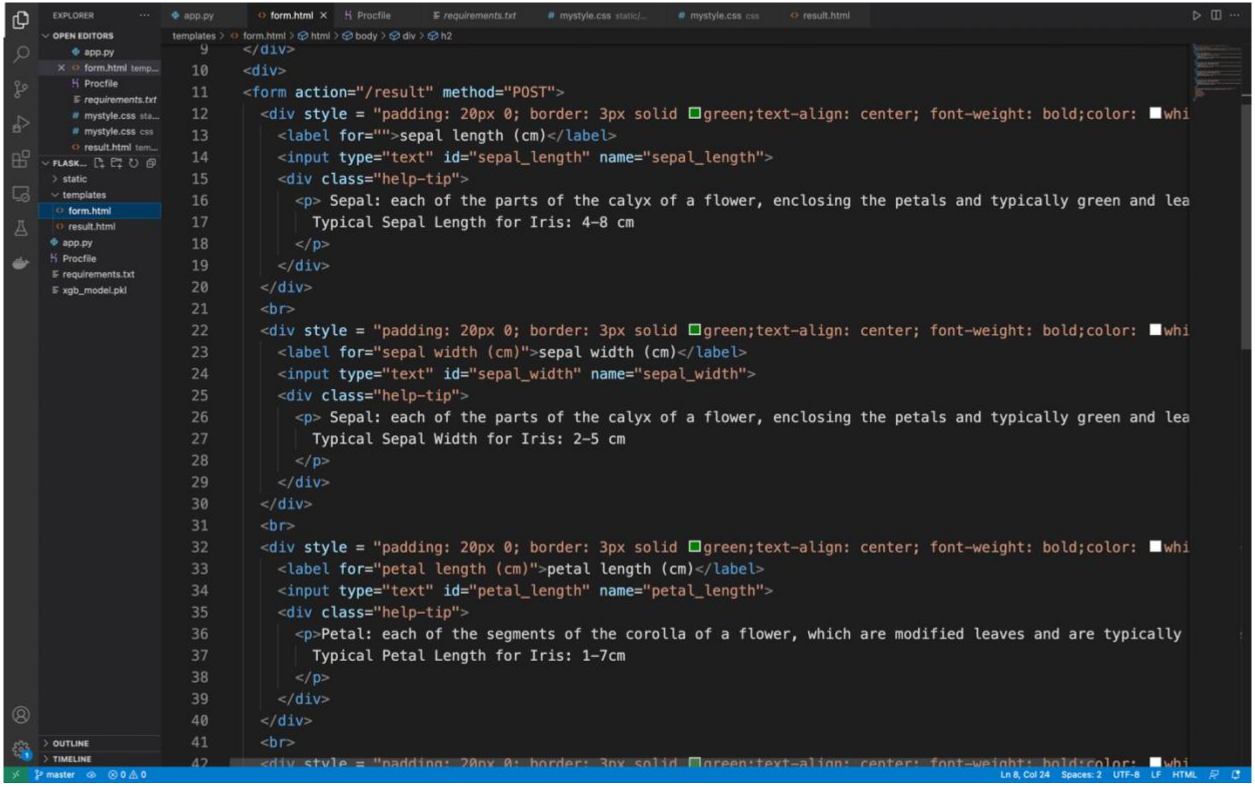
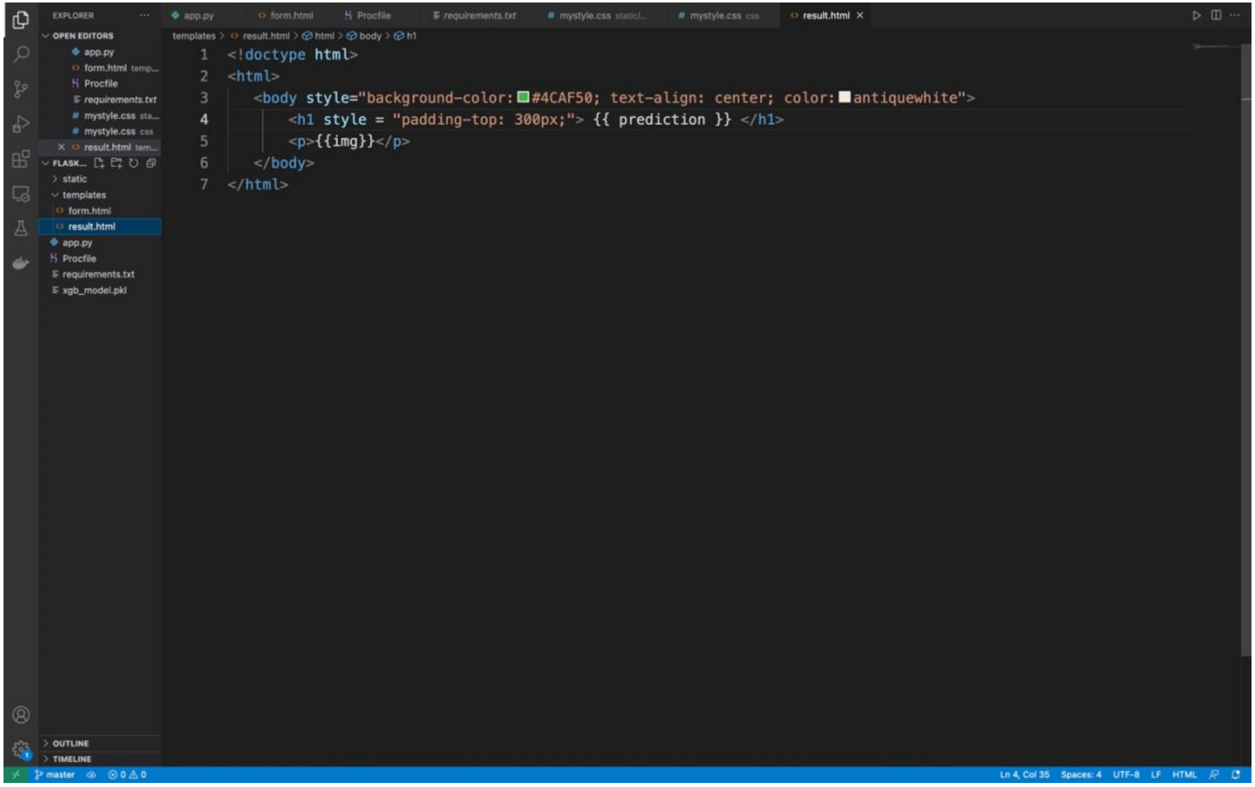
Created the form.html, this is the index page for our web application



