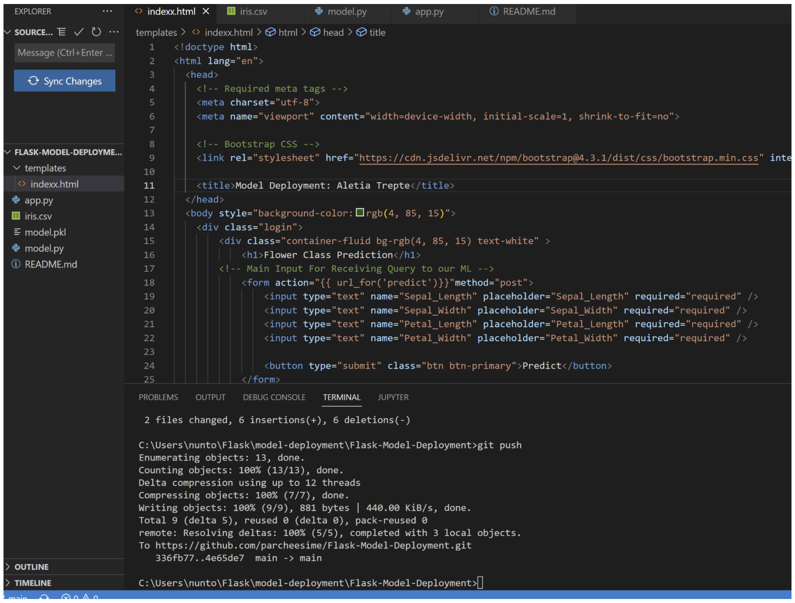
Deploying a Model

Name: Aletia Trepte
Batch: LISUM11: 30
Date: 07/28/2022
To: Glacier Data

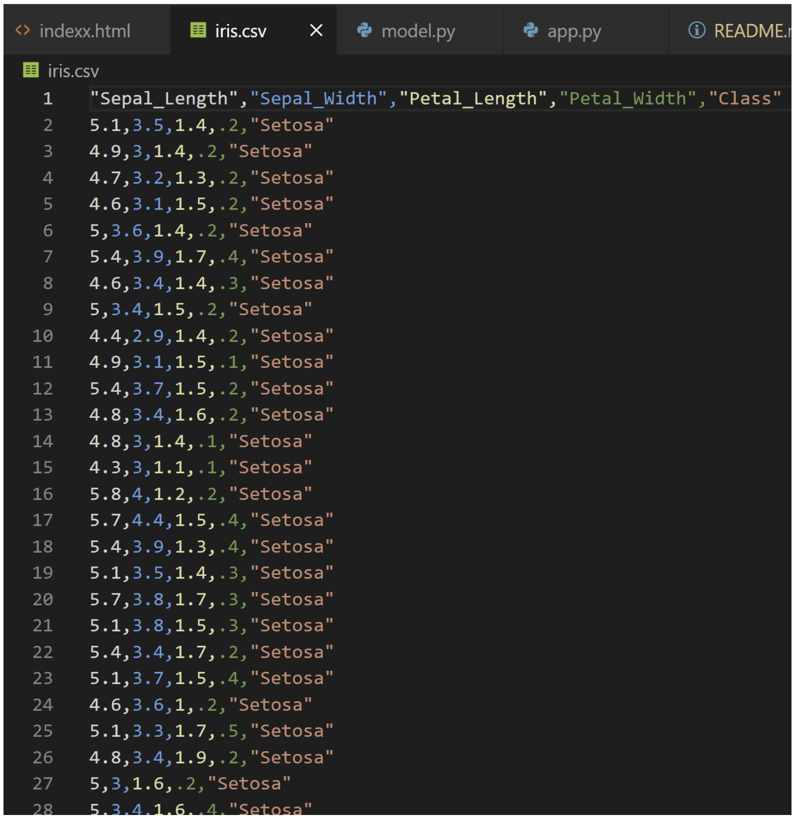
Readme file



```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" inte

    <title>Model Deployment: Aletia Trepte</title>
  </head>
  <body style="background-color:rgb(4, 85, 15)">
    <div class="login">
        <div class="container-fluid bg-rgb(4, 85, 15) text-white" >
            <h1>Flower Class Prediction</h1>
        <!-- Main Input For Receiving Query to our ML -->
            <form action="{{ url_for('predict')}}"method="post">
                <input type="text" name="Sepal_Length" placeholder="Sepal_Length" required="required" />
