Data Glacier

Week4: Deployment on Flask

Name: Munirah Alfehaid
Batch code: LISUM35
Submission date: 24/7/2024
Submitted to: Data Glacier

Introduction

This assignment involves building and deploying a machine learning model for predicting real estate prices based on various features. The primary objectives are to:

1. Train a regression model using a provided real estate dataset.
2. Save the trained model using `pickle` for later use.
3. Create a web application using Flask to accept input data, predict the house price per unit area, and display the results.

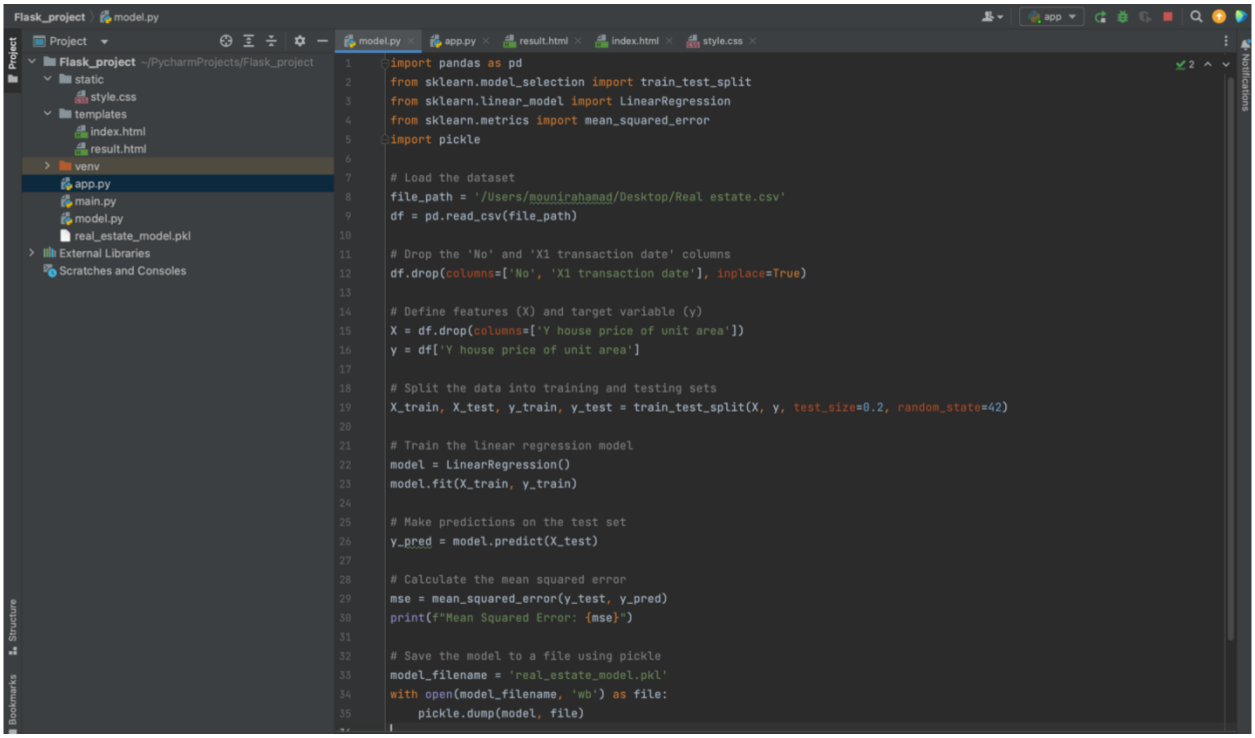
The dataset consists of information about house transactions, including the age of the house, distance to the nearest MRT station, number of convenience stores nearby, and the geographical coordinates (latitude and longitude). The target variable is the house price per unit area.

To complete this assignment, the project is structured with separate files for training the model, setting up the Flask web application, and defining HTML templates and CSS for the user interface. This clear structure ensures that the project is organized and maintainable.

The web application allows users to input the necessary features, predicts the house price using the trained model, and displays the prediction on a results page. This assignment demonstrates the practical application of machine learning in a real-world scenario, integrating model training, persistence, and deployment in a user-friendly web interface.

