**DEVSHELF – MID-TERM MEET**

**SITE SCULPTORS**

**OVERVIEW OF OUR PROGRESS :**

1. FRONTEND DEVELOPMENT PROGRESS :

1. HTML, CSS, JavaScript Basics : These foundational languages were used to create static web pages and basic interactivity.

2. React.js: React.js was chosen for its component-based architecture, which facilitates scalable and efficient frontend development.

Development Progress:

1. Setting Up React App:

- Utilized create-react-app to set up the project environment.

- Configured necessary dependencies and folder structure for organization.

2. Implementation of Functional Components:

- Developed login pages using functional components to handle user authentication.

- Created a home header component to provide navigation and user interface consistency.

-And added this repository of frontend to the team's github frontend branch

3. Key Concepts Implemented:

- Functional Components: Leveraged React's functional components for modular UI elements.

- State Management: Utilized useState hook for managing component-level state, ensuring dynamic user interactions.

- Routing: Implemented routing using React Router for navigating between different pages of the application.

- Styling: Applied CSS for styling components and maintaining a cohesive design across the application

1. BACKEND DEVELOPMENT PROGRESS :

1. MongoDB:

- MongoDB Concepts: Learned CRUD operations (Create, Read, Update, Delete).

- Tools Setup:

MONGO DB instances setup:

- MongoDB Compass: For visual data exploration.

- MongoDB Shell: For advanced command-line interactions.

- MongoDB Atlas: Set up for cloud-based database deployment.

- Database and Collection:

- Created a LIB database and a BOOKS collection.

- Inserted book details using MongoDB Compass.

2. Development Environment:

- Linked MongoDB to \*VS Code\* for streamlined development.

- Set up \*GitHub\* for version control.

3. Next Steps:

- Exploring further operations and integrations with MongoDB.

Conclusion:

The project's frontend development using React.js has laid a robust foundation for building a responsive and interactive Library Management System. The initial implementation of login pages and a home header using functional components demonstrates progress towards creating a user-friendly interface. Future iterations will further enhance functionality and user experience of library management system.