Good Old Friend – Simplifying Book Recommendation

ON

Submitted in partial fulfillment of the requirements of the degree of

Bachelor of Engineering

(Information Technology)

By

Name: - Shreyash Ganesh Dhekane

Roll: No: - 16

WebX CA Mini Project, Third Year (Semester-VI)

Under the guidance of

Prof. Dipti Karani

Department of Information Technology

Vivekanand Education Society's Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai

2024-2025

Vivekanand Education Society's
Institute of Technology

(Autonomous Institute Affiliated to University of Mumbai, Approved by AICTE & Recognised by Govt. of Maharashtra)
NAAC accredited with 'A' grade

V.E.S.
Since 1962

Certificate

This is to certify that **Mr. Shreyash Ganesh Dhekane Roll No.16** have completed the project report on the topic **Good Old Friend – Simplifying Book Recommendation** satisfactorily in partial fulfilment of the requirements for the award of Mini Project in WebX Lab of third Year, (Semester-VI) in Information Technology under the guidance of Mrs. Dipti Karani during the year 2024-2025 as prescribed by An Autonomous Institute Affiliated to University of Mumbai

Mrs. Dipti Karani

Supervisor/ Examiner

Table of Contents

Chapter 1: Introduction

1.1 Introduction

In today's digital era, the vast amount of information and resources available online has made discovering relevant content a daunting task. This problem is especially noticeable in the domain of books, where readers often struggle to find titles that match their interests, preferences, and past reading habits. Recommendation systems offer a solution by filtering and presenting personalized suggestions based on user behavior and item features.

This project presents a **Flask-based Book Recommendation System** that aims to enhance the user experience by providing relevant book suggestions using a **content-based filtering algorithm**. The system uses metadata from books such as title, author, and publisher to identify similar titles and recommend them to the user. Along with book recommendation features, the platform also includes user authentication through MongoDB Atlas, and a secure registration process that incorporates OTP (One-Time Password) verification via email to ensure the authenticity of users.

The platform is designed with a clean and responsive interface using Flask, and is integrated with machine learning models that utilize natural language processing (NLP) techniques to compute similarities between books. A structured database of books along with associated metadata and cover images is used for the recommendation engine. This system not only helps users discover books tailored to their tastes but also provides features like book upload, deletion, and a contact form for feedback.

1.2 Objective

The primary objectives of this project are:

- To design and implement a personalized book recommendation system using content-based filtering techniques.
- To create a user-friendly web interface that allows users to register, log in, receive recommendations, and interact with the platform.
- To implement user authentication and account management using MongoDB Atlas, providing scalability and cloud-based database access.
- To enhance security and trust by integrating OTP-based email verification during the user registration process.
- To allow users to contribute to the platform by uploading new books, and manage them via update/delete functionalities.
- To ensure the application is scalable, responsive, and easy to deploy, making it suitable for real-world use cases.

1.3 Motivation

The motivation behind this project stems from the increasing difficulty users face in choosing books due to the overwhelming number of titles available across genres and platforms. Existing recommendation systems used by major platforms often come with commercial interests, lacking transparency and flexibility for academic or small-scale deployment.

This project was envisioned to solve the following key problems:

- **Information overload**: Help users make better reading decisions by narrowing down options based on their preferences.
- **Lack of personalized systems in academia**: Develop a scalable and customizable recommendation platform suitable for learning and experimental purposes.
- **Secure and modern user management**: Leverage MongoDB Atlas for seamless cloud database integration and OTP-based email verification to ensure secure user onboarding.

Furthermore, as engineering students with an interest in machine learning and web development, we were driven to create a full-stack application that integrates practical concepts from both fields, giving us hands-on experience with real-world technologies and challenges.

1.4 Scope of the Work

The scope of this project covers the following key areas:

- **User Authentication and Management**:
  - Secure sign-up and login functionality using MongoDB Atlas.
  - Profile image upload and account update features.
  - Email OTP verification during registration to verify authenticity and prevent spam.

- **Book Recommendation Engine**:
  - Utilizes content-based filtering and NLP techniques (TF-IDF/Count Vectorizer and Cosine Similarity).
  - Recommends top 10 books based on user input.

- **Dynamic Web Interface**:
    - Built with Flask framework, rendering data dynamically using Jinja2 templating.
    - Interactive features like uploading new books, deleting existing ones, and displaying random book picks on the homepage.

- **Data Handling**:
    - Structured CSV datasets for books and their cover images.
    - Support for adding and deleting book entries dynamically through forms.

- **Email Communication & Contact Page**:
    - A dedicated contact page for users to submit feedback or queries.
    - Integration with email service for OTP and communication support.

The project is modular, scalable, and designed to be deployed in an academic or small business setting. Future work could include collaborative filtering, user review integration, and improved UI/UX with modern frontend frameworks.

Chapter 2: Literature Review

2.1 Introduction

In the ever-expanding digital ecosystem, information overload is a challenge faced by internet users, especially when it comes to making decisions on what to read, watch, or buy. This issue is most prevalent in platforms offering books, movies, and other media, where users have a wide variety of choices, often leading to analysis paralysis. A **Recommendation System** serves as a potential solution to this problem by offering personalized suggestions based on the users' previous preferences, behavior, and interactions.

