ICT ACADEMY OF KERALA

Summer Internship Report

Full Stack Application Development with ReactJS

Name of Team Member

Suryadas K P,Dhanush K Krishnankutty,Sijil George,Gokul Suresh,Lino Antony

GPTC PERUMBAVOOR

08.05.2023

EXECUTIVE SUMMARY

The Library App is a web application developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. It provides a platform for users to manage and organize their personal library collections. The app allows users to add, search, and categorize books, track reading progress, and create personalized bookshelves. This project demonstrates the implementation of a full-stack application using modern web development technologies.

INTRODUCTION

The Library App is designed to address the need for an efficient and user-friendly platform for book enthusiasts to manage their personal libraries. By utilizing the MERN stack, the application combines the power of a robust backend, database management, and a dynamic frontend interface to provide an intuitive library management solution.

OBJECTIVES

- Develop a user-friendly web application for managing personal libraries.
- Implement CRUD (Create, Read, Update, Delete) operations for books and bookshelves.
- Enable book search functionality using external APIs or database queries.
- Track and display reading progress for individual books.
- Categorize books based on genres, authors, or user-defined tags.
- Create an appealing and responsive user interface for seamless navigation

SCOPE AND DELIVERABLES

To develop the Library App using the MERN stack, the following materials and technologies are required:

1. MongoDB: A NoSQL database for storing book data and user information
2. Express.js: A web application framework for building the backend API.
3. React.js: A JavaScript library for building the frontend user interface.
4. Node.js: A runtime environment for executing server-side JavaScript code.
5. HTML, CSS, and JavaScript for frontend development.
6. Git and GitHub for version control and collaborative development.

METHODOLOGY

The development of the Library App follows an iterative and collaborative approach. The project is divided into the following phases:

- Requirement Gathering: Identify the specific features, functionalities, and user requirements for the library app.
- Database Design: Design the MongoDB schema for storing books, users, bookshelves, and related information.
- Backend Development: Use Express.js and Node.js to build the server-side API for handling CRUD operations and integrating with the database.
- Frontend Development: Utilize React.js to create a responsive user interface with features like book search, bookshelf management, and reading progress tracking.
- Integration and Testing: Integrate the frontend and backend components and perform rigorous testing to ensure proper functionality and error handling.
- Deployment: Deploy the Library App on a hosting platform or server to make it accessible to users.

PROJECT ACTIVITIES

Step 1: Project Setup

1. Set up the development environment by installing Node.js and MongoDB.
2. Create a new directory for your project.
3. Initialize a new Node.js project using npm or yarn.
4. Set up the basic folder structure for your project.

Step 2: Backend Development (Node.js and Express.js)

1. Create a new directory for the backend.
2. Set up the Express.js server.
3. Define the necessary routes and APIs for the library app, such as authentication, book management, user management, etc.
4. Connect to the MongoDB database using a MongoDB driver or an ODM (Object Data Modeling) library like Mongoose.
5. Implement the required CRUD (Create, Read, Update, Delete) operations for books, users, and other relevant entities.
6. Implement authentication and authorization mechanisms using JWT (JSON Web Tokens) or any other preferred method.
7. Write unit tests for your backend APIs using testing frameworks like Jest or Mocha.

Step 3: Frontend Development (React.js)

1. Create a new directory for the frontend.
2. Set up the React.js project using create-react-app or your preferred boilerplate.
3. Design and develop the user interface for the library app using React components, CSS, and any additional UI libraries like Material-UI or Bootstrap.
4. Set up routing using React Router to handle navigation between different pages or components.
5. Implement forms and input validation for adding/editing books, user registration, login, etc.

6. Integrate with the backend APIs to fetch and display data from the server.
7. Handle user interactions and events, such as book searches, book borrowing/returning, user authentication, etc.
8. Write unit tests for your React components using testing libraries like React Testing Library or Enzyme.

