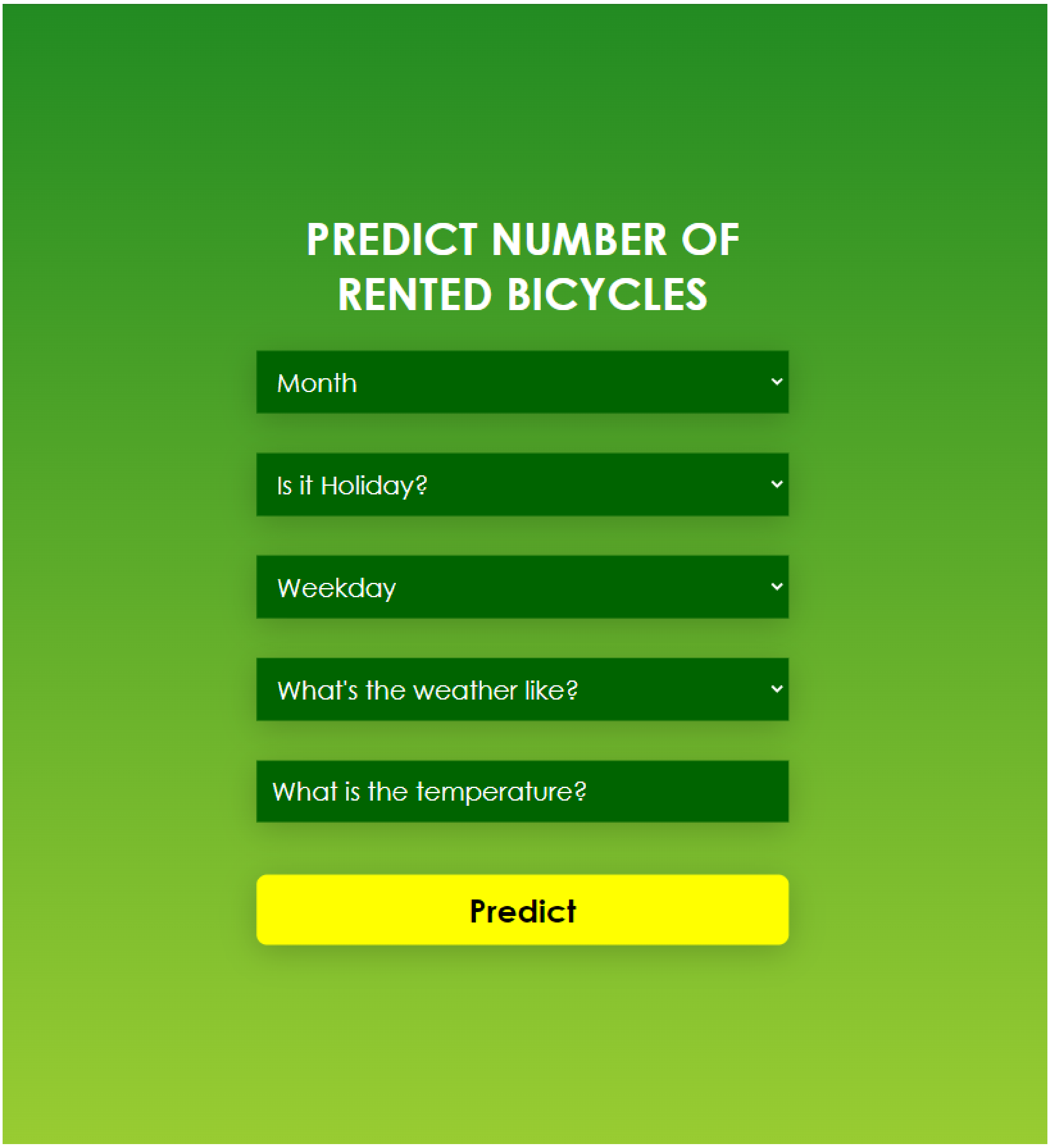


```
@app.route('/')
def home_page():
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    features = [x for x in request.form.values()]
    int_features = change_to_int(features)
    arr_features = [np.array(int_features)]
    prediction = model.predict(arr_features)
    # return render_template('index.html', prediction_text=int_features)
    return render_template('index.html', prediction_text='Number of rented bicycles should be {}'.format(round(prediction[0])))

if __name__ == "__main__":
    app.run(port=5000, debug=False)
```

