Name – Purvesh Mehta

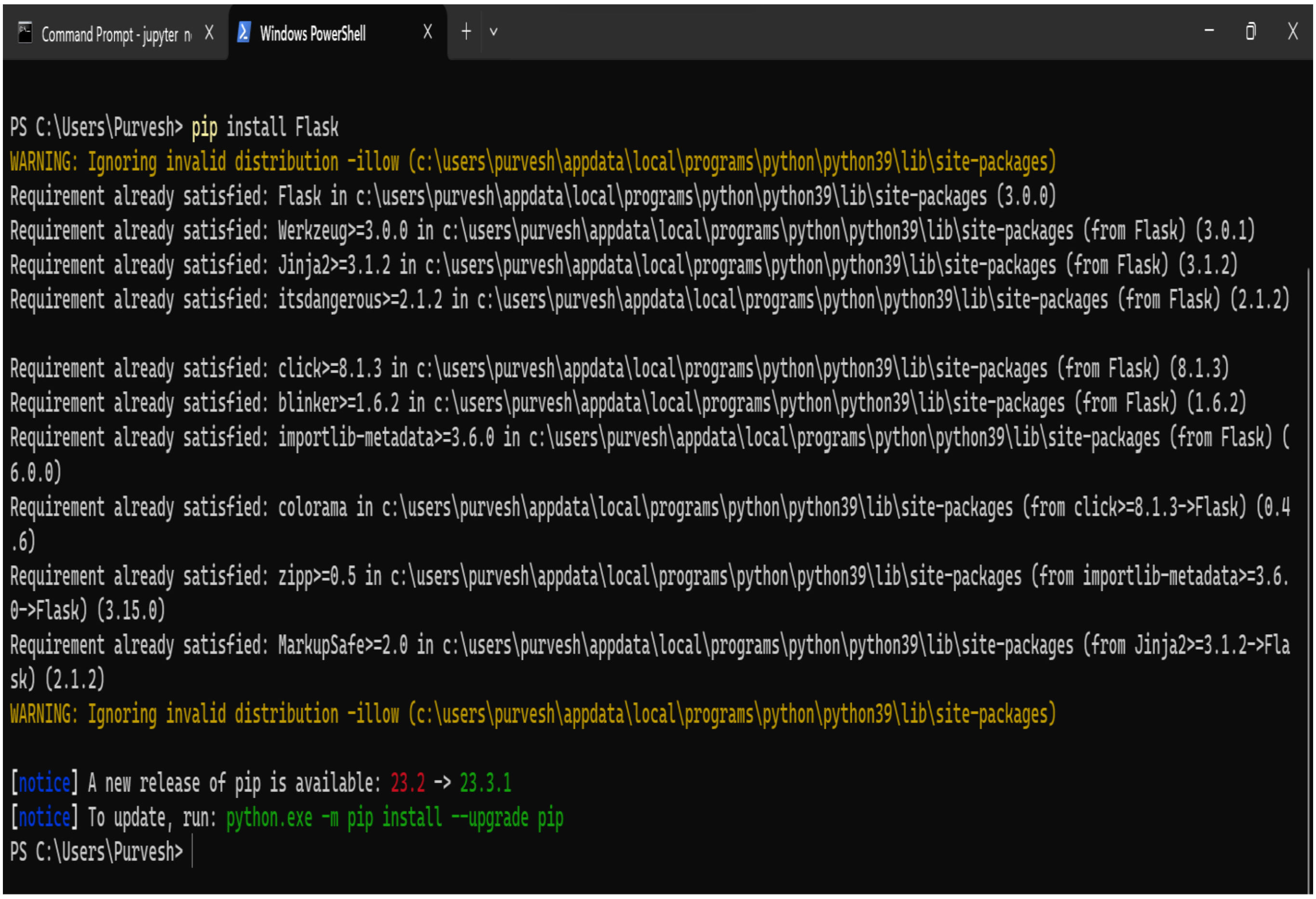
Batch Code – LISUM27

Submission Date – 27th November 2023

Submitted To – Data Glacier (Week 4 Task)

