INDEX

NAME: ESWARAVAKA GEETHIKA

REG NO:113323106029

USERNAME:aut113323eca16

DEPT:ECE

PHASE 5 :PROJECT DEMONSTRATION AND DOCUMENTATION

Title: AI-Powered Affordable House Finder

Abstract:

The AI Affordable House Finder project is developed to assist users in finding cost-effective housing options by utilizing artificial intelligence, geolocation services, and user-preference analytics. The system integrates real-time property listings, budget analysis, and personalized recommendations. This document covers the final phase, which includes a full system demonstration, technical documentation, performance results, and user feedback. The system aims to simplify housing searches while ensuring scalability, responsiveness, and data privacy.

1. Project Demonstration

Overview:
A live walkthrough of the AI Affordable House Finder system will be conducted, highlighting real-time listing queries, user preference matching, filtering logic, and affordability analysis.

Demonstration Details:

- **System Walkthrough:** From user login to house recommendation results.

- **AI Matching Logic:** Showcases how machine learning evaluates listings based on income, location preference, and amenities.
- **Affordability Analysis:** Demonstrates integration with cost-of-living APIs and user financial input.
- **Performance Metrics:** System responsiveness, real-time listing updates, and concurrent user handling.
- **Security & Privacy:** Explanation of user data handling, encryption, and privacy protocols.

Outcome:
Stakeholders will see the tool's effectiveness in helping users find affordable housing with AI-powered personalization and secure data handling.

2. Project Documentation

Overview:
Detailed documentation of system architecture, data sources, machine learning models, and usage guides.

Sections:

- **System Architecture:** Diagrams of backend, frontend, and third-party API integrations.
- **Code Documentation:** Well-commented source code for scraping, model training, filtering engine, and user interface.
- **User Guide:** Instructions for searching, filtering, and saving properties.
- **Admin Guide:** Managing listings, running updates, and monitoring user analytics.
- **Testing Reports:** Load tests, response time benchmarks, and model accuracy analysis.

Outcome:
Comprehensive guide for system use, modification, and scaling.

3. Feedback and Final Adjustments

Overview:
Feedback collected from users and evaluators to refine system usability and performance.

Steps:

- Collect feedback via surveys.
- Implement UI/UX improvements.
- Improve matching model accuracy.
- Conduct final performance testing.

Outcome:
A refined and validated system ready for broader release.

4. Final Project Report Submission

Overview:
Summarizes the entire development process, key challenges, outcomes, and lessons learned.

Sections:

- **Executive Summary**
- **Phase Breakdown:** From concept to AI integration and deployment.
- **Challenges & Solutions:** E.g., inconsistent listing formats, API rate limits, data privacy concerns.
- **Outcomes:** AI accuracy, user satisfaction rates, and system readiness.

Outcome:
A detailed final report covering the end-to-end project journey.

5. Project Handover and Future Works

Overview:
Outlines plans for system expansion and maintenance.

Handover Details:

- **Next Steps:** Suggestions for multilingual support, mobile app integration, and partnership with real estate platforms.
- **Documentation & Access:** Source code, credentials, and deployment instructions handed over.

Outcome:
System is handed off with a roadmap for future scaling and maintenance.

PROGRAM:

main.py

```python
import pandas as pd

# Simulated dataset
data = {
    'Location': ['City Center', 'Suburb', 'Countryside', 'Suburb',
        'City Center'],
    'Price': [500000, 250000, 150000, 200000, 550000],
    'Bedrooms': [3, 4, 3, 2, 3],
    'Bathrooms': [2, 2, 1, 1, 2],
    'SquareFeet': [1500, 2000, 1200, 1300, 1600]
}

# Create DataFrame
df = pd.DataFrame(data)

# Define affordable threshold
affordable_price = 300000

# Phase 5 AI: Advanced Filtering Logic
# Filter affordable houses with at least 3 bedrooms
affordable_houses = df[
    (df['Price'] <= affordable_price) & (df['Bedrooms'] >= 3)
]

# Sort by price ascending
affordable_houses = affordable_houses.sort_values(by='Price')
```

Output

```
=== Affordable Houses (Phase 5 AI Logic) ===
      Location   Price  Bedrooms  Bathrooms  SquareFeet
2  Countryside  150000         3          1        1200
1       Suburb  250000         4          2        2000

=== Code Execution Successful ===
```