Recommendation systems are widely used in e-commerce, streaming services, and educational platforms to provide content discovery. Content-based filtering (CBF), collaborative filtering (CF), and hybrid models are the core approaches in recommendation system research. The implementation of these systems has seen significant growth in recent years due to advancements in machine learning, natural language processing, and database management.

This chapter explores existing literature surrounding recommendation systems, particularly in the context of books. It will also define the challenges and gaps in existing systems that this project aims to address, emphasizing the need for secure and scalable solutions, such as MongoDB Atlas for database management and OTP verification for user registration.

2.2 Problem Definition

The rapid digitization of content has resulted in a surge of book titles available across multiple platforms. As a result, users often find it difficult to discover books that match their individual preferences. Traditional approaches to discovering books, such as browsing or searching through online databases, may not provide the level of personalization required. This leads to the following problems:

1. **Information Overload**: Users are overwhelmed with countless book options, which makes it difficult for them to choose a book that fits their interests and preferences.
2. **Lack of Personalization**: Many platforms lack intelligent algorithms that personalize recommendations based on the user's past behavior or preferences.
3. **Data Management and Security**: Ensuring user security and providing an intuitive registration process is a key challenge. Many systems lack proper authentication and security measures, resulting in potential risks related to user data.
4. **Scalability Issues**: Handling large amounts of data from a growing number of users and book titles demands an efficient and scalable database system. Traditional relational databases can struggle to keep up with the volume, which is why this project uses MongoDB Atlas as the solution.
5. **User Engagement and Trust**: The absence of robust security features like OTP-based email verification makes many platforms susceptible to fake registrations

and fraud.

This project aims to solve these problems by developing a book recommendation system that utilizes MongoDB Atlas for secure user authentication, employs a content-based filtering approach for book recommendations, and implements OTP verification during user registration to improve system security and reduce fraud.

2.3 Review of Literature Survey

Area of Study	Study/Reference	Key Insights	Application to the Project
Book Recommendation Systems	Melville et al. (2002)	Content-Based Filtering (CBF) recommends items based on item features (e.g., book metadata).	Used CBF approach in the project for recommending books based on metadata (e.g., author, publisher).
	Su & Khoshgoftaar (2009)	Collaborative Filtering (CF) uses user behavior to recommend items. However, suffers from the "cold start" problem.	Not implemented in this project due to the cold start issue; the project uses content-based filtering instead.
	Burke (2002)	Hybrid recommendation systems combine multiple algorithms to enhance recommendation accuracy.	The project uses a simple content-based approach for efficiency but recognizes the potential of hybrid methods.
Security in Web Applications	Bonneau et al. (2015)	Traditional password-based authentication is vulnerable to various attacks. Multi-factor authentication improves security.	The project uses OTP-based authentication to provide an extra layer of security during user registration.
	Juels & Pappu (2003)	OTP (One-Time Password) systems offer better security than traditional passwords by sending temporary codes via email.	The project integrates OTP verification for secure user registration, improving account protection.
MongoDB Atlas for Authentication	MongoDB Documentation	MongoDB Atlas offers scalability, security, and flexibility for large datasets with unstructured data.	The project uses MongoDB Atlas for user authentication, taking advantage of its scalability, security, and ease of use.

Chapter 3: Design and Implementation

3.1 Introduction

The design and implementation phase of the **Book Recommendation System** is crucial for translating the project's objectives into functional, real-world software. This chapter will describe the overall design approach, the architecture, and the required system specifications. Additionally, it includes setup instructions for deploying both the frontend and backend systems and explains the directory structure to organize the project efficiently.