Part – 1

Installing Flask



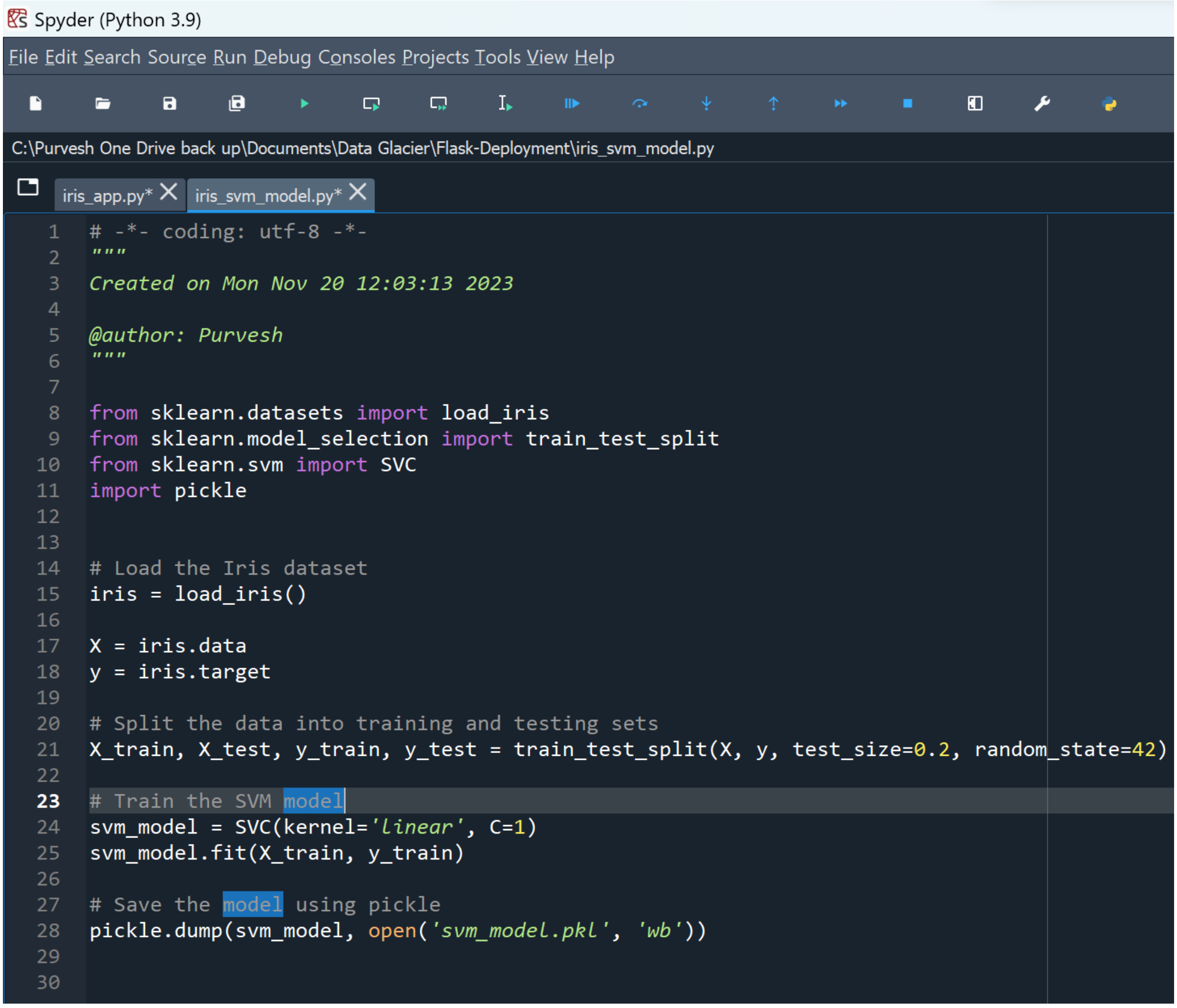
```
PS C:\Users\Purvesh> pip install Flask
WARNING: Ignoring invalid distribution -illow (c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages)
Requirement already satisfied: Flask in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (3.0.0)
Requirement already satisfied: Werkzeug>=3.0.0 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (3.0.1)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (2.1.2)

Requirement already satisfied: click>=8.1.3 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (8.1.3)
Requirement already satisfied: blinker>=1.6.2 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (1.6.2)
Requirement already satisfied: importlib-metadata>=3.6.0 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Flask) (
6.0.0)
Requirement already satisfied: colorama in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from click>=8.1.3->Flask) (0.4
.6)
Requirement already satisfied: zipp>=0.5 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from importlib-metadata>=3.6.
0->Flask) (3.15.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages (from Jinja2>=3.1.2->Fla
sk) (2.1.2)
WARNING: Ignoring invalid distribution -illow (c:\users\purvesh\appdata\local\programs\python\python39\lib\site-packages)

[notice] A new release of pip is available: 23.2 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\Purvesh>
```

1 Instaling Flask

Part – 2

The provided Python script utilizes the scikit-learn library to load the Iris dataset, a well-known dataset in machine learning. It then splits the dataset into training and testing sets, with 80% of the data used for training and 20% for testing. The script employs a Support Vector Machine (SVM) algorithm with a linear kernel and regularization parameter C=1 for training.

