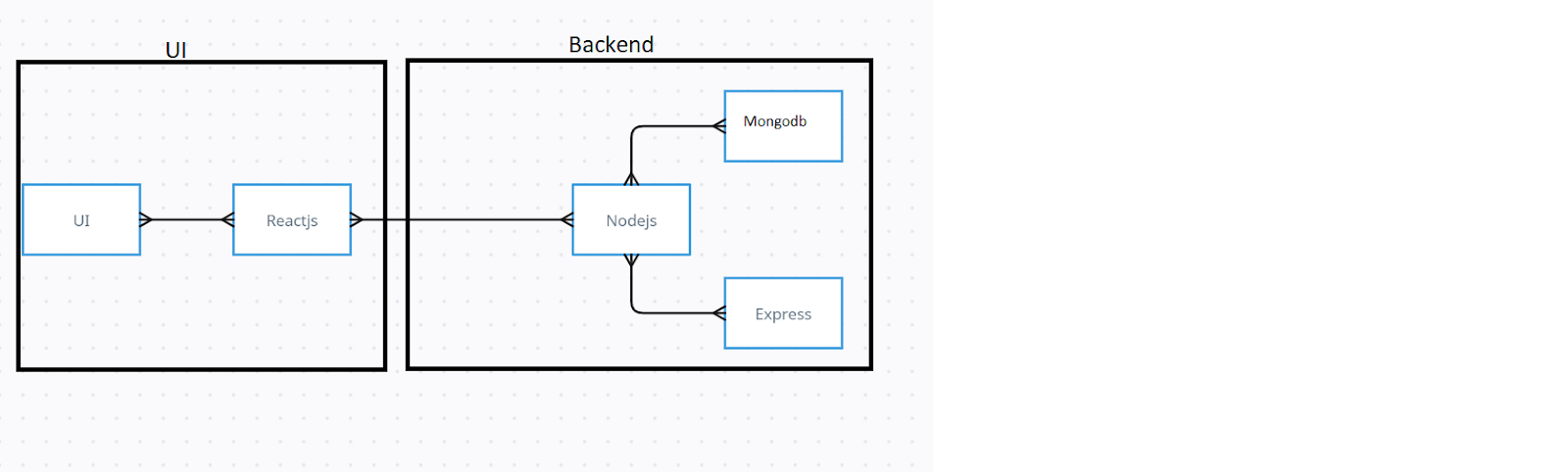
**BOOK A DOCTOR USING MERN**

**INTRODUCTION**

Booking a doctor appointment has never been easier. With our convenient online platform, you can quickly and effortlessly schedule your appointments from the comfort of your own home. No more waiting on hold or playing phone tag with busy receptionists. Our user-friendly interface allows you to browse through a wide range of doctors and healthcare providers, making it simple to find the perfect match for your needs.

With our advanced booking system, you can say goodbye to the hassle of traditional appointment booking. Our platform offers real-time availability, allowing you to choose from a range of open slots that fit your schedule. Whether you prefer early morning, evening, or weekend appointments, we have options to accommodate your needs.

**TECHNICAL ARCHITECTURE**



The technical architecture of our Book a Doctor app follows a client-server model, where the frontend serves as the client and the backend acts as the server. The frontend encompasses not only the user interface and presentation but also incorporates the axios library to connect with backend easily by using RESTful Apis.