3.2 Proposed Design

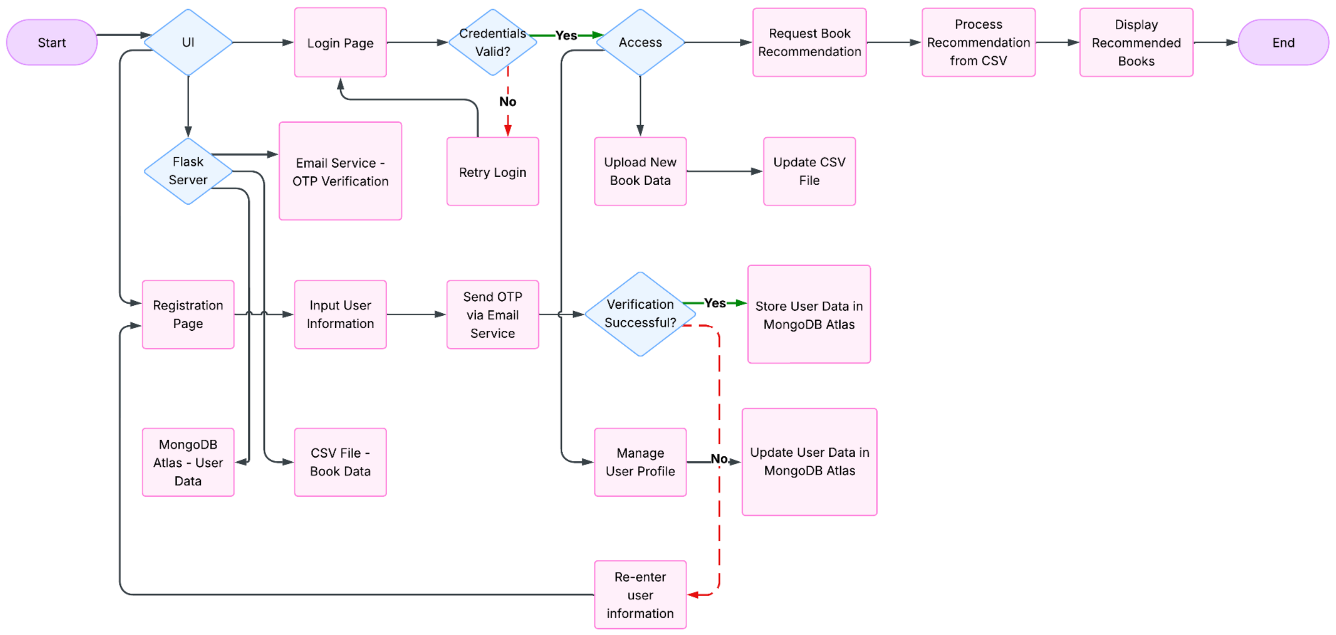
The Book Recommendation System has been designed with user interaction and experience at its core, ensuring an easy-to-use interface while providing a robust backend to handle operations such as user authentication, book data management, and personalized recommendations.

- **Frontend Design**:
    - The frontend will use HTML and CSS to create an interactive and responsive user interface.
    - Users will interact with a registration page, login page, book recommendation page, and an account management page.
    - For displaying book recommendations, data will be retrieved from the backend, processed for recommendations, and presented in a clear format.

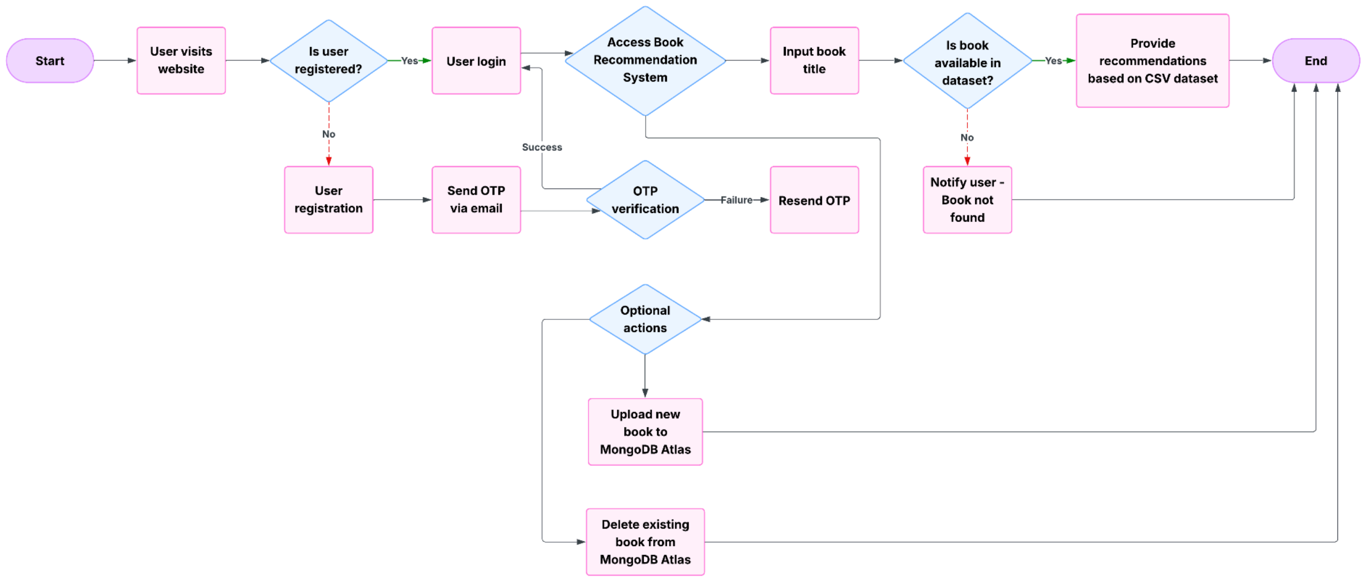
- **Backend Design**:
    - Flask serves as the backend framework, handling user authentication, OTP verification, data storage, and recommendations.
    - MongoDB Atlas is used for storing user data, including email and hashed passwords for authentication, ensuring scalability and security.
    - The recommendation engine uses a content-based filtering model, where users can input a book name, and the system will recommend similar books from a pre-existing dataset.
    - OTP verification is incorporated during user registration to enhance security and prevent unauthorized access.

