**Design Document for AI Book Recommender**

**Introduction**

The AI Book Recommender project aims to address the need for book recommendations based on user-specified genres. By leveraging the Google Books API and a text generation pipeline, the application fetches and filters top books, ultimately helping users find the best book for their needs. The primary goal is to enhance user experience by providing a seamless interface for discovering and selecting books.

**Objective**

To develop an application that fetches the top 100 books in a specified genre and narrows down to the top 10 books, helping users select the best book from the list.

**Approach**

1. **Use of Google Books API**:
   * The Google Books API is a robust and comprehensive source of book information. It allows querying books by genre and fetching detailed information about them, making it a suitable choice for this project.
2. **Streamlit for UI Development**:
   * Streamlit is chosen for its simplicity and rapid development capabilities. It allows the creation of interactive web applications with minimal code, ensuring a user-friendly interface for the book recommender system.
3. **Session State Management**:
   * Streamlit's session state is utilized to manage and store the fetched book data and user inputs. This ensures that the state of the application is preserved across different user actions (e.g., fetching books, filtering top books).
4. **Pipeline Initialization with Hugging Face**:
   * Although not directly used in the current implementation, the Hugging Face text generation pipeline is initialized for potential future enhancements, such as generating summaries or reviews for the books.

**Workflow**

1. **User Input**:
   * The user inputs a genre in the text field.
2. **Fetching Top 100 Books**:
   * On clicking "Fetch Top 100 Books", the application sends a request to the Google Books API, fetching up to 100 books based on the specified genre. The results are stored in the session state.
3. **Displaying Top 100 Books**:
   * The fetched books are displayed to the user with an index.
4. **Filtering Top 10 Books**:
   * On clicking "Filter Top 10 Books", the top 10 books from the fetched list are displayed.
5. **Selecting and Displaying the Top Book**:
   * The application automatically selects the top book from the filtered list and displays it to the user, concluding with a thank you message.

**Design Decisions**

1. **Use of Google Books API**:
   * The decision to use Google Books API was driven by its extensive database and reliability. It provides a wide range of books and detailed metadata, making it an ideal choice for this application.
2. **Streamlit**:
   * Chosen for its ability to quickly develop and deploy web applications, Streamlit provides an easy-to-use interface for users and allows seamless interaction with the backend logic.
3. **Automatic Top Book Selection**:
   * To enhance user experience and reduce decision fatigue, the application automatically selects and displays the top book after filtering the top 10 books. This decision streamlines the user journey and provides immediate value.
4. **Simplified User Interaction**:
   * The user interface is designed to be intuitive and straightforward, with minimal steps required to fetch and view the top book recommendations.

**Conclusion**

The AI Book Recommender application leverages the strengths of the Google Books API and Streamlit to provide a user-friendly platform for book discovery. The design choices ensure a seamless user experience, from fetching a broad list of books to narrowing down to the best book in the specified genre. This approach addresses the need for efficient and accurate book recommendations, enhancing the overall user engagement with the application.