## **Design Document**

### 

### **Project Overview**

The Book Recommendation System is a web application built using Streamlit that recommends books based on user-specified genres and preferences. It leverages the Huggingface API to generate lists of popular books and refines the list based on user preferences to provide a personalized recommendation.

### **Goals and Objectives**

1. **User-Friendly Interface**: Provide a simple, interactive UI for users to input genres and preferences.
2. **Accurate Recommendations**: Use AI tools to generate, filter, and refine book lists to recommend the best books based on user input.
3. **Scalability**: Ensure the system can handle various genres and user preferences without performance degradation.

### **Components and Functionality**

1. **GenreSearchTool**:
   * **Purpose**: Fetches a list of 100 popular book titles for a given genre.
   * **Method**: Uses the Huggingface API to generate text based on the user's genre input.
2. **TopTenFilterTool**:
   * **Purpose**: Filters the list of 100 books to select the top 10 most popular books.
   * **Method**: Uses another Huggingface API prompt to refine the list.
3. **UserPreferenceTool**:
   * **Purpose**: Selects the best book from the top 10 list based on user-specified reading preferences.
   * **Method**: Generates a prompt including the user's preferences and the list of top 10 books to get a personalized recommendation.

### 

### 

### 

### 

### 

### **Design Decisions**

1. **Use of Huggingface API**:
   * **Reason**: Huggingface provides a robust API for natural language processing and text generation, which is essential for generating and refining book lists.
   * **Benefit**: Leveraging a powerful NLP model ensures the recommendations are relevant and high-quality.
2. **Streamlit for UI**:
   * **Reason**: Streamlit allows for rapid development of web applications with a focus on simplicity and interactivity.
   * **Benefit**: Streamlit's intuitive design makes it easy for users to interact with the application and get real-time results.
3. **Environment Variables**:
   * **Reason**: Securely store and access the Huggingface API token.
   * **Benefit**: Keeps sensitive information out of the source code and allows for easy configuration.

### 

### **Future Improvements**

1. **Enhance the Recommendation Algorithm**: Incorporate more sophisticated algorithms to improve recommendation accuracy.
2. **User Authentication**: Allow users to create accounts and save their preferences and reading history.
3. **Feedback Mechanism**: Collect user feedback to further refine and improve recommendations.