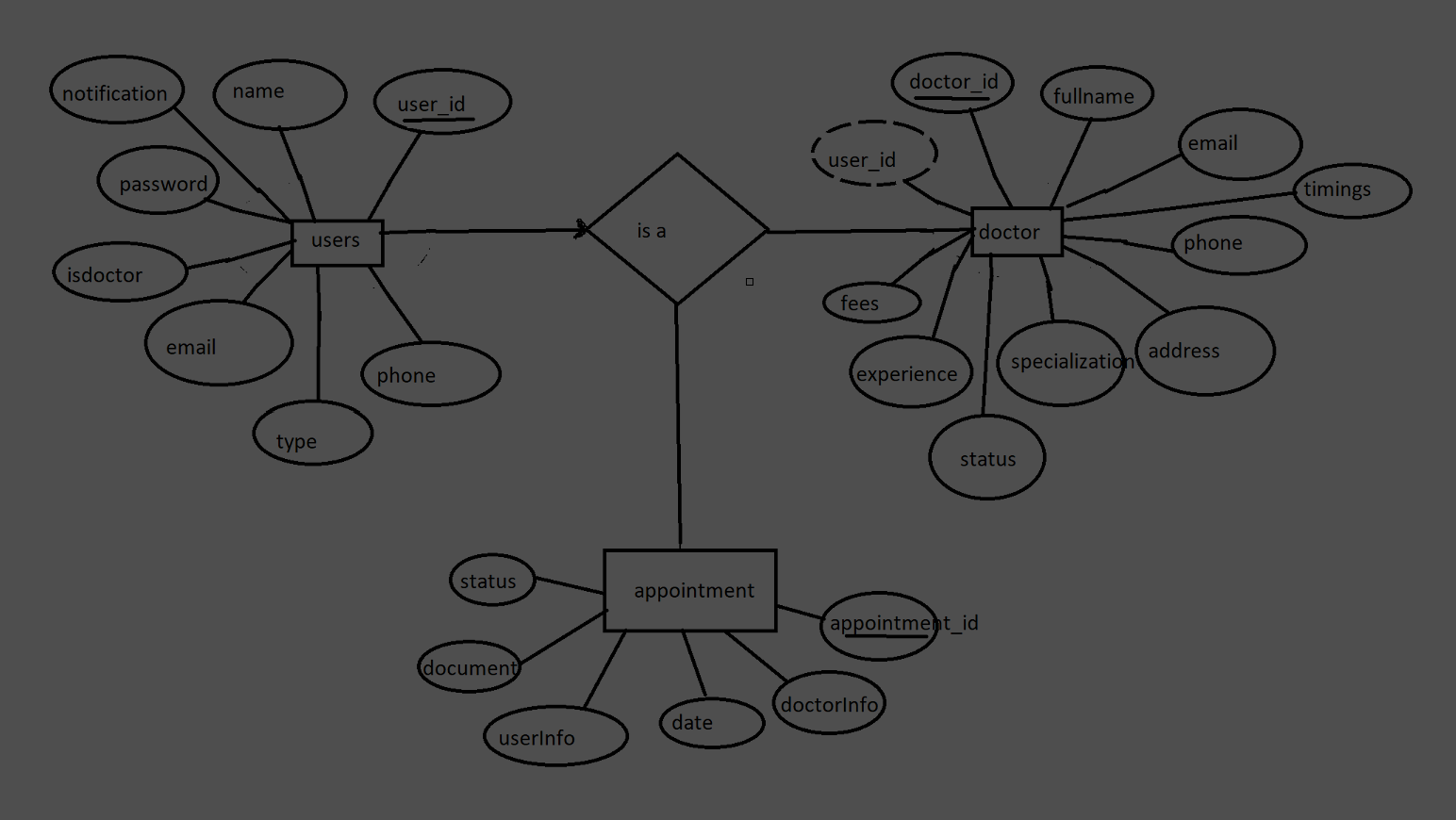
The frontend utilizes the bootstrap and material UI library to establish real-time and better UI experience for any user whether it is admin, doctor and ordinary user working on it.

On the backend side, we employ Express.js frameworks to handle the server-side logic and communication.

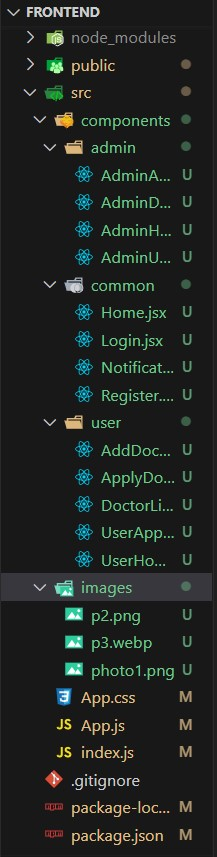
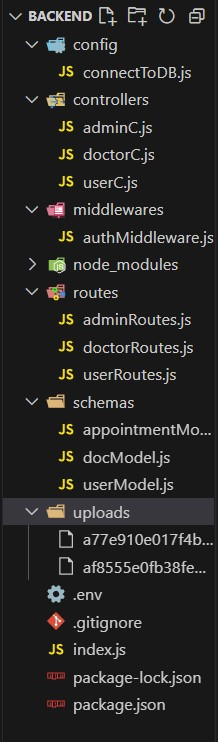
For data storage and retrieval, our backend relies on MongoDB. MongoDB allows for efficient and scalable storage of user data, including user profiles, for booking room, and adding room, etc. It ensures reliable and quick access to the necessary information.

Together, the frontend and backend components, along with moment, Express.js, and MongoDB, form a comprehensive technical architecture for our Book a Doctor app. This architecture enables real-time communication, efficient data exchange, and seamless integration, ensuring a smooth and immersive booking an appointment and many more experience for all users.

**ER DIAGRAM**

****

**PROJECT STRUCTURE**

The first image is of frontend part which is showing all the files and folders that have been used in UI development

The second image is of Backend part which is showing all the files and folders that have been used in backend development

**PRE-REQUISTIC:**

Here are the key prerequisites for developing a full-stack application using Node.js, Express.js, MongoDB, React.js:

✔**Node.js and npm:**

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the server-side. It provides a scalable and efficient platform for building network applications.

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

Download: https://nodejs.org/en/download/

Installation instructions: <https://nodejs.org/en/download/package-manager/>

**npm init**

✔**Express.js:**

Express.js is a fast and minimalist web application framework for Node.js. It simplifies the process of creating robust APIs and web applications, offering features like routing, middleware support, and modular architecture.

Install Express.js, a web application framework for Node.js, which handles server-side routing, middleware, and API development.