3.3 Architecture Diagram

- **System Architecture**


Start
UI
Login Page
Credentials Valid?
Yes
No
Access
Request Book Recommendation
Process Recommendation from CSV
Display Recommended Books
End
Flask Server
Email Service - OTP Verification
Retry Login
Upload New Book Data
Update CSV File
Registration Page
Input User Information
Send OTP via Email Service
Verification Successful?
Yes
Store User Data in MongoDB Atlas
MongoDB Atlas - User Data
CSV File - Book Data
Manage User Profile
No
Update User Data in MongoDB Atlas
Re-enter user information

- **Basic Workflow**


Start
User visits website
Is user registered?
Yes
User login
Access Book Recommendation System
Input book title
Is book available in dataset?
Yes
Provide recommendations based on CSV dataset
End
No
User registration
Send OTP via email
Success
OTP verification
Failure
Resend OTP
No
Notify user - Book not found
Optional actions
Upload new book to MongoDB Atlas
Delete existing book from MongoDB Atlas

3.4 System Requirements: - Hardware & Software

Hardware Requirements:

- **Server/Local Machine**:
  - Processor: Minimum 2.0 GHz Dual-Core Processor
  - RAM: Minimum 4GB of RAM
  - Disk Space: At least 1GB of available disk space
  - Network: Stable internet connection for MongoDB Atlas connectivity

Software Requirements:

- **Frontend**:
  - HTML, CSS, JavaScript
  - Flask framework (for the integration with Python)
- **Backend**:
  - Python 3.x (For Flask)
  - Flask framework
  - MongoDB Atlas (for cloud-based database management)
  - Necessary Python packages (Flask, Flask-Mail, Flask-Login, Flask-WTF, Flask-SQLAlchemy, Pandas, NumPy)
- **Database**:
  - MongoDB Atlas (for cloud-based database)
- **Development Tools**:
  - Code editor: VS Code, PyCharm, Sublime Text, or any IDE of choice
  - Terminal (for command-line operations)

3.5 Setup Instructions

Frontend Setup (HTML, CSS, Flask)

Steps:

1. **Install Python:**Download from: https://www.python.org/downloads/
   Verify installation: python --version

2. **Navigate to frontend directory:**
   cd frontend

3. **Install Python dependencies:**
   pip install -r requirements.txt

4. **Run the Flask development server:**
   python app.py

5. **Frontend will be accessible via:**
   http://localhost:5000

Backend Setup (Flask - Python)

Steps:

1. **Install Python:**Download from: https://www.python.org/downloads/
   Verify installation: python --version

2. **Navigate to backend directory:**
   cd backend

3. **Create a virtual environment:**
   python -m venv venv

4. **Activate virtual environment:**
   Windows: venv\Scripts\activate

5. **Install Python dependencies:**
   pip install -r requirements.txt

6. **Configure environment variables:**
   - Create a .env file in the backend directory.
   - Add MongoDB URI and secret keys:
     MONGO_URI=mongodb+srv://<username>:<password>@cluster.mongodb.net/<your-database>
     SECRET_KEY=your-secret-key

7. **Start the Flask server:**
   python app.py

8. **Backend will run on:**
   http://localhost:5000

Database Setup (MongoDB Atlas)

Steps:

1. **Set up a cluster**:
   - Create a new project and cluster in MongoDB Atlas.
   - Choose the free-tier cluster.

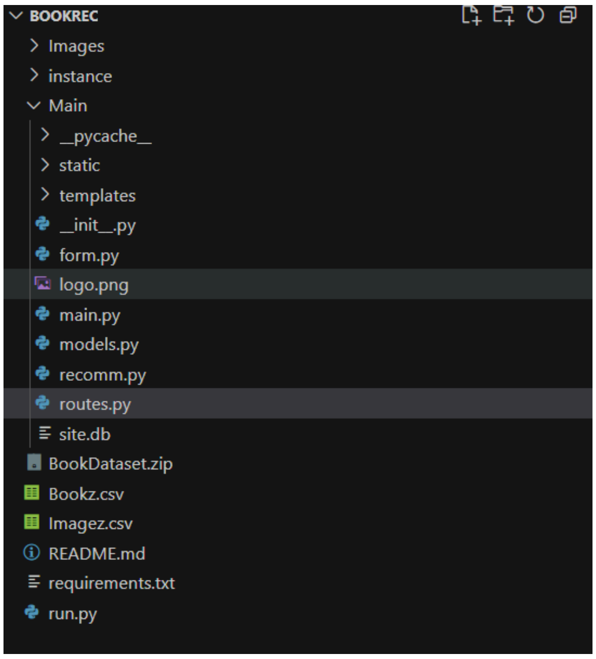
2. **Configure security settings**:

   - Add a database user and configure access permissions.
   - Whitelist your current IP address.