                <input type="text" name="Sepal_Width" placeholder="Sepal_Width" required="required" />
                <input type="text" name="Petal_Length" placeholder="Petal_Length" required="required" />
                <input type="text" name="Petal_Width" placeholder="Petal_Width" required="required" />

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

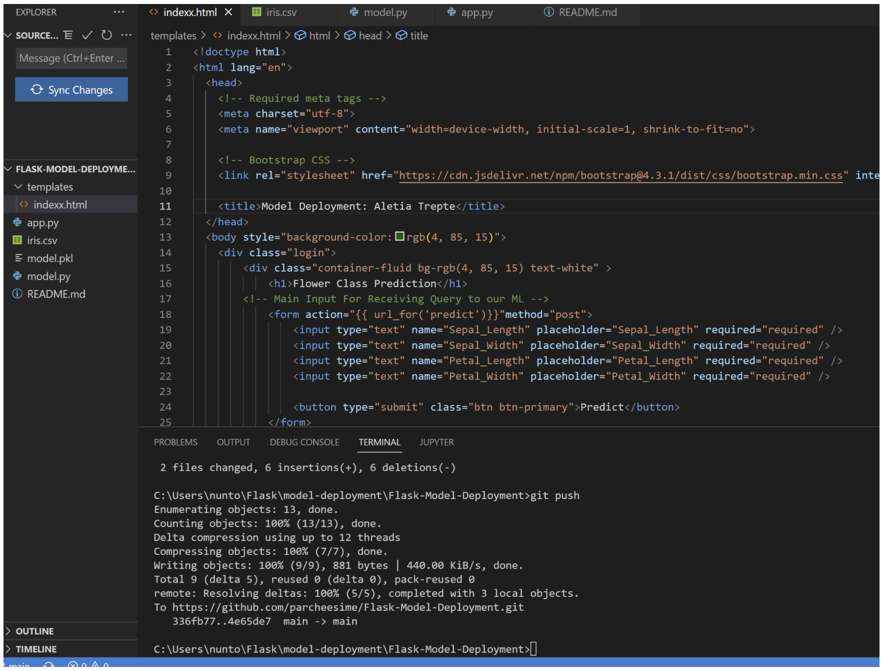
```
 2 files changed, 6 insertions(+), 6 deletions(-)

C:\Users\nunto\Flask\model-deployment\Flask-Model-Deployment>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 881 bytes | 440.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
To https://github.com/parcheesime/Flask-Model-Deployment.git
   336fb77..4e65de7  main -> main