After the training process, the script saves the trained SVM model using the pickle module, resulting in a file named 'svm_model.pkl'. This saved model is later used for predictions using Flask


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Purvesh One Drive back up\Documents\Data Glacier\Flask-Deployment\iris_svm_model.py
iris_app.py*
iris_svm_model.py*

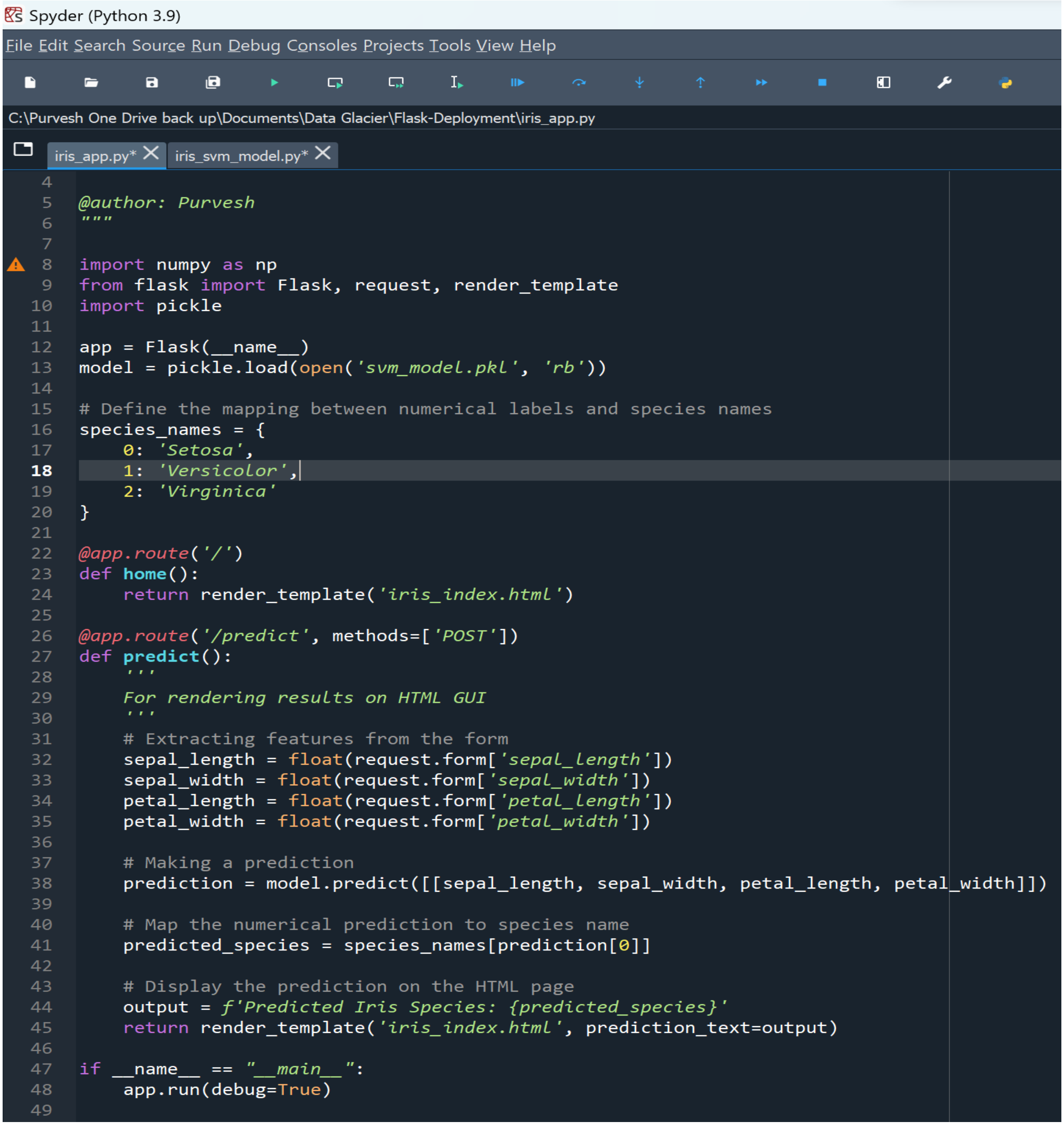
```python
# -*- coding: utf-8 -*-
"""
Created on Mon Nov 20 12:03:13 2023

@author: Purvesh
"""

from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import pickle


# Load the Iris dataset
iris = load_iris()

X = iris.data
y = iris.target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the SVM model
svm_model = SVC(kernel='linear', C=1)
svm_model.fit(X_train, y_train)

# Save the model using pickle
pickle.dump(svm_model, open('svm_model.pkl', 'wb'))


```

2 SVM Code

Part – 3

Flask Code

This Flask web application serves as a platform to make predictions on Iris flower species based on user input. It uses a pre-trained SVM model, loaded from the 'svm_model.pkl' file, to classify the Iris species. The input features, representing sepal and petal dimensions and two more, are extracted from a form submitted by the user. The application incorporates a mapping between numerical labels and species names to provide more interpretable predictions. After receiving the input, the script predicts the Iris species and displays the result in the HTML template. The species name is then presented as the output, enhancing the user experience by providing a more meaningful prediction. The Flask app runs in debug mode, allowing for easy development and testing. This deployment is a practical demonstration of using machine learning models in real-world applications through a web interface.


Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Purvesh One Drive back up\Documents\Data Glacier\Flask-Deployment\iris_app.py
iris_app.py*
iris_svm_model.py*

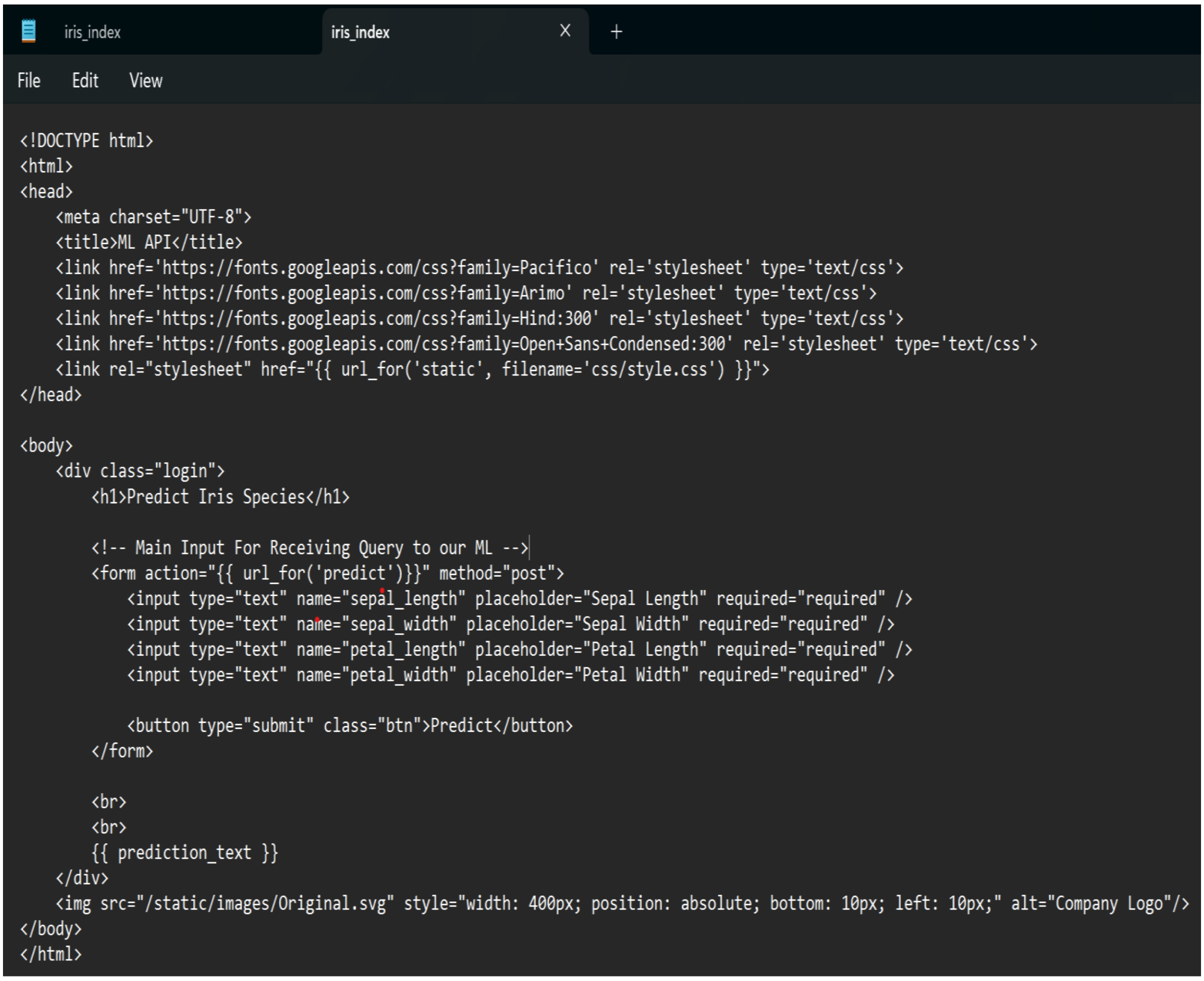
```python

@author: Purvesh
"""

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

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

# Define the mapping between numerical labels and species names
species_names = {
    0: 'Setosa',
    1: 'Versicolor',
    2: 'Virginica'
}

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

@app.route('/predict', methods=['POST'])
def predict():
    '''
    For rendering results on HTML GUI
    '''
    # Extracting features from the form
    sepal_length = float(request.form['sepal_length'])
    sepal_width = float(request.form['sepal_width'])
    petal_length = float(request.form['petal_length'])
    petal_width = float(request.form['petal_width'])

    # Making a prediction
    prediction = model.predict([[sepal_length, sepal_width, petal_length, petal_width]])

    # Map the numerical prediction to species name
    predicted_species = species_names[prediction[0]]

    # Display the prediction on the HTML page
    output = f'Predicted Iris Species: {predicted_species}'
    return render_template('iris_index.html', prediction_text=output)

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