3. **Create a new database**:
   - Name: `bookRecommendationDB`

4. **Get the connection string**:
   mongodb+srv://<username>:<password>@cluster.mongodb.net/bookRecommendationDB

3.6 Project Directory Structure



```
BOOKREC
  > Images
  > instance
  v Main
    > __pycache__
    > static
    > templates
      __init__.py
      form.py
      logo.png
      main.py
      models.py
      recomm.py
      routes.py
      site.db
  BookDataset.zip
  Bookz.csv
  Imagez.csv
  README.md
  requirements.txt
  run.py
```

Chapter 4: Results and Discussions

4.1 Introduction

The Results and Discussions chapter provides an analysis of the outcomes achieved after implementing the Book Recommendation System. It highlights the key functionalities and features implemented in the system, as well as the overall performance. This chapter also evaluates the effectiveness of the recommendation system, the accuracy of user authentication, and the implementation of OTP verification. In addition, the analysis will also cover potential challenges faced and how they were addressed during the project development.

4.2 Github Link

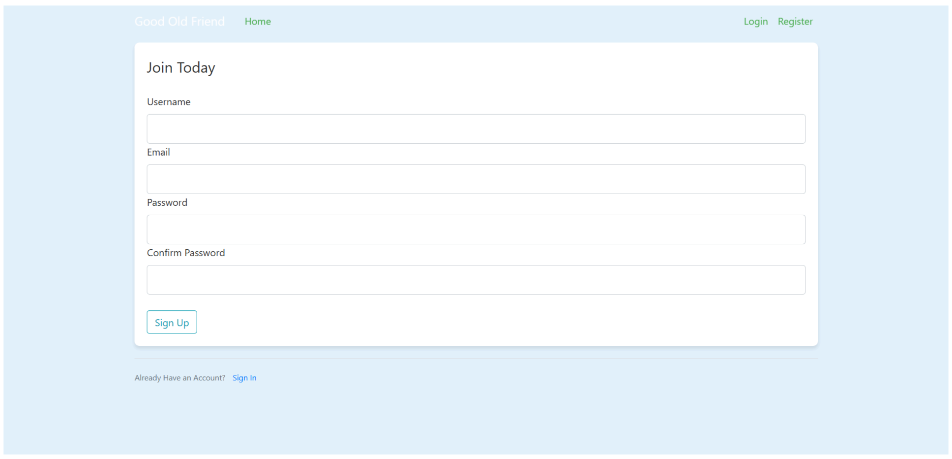
The complete source code and detailed instructions for the Book Recommendation System are available on GitHub. You can access the repository to explore the code, issues, pull requests, and further documentation for the project.

- **GitHub Repository Link**: https://github.com/shreyashdhekanevesit/webxca2.git

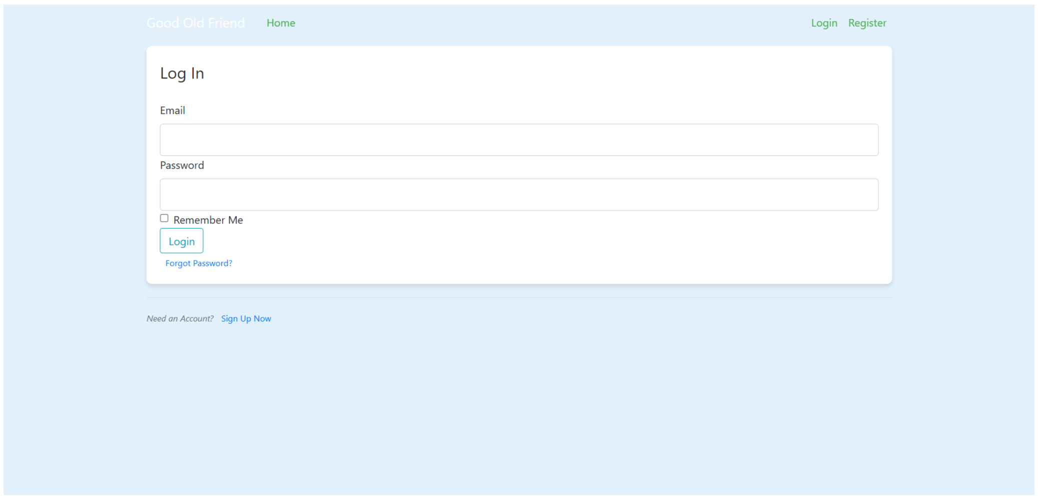
The repository contains the following:

- **Frontend**: The HTML, CSS, and JavaScript files required to run the user interface.
- **Backend**: The Flask-based code responsible for handling logic, user authentication, and OTP verification.
- **Database Setup**: Configuration details for MongoDB Atlas integration.
- **Dependencies**: A list of required libraries and modules specified in the `requirements.txt`.