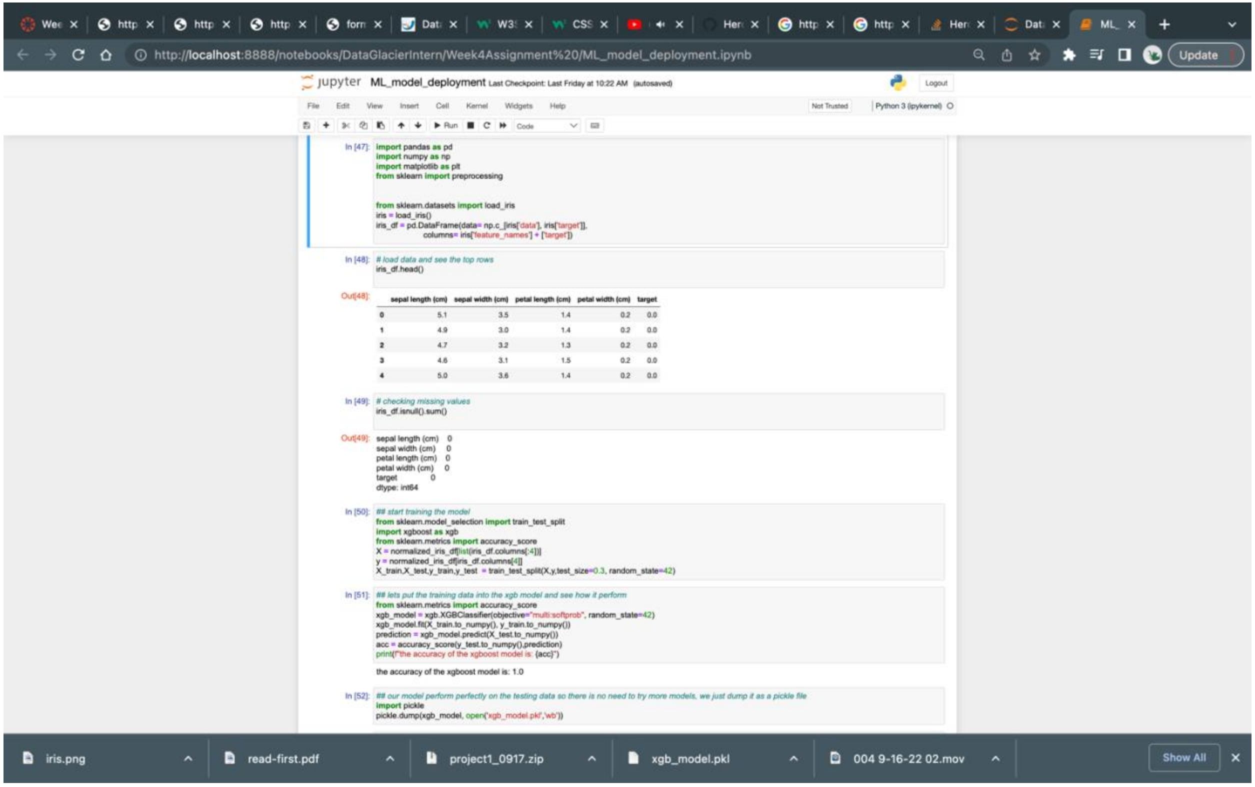
```
</div>
<div>
<form action="/result" method="POST">
  <div style = "padding: 20px 0; border: 3px solid green;text-align: center; font-weight: bold;color: whi
    <label for="">sepal length (cm)</label>
    <input type="text" id="sepal_length" name="sepal_length">
    <div class="help-tip">
      <p> Sepal: each of the parts of the calyx of a flower, enclosing the petals and typically green and lea
        Typical Sepal Length for Iris: 4-8 cm
      </p>
    </div>
  </div>
  <br>
  <div style = "padding: 20px 0; border: 3px solid green;text-align: center; font-weight: bold;color: whi
    <label for="sepal width (cm)">sepal width (cm)</label>
    <input type="text" id="sepal_width" name="sepal_width">
    <div class="help-tip">
      <p> Sepal: each of the parts of the calyx of a flower, enclosing the petals and typically green and lea
        Typical Sepal Width for Iris: 2-5 cm
      </p>
    </div>
  </div>
  <br>
  <div style = "padding: 20px 0; border: 3px solid green;text-align: center; font-weight: bold;color: whi
    <label for="petal length (cm)">petal length (cm)</label>
    <input type="text" id="petal_length" name="petal_length">
    <div class="help-tip">
      <p>Petal: each of the segments of the corolla of a flower, which are modified leaves and are typically
        Typical Petal Length for Iris: 1-7cm
      </p>
    </div>
  </div>
  <br>
  <div style = "padding: 20px 0; border: 3px solid green;text-align: center; font-weight: bold;color: whi
```