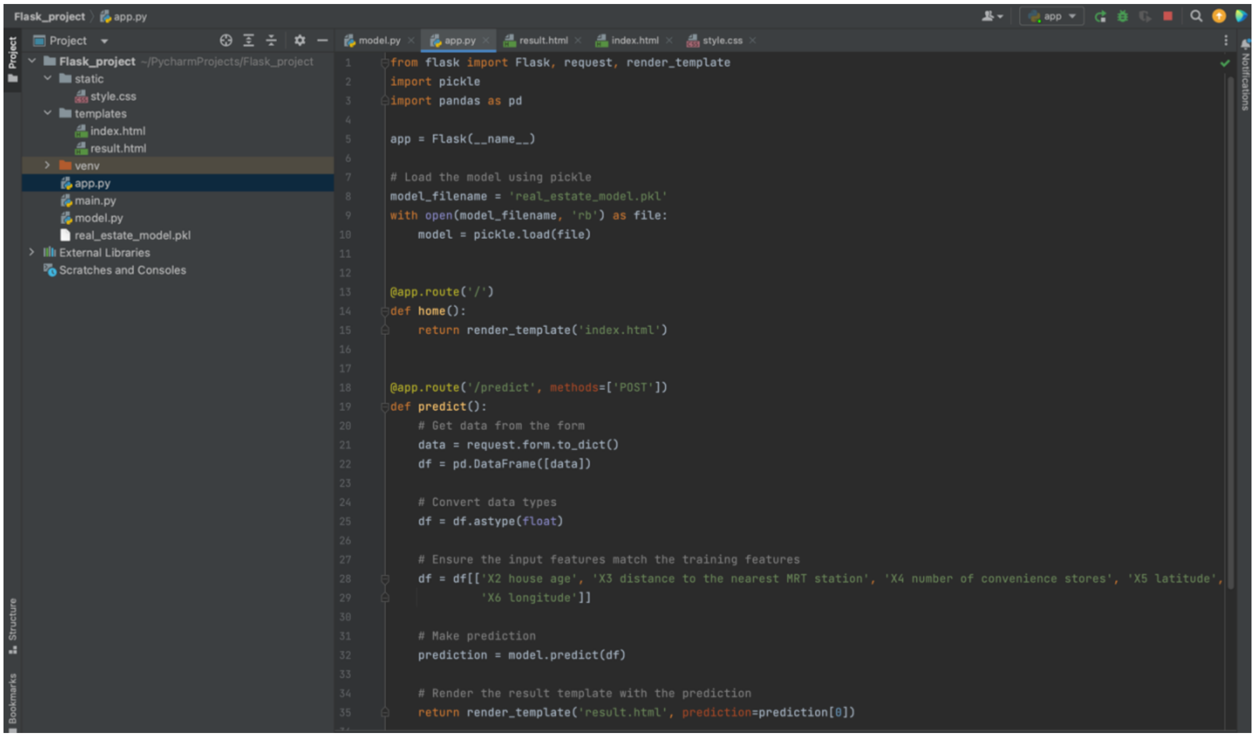
Snapshot of deployment steps:

