E-Book Management System

# 1. Introduction

Project Title: E-Book Management System  
Team Members:

1. Aswin.T(Team leader)
2. Sanjay.K
3. Santhosh S.M
4. Deepan.K
5. Dhesingu.S

# 2. Project Overview

Purpose: The E-Book Management System is designed to help users manage, upload, and read e-books online. It provides a user-friendly interface to organize e-books, categorize them, and access them from anywhere.  
Features:  
- User registration and login  
- E-book upload and management  
- E-book categorization and search  
- User-friendly reading interface  
- Admin dashboard for content management

# 3. Architecture

Frontend: The frontend is developed using React.js, providing a responsive and dynamic user interface.  
Backend: The backend is built with Node.js and Express.js to handle API requests and business logic.  
Database: MongoDB is used as the database to store user data, e-book details, and categories.

# 4. Setup Instructions

Prerequisites:  
- Node.js  
- MongoDB  
- Git  
Installation:  
1. Clone the repository: `git clone <repository-url>`  
2. Navigate to the project directory.  
3. Install dependencies:  
 - For frontend: `cd client && npm install`  
 - For backend: `cd server && npm install`  
4. Set up environment variables in a `.env` file.  
5. Start the servers:  
 - Frontend: `npm start`  
 - Backend: `npm run dev`

# 5. Folder Structure

Client:  
- public/  
- src/  
- components/  
- pages/  
- utils/  
Server:  
- controllers/  
- models/  
- routes/  
- middleware/  
- config/

# 6. Running the Application

To run the application locally:  
Frontend: `npm start` in the client directory  
Backend: `npm run dev` in the server directory

# 7. API Documentation

Endpoints:  
1. User Authentication:  
- POST `/api/auth/register` - Register a new user  
- POST `/api/auth/login` - User login  
2. E-Book Management:  
- GET `/api/books` - Retrieve all e-books  
- POST `/api/books` - Upload a new e-book  
- DELETE `/api/books/:id` - Delete an e-book

# 8. Authentication

Authentication is handled using JWT (JSON Web Tokens). Users receive a token upon login, which is required to access protected routes. Tokens are stored in HTTP-only cookies for security.