Created result page, this is what the page going to take us after we click submit



```
<!doctype html>
<html>
    <body style="background-color: #4CAF50; text-align: center; color: antiquewhite">
        <h1 style = "padding-top: 300px;"> {{ prediction }} </h1>
        <p>{{img}}</p>
    </body>
</html>
```

Created the model for Iris prediction, the model is xgboost.


http://localhost:8888/notebooks/DataGlacierIntern/Week4Assignment%20/ML_model_deployment.ipynb
jupyter ML_model_deployment Last Checkpoint: Last Friday at 10:22 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
iris.png
read-first.pdf
project1_0917.zip
xgb_model.pkl
004 9-16-22 02.mov
Show All

In [47]:
```python
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn import preprocessing


from sklearn.datasets import load_iris
iris = load_iris()
iris_df = pd.DataFrame(data= np.c_[iris['data'], iris['target']],
                       columns= iris['feature_names'] + ['target'])
```

In [48]:
```python
# load data and see the top rows
iris_df.head()
```

Out[48]:
```
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  target
0                5.1               3.5                1.4               0.2     0.0
1                4.9               3.0                1.4               0.2     0.0
2                4.7               3.2                1.3               0.2     0.0
3                4.6               3.1                1.5               0.2     0.0
4                5.0               3.6                1.4               0.2     0.0
```