1-model.py

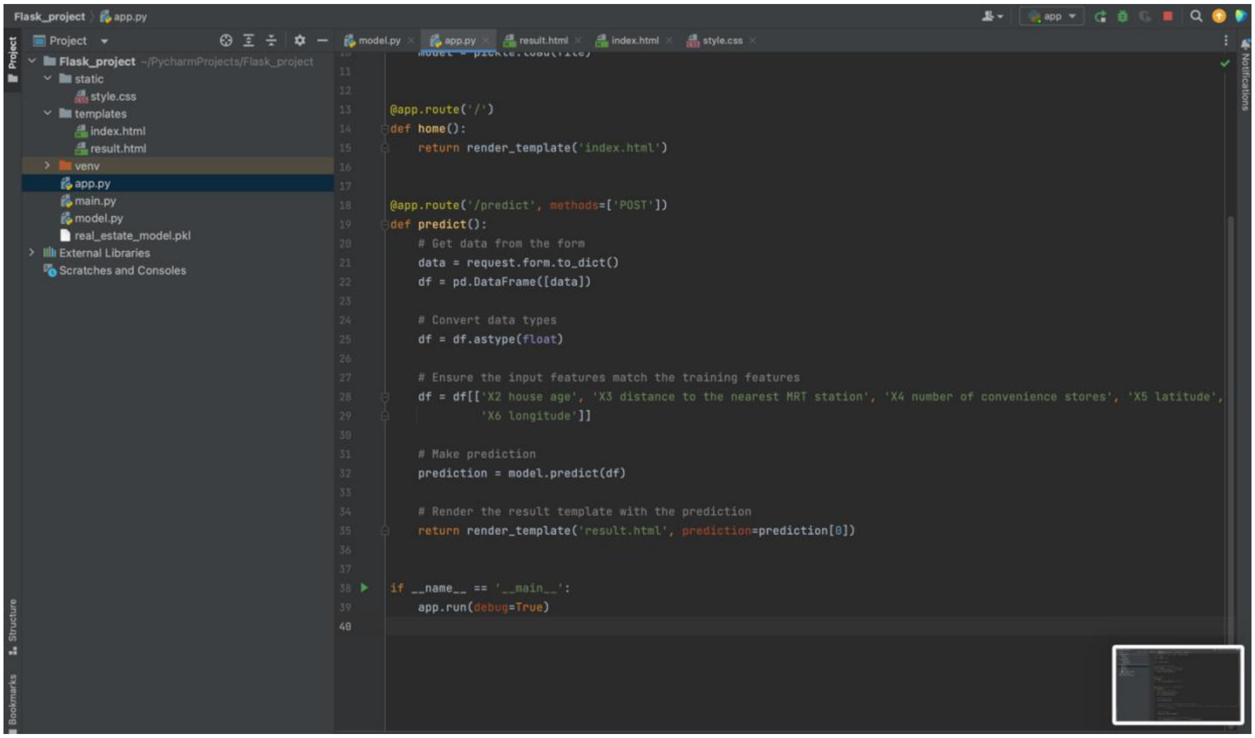


```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import pickle

# Load the dataset
file_path = '/Users/mounirahamad/Desktop/Real estate.csv'
df = pd.read_csv(file_path)

# Drop the 'No' and 'X1 transaction date' columns
df.drop(columns=['No', 'X1 transaction date'], inplace=True)

# Define features (X) and target variable (y)
X = df.drop(columns=['Y house price of unit area'])
y = df['Y house price of unit area']

# 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 linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Calculate the mean squared error
mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")

# Save the model to a file using pickle
model_filename = 'real_estate_model.pkl'
with open(model_filename, 'wb') as file:
    pickle.dump(model, file)
```

2-app.py



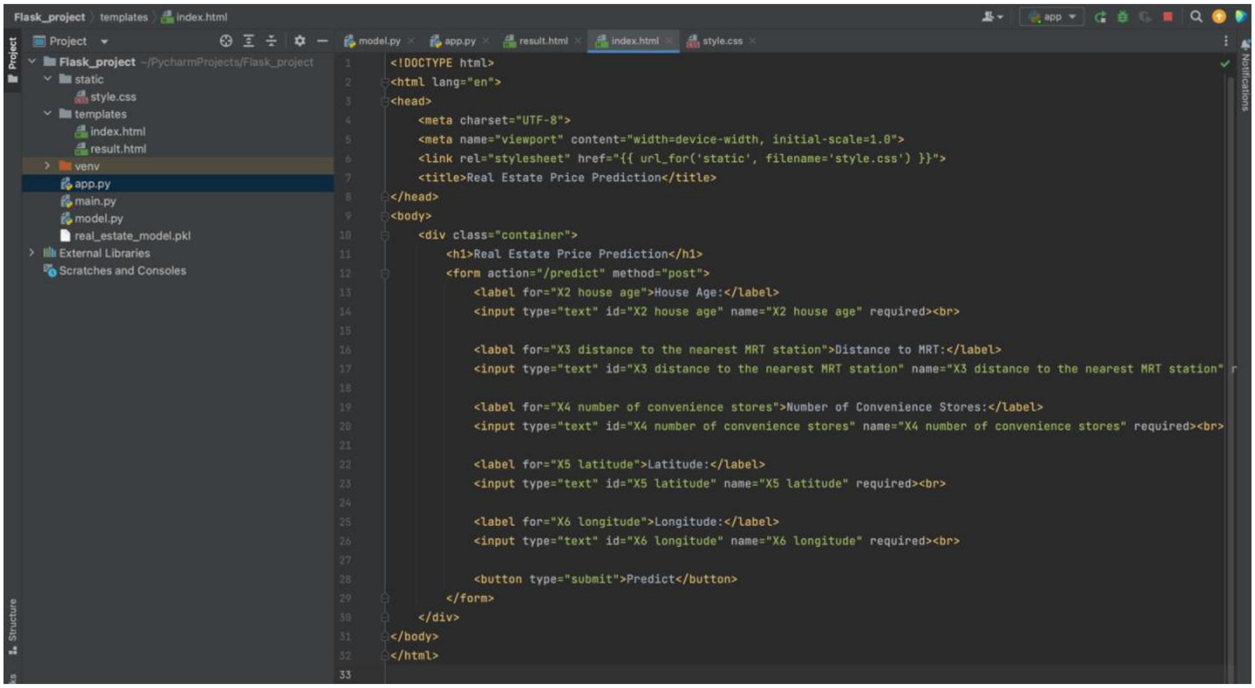
```python
from flask import Flask, request, render_template
import pickle
import pandas as pd

app = Flask(__name__)

# Load the model using pickle
model_filename = 'real_estate_model.pkl'
with open(model_filename, 'rb') as file:
    model = pickle.load(file)


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


@app.route('/predict', methods=['POST'])
def predict():
    # Get data from the form
    data = request.form.to_dict()
    df = pd.DataFrame([data])

    # Convert data types
    df = df.astype(float)

    # Ensure the input features match the training features
    df = df[['X2 house age', 'X3 distance to the nearest MRT station', 'X4 number of convenience stores', 'X5 latitude',
             'X6 longitude']]

    # Make prediction
    prediction = model.predict(df)

    # Render the result template with the prediction
    return render_template('result.html', prediction=prediction[0])
```