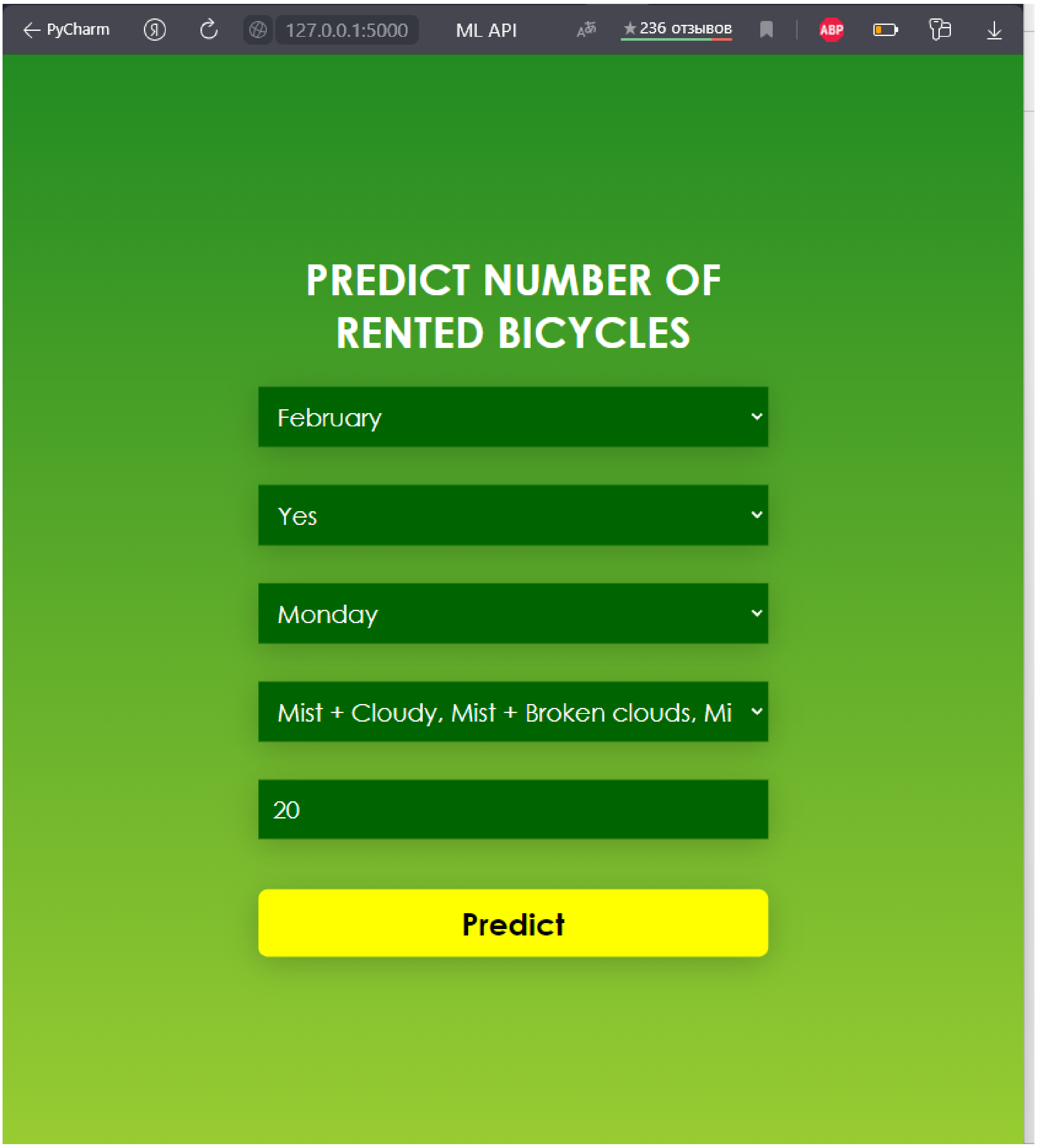
5. Check the work of flask



```
(venv) C:\Users\79818\data_glacier_internship\Internship\Week4>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: off
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
```