C:\Users\nunto\Flask\model-deployment\Flask-Model-Deployment>
```

Iris dataset



```
"Sepal_Length","Sepal_Width","Petal_Length","Petal_Width","Class"
5.1,3.5,1.4,.2,"Setosa"
4.9,3,1.4,.2,"Setosa"
4.7,3.2,1.3,.2,"Setosa"
4.6,3.1,1.5,.2,"Setosa"
5,3.6,1.4,.2,"Setosa"
5.4,3.9,1.7,.4,"Setosa"
4.6,3.4,1.4,.3,"Setosa"
5,3.4,1.5,.2,"Setosa"
4.4,2.9,1.4,.2,"Setosa"
4.9,3.1,1.5,.1,"Setosa"
5.4,3.7,1.5,.2,"Setosa"
4.8,3.4,1.6,.2,"Setosa"
4.8,3,1.4,.1,"Setosa"
4.3,3,1.1,.1,"Setosa"
5.8,4,1.2,.2,"Setosa"
5.7,4.4,1.5,.4,"Setosa"
5.4,3.9,1.3,.4,"Setosa"
5.1,3.5,1.4,.3,"Setosa"
5.7,3.8,1.7,.3,"Setosa"
5.1,3.8,1.5,.3,"Setosa"
5.4,3.4,1.7,.2,"Setosa"
5.1,3.7,1.5,.4,"Setosa"
4.6,3.6,1,.2,"Setosa"
5.1,3.3,1.7,.5,"Setosa"
4.8,3.4,1.9,.2,"Setosa"
5,3,1.6,.2,"Setosa"
5,3.4,1.6,.4,"Setosa"
```

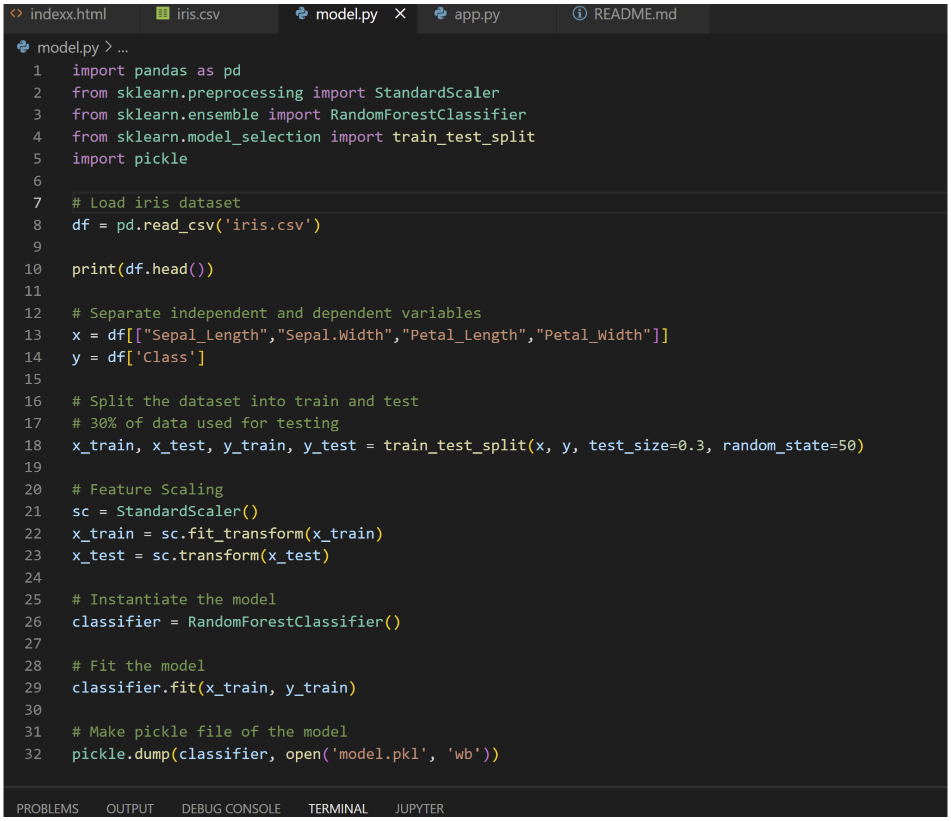
Index.html



```html
<!doctype html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css" inte

    <title>Model Deployment: Aletia Trepte</title>
  </head>
  <body style="background-color: rgb(4, 85, 15)">
    <div class="login">
      <div class="container-fluid bg-rgb(4, 85, 15) text-white" >
        <h1>Flower Class Prediction</h1>
      <!-- Main Input For Receiving Query to our ML -->
        <form action="{{ url_for('predict')}}"method="post">
          <input type="text" name="Sepal_Length" placeholder="Sepal_Length" required="required" />
          <input type="text" name="Sepal_Width" placeholder="Sepal_Width" required="required" />
          <input type="text" name="Petal_Length" placeholder="Petal_Length" required="required" />
          <input type="text" name="Petal_Width" placeholder="Petal_Width" required="required" />

