The DoctorOnCall System facilitates patients in scheduling appointments with preferred doctors or Emergency Appointments set up via an online platform. Developed using the React, Nodejs, and Prisma stack, it offers a professional website with dynamic functionalities. Key features include dynamic home, Email Notification, filtering doctors, setting up appointments and patients/doctors dashboards, secure user login with validation mechanisms, appointment scheduling with access to doctor details, patient appointment management, doctor selection, and overall industries level code splitting followed.

**What Features Will You Find Here:**

* Dynamic Homepage: Conveniently gathers essential information on a single page.
* Easy Appointment Setup: Seamlessly schedule appointments with doctors, featuring selectable available dates and time ranges.
* Dynamic Appointment Management: Flexible time and date range selection for appointments.
* Doctor Selection: Patients can choose their preferred doctor for appointments.
* Industry-standard Code Quality: Utilization of top-notch code conventions, code splitting, and TypeScript in backend development.
* Email Notifications: Automated email notifications for appointment setup and invoicing.
* Doctor Dashboard: Dynamic dashboard for doctors to access patient information and provide online treatment with customizable prescriptions.
* Patient Dashboard: Patients can track their treatment progress, view medications, prescriptions, and appointment details from a dynamic dashboard.
* Appointment Tracking: Track appointment status using a unique tracking ID directly from the homepage.
* Dynamic Prescription and Invoice Generation: Customized prescriptions and invoices tailored to each appointment.
* User Authentication: Dynamic authentication system allowing user sign-in, sign-up, password recovery, and email verification (for doctors).
* Dynamic Doctor Filtering: Advanced filtering options for doctors and appointment scheduling, along with blogging and service features.
* Ongoing Development: Continued enhancements with an evolving admin system.
* Best Practices: Adherence to industry best practices, ensuring reusable and maintainable code.

Getting Started

To begin using the DoctorOnCall System, follow these simple steps:

* Install Node.js (globally)

npm install npm@latest -g

* Prisma CLI: If you're planning to work on the backend part, you'll need to have Prisma CLI installed globally. You can install it using npm:

npm install -g prisma

* TypeScript (optional): TypeScript is used in the backend, ensure that you have TypeScript installed globally. You can install it using npm:

npm install -g typescript

The front-end and Backend code are in the same directory, with the Backend API located at the ./api directory

**# Setup Documentation**

### Install Frontend

cd Doctor-Appointment

- npm install

- npm start

### Install Backend

cd api

- npm install

### Setup Database

1. Rename .env.example to .env (remove .example).

2. Create a PostgreSQL Database (Railway if you don't have one installed locally):

- Create an account at https://railway.app/.

- Navigate to the New Section > Database > Add PostgreSQL.

- Select your created database and go to the Variables tab.

- Copy DATABASE\_PUBLIC\_URL.

- Paste the database URL into the .env file.

### Install Prisma

- npm install -g prisma

- npx prisma generate

- npx prisma migrate dev

### Setup Google App Password (For Email Notification)

1. Go to Google Account settings at https://myaccount.google.com/security?hl=en.

2. Navigate to Security > 2-Step Verification.

3. Scroll to the bottom of the page and find App passwords.

4. Select your project name and copy the generated password.

5. Paste the app password into .env as EMAIL\_PASS.

### Setup Cloudinary to Upload Image

1. Create a Cloudinary Account at https://cloudinary.com/.

2. Login to your Cloudinary Account and copy all the credentials (e.g., Cloud name, API key, API secret).

3. Paste those credentials into the .env file.

### Start Backend

npm run dev