In [49]:
```python
# checking missing values
iris_df.isnull().sum()
```

Out[49]:
```
sepal length (cm)    0
sepal width (cm)     0
petal length (cm)    0
petal width (cm)     0
target               0
dtype: int64
```

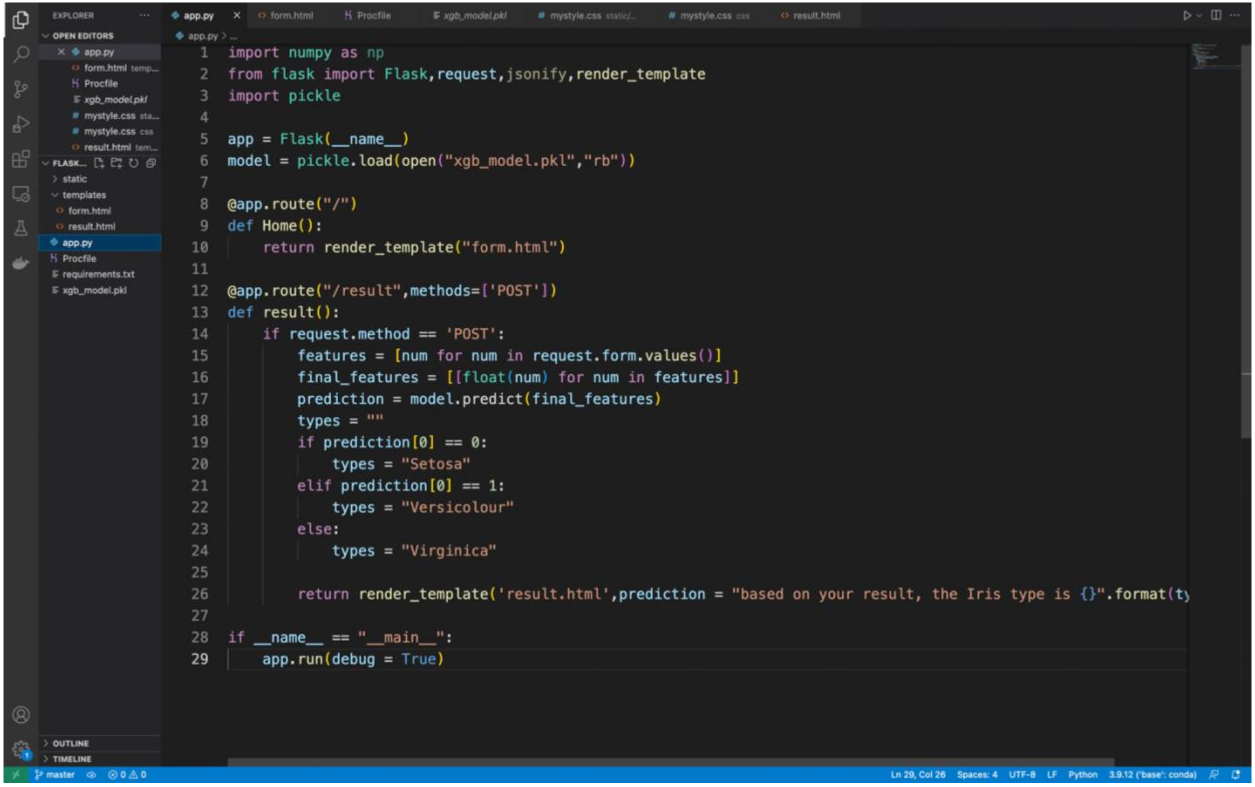
In [50]:
```python
## start training the model
from sklearn.model_selection import train_test_split
import xgboost as xgb
from sklearn.metrics import accuracy_score
X = normalized_iris_df[list(iris_df.columns[:4])]
y = normalized_iris_df[iris_df.columns[4]]
X_train,X_test,y_train,y_test  = train_test_split(X,y,test_size=0.3, random_state=42)
```