```

3 Flask Code

Part – 4

Index File for Flask Application

This HTML code defines the structure of a web page designed for predicting Iris species through a machine learning model. The page includes a form with input fields for sepal and petal dimensions, allowing users to submit queries. Upon form submission, the Flask app processes the input, predicts the Iris species, and displays the result below the form. The page is visually appealing, utilizing Google Fonts and custom styling through an external CSS file. The "Predict" button triggers the prediction process, providing a user-friendly interface for interacting with the deployed machine learning model.



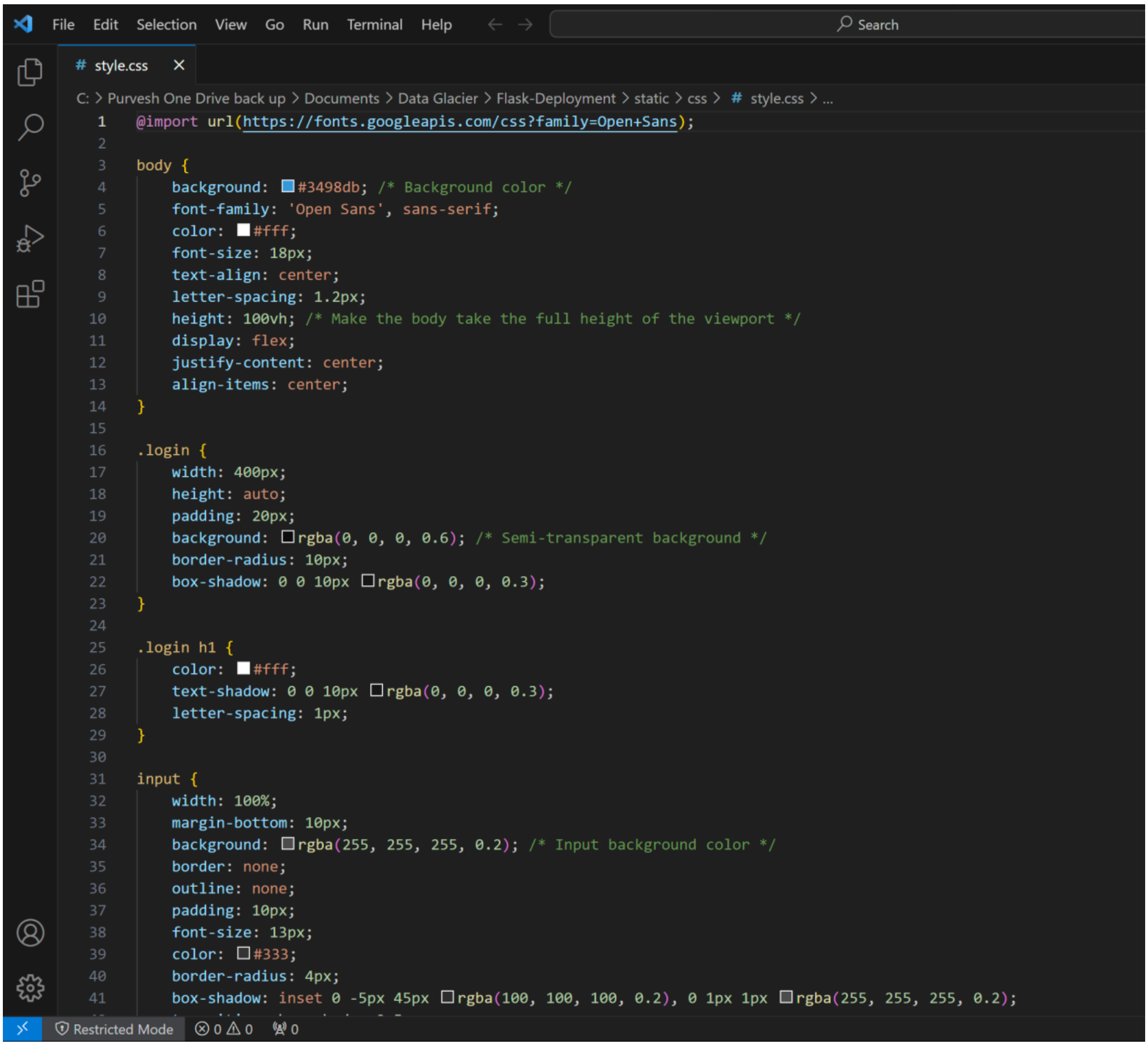
```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>ML API</title>
    <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
    <link href='https://fonts.googleapis.com/css?family=Hind:300' 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 Iris Species</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">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>
```

4 Index.html

Part – 5

Styling Enhancement for Visual Appeal

‘style.css’ was incorporated to refine the CSS styling. It uses a visually appealing background, and the layout of the page was adjusted for better alignment, this step was done to improve the user experience.