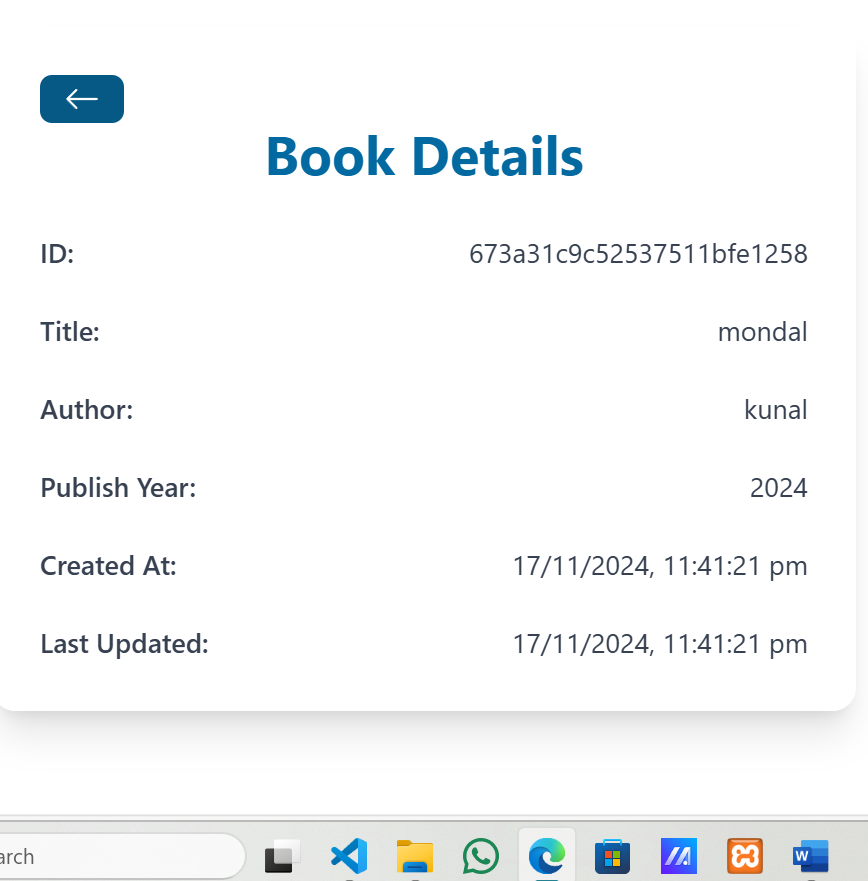
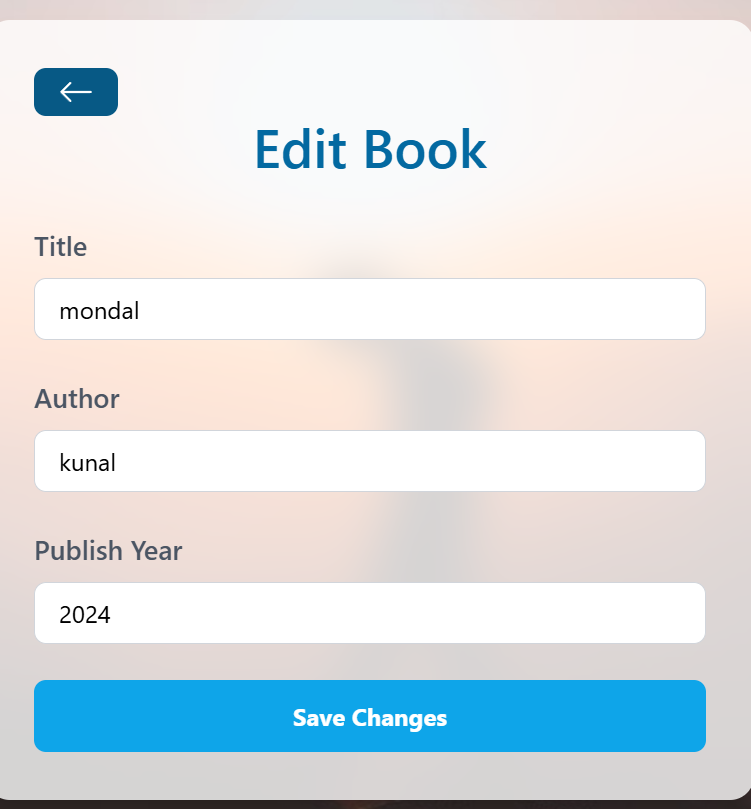
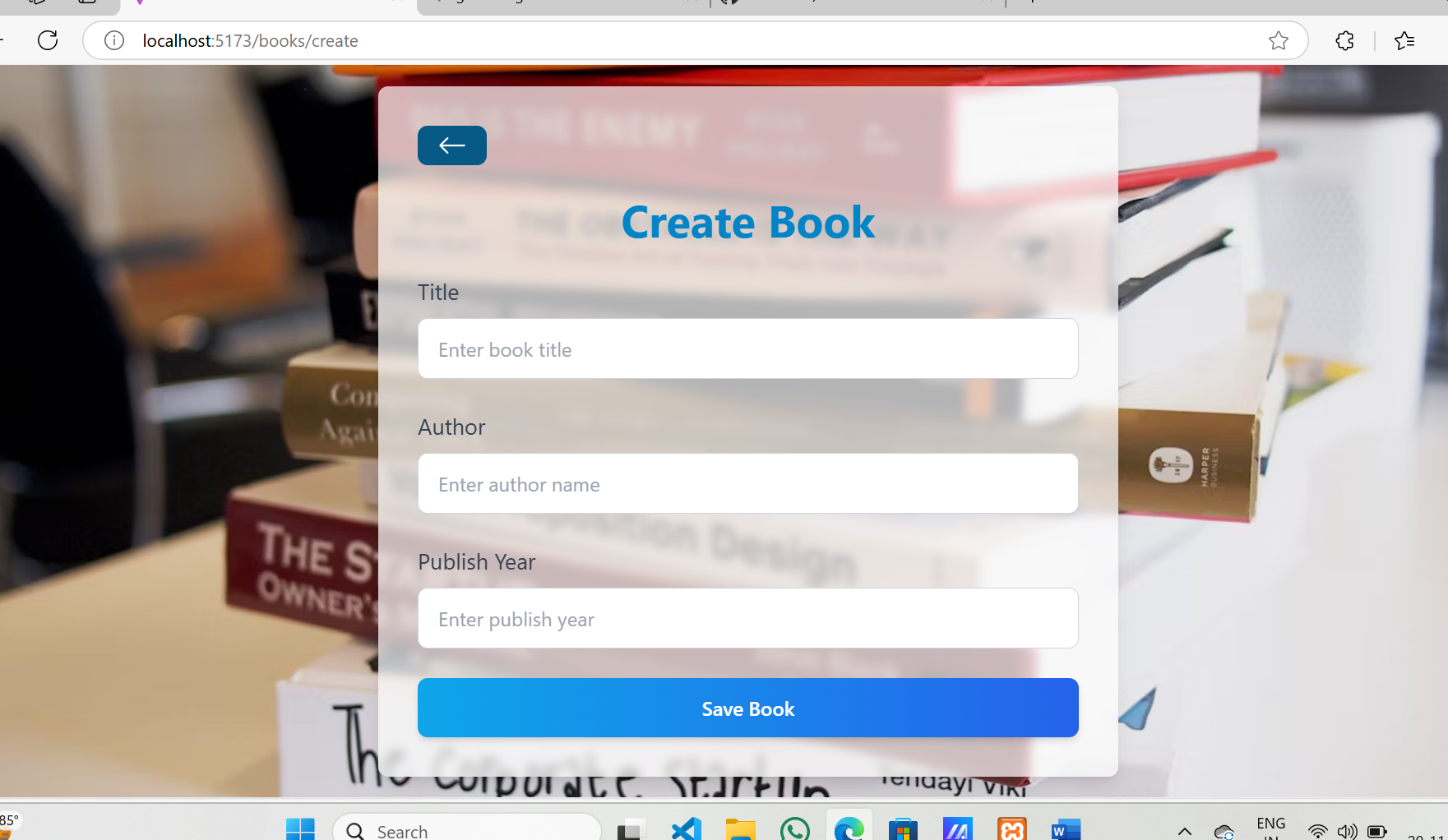
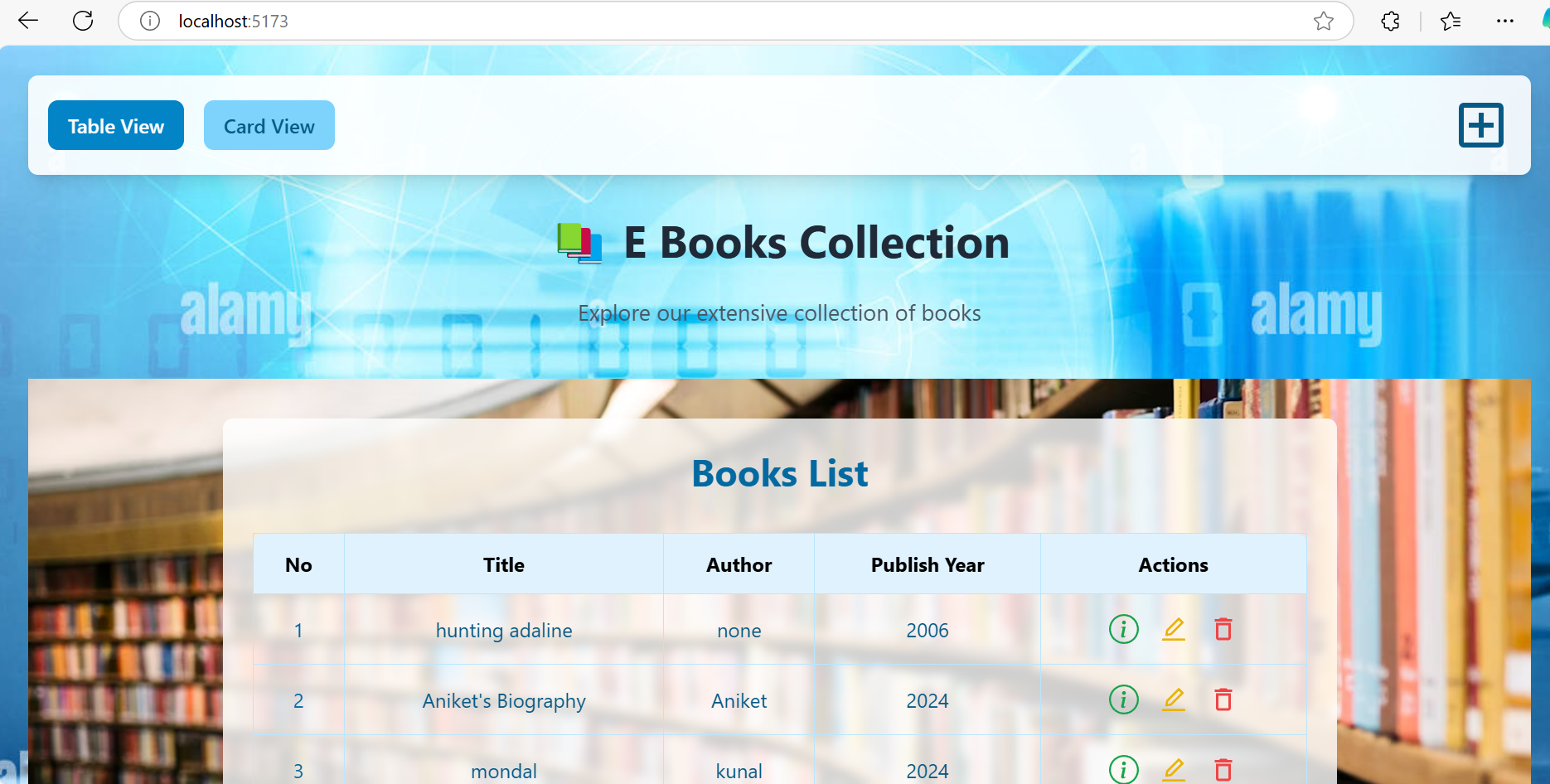
# 9. User Interface

The UI is designed with a focus on simplicity and ease of use. It includes a home page, user dashboard, book listing, and reading interface.

# 10. Testing

Testing Strategy:  
- Unit testing for backend APIs using Jest  
- Component testing for frontend using React Testing Library

# 11. Screenshots or Demo



# 12. Known Issues

1. PDF rendering issues on older browsers.  
2. Large file uploads may take longer than expected due to server limitations.

# 13. Future Enhancements

Potential improvements:  
- Implementing a recommendation system for e-books based on user preferences  
- Adding support for additional file formats like EPUB  
- Enhancing search functionality with filters and tags  
- Mobile app version for offline access