In [51]:
```python
## lets put the training data into the xgb model and see how it perform
from sklearn.metrics import accuracy_score
xgb_model = xgb.XGBClassifier(objective="multi:softprob", random_state=42)
xgb_model.fit(X_train.to_numpy(), y_train.to_numpy())
prediction = xgb_model.predict(X_test.to_numpy())
acc = accuracy_score(y_test.to_numpy(),prediction)
print(f"the accuracy of the xgboost model is: {acc}")
```

```
the accuracy of the xgboost model is: 1.0
```

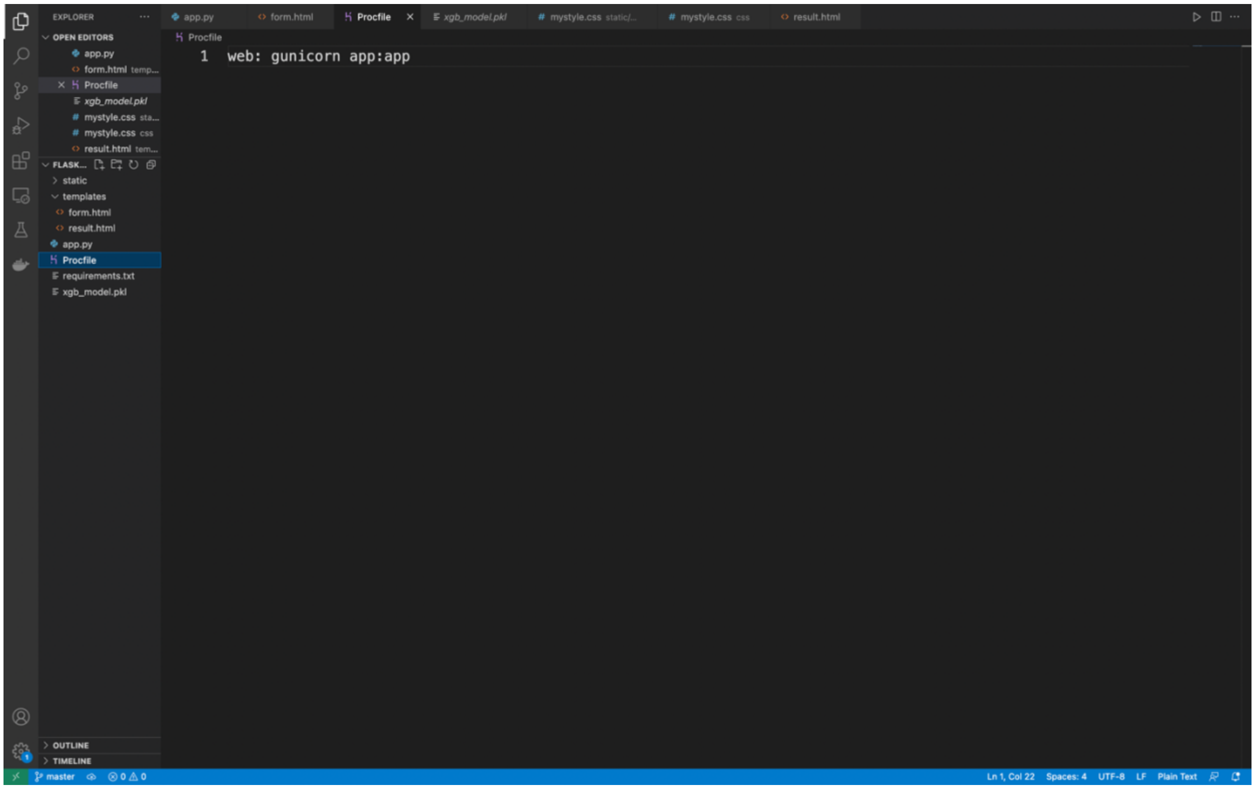
In [52]:
```python
## our model perform perfectly on the testing data so there is no need to try more models, we just dump it as a pickle file
import pickle
pickle.dump(xgb_model, open('xgb_model.pkl','wb'))
```

