untitled

High-Level Design (HLD): Movie Recommendation System

1. **User Interface (UI):**
    - The system utilizes a web-based graphical user interface (GUI) developed using Flask for server-side rendering of HTML templates.
    - Users interact with the application through a browser to select the type of movie recommendation.
2. **Backend (Flask):**
    - The Flask framework is employed to handle routing, form submissions, and rendering HTML templates.
    - Different routes are defined for each recommendation type.
3. **Data Processing:**
    - Pandas library is used for data processing and manipulation.
    - Movie and rating data are loaded from CSV files, providing the necessary input for recommendation algorithms.
4. **Recommendation Modules:**
    - **Popularity-Based:**
        - Utilizes the `popularity_recommender` function in `app.py`.
        - Filters movies by genre and ratings threshold, providing recommendations based on popularity.
    - **Content-Based:**

        - Relies on the `content_recommender` function in `app.py`.
        - Implements a content-based recommendation algorithm based on movie genres.
    - **Collaborative-Based:**
        - Utilizes the `collaborative_recommender` function in `app.py`.
        - Implements collaborative filtering to suggest movies based on user similarity.

5. **HTML Templates:**
    - HTML templates are used for the frontend to present a visually appealing and user-friendly interface.
    - Bootstrap is employed for responsive design and styling.
6. **Main Application (main.py):**
    - Handles the main application logic, including route definitions and form submissions.
    - Renders the main page allowing users to select the type of recommendation.
7. **External Recommendation Functions (app.py):**
    - Contains the recommendation functions (`popularity_recommender`, `content_recommender`, `collaborative_recommender`).
    - These functions provide the core logic for generating movie recommendations.
8. **Template Files (in the 'templates' directory):**
    - `main.html`: The main page for users to choose the recommendation type.

- `result_popularity.html` : Displays Popularity-Based recommendations.
- `content_result.html` : Displays Content-Based recommendations.
- `collaborative_result.html` : Displays Collaborative-Based recommendations.

9. **Bootstrap and jQuery:**
   - Bootstrap is included for styling and responsive design.
   - jQuery is used to enable Bootstrap functionality.
10. **Dependencies:**
    - Flask, Pandas, Bootstrap, jQuery, and other required libraries are used.
    - External libraries handle collaborative filtering and content-based algorithms.
11. **Testing:**
    - The system should be thoroughly tested for different scenarios, ensuring accurate and reliable movie recommendations.
12. **Scalability and Maintenance:**
    - The modular design allows for easy addition of new recommendation algorithms.
    - Codebase organization facilitates maintainability and future enhancements.
13. **Deployment:**
    - The application can be deployed locally or on a server for wider access.