4.3 Results of Implementation



Registration page for the user to register himself on the platform using email id


Good Old Friend Home Login Register

Log In

Email

Password

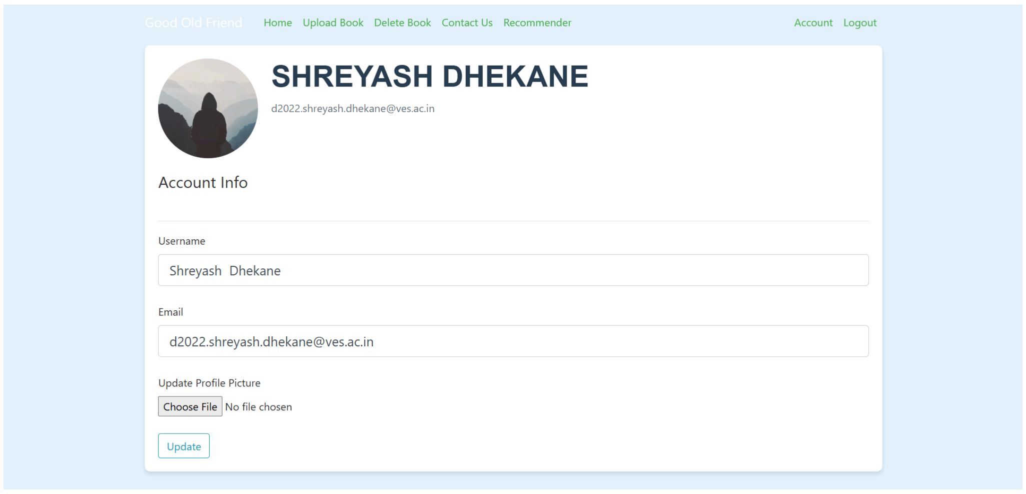
Remember Me

Login

Forgot Password?

Need an Account? Sign Up Now

Login Page to login after creation of profile after registration


Good Old Friend Home Upload Book Delete Book Contact Us Recommender Account Logout

SHREYASH DHEKANE

d2022.shreyash.dhekane@ves.ac.in

Account Info

Username

Shreyash Dhekane

Email

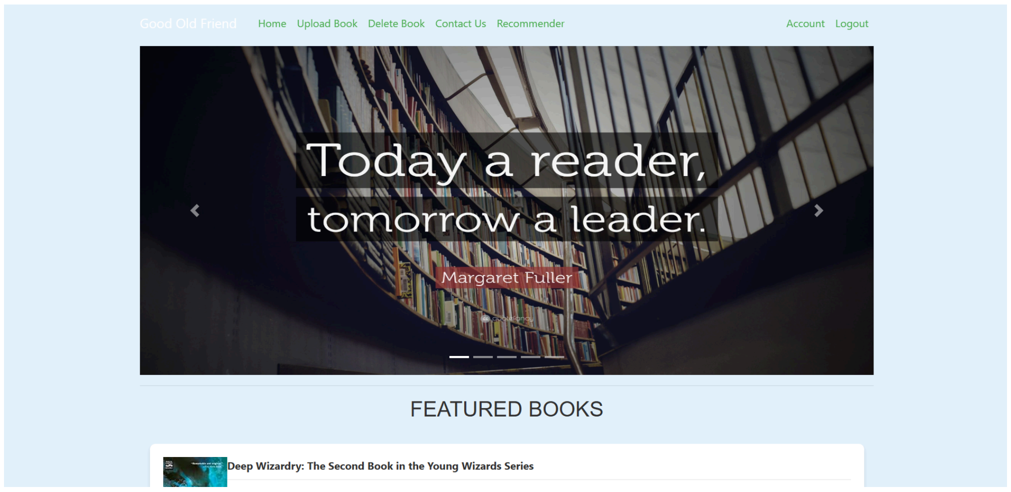
d2022.shreyash.dhekane@ves.ac.in

Update Profile Picture

Choose File No file chosen

Update

Account page to update information about the user


Good Old Friend Home Upload Book Delete Book Contact Us Recommender Account Logout

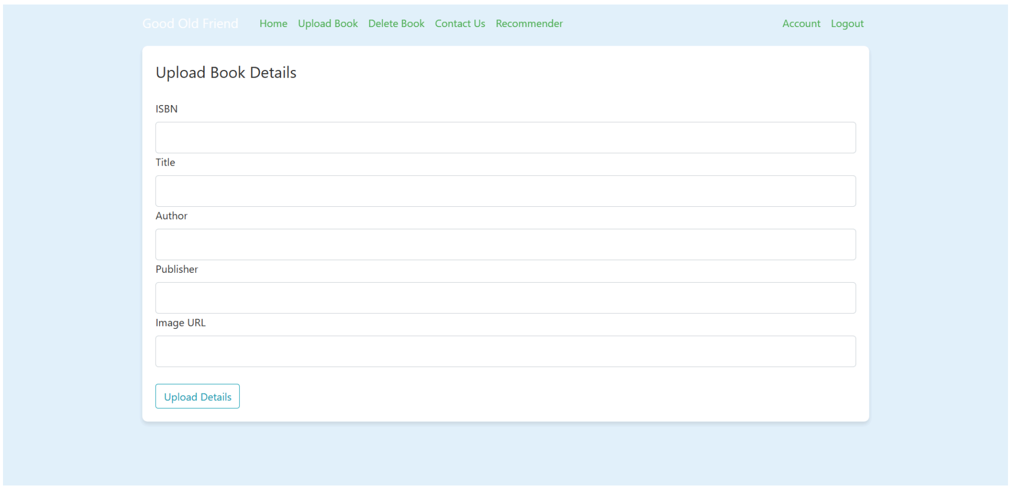
Today a reader, tomorrow a leader.