Write the app.py file, it is a written using flask



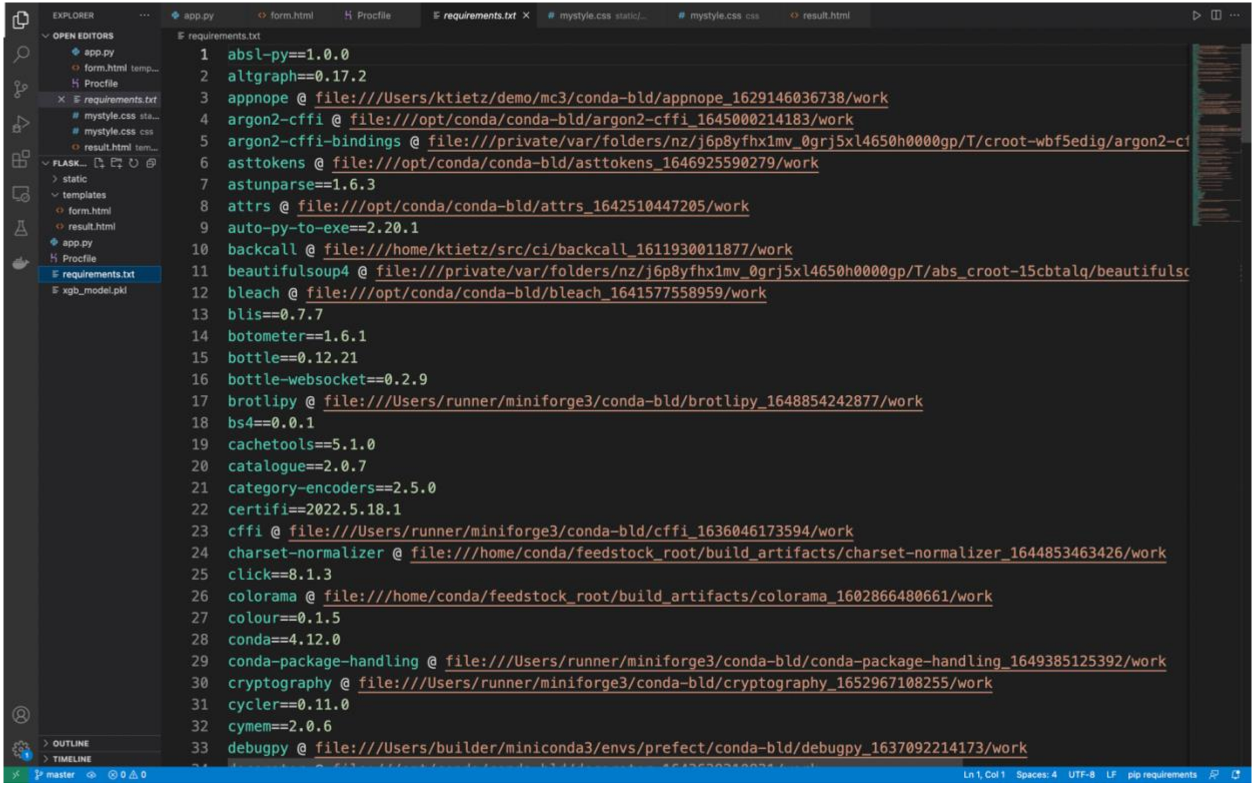
```python
import numpy as np
from flask import Flask,request,jsonify,render_template
import pickle

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

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

@app.route("/result",methods=['POST'])
def result():
    if request.method == 'POST':
        features = [num for num in request.form.values()]
        final_features = [[float(num) for num in features]]