Installation: Open your command prompt or terminal and run the following command:

**npm install express**

✔**MongoDB:**

MongoDB is a flexible and scalable NoSQL database that stores data in a JSON-like format. It provides high performance, horizontal scalability, and seamless integration with Node.js, making it ideal for handling large amounts of structured and unstructured data.

Set up a MongoDB database to store your application's data.

Download: https://www.mongodb.com/try/download/community

Installation instructions: https://docs.mongodb.com/manual/installation/

✔**Moment.js:**

Momentjs is a JavaScript package that makes it simple to parse, validate, manipulate, and display date/time in JavaScript. Moment. js allows you to display dates in a human-readable format based on your location. Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: [https://momentjs.com/](https://momentjs.com/%20)

✔**React.js:**

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making it easier to build dynamic and responsive web applications.

Install React.js, a JavaScript library for building user interfaces.

Follow the installation guide: <https://reactjs.org/docs/create-a-new-react-app.html>

✔**Antd:**

Ant Design is a React. js UI library that contains easy-to-use components that are useful for building interactive user interfaces. It is very easy to use as well as integrate. It is one of the smart options to design web applications using react.

Follow the installation guide: <https://ant.design/docs/react/introduce>

✔**HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential.

✔**Database Connectivity**: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations. To Connect the Database with Node JS go through the below provided link:

https://www.section.io/engineering-education/nodejs- mongoosejs-mongodb/

✔**Front-end Framework**: Utilize Reactjs to build the user-facing part of the application, including entering booking room, status of the booking, and user interfaces for the admin dashboard.

For making better UI we have also used some libraries like material UI and boostrap.

✔**Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at: https://git-scm.com/downloads

✔**Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

• Visual Studio Code: Download from <https://code.visualstudio.com/download>

To run the existing Video Conference App project downloaded from GitHub:

Follow below steps:

Clone the Repository:

* Open your terminal or command prompt.
* Navigate to the directory where you want to store the e-commerce app.
* Execute the following command to clone the repository:

**git clone**: [**LINK**](https://github.com/awdhesh-student/bookdoctor.git)

Install Dependencies:

• Navigate into the cloned repository directory:

cd book-a-doctor

• Install the required dependencies by running the following commands:

cd frontend

npm install

cd ../backend

npm install

Start the Development Server:

• To start the development server, execute the following command:

npm start

• The book a doctor app will be accessible at <http://localhost:3000>

You have successfully installed and set up the online complaint registration and management app on your local machine. You can now proceed with further customization, development, and testing as needed.

**Roles and Responsibilities:**

The project has 2 type of user – Customer and Doctor and other will be Admin which takes care to all the user. The roles and responsibilities of these two types of users can be inferred from the API endpoints defined in the code. Here is a summary:

Customer/Ordinary:

1. Create an account and log in to the system using their email and password.
2. They will be shown automatically all the doctors in their dashboard.
3. After clicking on the Book Now, a form will generate in which date of appointment and documents need to send.
4. They can sees the status of their appointment and can get a notification if the appointment is schedule or not.
5. The user can also cancel it`s booking in booking history page and can change the status of booking.

Admin:

1. Manage and monitor the overall operation of the appointment and the type of users and doctors to the application.
2. He monitors the applicant of all doctors and approve them and then doctors are registered in the app.
3. Implement and enforce platform policies, terms of service, and privacy regulations.

Doctor:

1. Gets the approval from the admin for his doctor account.
2. Manages all the appointments that are getting from the users

**Project Flow:**

Before starting to work on this project, let’s see the demo.

**Project demo:** [**LINK**](https://drive.google.com/file/d/1wsPyhAY9dU5fAfz4hN_-gbBBmAiPY9OA/view?usp=drive_link)

Use the code in: [**LINK**](https://github.com/awdhesh-student/bookdoctor.git)

or follow the videos below for better understanding.

**Milestone 1: Project setup and configuration.**

* **Folder setup:**

1. Create frontend and
2. Backend folders

* **Installation of required tools:**

1. Open the frontend folder to install necessary tools

For frontend, we use:

* React
* Bootstrap
* Material UI
* Axios
* Moment
* Antd
* mdb-react-ui-kit
* react-bootstrap

2. Open the backend folder to install necessary tools

For backend, we use:

* cors
* bcryptjs
* express
* dotenv
* mongoose
* Moment
* Multer
* Nodemon
* jsonwebtoken

**Milestone 2: Backend Development**

* **Setup express server**

1. Create index.js file in the server (backend folder).
2. define port number, mongodb connection string and JWT key in env file =  to access it.
3. Configure the server by adding cors, body-parser.

* **Configure MongoDB**

1. Import mongoose.
2. Add database connection from config.js file present in config folder
3. Create a model folder to store all the DB schemas like user, doctor and appointment schemas.

* **Add authentication:** for this,

You need to make middleware folder and in that make authMiddleware.js file for the authentication of the projects and can use in.

v