Margaret Fuller

FEATURED BOOKS

Deep Wizardry: The Second Book in the Young Wizards Series

Homepage to see general recommendations of books for visitors



Upload Book Details

ISBN

Title

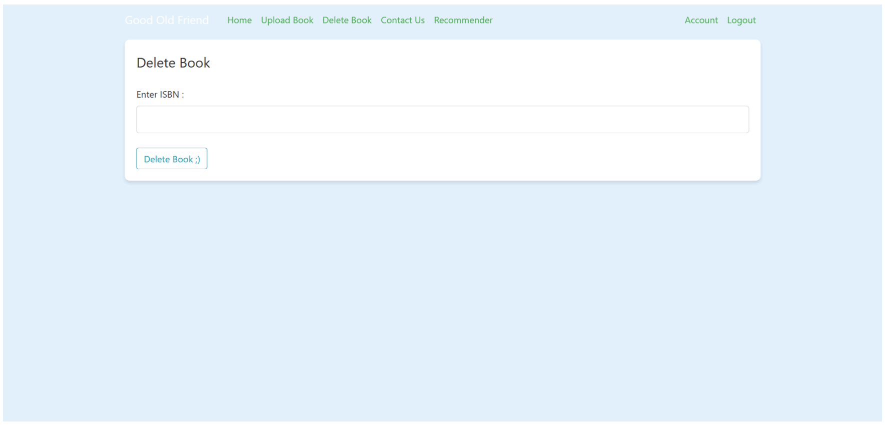
Author

Publisher

Image URL

Upload Details

Functionality to add books for new publications

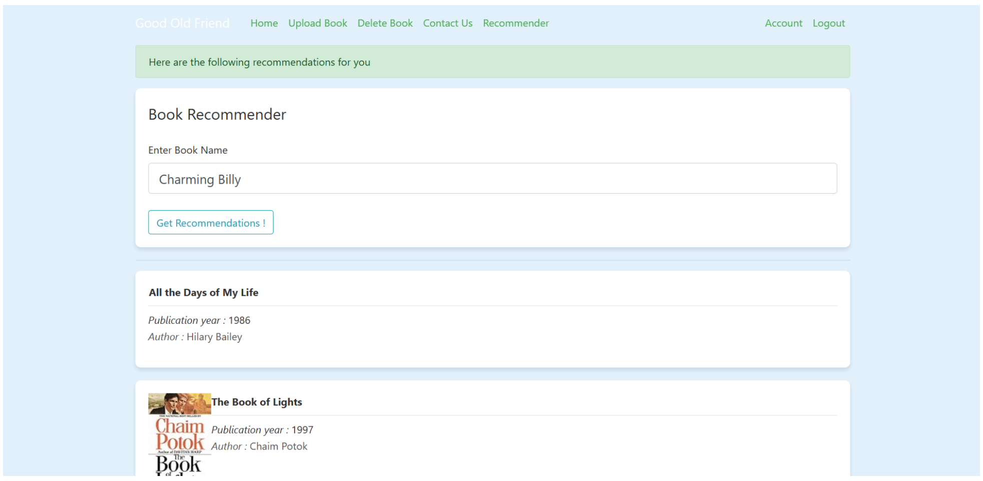


Delete Book

Enter ISBN :

Delete Book ;)

Delete listing of books that are irrelevant to current scenario



Here are the following recommendations for you

Book Recommender

Enter Book Name

Charming Billy

Get Recommendations !

All the Days of My Life

Publication year : 1986
Author : Hilary Bailey

The Book of Lights

Publication year : 1997
Author : Chaim Potok

book recommendation page that gives 10 similar books to the book you enter (using cosine similarity)

4.4 Results Analysis

The results of the implementation were evaluated based on various factors:

1. **User Authentication and OTP Verification**:
    - The system's authentication mechanism using **MongoDB Atlas** worked smoothly. Upon user registration, an OTP was successfully sent to the email address provided by the user.
    - The OTP verification process was effective in ensuring that only legitimate users could create accounts.
2. **Recommendation Engine**:
    - The **book recommendation system** was able to return relevant books based on the input provided by the user.
    - **Accuracy**: The recommendations were generally accurate due to the content-based filtering model, which considers book metadata such as title and author.
    - **Response Time**: The response time for fetching recommendations was quick, ensuring a good user experience.
3. **Database Management**:
    - **MongoDB Atlas** provided reliable and efficient storage for user data and book information, ensuring that both user details and book recommendations could be retrieved with minimal latency.
    - **Scalability**: The system is designed to handle an increasing number of users and books, as **MongoDB Atlas** provides cloud-based scalability.
4. **Frontend and User Interface**:
    - The user interface was intuitive and responsive, offering a smooth user experience for different devices. Users could easily navigate through the pages and access features such as registration, login, and recommendations.
5. **Challenges and Limitations**:
    - **Challenge 1**: One of the challenges faced during the implementation was integrating OTP verification with the backend and ensuring the security of user data. This was resolved by using **Flask-Mail** for sending OTPs and ensuring proper encryption techniques were implemented.
    - **Challenge 2**: Another challenge was ensuring the recommendation engine was fast and efficient. To address this, we optimized the querying mechanism in the backend and used efficient data structures for storing book information.