        prediction = model.predict(final_features)
        types = ""
        if prediction[0] == 0:
            types = "Setosa"
        elif prediction[0] == 1:
            types = "Versicolour"
        else:
            types = "Virginica"

        return render_template('result.html',prediction = "based on your result, the Iris type is {}".format(ty

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

Created Procfile for heroku



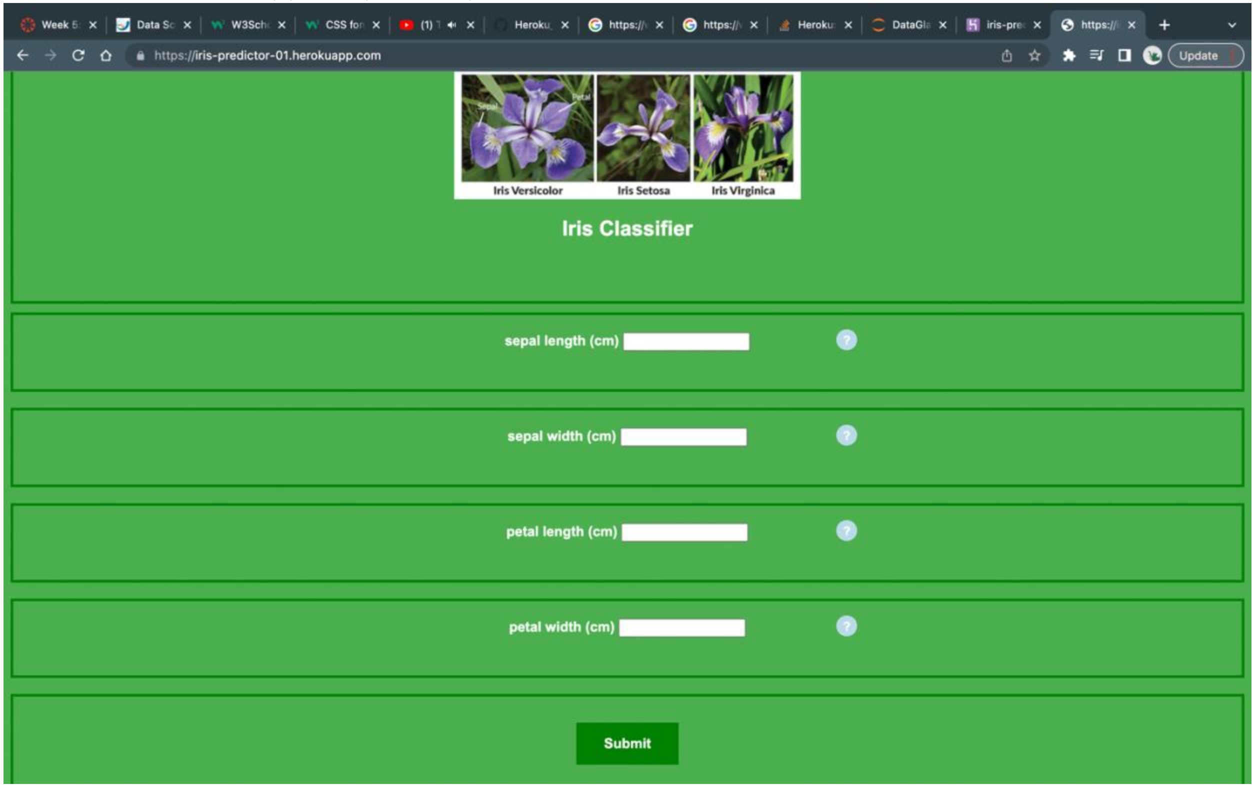
```
web: gunicorn app:app
```

Created requirements.txt