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


@app.route('/predict', methods=['POST'])
def predict():
    # Get data from the form
    data = request.form.to_dict()
    df = pd.DataFrame([data])

    # Convert data types
    df = df.astype(float)

    # Ensure the input features match the training features
    df = df[['X2 house age', 'X3 distance to the nearest MRT station', 'X4 number of convenience stores', 'X5 latitude',
             'X6 longitude']]

    # Make prediction
    prediction = model.predict(df)

    # Render the result template with the prediction
    return render_template('result.html', prediction=prediction[0])
```

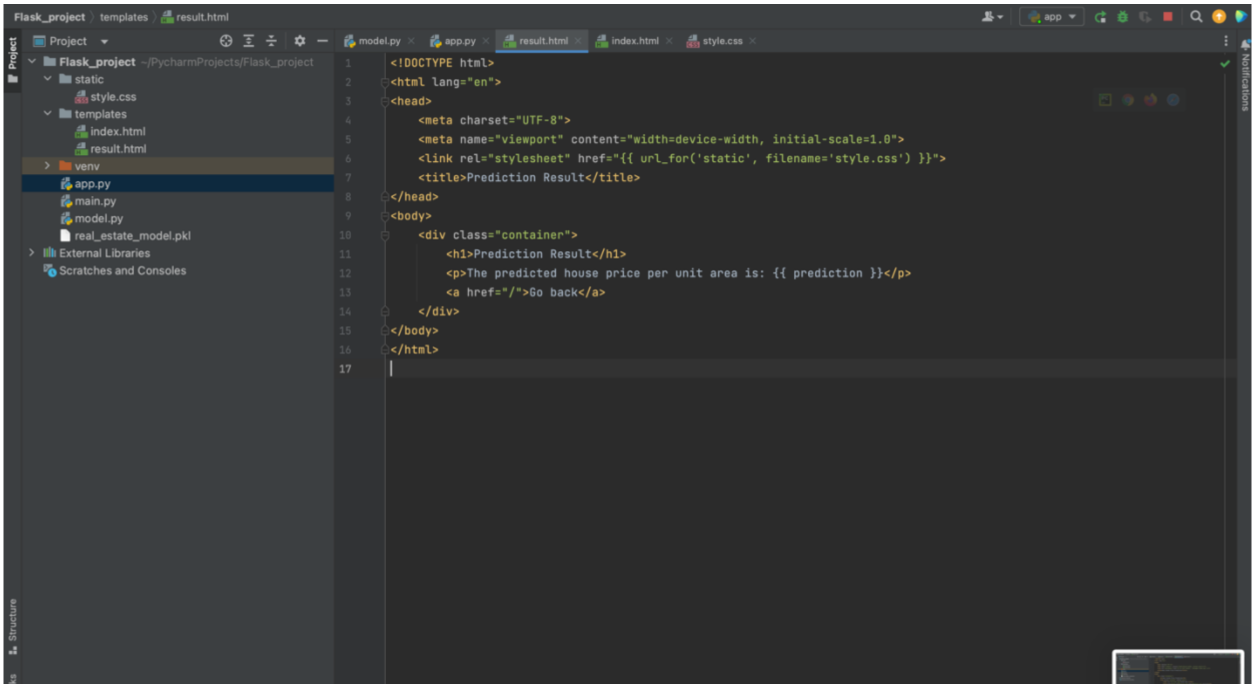
```python
if __name__ == '__main__':
    app.run(debug=True)
```

3-index.html



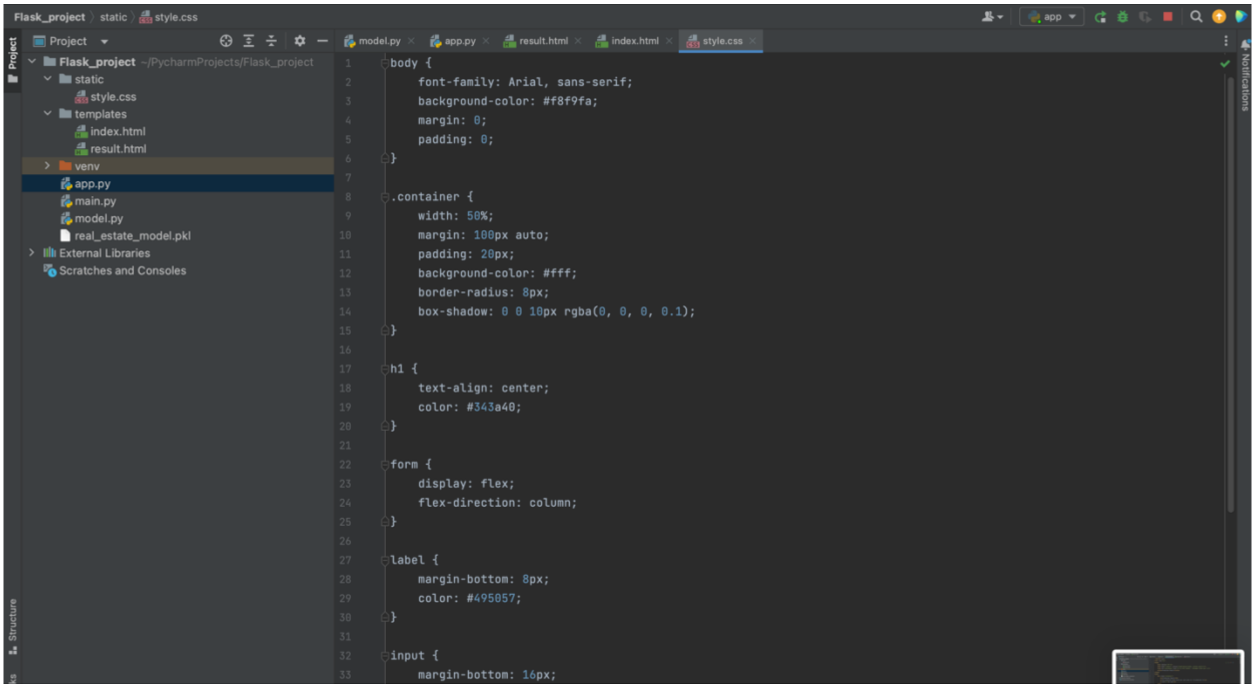
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    <title>Real Estate Price Prediction</title>
</head>
<body>
    <div class="container">
        <h1>Real Estate Price Prediction</h1>
        <form action="/predict" method="post">
            <label for="X2 house age">House Age:</label>
            <input type="text" id="X2 house age" name="X2 house age" required><br>

            <label for="X3 distance to the nearest MRT station">Distance to MRT:</label>
            <input type="text" id="X3 distance to the nearest MRT station" name="X3 distance to the nearest MRT station" r

            <label for="X4 number of convenience stores">Number of Convenience Stores:</label>
            <input type="text" id="X4 number of convenience stores" name="X4 number of convenience stores" required><br>

            <label for="X5 latitude">Latitude:</label>
            <input type="text" id="X5 latitude" name="X5 latitude" required><br>

            <label for="X6 longitude">Longitude:</label>
            <input type="text" id="X6 longitude" name="X6 longitude" required><br>

            <button type="submit">Predict</button>
        </form>
    </div>
</body>
</html>
```

