📘 Assignment: Simulate a Progressive Personalized Recommendation System

You're tasked with building a **basic AI-powered recommendation module** that evolves over time based on **user interactions**.

✅ Problem Statement

Imagine you're designing a recommendation engine for a shopping app. At first, the app has *no deep user knowledge*—it recommends based on general trends (content + collaborative filtering). But over time, it starts building a **lightweight user knowledge graph** based on what the user clicks or likes.

🎯 Goal

Build a **Python-based script or minimal API** that demonstrates:

1. **Initial Recommendations**:
    - Use a **hybrid approach** (simple collaborative + content filtering) to recommend items to a new user.
    - Data can be faked or minimal (e.g., a small product dataset with categories, tags, and sample user interactions).
2. **User Click Simulation**:
    - Simulate the user clicking/liking a few products (add those product IDs or tags to a "user profile").
3. **User Knowledge Graph (Simplified)**:
    - Based on the clicked items, update a lightweight **user profile** (e.g., preferred categories, tags, styles).
4. **Personalized Recommendations** (Round 2):
    - Now generate **personalized recommendations** using the user profile.

    - Still mix in a few **trendy or collaborative-based** recommendations to simulate hybrid logic.

📂 Expected Structure of Submission

- `products.csv`: Small dataset of 20–30 products with fields like `product_id`, `title`, `tags`, `category`, `popularity_score`
- `users.csv`: Minimal user interaction data (can be simulated)
- `recommend.py`: Python script or notebook that:
    - Shows Round 1 recommendations (cold start)
    - Simulates user clicking 2–3 products
    - Updates user profile
    - Shows Round 2 personalized recommendations
- `README.md`: Max 1-page explanation:
    - What logic was used for content filtering?
    - How did you simulate collaborative filtering?
    - How is the user profile being stored/updated?

✅ How You Can Evaluate Easily

You just need to run the script and check:

- Do Round 2 recommendations logically evolve from Round 1?
- Are their explanations clear in `README.md`?
- Is there **code clarity**, even if it's basic?

Bonus points if:

- They modularize their logic (functions/classes)
- They use basic visualization to show user profile growth
- They simulate more than one user to show collaborative logic

---

✨ Sample Output (Ideal Flow)

```
--- ROUND 1: Cold Start Recommendations ---
Recommended: [Product 12, Product 5, Product 8]

User clicked: [Product 5, Product 8]

--- UPDATED USER PROFILE ---
Preferred tags: ['boho', 'pastel', 'summer']
Liked categories: ['tops', 'skirts']

--- ROUND 2: Personalized Recommendations ---
Recommended: [Product 17 (based on profile), Product 2 (popular),
Product 6 (content match)]
```

NOTE:

Please don’t just copy paste from ChatGPT and give us. Just do it by yourself or take ideas from GPT. But we don’t want assignments which are directly chat gpt copy pasted.