File
Edit
Selection
View
Go
Run
Terminal
Help
Search
# style.css
C: > Purvesh One Drive back up > Documents > Data Glacier > Flask-Deployment > static > css > # style.css > ...
Restricted Mode
⊗ 0 ⚠ 0
0

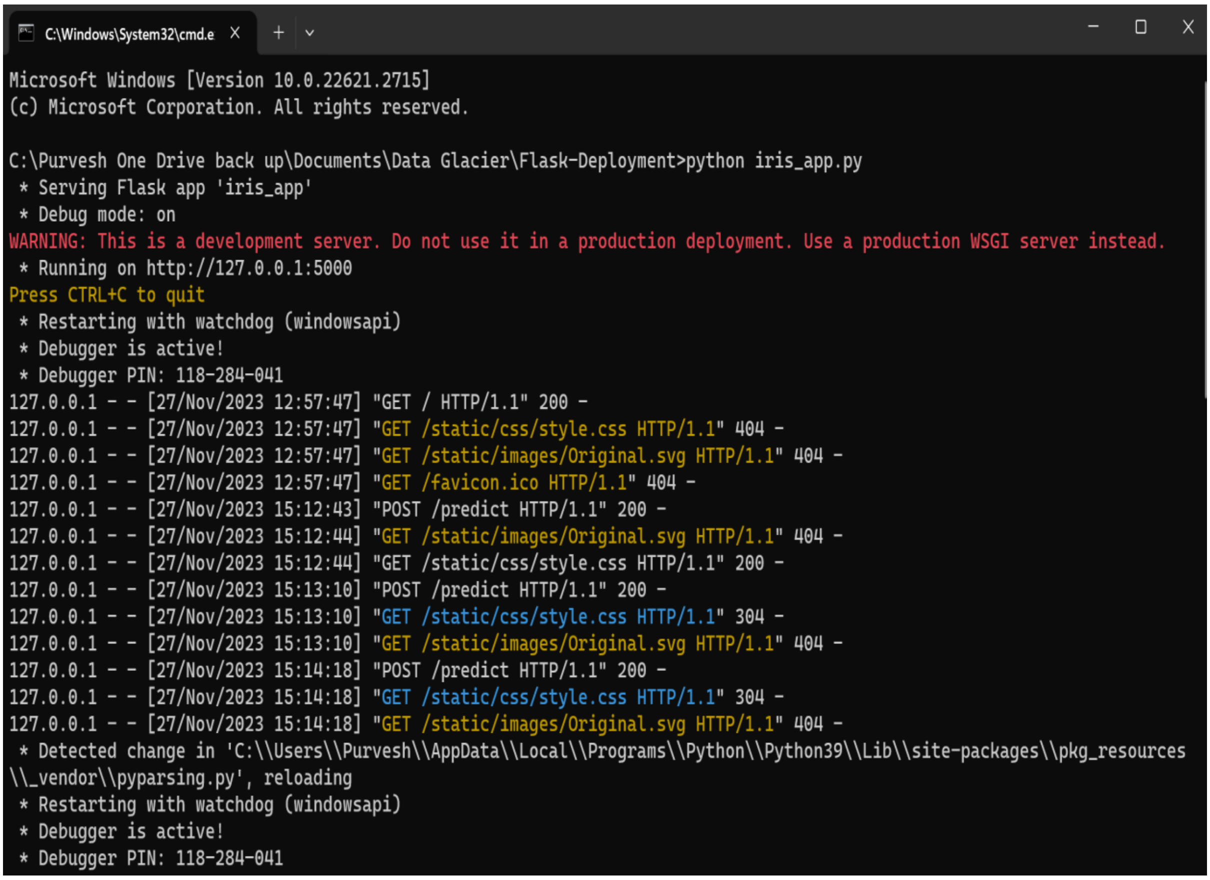
```css
@import url(https://fonts.googleapis.com/css?family=Open+Sans);

body {
    background: #3498db; /* Background color */
    font-family: 'Open Sans', sans-serif;
    color: #fff;
    font-size: 18px;
    text-align: center;
    letter-spacing: 1.2px;
    height: 100vh; /* Make the body take the full height of the viewport */
    display: flex;
    justify-content: center;
    align-items: center;
}

.login {
    width: 400px;
    height: auto;
    padding: 20px;
    background: rgba(0, 0, 0, 0.6); /* Semi-transparent background */
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
}

.login h1 {
    color: #fff;
    text-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
    letter-spacing: 1px;
}

input {
    width: 100%;
    margin-bottom: 10px;
    background: rgba(255, 255, 255, 0.2); /* Input background color */
    border: none;
    outline: none;
    padding: 10px;
    font-size: 13px;
    color: #333;
    border-radius: 4px;
    box-shadow: inset 0 -5px 45px rgba(100, 100, 100, 0.2), 0 1px 1px rgba(255, 255, 255, 0.2);
```

5 Style.css

Part – 6

Deployment on a Local Machine

Using command prompt and python to run the application.