Step 4: Integration and Deployment

1. Ensure that the backend and frontend projects are working correctly independently.
2. Integrate the frontend and backend by configuring the necessary CORS (Cross-Origin Resource Sharing) settings and API endpoints.
3. Test the integrated application to ensure seamless communication between the frontend and backend.
4. Optimize and build the frontend project for production using tools like webpack or Create React App.
5. Deploy the backend and frontend to appropriate hosting platforms, such as Heroku, AWS, or Netlify.
6. Set up a continuous integration/continuous deployment (CI/CD) pipeline to automate the deployment process.
7. Monitor and troubleshoot any issues that may arise in the deployed application.

Step 5: Additional Features and Enhancements (Optional)

1. Implement additional features like book recommendations, user reviews, book ratings, etc.
2. Improve the user interface and experience by adding animations, responsive design, or accessibility features.
3. Implement advanced search functionalities using filters, sorting, and pagination.
4. Integrate third-party APIs for additional data sources or services, such as book metadata or online book borrowing systems.

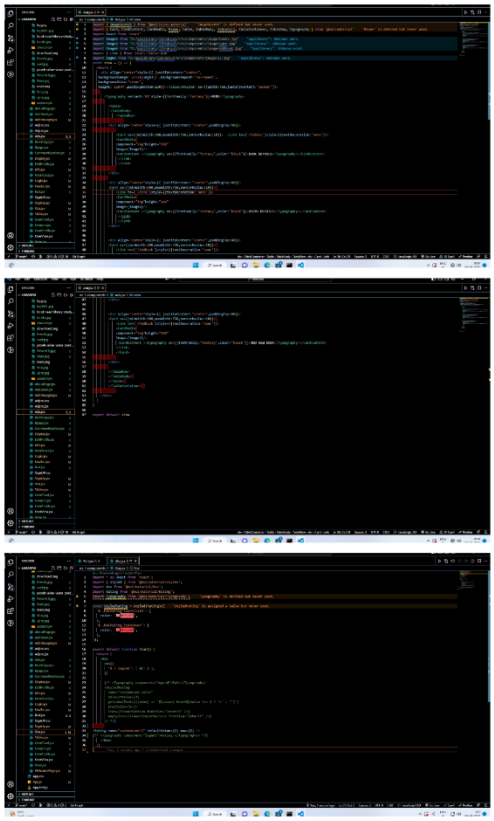
RESULTS & FINDINGS

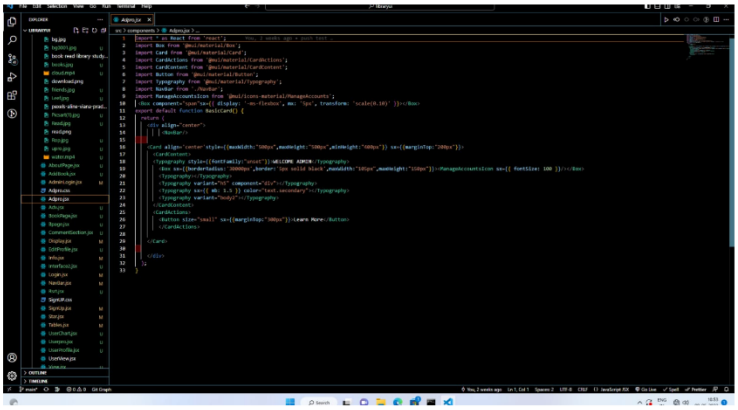
The Library App successfully provides users with a comprehensive solution for managing their personal libraries. Users can easily add, search, and categorize books, track reading progress, and create customized bookshelves. The app offers a user-friendly interface and efficient data management, enhancing the overall library management experience.

