Scheduling Platform

(Cal.com Clone)

SDE Intern Fullstack Assignment

Description

Build a functional scheduling/booking web application that closely replicates Cal.com's design and user experience. The application should allow users to create event types, set their availability, and let others book time slots through a public booking page. Your implementation should visually resemble Cal.com's UI patterns, layout structure, and interaction design.

AI Tools Usage

You are **allowed and encouraged** to use AI tools such as ChatGPT, Claude, GitHub Copilot, Cursor, or any other AI assistants for development. However, you **must understand every line of code** you submit and be prepared to explain your implementation decisions during the evaluation interview.

Technical Stack

- **Frontend:** React.js or Next.js (Single Page Application)
- **Backend:** Node.js with Express.js OR Python with FastAPI/Django
- **Database:** PostgreSQL or MySQL (design your own schema)

Core Features (Must Have)

1. Event Types Management

- Create event types with title, description, duration (in minutes), and URL slug
- Edit and delete existing event types
- List all event types on a dashboard
- Each event type should have a unique public booking link

2. Availability Settings

- Set available days of the week (e.g., Monday to Friday)
- Set available time slots for each day (e.g., 9:00 AM - 5:00 PM)
- Set timezone for the availability schedule

3. Public Booking Page

- Calendar view to select a date
- Display available time slots based on availability settings
- Booking form to collect booker's name and email
- Prevent double booking of the same time slot
- Booking confirmation page with event details

4. Bookings Dashboard

- View upcoming bookings
- View past bookings
- Cancel a booking

Good to Have (Bonus)

- Responsive design (mobile, tablet, desktop)
- Multiple availability schedules
- Date overrides (block specific dates or set different hours)
- Rescheduling flow for existing bookings
- Email notifications on booking confirmation/cancellation
- Buffer time between meetings
- Custom booking questions

Important Notes

- **UI Design:** Your application should closely resemble Cal.com's design. Study Cal.com's UI carefully before starting - visit cal.com to understand the user flows and interface patterns.
- **No Login Required:** Assume a default user is logged in for the admin side (event types, availability, bookings dashboard). The public booking page is accessible without login.
- **Sample Data:** Seed your database with sample event types and a few bookings.
- **Database Design:** Design your own database schema. This will be evaluated.
- **README File:** Include setup instructions, tech stack used, and any assumptions made.
- **Original Work:** Plagiarism from existing repositories will result in immediate disqualification.

Submission

- Upload your code to GitHub and ensure the repository is **public**
- Deploy your application (Vercel, Netlify, Render, Railway, or any cloud service)
- Submit both the GitHub repository link and the deployed application link

Evaluation Criteria

Criteria	What We Look For
Functionality	All core features working correctly
UI/UX	Visual similarity to Cal.com's design and UX patterns
Database Design	Well-structured schema with proper relationships
Code Quality	Clean, readable, and well-organized code
Code Modularity	Proper separation of concerns, reusable components
Code Understanding	Ability to explain your code during evaluation

Timeline

Submission Deadline: 2 days from the date of receiving this assignment

Good luck!