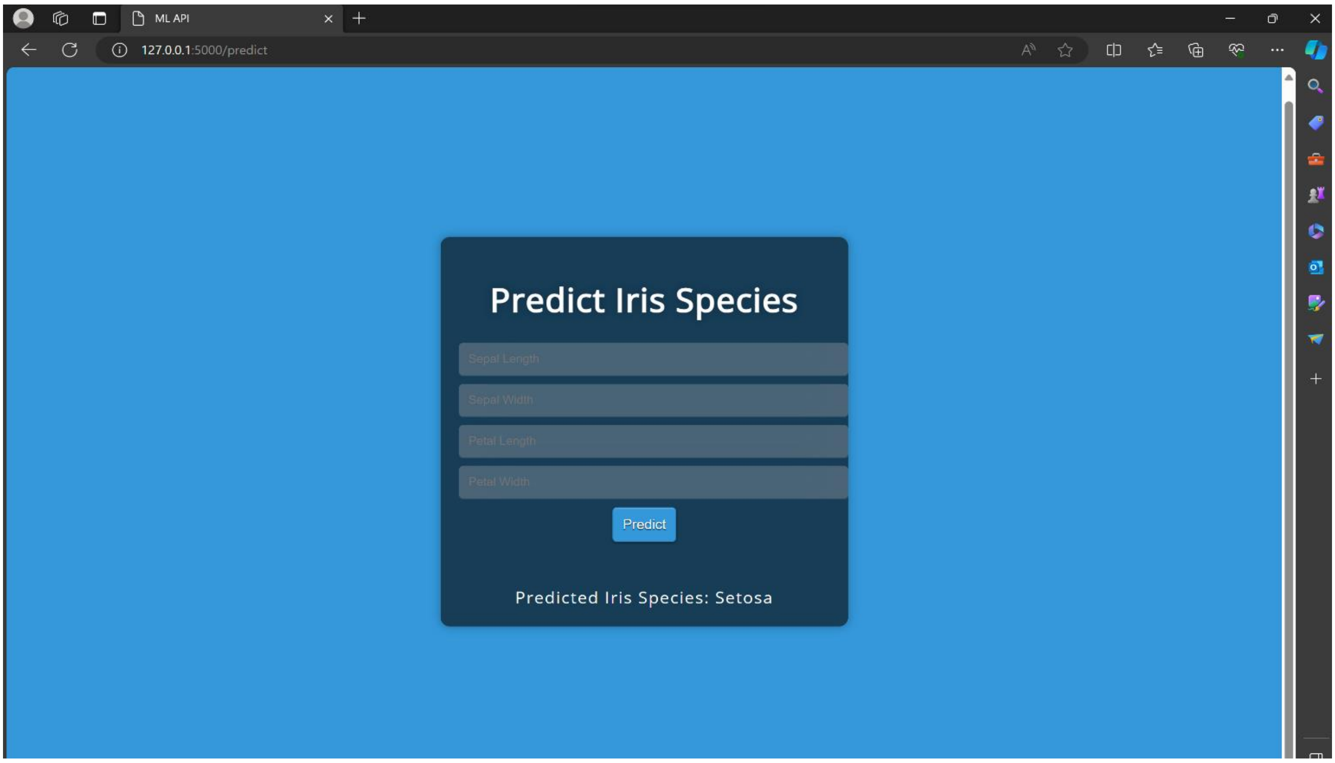


```
Microsoft Windows [Version 10.0.22621.2715]
(c) Microsoft Corporation. All rights reserved.

C:\Purvesh One Drive back up\Documents\Data Glacier\Flask-Deployment>python iris_app.py
 * Serving Flask app 'iris_app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 118-284-041
127.0.0.1 - - [27/Nov/2023 12:57:47] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [27/Nov/2023 12:57:47] "GET /static/css/style.css HTTP/1.1" 404 -
127.0.0.1 - - [27/Nov/2023 12:57:47] "GET /static/images/Original.svg HTTP/1.1" 404 -
127.0.0.1 - - [27/Nov/2023 12:57:47] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [27/Nov/2023 15:12:43] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [27/Nov/2023 15:12:44] "GET /static/images/Original.svg HTTP/1.1" 404 -
127.0.0.1 - - [27/Nov/2023 15:12:44] "GET /static/css/style.css HTTP/1.1" 200 -
127.0.0.1 - - [27/Nov/2023 15:13:10] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [27/Nov/2023 15:13:10] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [27/Nov/2023 15:13:10] "GET /static/images/Original.svg HTTP/1.1" 404 -
127.0.0.1 - - [27/Nov/2023 15:14:18] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [27/Nov/2023 15:14:18] "GET /static/css/style.css HTTP/1.1" 304 -
127.0.0.1 - - [27/Nov/2023 15:14:18] "GET /static/images/Original.svg HTTP/1.1" 404 -
 * Detected change in 'C:\\Users\\Purvesh\\AppData\\Local\\Programs\\Python\\Python39\\Lib\\site-packages\\pkg_resources
\\_vendor\\pyparsing.py', reloading
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 118-284-041
```