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

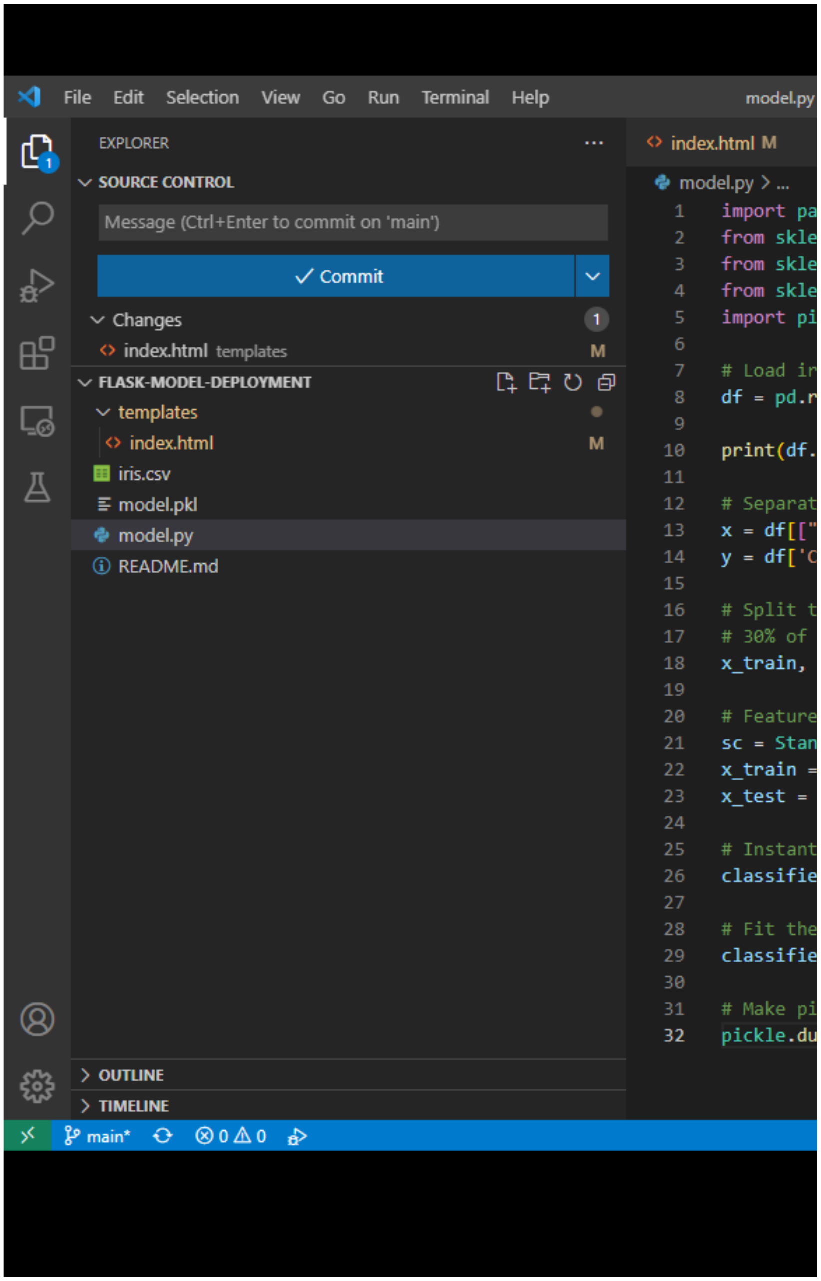
```
 2 files changed, 6 insertions(+), 6 deletions(-)

C:\Users\nunto\Flask\model-deployment\Flask-Model-Deployment>git push
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (9/9), 881 bytes | 440.00 KiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 3 local objects.
To https://github.com/parcheesime/Flask-Model-Deployment.git
   336fb77..4e65de7  main -> main

C:\Users\nunto\Flask\model-deployment\Flask-Model-Deployment>
```

model.py



```python
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
import pickle

# Load iris dataset
df = pd.read_csv('iris.csv')

print(df.head())

# Separate independent and dependent variables
x = df[["Sepal_Length","Sepal.Width","Petal_Length","Petal_Width"]]
y = df['Class']

# Split the dataset into train and test
# 30% of data used for testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=50)

# Feature Scaling
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

# Instantiate the model
classifier = RandomForestClassifier()

# Fit the model
classifier.fit(x_train, y_train)

# Make pickle file of the model
pickle.dump(classifier, open('model.pkl', 'wb'))
```

