Techplement Internship Report
Clinic Management System

Submitted by Team Web 09:

VISHWESH VISHNU AMBI
Dhinesh Kumar
Simran Kaur
Akhila Vardolu
Rohit Shaw
Pratap Sikdar

Internship duration:

1st May 2024 to 30th May 2024

Project Overview:

The primary objective of our project is to develop a comprehensive web application designed to streamline the process of booking appointments with doctors and managing patient interactions within a hospital or healthcare facility. This application will provide an intuitive user interface for patients, doctors, and administrators, ensuring a seamless experience from appointment scheduling to consultation. Our Online Doctor Appointment and Patient Management Web Application aspires to revolutionize the healthcare experience by leveraging technology to create a seamless, user-friendly platform that benefits patients, doctors, and administrators alike. By following a structured development process and maintaining a focus on user needs, we aim to deliver an application that significantly improves the efficiency and quality of healthcare services.

Project Team:

Vishwesh Vishnu Ambi: - Front End Developer

- Responsible for designing and implementing the user interface using React.
- Ensures a smooth and intuitive user experience for patients, doctors, and administrators.

Dhinesh Kumar: - Front End Developer

- Collaborates on the development of the user interface using React.
- Focuses on creating responsive and accessible design elements.

Simran Kaur:- Front End Developer

- Works on the visual design and front-end functionality using React.
- Assists in integrating front-end components with backend services.

Pratap Sikdar: - Frontend + Backend Develop

- Engages in developing and maintaining both front-end and backend systems.
- Develops both front-end features using React and backend services using Node.js and Express.js.
- Focuses on the overall architecture and implementation of full-stack features.

Akhila Vardolu: - Frontend + Backend Developer

- Contributes to the development of front-end interfaces and backend functionality
- Ensures seamless integration between front-end and backend components.
- Focuses on the overall architecture and implementation of full-stack features

Rohit Shaw: - Frontend + Backend Developer

- Contributes to the development of front-end interfaces and backend functionality.
- Focuses on the overall architecture and implementation of full-stack features.
- Ensures seamless integration between front-end and backend components.

Project Technologies:

Frontend:

- ReactJs
- CSS
- Bootstrap

Backend:

- NodeJs
- ExpressJs

Database:

- MongoDB

Project Setup:

- npx create-react-app clinicmangmentsystem
- npm install bcryptjs colors concurrently dotenv express jsonwebtoken mongoose morgan nodemon
- npm install bootstrap
- npm install reduxjs/toolkit antd axios moment react react-dom react-redux react-router-dom react-spinners
- To run the project: npm run dev

Description of the packages used:

- bcryptjs: Library to hash passwords.
- colors: Library for adding colors to console logs.
- concurrently: Utility to run multiple commands concurrently.
- dotenv: Module that loads environment variables from a .env file into process.env.
- express: Web application framework for Node.js.
- jsonwebtoken: Library to work with JSON Web Tokens.
- mongoose: MongoDB object modeling tool for Node.js.
- morgan: HTTP request logger middleware for Node.js.
- nodemon: Utility that monitors for any changes in your source and automatically restarts your server.
- @reduxjs/toolkit: Official, opinionated, batteries-included toolset for efficient Redux development.
- antd: A popular React UI framework.
- axios: Promise-based HTTP client for the browser and Node.js.
- moment: Parse, validate, manipulate, and display dates and times in JavaScript.
- react-dom: This package serves as the entry point to the DOM and server renderers for React.
- react-redux: Official React bindings for Redux.
- react-router-dom: DOM bindings for React Router.
- react-spinners: A collection of loading spinner components for React.

Project Resources/ Reference:

- https://www.geeksforgeeks.org/mern-stack/
- https://stackoverflow.com/questions/tagged/mern
- https://chat.openai.com/ (For Error Handling)
- https://www.mongodb.com/docs/atlas/
- https://iconscout.com
- https://docpulse.com/
- https://www.adroitinfosystems.com/products/ehospital-systems/
- https://getbootstrap.com/

Project Risks

Connectivity with Database

- Risk: Difficulty in establishing and maintaining connectivity with the database could lead to delays or errors in data retrieval and storage.

Mitigation Strategy:

- Ensure thorough testing of database connectivity during development.
- Implement retry mechanisms and error handling to handle intermittent connection issues.
- Utilize database monitoring tools to promptly identify and resolve connectivity issues.

API Naming Format

- Risk: Inconsistent or unclear API naming conventions may lead to confusion among developers and hinder future maintenance and scalability.

Mitigation Strategy:

- Establish clear and consistent naming conventions for APIs based on industry standards or team agreements.
- Document API endpoints and naming conventions comprehensively for reference by all team members.
- Conduct regular code reviews to ensure adherence to naming conventions.

Limited Device Compatibility

- Risk: The application is only optimized for big screens and lacks responsiveness for mobile or tablet devices, potentially leading to a poor user experience for a significant portion of users.

Mitigation Strategy:

- Conduct a thorough assessment of user demographics and device usage statistics to prioritize device compatibility.
- Implement responsive design principles and frameworks (e.g., Bootstrap, CSS media queries) to ensure the application adapts to various screen sizes.

Real-time Date and Time Management

- Risk: Managing real-time date and time functionality, especially across different time zones, may pose challenges and require additional effort to implement and maintain.

Mitigation Strategy:

- Utilize reliable third-party libraries or packages (such as Moment.js) to simplify date and time management tasks.
- Thoroughly test date and time functionality across various scenarios, including different time zones and daylight saving time transitions.
- Implement error handling and fallback mechanisms to gracefully handle unexpected issues related to date and time management.