6 Deployment of Flask

Part – 7

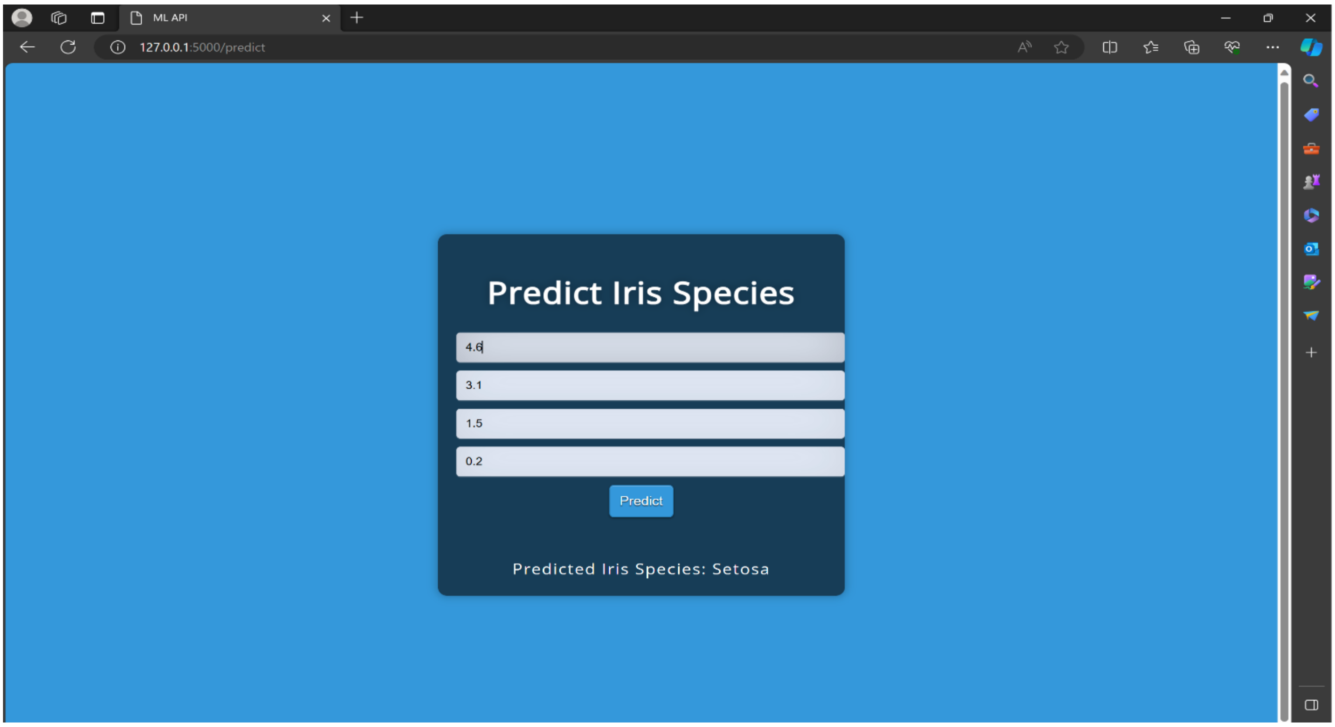
Using the address to run the application.



Predicted Iris Species: Setosa

Part – 8

Testing model prediction on flask



4.6

3.1

1.5

0.2

Predicted Iris Species: Setosa

Part – 9

Directory Folder Structure

```
- project_folder
  - iris_app.py
  - static
    - css
      - style.css
    - images
      - Original.svg
  - templates
    - iris_index.html
  - svm_model.pkl
```

Data Intake Report

Name: Deployment on Flask

Report date: 27th November 2023

Internship Batch: LISUM27

Version:

Data intake by:

Data intake reviewer:

Data storage location: https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html

Tabular data details:

Total number of observations	150
Total number of files	1
Total number of features	4
Base format of the file	.csv
Size of the data	< 5 KB