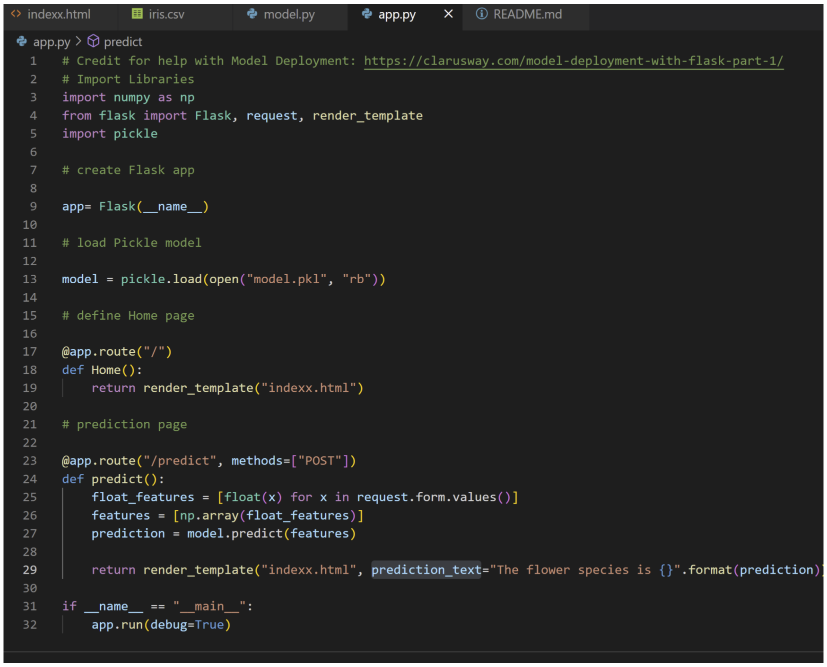
Pickle File



```
import pa
from skle
from skle
from skle
import pi

# Load ir
df = pd.r

print(df.

# Separat
x = df[["
y = df['C

# Split t
# 30% of
x_train,

# Feature
sc = Stan
x_train =
x_test =

# Instant
classifie

# Fit the
classifie

# Make pi
pickle.du
```