4-result.html



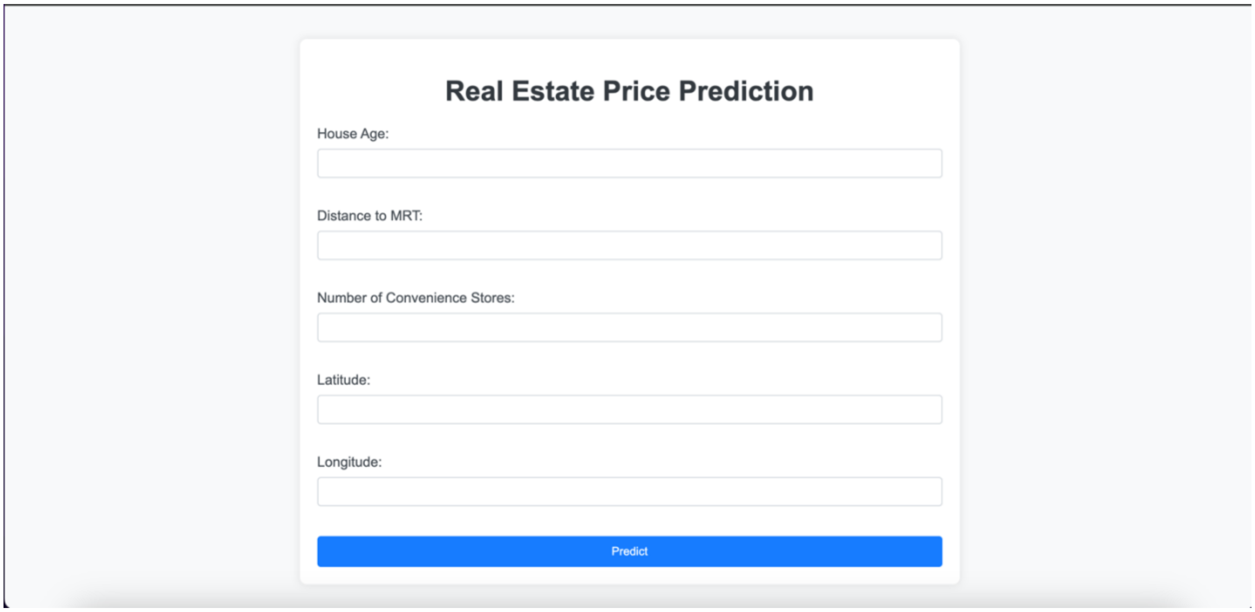
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
    <title>Prediction Result</title>
</head>
<body>
    <div class="container">
        <h1>Prediction Result</h1>
        <p>The predicted house price per unit area is: {{ prediction }}</p>
        <a href="/">Go back</a>
    </div>
</body>
</html>
```

5-style.css



```css
body {
    font-family: Arial, sans-serif;
    background-color: #f8f9fa;
    margin: 0;
    padding: 0;
}

.container {
    width: 50%;
    margin: 100px auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