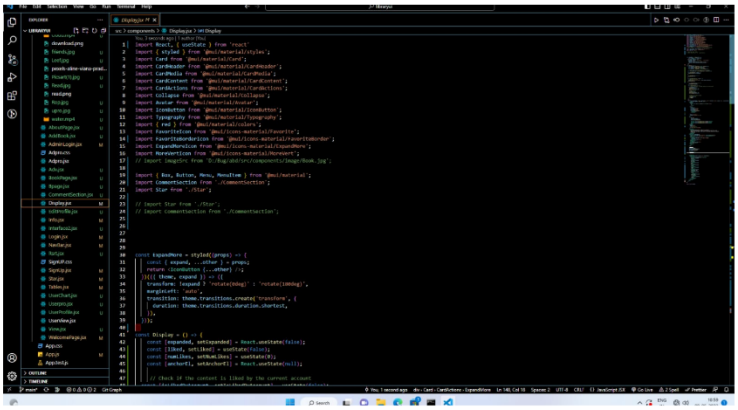
CONCLUSION

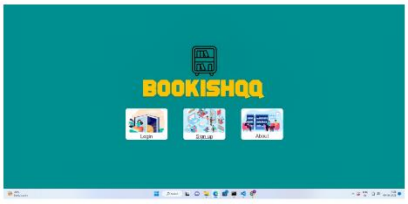
The Library App developed using the MERN stack demonstrates the effectiveness of combining modern web technologies for creating a feature-rich and user-friendly library management solution. The project achieves its objectives of providing users with a seamless experience in organizing and managing their personal book collections. By leveraging the MERN stack, the app delivers a robust backend, efficient database management, and an intuitive frontend, making it a valuable tool for book enthusiasts.

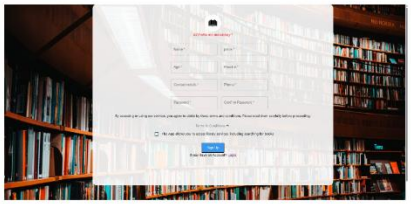
APPENDIX



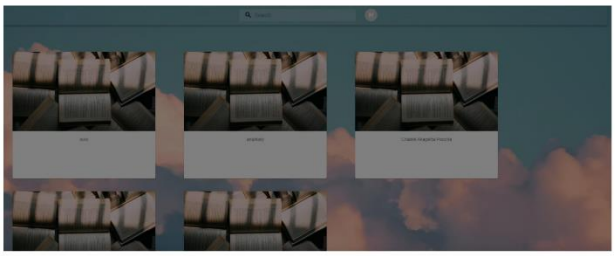




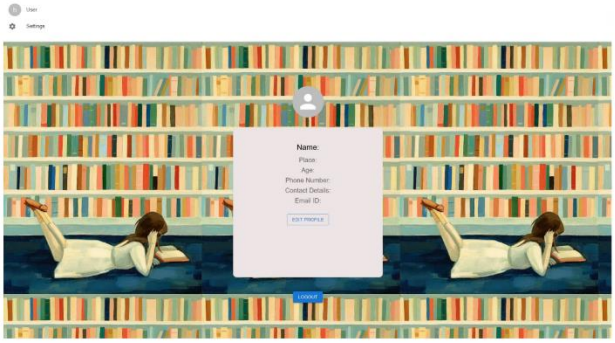

BOOKISHQQ
Login
Signup
About

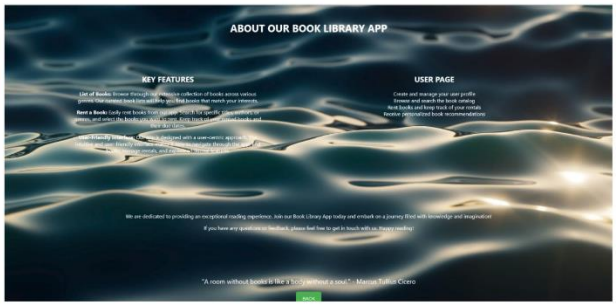



Login




Book Cover
solo
5
Genre: comedy
Author: Arav Dayal
ISBN: 3256457
Publication Year: 2022
Price: 4065
Description: it is a good
BACK


User
Settings
Name:
Place
Age
Phone Number:
Contact Details:
Email ID:
EDIT PROFILE
LOGOUT


ABOUT OUR BOOK LIBRARY APP
KEY FEATURES
USER PAGE
"A room without books is like a body without a soul." - Marcus Tullius Cicero
BACK