Overview of the Full-Stack Project

Project Objective:

The project creates a health care management application where a patient makes appointments with doctors, can view prescriptions made from them, makes payments, and submits reviews. Doctors run their clinics, manage appointments, and make prescriptions for different treatments.

1. Backend

In building the backend with Node.js, Prisma ORM is employed to interact with the database in the development. It is connected to a MySQL database and essentially addresses models such as Patient and Doctor, Appointments, Prescriptions to deal with significant healthcare-related data.

Key Components

Authentication: JWT-based authentication through the Auth model

Database Models: Patients, Doctors, Appointments, Prescriptions, Reviews, etc, which include the relational structure of the healthcare data

API Endpoints:

Patient Management-for example, POST /patients, GET /patients/:id

Doctor Management-for example, POST /doctors, GET /doctors/:id

Appointment Scheduling-for example, POST /appointments

Prescription Management-for example, POST /prescriptions

Database Design: We are using the ORM, Prisma with models which are linked up with entities like Patient, Doctor, Appointments, and Prescriptions. Every model is related using foreign key relations for integrity of user data as well as appointments.

2. Frontend:

The frontend is built with React.js for a dynamic and responsive user interface. Some of the most important features include:

UI Design:Tailwind CSS for smooth and responsive layouts. The design will be ease-of-use-oriented with least user effort while scheduling appointments and managing health information.

User Experience:Dashboard pages for Patients and doctors. Patients can log in to their account to have an overview of appointment histories, manage profile, and write reviews. Doctors can log in to the doctor's dashboard to look over the clinic schedule and prescriptions.

Technologies:React.js, Tailwind CSS, Axios(API calls)

3.Hosting/Deployment:

Vercel hosts the application for frontend deployments while hosting the backend on AWS EC2.

Deployment

Backend: dockerized node.js server deployed to AWS EC2.

Frontend: Continuous deployment using Vercel. Automatic build and deployment on push to main.

Environment Variables: Through the use of .env files, manages the database and API keys securely.

4. Project Link

https://clouldplaydoctor.online/