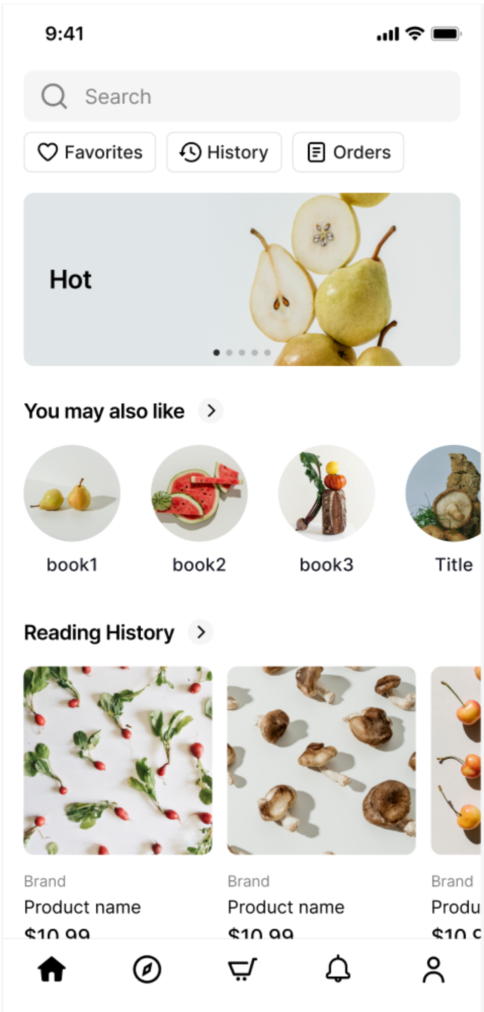
Z5526892
Xinghua Wu

Project Design

1.Scope

This project aims to design a personalized e-book recommender system tailored for book enthusiasts. The goal is to provide accurate book recommendations based on user reading habits.

- On mobile devices, the system recommends 3 books across all categories in a single view.
- On the web platform, it recommends 3 books per category with multi-category display.
- A simulated mobile UI is illustrated below,



showing book details with functionalities such as rating, like, dislike, save-to-favorites, and “not interested” tagging.

In a dynamic environment:

- User actions such as rating, reviewing, and bookmarking are incrementally updated into the interaction matrix.
- When a new book is added, an embedding vector is generated immediately.
- In the knowledge graph, the edge weights are dynamically updated based on new user behaviors.

To address the cold-start problem, the system supports:

- Interest-based onboarding questions during registration (e.g., selecting favorite genres, authors, or booklists), which are mapped to structured tags.
- Or, a LLM-driven conversational onboarding, which summarizes user preferences through interactive dialogue.

The system may adopt a monetization model based on either:

- a 5% commission per e-book purchase, or
- a subscription membership fee.

2. Dataset

We utilize the following datasets:

- Goodreads Book Graph (~2.3 million books, 6 million ratings)
- Project Gutenberg (free book texts)
- OpenLibrary API/dumps (book metadata and tags)

Module	Required Fields	Dateset
Collaborative Filtering (CF)	user_id, item_id (book_id or ISBN), rating or implicit behavior (clicks, favorites)	goodreads_interactions.csv, BX-Book-Ratings.csv, amazon_reviews
Content-Based Recommendation (CBF)	title, description/full_text, genres/tags, review_text	goodreads_books.json.gz, Gutenberg full text, Amazon reviews
Knowledge Graph (KG)	author, genre, tags, publisher, series, publication_year; named entities	goodreads_book_authors.json.gz, OpenLibrary subjects, user-defined triplets

Limitations

- Goodreads:
    - The description field is inconsistent in quality—some entries are too short or missing.
    - Tags (genres / shelves) are crowd-sourced, resulting in redundancy and ambiguity.
- Project Gutenberg:
    - Lacks user interactions (no ratings or clicks).
    - Full texts are lengthy and unstructured, requiring preprocessing for downstream tasks.

3.Methods

The system adopts a multi-source hybrid recommendation approach:

- Collaborative Filtering: Utilizes user-user or item-item similarity based on historical interactions.
- Content-Based Filtering: Embeddings of book texts/descriptions are used for semantic similarity.
- Knowledge Graph Reasoning:
    - Construct a user-specific subgraph based on reading history.
    - Use it for:
        - Direct entity recall (e.g., connected books, themes).
        - Feature augmentation for other recall paths (e.g., tag-based recall).
- Multi-path Recall Merging:
    - Integrate results from CF, CBF, and KG-based recall.
    - Use LLMs for reranking, to form the final personalized recommendation list.

Throughout the pipeline, LLMs are used for:

- User modeling (preference extraction, cold-start profiling, interest summarization)
- Content understanding (embedding generation, tag extraction)
- Graph construction
- Reranking and explanation generation

4. Evaluation

1) Model Evaluation (Offline)

We evaluate recommendation quality using:

- Precision@K & Recall@K: Accuracy and coverage of the top-K recommendations.
- NDCG (Normalized Discounted Cumulative Gain): Accounts for position in the ranked list.
- Coverage: Measures how diverse the recommended content is.
- MAP（Mean Average Precision）
- Hit Rate

We adopt a user-based leave-one-out strategy:

For each user, one known interaction is held out as the test case.

2) User-Centric Evaluation (Simulated Real-World Use)

Beyond algorithmic metrics, user experience is key. We evaluate through:

- User behavior feedback:
  - Rating frequency
  - Save-to-favorites ratio
  - Book detail view rate
- Subjective feedback (via post-test questionnaire):
  - Relevance of recommendations
  - Diversity and satisfaction
  - Willingness to use the system long term
  - Suggestions for improvement

3) Trade-off Considerations

We aim to balance accuracy and diversity:

- Higher precision may lead to content homogeneity.
- Excessive diversity can reduce personalization.
- We will adopt a weighted metric ensemble to balance both aspects.
- We employ Precision@5 and NDCG as primary evaluation metrics to assess the accuracy of recommendations. Additionally, we consider the diversity and coverage of the recommended content to evaluate the system's exploration

capability, aiming to achieve an effective balance between personalization and novelty.

4) Efficiency and Responsiveness

The system uses an offline-online hybrid architecture:

- Offline:
  - Periodic training of user/item embeddings
  - Update knowledge graph structure
- Online:
  - Real-time vector-based nearest-neighbor recall (e.g., via FAISS)
  - Near-instantaneous Top-N recommendations

5) User Simulation Study

We designed a simulated user interface to demonstrate the recommendation workflow:

1. The user receives a recommendation list.
2. Interacts with a book (e.g., clicks, likes, skips).
3. Provides feedback via questionnaire, including:
   - "Did the recommendations match your interest?"
   - "How would you rate the recommendation quality?"
   - "Would you use this system regularly?"
   - "What improvements would you suggest?"