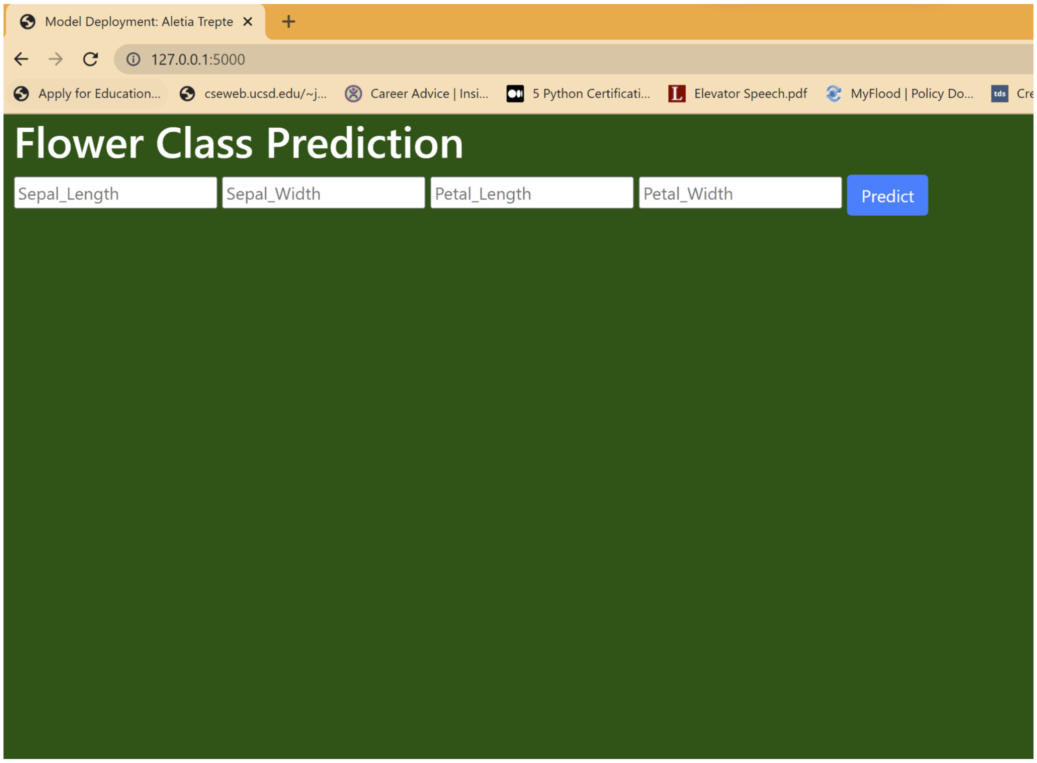
Deploying the model

app.py

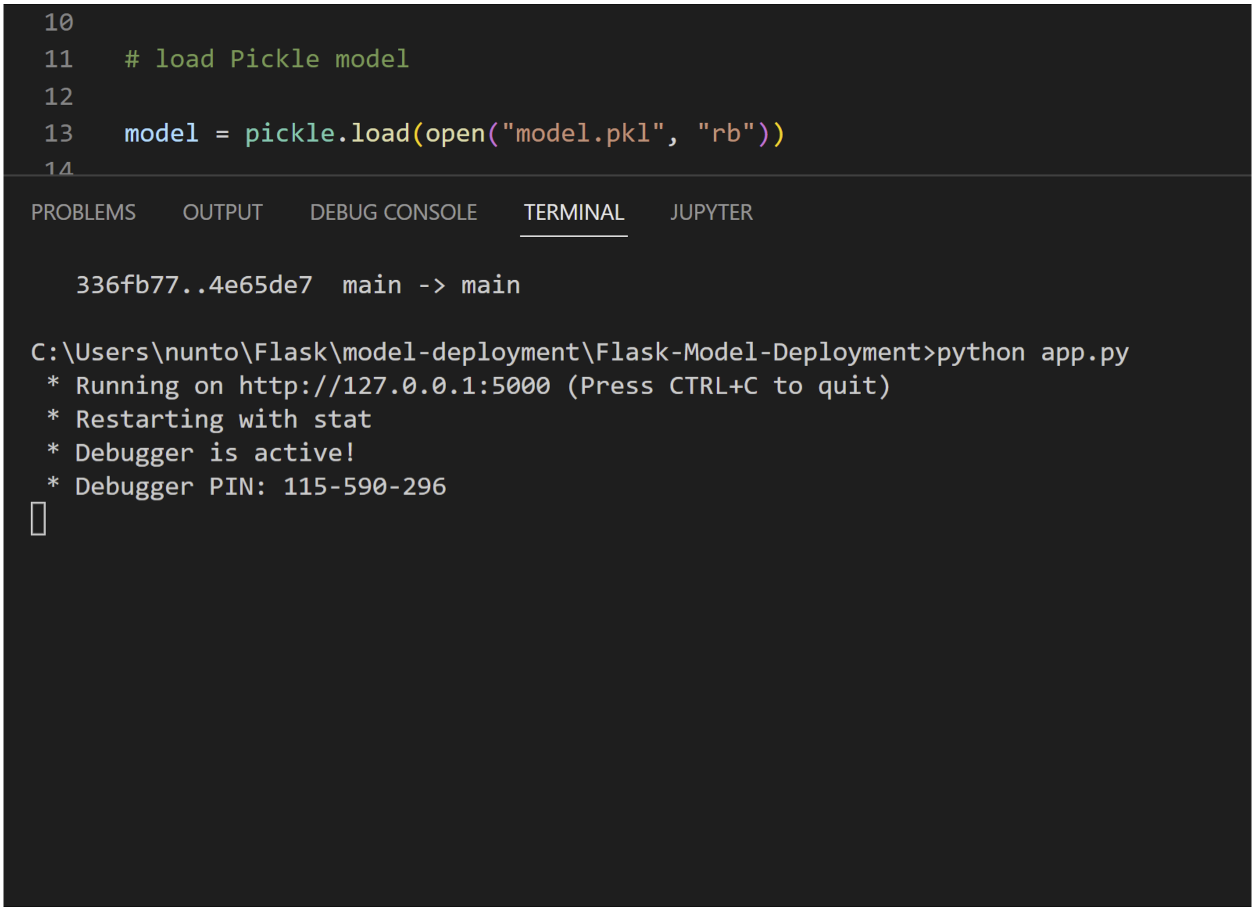


```
# Credit for help with Model Deployment: https://clarusway.com/model-deployment-with-flask-part-1/
# Import Libraries
import numpy as np
from flask import Flask, request, render_template
import pickle

# create Flask app

app= Flask(__name__)

# load Pickle model

model = pickle.load(open("model.pkl", "rb"))

# define Home page

@app.route("/")
def Home():
