REPORT

Summer Of Innovation - DevShelf

Team: **TechX4**

Project Name:

- Library Management System for academic institutions.

Team:

- Samartha Tripathi (Team Lead)
- Vidit Parikh
- Aditya Mishra
- Harsh Chauhan

Introduction:

This report details the step wise general idea we plan to follow for the development of a Library Management System (LMS) using the MERN stack (MongoDB, Express.js, React.js, Node.js). The LMS is designed to facilitate the efficient management of a library's resources (and obviously compete in the SoI :)) and provide a seamless experience for both administrators (the senior PS) and users. The system will be deployed finally on Vercel, and a GitHub repository has been created to track the progress and updates of the project.

Project Overview:

The Library Management System includes two main interfaces:

Admin Interface:

- **User Management**: Add, update, delete, and view user accounts.
- **Book Management**: Add, update, delete, and view book details.
- **Borrow/Return Management**: Track which books are borrowed and manage the return process.
- **Dashboard**: View summary statistics and system alerts

User Interface:

- **Book Search and Browse**: Search for books by ISBN, title, author, genre, or keyword.
- **Book Details**: View detailed information about each book.
- **Borrowing Books**: Borrow books and view current borrowings.
- **Account Management**: Update personal details and view borrowing history.

Technologies Used:

Frontend:

- **React.js:** For building the user interface.
- **Tailwind:** CSS Library for common elements.
- **Bootstrap:** CSS Library for common elements.
- **Redux:** For state management.
- **React Router:** For handling routing in the application.
- **Material-UI:** For styling and responsive design components.

Backend:

- **Node.js:** For the server-side runtime environment.
- **Express.js:** For building the RESTful API.
- **MongoDB:** For the database to store user and book information.
- **Mongoose:** For object data modeling (ODM) to interact with MongoDB.

Deployment:

- **Vercel:** For deploying the frontend and backend of the application.

Additional Tools:

- **GitHub:** For version control and collaboration.
- **Postman:** For API testing.
- **JWT (JSON Web Tokens):** For authentication and authorization.
- **Bcrypt.js:** For password hashing

Entity-Relationship Diagram:

https://whimsical.com/lib-mgmt-7THuKu4RdKSQHnDzHsdYTx

System Architecture:

Frontend:

The frontend is built using React.js, which communicates with the backend API to fetch and update data. Redux is used for state management to ensure a predictable state container, while React Router handles the routing within the application. Material-UI is used to create a responsive and user-friendly interface.

Backend:

The backend is built with Node.js and Express.js to create a RESTful API. MongoDB serves as the database to store information about users, books, and transactions. Mongoose is used to interact with the MongoDB database using schemas and models.

Authentication:

Authentication is handled using JWT. Upon login, a token is generated and sent to the client, which is then stored in local storage. This token is used for all subsequent requests to ensure the user is authenticated.

Deployment:

The application is deployed on Vercel, which provides a seamless platform for hosting both the frontend and backend. Continuous deployment is set up through GitHub, allowing for automatic updates whenever changes are pushed to the repository.

Project Management:

Repository:

The project is hosted on GitHub and is updated frequently. The repository includes all the source code, documentation, and project management files.
Repository URL: https://github.com/GruffElixir/SOI_WebDev_2024

Updates and Collaborations:

Regular updates are pushed to the GitHub repository to ensure all team members are working with the latest version of the code. Issues and pull requests are used to manage tasks and collaborate effectively. Also, we have added our mentor to the repository as a collaborator.

Thank You!