**Visualizing Housing Market Trends: An Analysis of Sale Prices and Features Using Tableau**

Team ID: LTVIP2025TMID51194

# 1. HTML - index.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Housing Market Trends - Upload Data</title>  
</head>  
<body>  
 <h2>Upload Housing Dataset (CSV)</h2>  
 <form action="/upload" method="post" enctype="multipart/form-data">  
 <input type="file" name="file" accept=".csv" required>  
 <input type="submit" value="Upload and Analyze">  
 </form>  
</body>  
</html>

# 2. HTML - result.html

<!DOCTYPE html>  
<html>  
<head>  
 <title>Analysis Result</title>  
</head>  
<body>  
 <h2>Dashboard: Housing Market Insights</h2>  
 <p>Analysis based on uploaded dataset.</p>  
 <img src="/static/price\_trends.png" width="600">  
 <br><a href="/">Upload Another File</a>  
</body>  
</html>

# 3. Dataset Upload & Visualization Code (Flask + Matplotlib)

from flask import Flask, render\_template, request  
import pandas as pd  
import matplotlib.pyplot as plt  
import os  
  
app = Flask(\_\_name\_\_)  
UPLOAD\_FOLDER = 'static'  
app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER  
  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
@app.route('/upload', methods=['POST'])  
def upload\_file():  
 file = request.files['file']  
 if file and file.filename.endswith('.csv'):  
 filepath = os.path.join(app.config['UPLOAD\_FOLDER'], file.filename)  
 file.save(filepath)  
  
 df = pd.read\_csv(filepath)  
 df = df.dropna()  
 df['YearBuilt'] = pd.to\_numeric(df['YearBuilt'], errors='coerce')  
 avg\_price\_by\_year = df.groupby('YearBuilt')['SalePrice'].mean().sort\_index()  
  
 plt.figure(figsize=(10, 6))  
 avg\_price\_by\_year.plot()  
 plt.title('Average Sale Price by Year Built')  
 plt.xlabel('Year Built')  
 plt.ylabel('Average Sale Price')  
 plt.grid(True)  
 plt.tight\_layout()  
 graph\_path = os.path.join(app.config['UPLOAD\_FOLDER'], 'price\_trends.png')  
 plt.savefig(graph\_path)  
 plt.close()  
  
 return render\_template('result.html', graph\_url=graph\_path)  
 return "Invalid file format", 400  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)