**Project Title: LLM-Powered Summer Home Recommender**

**Overview:** Design and implement a Python-based command-line application that acts as a personalized recommender system for vacation rentals. Inspired by Airbnb, this app helps users find ideal summer homes based on their preferences, budgets, and group sizes. The system will simulate a real estate/vacation rental platform and optionally integrate with an LLM (e.g., OpenRouter.ai) to generate travel descriptions, activity suggestions, or rental highlights.

**Use Case Story:** Imagine an MMA student who just graduated. They’ve been grinding hard all year and finally want to take a well-deserved break. They have a budget, they know their vibe (e.g., mountains vs. beaches, remote vs. city), and they want a data-driven way to choose the perfect spot. This app helps them find the best match.

**Functional Requirements:**

**1. User Profile Management**

* Attributes:
  + user\_id (unique identifier)
  + name
  + group size
  + preferred environment (e.g., mountain, lake, beach, city)
  + budget range
  + travel dates (optional)
* Actions:
  + Create, view, edit, delete profile

**2. Property Listings**

* Sample dataset (JSON or CSV) of properties:
  + location
  + type (cabin, condo, house, etc.)
  + nightly price
  + features (e.g., hot tub, WiFi, pet friendly)
  + tags (e.g., remote, family-friendly, nightlife)
* Optionally: generate sample listings via LLM prompts and store as JSON/CSV

**3. Recommender Logic**

* Use vectorized matching (Pandas/NumPy)
* Rank properties by fit score based on user preferences and budget
* Display top-N recommended listings

**4. LLM API Integration (Optional but encouraged)**

* Generate travel blurbs, suggested activities, or Airbnb-style property descriptions
* Generate synthetic property listings or user inputs via prompt (e.g., "Generate 5 fake property listings in Ontario for under $200/night with lake access")
* Example: “Write a fun intro for a user looking to travel to a mountain cabin with 4 friends under $200/night.”

**5. Command line interface (CLI) Application**

* Menu-driven app with commands:
  + create\_user, edit\_profile, view\_properties, get\_recommendations, llm\_summary, exit
* Optional: interactive chatbot-style interface to collect user preferences and generate JSON data dynamically via LLM

**Bonus Features (Optional)**

* GUI version (Tkinter or Streamlit)
* Search and filter listings
* Simulate booking flow or save properties
* Map integration or geolocation clustering (optional stretch goal)

**Deliverables**

1. **Source Code**: Python scripts with clean structure and comments
2. **Documentation**:
   * README file with usage guide
   * Setup instructions
   * LLM usage instructions (if integrated)
3. **Presentation** (10 minutes):
   * Live demo
   * Recommender logic explanation
   * Optional LLM-generated enhancements

**Grading Breakdown**

* Functionality (50%)
* Code Quality (30%)
* Documentation (10%)
* Presentation (10%)

**Timeline**

* **Final Submission Deadline**: Wednesday, August 27, 2025
* **Opt-Out Demo Deadline**: Wednesday, August 27, 2025 (Zoom or live)

**Opt-Out Project Specification (Individual Work Only)**

This option is available only to students working individually. Students with a strong CS/programming background are **discouraged** from joining groups. If opting out, you must develop an extended version of the recommender system, including:

**Core Requirements:**

* User authentication and session tracking
* Visual dashboards (e.g., price distribution, location clusters)
* Collaborative filtering or advanced multi-factor scoring model
* Fully responsive GUI (e.g., Streamlit, Flask, or custom interface)
* LLM-generated datasets or chatbot-style interaction to populate user/property info

**Documentation:**

* Inline code comments and a detailed README
* System design diagram or architecture notes
* Explanation of recommender algorithm and LLM usage

**Testing:**

* Unit and integration tests

**Presentation:**

* 10–15 minute live demo by Wednesday, August 27, 2025 (or earlier on Zoom)
* Showcase improvements and performance evaluation

Final submission and demo: **Wednesday, August 27, 2025**