    return render_template("indexx.html")

# prediction page

@app.route("/predict", methods=["POST"])
def predict():
    float_features = [float(x) for x in request.form.values()]
    features = [np.array(float_features)]
    prediction = model.predict(features)

    return render_template("indexx.html", prediction_text="The flower species is {}".format(prediction)

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

Browser



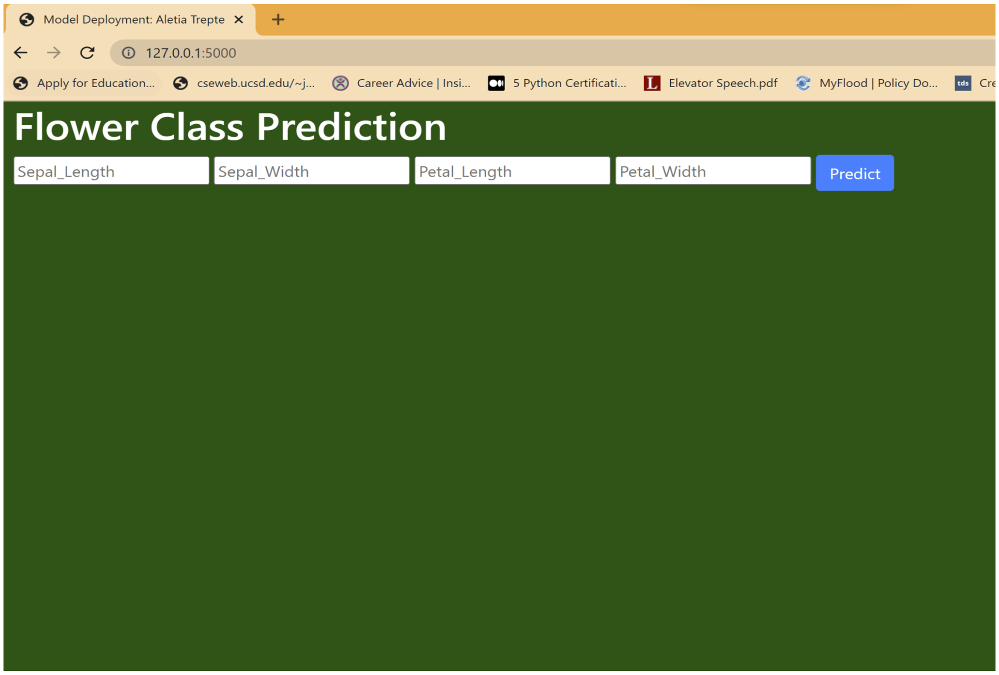
Run app.py