```
absl-py==1.0.0
altgraph==0.17.2
appnope @ file:///Users/ktietz/demo/mc3/conda-bld/appnope_1629146036738/work
argon2-cffi @ file:///opt/conda/conda-bld/argon2-cffi_1645000214183/work
argon2-cffi-bindings @ file:///private/var/folders/nz/j6p8yfhx1mv_0grj5xl4650h0000gp/T/croot-wbf5edig/argon2-cf
asttokens @ file:///opt/conda/conda-bld/asttokens_1646925590279/work
astunparse==1.6.3
attrs @ file:///opt/conda/conda-bld/attrs_1642510447205/work
auto-py-to-exe==2.20.1
backcall @ file:///home/ktietz/src/ci/backcall_1611930011877/work
beautifulsoup4 @ file:///private/var/folders/nz/j6p8yfhx1mv_0grj5xl4650h0000gp/T/abs_croot-15cbtalq/beautifulsc
bleach @ file:///opt/conda/conda-bld/bleach_1641577558959/work
blis==0.7.7
botometer==1.6.1
bottle==0.12.21
bottle-websocket==0.2.9
brotlipy @ file:///Users/runner/miniforge3/conda-bld/brotlipy_1648854242877/work
bs4==0.0.1
cachetools==5.1.0
catalogue==2.0.7
category-encoders==2.5.0
certifi==2022.5.18.1
cffi @ file:///Users/runner/miniforge3/conda-bld/cffi_1636046173594/work
charset-normalizer @ file:///home/conda/feedstock_root/build_artifacts/charset-normalizer_1644853463426/work
click==8.1.3
colorama @ file:///home/conda/feedstock_root/build_artifacts/colorama_1602866480661/work
colour==0.1.5
conda==4.12.0
conda-package-handling @ file:///Users/runner/miniforge3/conda-bld/conda-package-handling_1649385125392/work
cryptography @ file:///Users/runner/miniforge3/conda-bld/cryptography_1652967108255/work
cycler==0.11.0
cymem==2.0.6
debugpy @ file:///Users/builder/miniconda3/envs/prefect/conda-bld/debugpy_1637092214173/work
```

Make sure our webapp run perfectly


https://iris-predictor-01.herokuapp.com

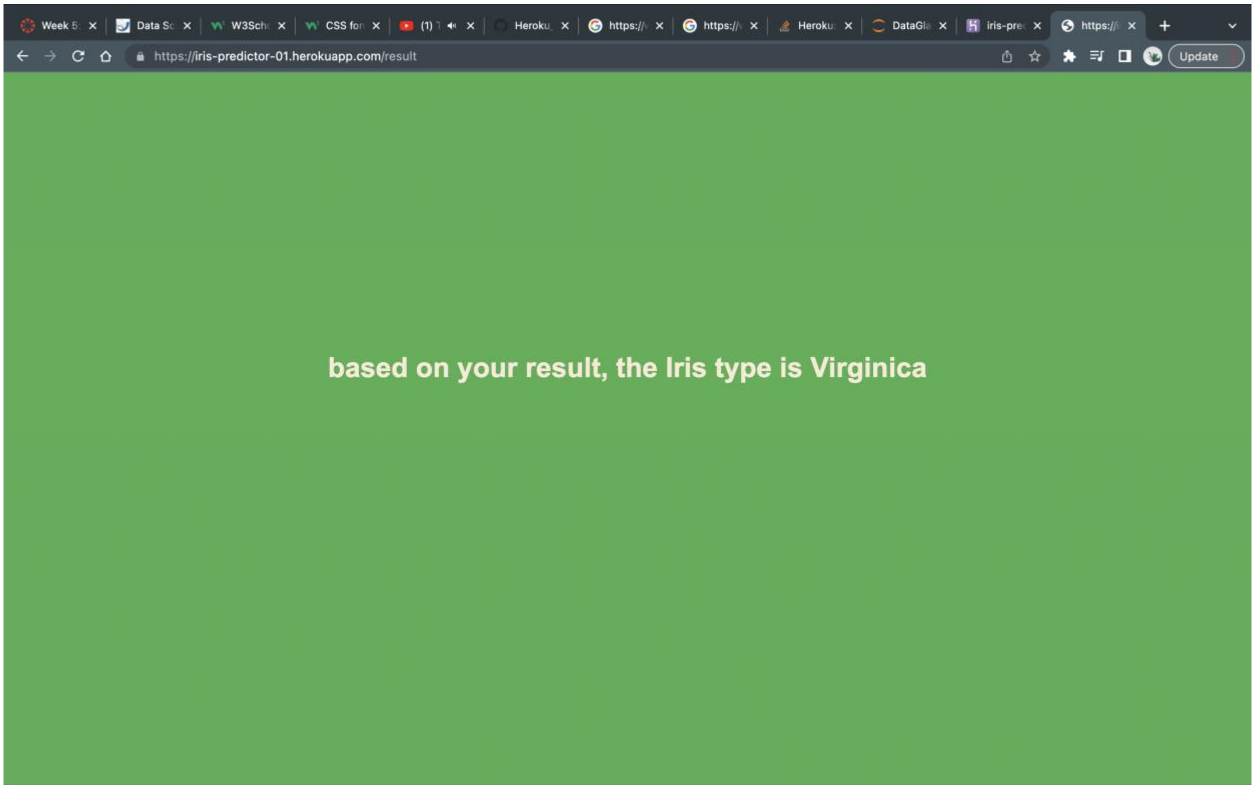
Iris Classifier

sepal length (cm)

sepal width (cm)

petal length (cm)

petal width (cm)


https://iris-predictor-01.herokuapp.com/result

based on your result, the Iris type is Virginica