PREDICT NUMBER OF RENTED BICYCLES



PREDICT NUMBER OF RENTED BICYCLES

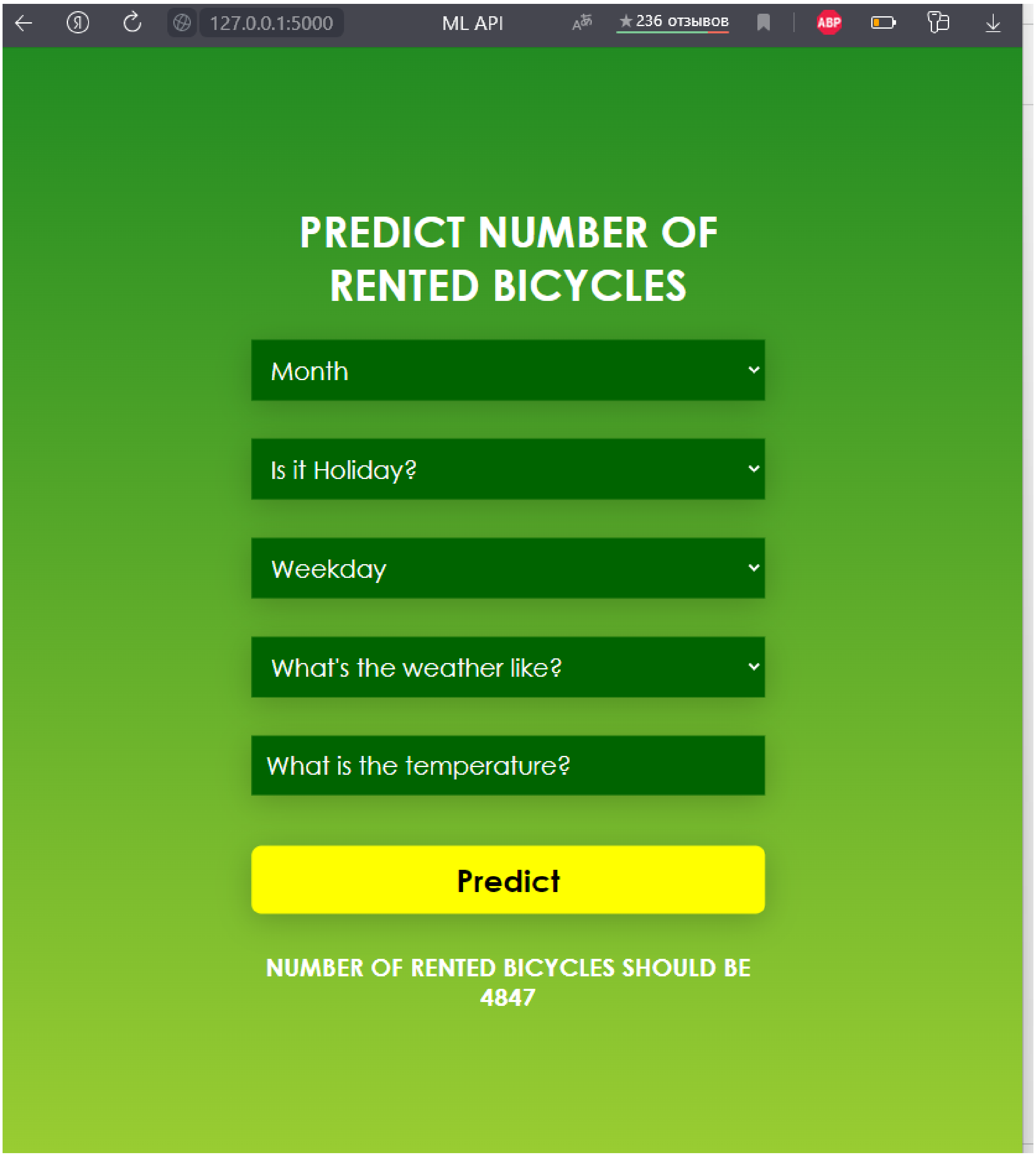
February

Yes

Monday

Mist + Cloudy, Mist + Broken clouds, Mi

20

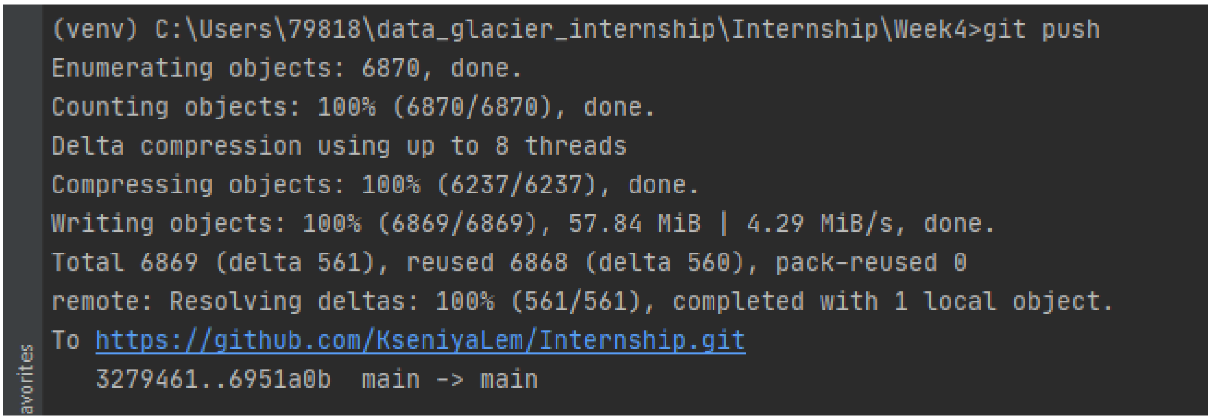


PREDICT NUMBER OF RENTED BICYCLES

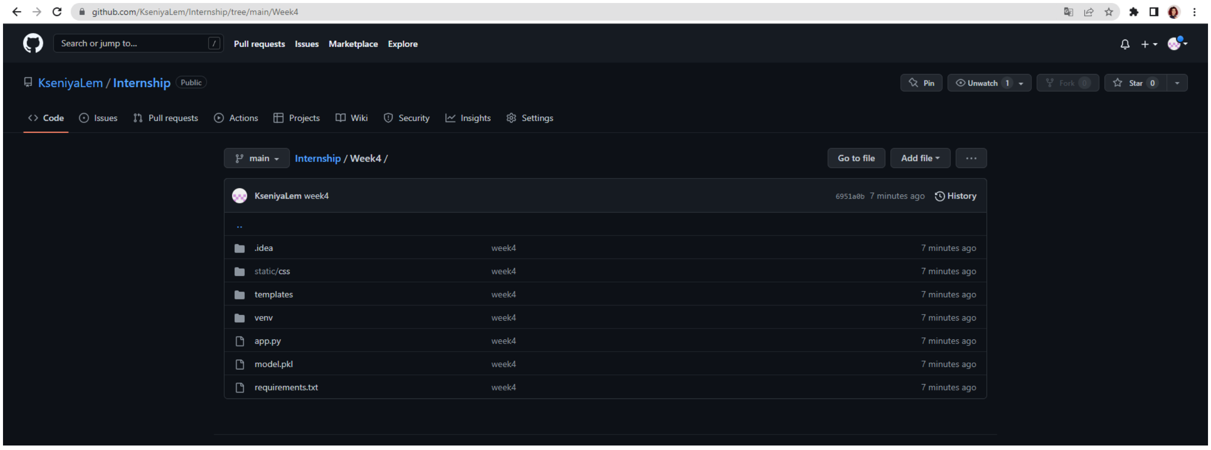
NUMBER OF RENTED BICYCLES SHOULD BE 4847

6. submitted to Github: https://github.com/KseniyaLem/Internship/tree/main/Week4

```
git add *
git commit -a -m week4
git push
```



```
(venv) C:\Users\79818\data_glacier_internship\Internship\Week4>git push
Enumerating objects: 6870, done.
Counting objects: 100% (6870/6870), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6237/6237), done.
Writing objects: 100% (6869/6869), 57.84 MiB | 4.29 MiB/s, done.
Total 6869 (delta 561), reused 6868 (delta 560), pack-reused 0
remote: Resolving deltas: 100% (561/561), completed with 1 local object.
To https://github.com/KseniyaLem/Internship.git
   3279461..6951a0b  main -> main
```



KseniyaLem week4 6951a0b 7 minutes ago

..		
.idea	week4	7 minutes ago
static/css	week4	7 minutes ago
templates	week4	7 minutes ago
venv	week4	7 minutes ago
app.py	week4	7 minutes ago
model.pkl	week4	7 minutes ago
requirements.txt	week4	7 minutes ago