6. **Future Improvements**:

    - **Accuracy of Recommendations**: Future work can explore the implementation of a hybrid recommendation system that combines content-based filtering with collaborative filtering to improve recommendation accuracy.
    - **Advanced OTP Features**: Integration of multi-factor authentication (MFA) could further enhance the security of the user authentication system.
    - **UI Enhancements**: The user interface can be further refined to include more interactive elements and better user feedback mechanisms.

This chapter concludes the results and discussions related to the **Book Recommendation System**. The system successfully meets the objectives set forth at the beginning of the project, offering a functional recommendation engine and secure user authentication. Future improvements could enhance the system's performance and usability further.

Chapter 5: Conclusion and Future Scope

5.1 Conclusion

The Book Recommendation System has been successfully implemented using Flask for the backend and MongoDB Atlas for the database, with additional functionality for secure user authentication and OTP verification. The system was designed to provide users with personalized book recommendations based on their input, leveraging content-based filtering techniques. This project addressed key requirements such as ensuring security through OTP verification and offering a user-friendly interface for easy interaction with the system.

The implementation of MongoDB Atlas allowed for efficient data storage and scalability, while the Flask framework provided a lightweight yet powerful backend. The integration of user authentication with OTP verification ensures that only legitimate users can access the system, further enhancing the security of the platform.

The recommendation engine provided relevant results based on book input, and the system was able to handle both user management and book data efficiently. Overall, the project met its objectives by offering a functional book recommendation service that is both secure and scalable.

5.2 Future Scope

While the current Book Recommendation System serves its purpose effectively, there are several opportunities to enhance and expand its functionality. Below are some potential improvements and future developments:

1. **Enhanced Recommendation System**:
    - Hybrid Recommendation Model: Currently, the system uses a content-based filtering approach. Future improvements could involve incorporating collaborative filtering or a hybrid model that combines both content-based and collaborative filtering. This would improve the recommendation accuracy by analyzing user behavior and preferences.
    - Deep Learning Models: Integration of machine learning models such as neural networks for book recommendation could offer more precise results, especially for complex datasets.

2. **Personalized User Experience**:
    - User Profiles: Creating detailed user profiles based on their reading history, preferences, and ratings can help generate more personalized recommendations.
    - User Feedback Loop: Allowing users to provide feedback on the recommendations (e.g., like/dislike or rate a book) can enhance the system's understanding of their preferences and improve the recommendations over time.

3. **Multi-Factor Authentication (MFA)**:

- For added security, implementing multi-factor authentication (MFA) would enhance the user login process, ensuring that even if one layer of security is compromised, the user is still protected by another.

4. **Scalability and Performance**:

    - Cloud Deployment: The system could be deployed to a cloud service such as AWS or Google Cloud for better scalability and to ensure high availability of the service as the user base grows.
    - Caching Mechanism: To improve the speed of recommendations, implementing a caching mechanism such as Redis could reduce the number of database calls, thus improving the performance of the system.

5. **Mobile Application**:

    - Developing a mobile version of the recommendation system using frameworks like Flutter or React Native could provide users with access to recommendations on the go. This would increase the accessibility and user engagement of the platform.

6. **Advanced Search Features**:

    - The system could include advanced search features, allowing users to search for books based on categories such as genre, language, and publication year. This would further personalize the book discovery experience.

Chapter 6: References

1. M. Beech, "Key issue – How to share and discuss your research successfully online," *Insights the UKSG Journal*, vol. 27, no. 1, pp. 92-95, 2014.

2. W. Yan and Y. Zhang, "Participation, academic influences and interactions: A comparison of Chinese and U.S. research universities on ResearchGate," *Canadian Journal of Information and Library Science*, vol. 44, no. 2/3, pp. 31-49, 2021.

3. J. Lee, S. Oh, H. Dong, F. Wang, and G. Burnett, "Motivations for self-archiving on an academic social networking site: A study on ResearchGate," *Journal of Information Science*, vol. 40, no. 6, pp. 753-767, 2019.

4. R. K. Bhardwaj, "Academic social networking sites: Comparative analysis of ResearchGate, Academia.edu, Mendeley and Zotero," *International Journal of Information and Library Science*, vol. 6, no. 1, pp. 1-9, 2017.

5. A. S. Salehahmadi, M. R. H. R. Sadeghi, and A. M. M. Zand, "The role of academic social networking sites in scholarly communication: A review," *Journal of Information Science and Technology*, vol. 10, no. 2, pp. 99-105, 2015.

6. R. K. Ghosh and P. K. Biswas, "Use of academic social networks by researchers in India," *Library Hi Tech News*, vol. 33, no. 8, pp. 18-24, 2016.

7. D. M. Shapiro, "The rise of social media in academia," *Journal of Social Media in Society*, vol. 7, no. 2, pp. 36-54, 2018.