```
# load Pickle model

model = pickle.load(open("model.pkl", "rb"))
```

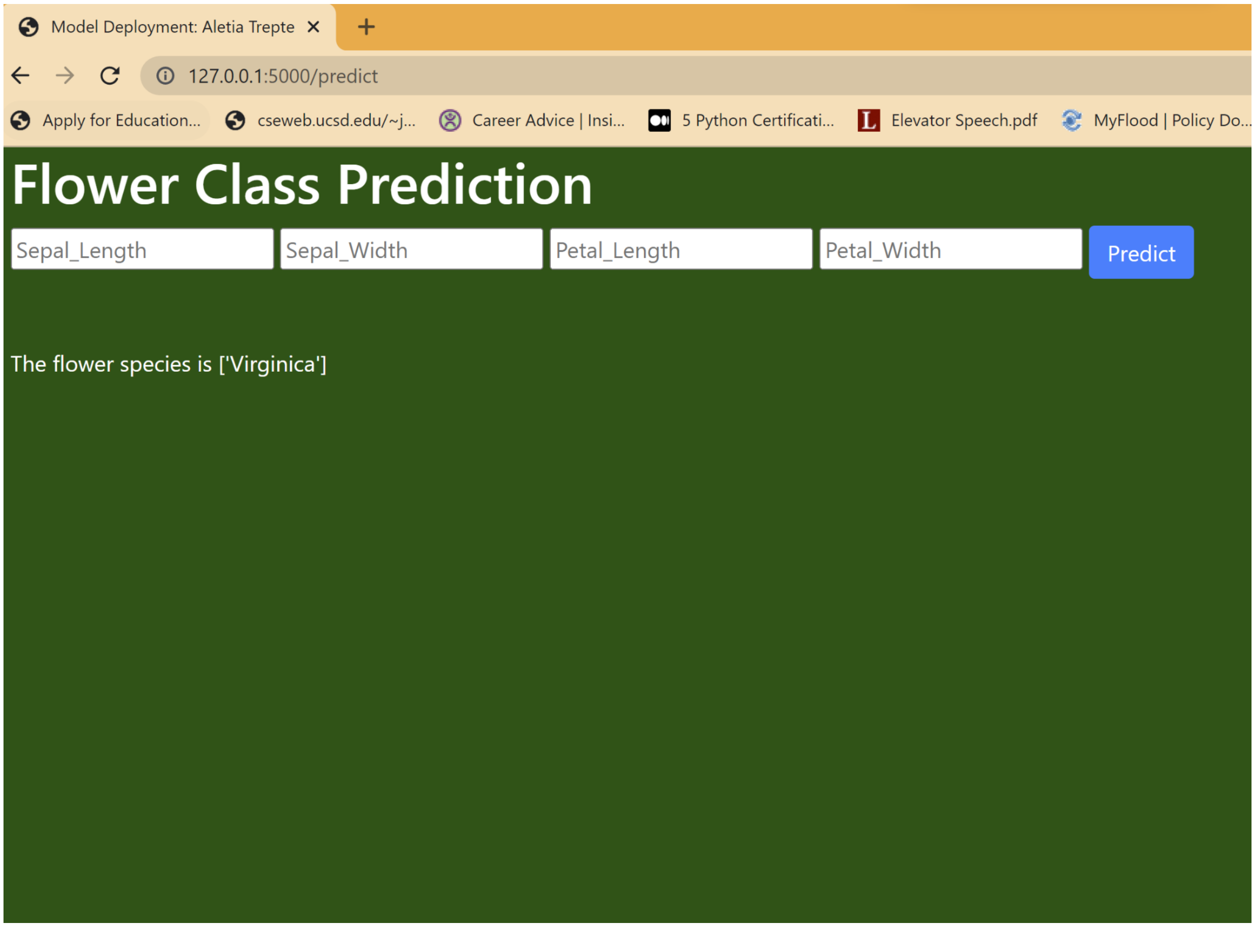
```
   336fb77..4e65de7  main -> main

C:\Users\nunto\Flask\model-deployment\Flask-Model-Deployment>python app.py
 * Running on http://127.0.0.1:5000 (Press CTRL+C to quit)
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 115-590-296
```

Web page output



Flower Class Prediction

Get prediction



Flower Class Prediction

Sepal_Length Sepal_Width Petal_Length Petal_Width

The flower species is ['Virginica']