h1 {
    text-align: center;
    color: #343a40;
}

form {
    display: flex;
    flex-direction: column;
}

label {
    margin-bottom: 8px;
    color: #495057;
}

input {
    margin-bottom: 16px;
```

6-Final result:



Real Estate Price Prediction

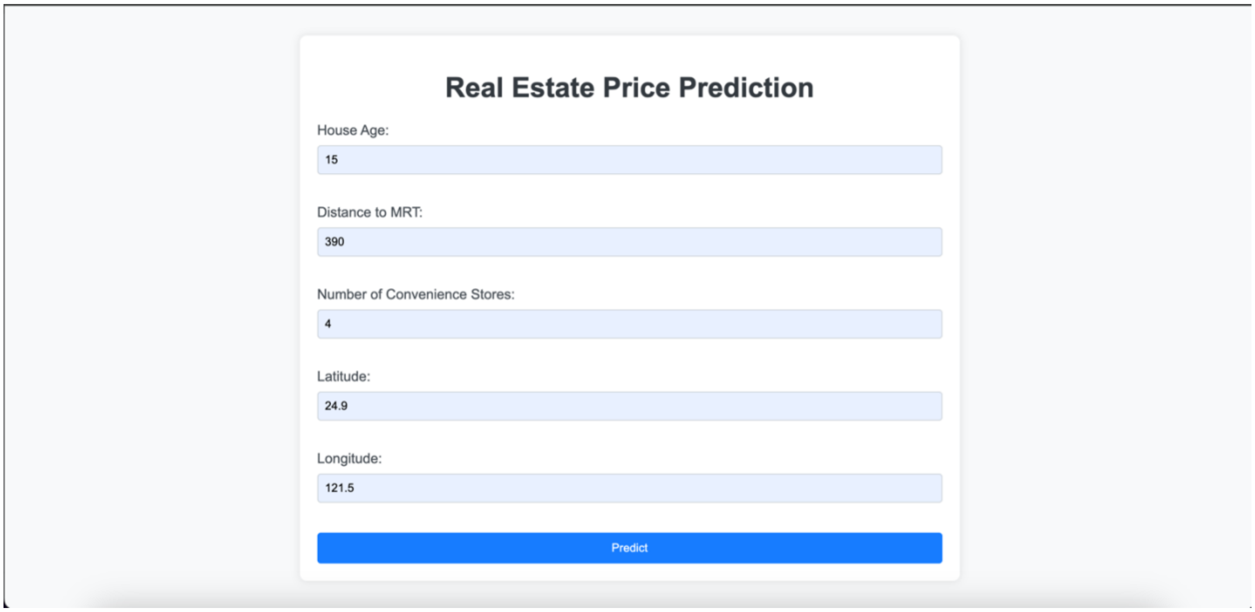
House Age:

Distance to MRT:

Number of Convenience Stores:

Latitude:

Longitude:



Real Estate Price Prediction

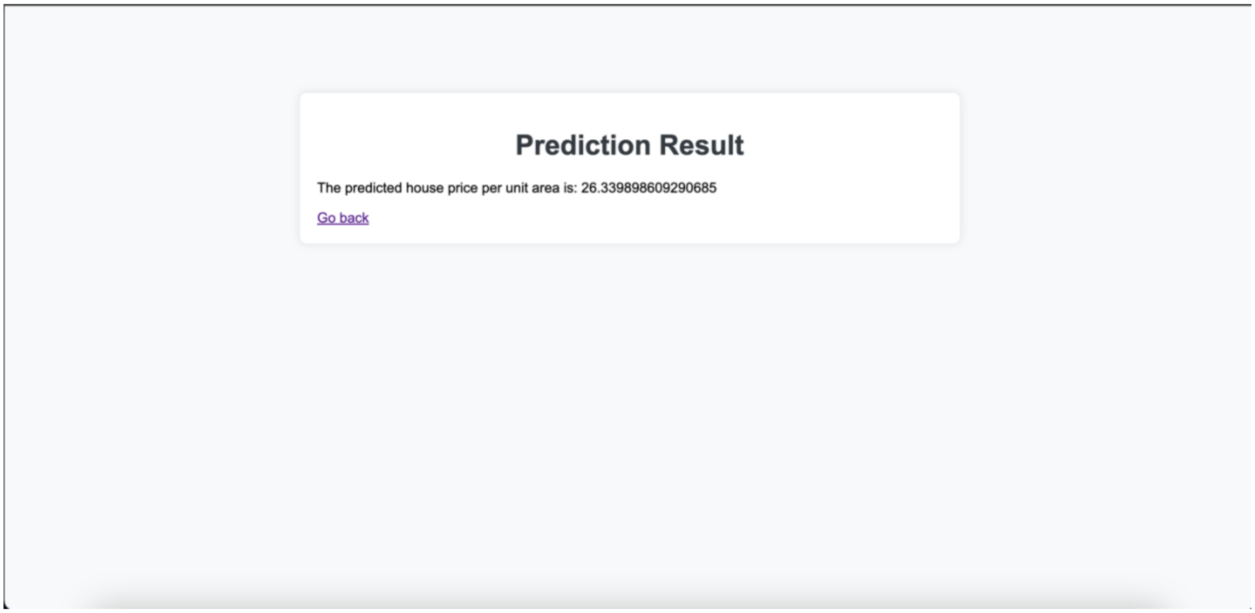
House Age: 15

Distance to MRT: 390

Number of Convenience Stores: 4

Latitude: 24.9

Longitude: 121.5



Prediction Result

The predicted house price per unit area is: 26.339898609290685