NAME - SIMRAN CHAWLA
EMAIL ID - 20051106@kiit.ac.in
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

REACT SCHEDULING AND BOOKING APPLICATION

Table of Contents

Introduction

The Scheduling and Booking Application is designed to facilitate the setting of availability by User 1 and the booking of available time slots by other users (User 2, User 3, etc.). The application uses React for the frontend, Node.js and Express for the backend, and MongoDB for data storage.

System Design

Architecture Overview

The application follows a client-server architecture:

- **Frontend**: React application that uses react-big-calendar for the calendar interface and Axios for making HTTP requests.
- **Backend**: Node.js and Express application that handles API requests and interacts with the MongoDB database.
- **Database**: MongoDB for storing user availability and booking data.

Component Breakdown

Frontend Components:

- App.js: Main entry point of the React application.
- Calendar.js: Component for displaying and managing the calendar.
- AvailabilityForm.js: Form for User 1 to set availability.
- BookingForm.js: Form for other users to book time slots.
- api.js: Module for API calls.

Backend Components:

- server.js: Main server file.
- routes/availabilities.js: Routes for handling availability-related API requests.
- routes/bookings.js: Routes for handling booking-related API requests.
- models/Availability.js: Mongoose model for availability.
- models/Booking.js: Mongoose model for bookings.

Implementation Details

Frontend:

1. Project Structure:

```
frontend/
 ├── public/
 ├── src/
|    ├── components/
|    |    ├── App.js
|    |    ├── Calendar.js
|    |    ├── AvailabilityForm.js
|    |    ├── BookingForm.js
|    ├── api/
|    |   └── api.js
|    ├── index.js
 ├── .env
 ├── package.json
```

2. Important Files:
   - App.js: Initializes the main components and routes.
   - Calendar.js: Uses react-big-calendar to display the calendar and manage events.
   - AvailabilityForm.js: Form for User 1 to submit availability.
   - BookingForm.js: Form for other users to book time slots.
   - api.js: Handles all API calls to the backend.

Backend:

1. Project Structure:

```
backend/
 ├── routes/
|    ├── availabilities.js
|    ├── bookings.js
 ├── models/
|    ├── Availability.js
|    ├── Booking.js
 ├── .env
 ├── server.js
 ├── package.json
```

2. Important Files:
   - server.js: Sets up Express server, connects to MongoDB, and initializes routes.
   - availabilities.js: Defines API endpoints for managing availability.

- bookings.js: Defines API endpoints for managing bookings.
- Availability.js: Mongoose schema and model for availability.
- Booking.js: Mongoose schema and model for bookings.

Database Schema

Availability Schema:

```
const mongoose = require('mongoose');
const availabilitySchema = new mongoose.Schema({
  userId: String,
  start: Date,
  end: Date,
});
module.exports = mongoose.model('Availability', availabilitySchema);
```

Booking Schema:

```
const mongoose = require('mongoose');
const bookingSchema = new mongoose.Schema({
  userId: String,
  availabilityId: String,
  start: Date,
  end: Date,
});
module.exports = mongoose.model('Booking', bookingSchema);
```

Setup Instructions

Prerequisites

- Node.js (v16 or later recommended)
- MongoDB (local or cloud instance)
- npm or yarn

Backend Setup

- Clone the repository
- Install Dependencies
- Setup environment variables
- Run the backend server

Frontend Setup

- Navigate to frontend directory
- Install Dependencies
- Setup environment variables
- Run the frontend server

Usage Instructions

User 1 (Scheduler):

- Navigate to the availability setting form.
- Enter available time slots.
- Submit to save availability.

Other Users (Bookers):

- View available time slots on the calendar.
- Select a time slot and book it.