Project Id: (MCA2024MPID06)

Subject code:KCA353

Project Report

on

Time Align: Automated Meeting Scheduler Application

Submitted in Partial Fulfillment of the Requirement For the Degree of

Master of Computer Application

By

Aditi Gupta (2302900140004)

Khushi Chaudhary (2302900140018)

Tusharika Jain (2302900140035)

Under the Supervision of

Ms. Monika Gaur

(Assistant professor)

ABES INSTITUTE OF TECHNOLOGY, GHAZIABAD

AFFILIATED TO

Dr A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, UTTAR PRADESH, LUCKNOW

(ODD SEM, 2024-25)

DECLARATION

We here by declare that the work presented in this report entitled “TIME ALIGN : Automated Meeting Scheduler Application ”, was carried out by us. We have not submitted the matter embodied in this report for the award for any other degree or diploma of any other university or institute. We have given due credit to the original authors/sources for all the word, ideas, diagrams, graphics, computer programs, experiments, results that are not our original contribution. We have used quotation marks to identify verbatim sentences and giving credit to the original authors/sources.

We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, we shall be fully responsible and answerable.

Name: Aditi Gupta
Roll No.: 2302900140004
Master of Computer Applications (Candidate Signature)

Name: Tusharika Jain
Roll No.: 2302900140035
Master of Computer Applications (Candidate Signature)

Name: Khushi Chaudhary
Roll No.: 2302900140018
Master of Computer Applications (Candidate Signature)

CERTIFICATE

Certified that **Aditi Gupta (Roll no. 2302900140004), Tusharika Jain (2302900140035) and Khushi Chaudhary (2302900140018)** has carried out the research work presented in this thesis entitled “Time Align: A Meeting Scheduler Web Application ” for the award of Master of Computer Application from Dr. APJ Abdul Kalam Technical University, Lucknow under my/our supervision. The report embodies result of original work, and studies are carried out by the student himself/herself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Mrs. Monika Gaur
Assistant Professor
Department of MCA

Prof. (Dr.) Sheelesh Kumar Sharma
Professor and Head
Department of MCA

ABES Institute of Technology, Ghaziabad

Date:
Place: Ghaziabad

ACKNOWLEDGEMENT

With deep gratitude, we express our earnest thanks to our esteemed supervisor Prof. Monika Gaur, Department of Master of Computer Application for her constant involvement, energetic efforts, and proficient guidance, which gave us direction and body to work, respond here. Without her counsel and encouragement, it would have been impossible to complete the project work in this manner.

I wish to express my gratitude to the Director and Dean Academics of ABES IT for their intellectual support throughout this work. We are thankful to all the faculty members of the MCA Department, especially for their support and advice during our project work. We also want to thank our friends for their valuable support whenever we needed it. We would like to thank all those people who have helped us in some way or the other in my project work.

Lastly, and most importantly, we thank our parents for their moral support and encouragement towards completing my project successfully. In the last, we want to thank Almighty God.

Aditi Gupta MCA (Final Year)
Roll No: 2302900140004
Date: Department of MCA
Place: Ghaziabad ABES Institute of Technology, Ghaziabad

Tusharika Jain MCA (Final Year)
Roll No: 2302900140035
Date: Department of MCA
Place: Ghaziabad ABES Institute of Technology, Ghaziabad

Khushi Chaudhary MCA (Final Year)
Roll No: 2302900140018
Date: Department of MCA
Place: Ghaziabad ABES Institute of Technology, Ghaziabad

ABSTRACT

The “Meeting Scheduler project” aims to develop an intelligent, automated system that streamlines the process of scheduling meetings by minimizing manual effort and reducing scheduling conflicts . This project seeks to address the challenges of coordinating participants' availability, managing different time zones, and avoiding overlapping commitments.

In today’s fast-paced environment, effective time management and collaboration are crucial for organizational success. This abstract presents a Meeting Scheduler Application designed to streamline the process of scheduling meetings across diverse teams and time zones. The application integrates a user-friendly interface with intelligent algorithms to automate the selection of optimal meeting times based on participants’ availability, preferences, and time zone considerations.

The proposed solution integrates seamlessly with existing calendar systems to provide real-time visibility into participants’ availability. Additional features include automated reminders, recurring meeting setups, and resource management for room bookings or video conferencing links.

The ultimate goal of this project is to improve productivity and collaboration by offering a user friendly interface that simplifies meeting scheduling, enhances communication, and ensures that meetings are set at the most convenient times for all participants.

LIST OF FIGURES

DESCRIPTION PAGE NO.

TABLE OF CONTENTS

DESCRIPTION Page No.

CHAPTER 1 : INTRODUTION

1.1 OVERVIEW :

In the contemporary workplace, effective communication and collaboration are paramount for achieving organizational goals. With teams often spread across different locations and time zones, scheduling meetings can become a significant challenge. Traditional methods, such as back-and forth emails or messaging, can lead to confusion, wasted time, and missed opportunities for collaboration. The Meeting Scheduler Application aims to address these challenges by providing a streamlined, efficient solution for scheduling meetings. This application leverages technology to simplify the scheduling process, enabling users to coordinate their availability effortlessly. By integrating with existing calendar systems, the application allows users to see real-time availability and suggest optimal meeting times that accommodate all participants. Key functionalities of the Meeting Scheduler Application include automated time zone adjustments, customizable reminders, and integration with popular communication platforms. These features not only enhance user experience but also ensure that meetings are scheduled in a way that maximizes participation and minimizes conflicts. As organizations increasingly adopt remote and hybrid work models, the need for an effective scheduling tool becomes even more critical. This application not only saves time but also fosters a culture of collaboration and engagement among team members, ultimately contributing to improved productivity and successful outcomes.

1.2 OBJECTIVE & AIM :

The objective of the web-based meeting scheduling application is to simplify and streamline the process of organizing, scheduling, and managing meetings. The application aims to enhance productivity and reduce the time spent coordinating availability, sending invitations, and managing follow-ups by providing an intuitive, automated, and integrated platform.

The aim of the application is to provide users with:

Ease of Scheduling: An intuitive interface that enables users to schedule meetings efficiently.

Seamless Integration: Integration with popular calendar systems to check availability and avoid conflicts.

Automation: Automatic notifications, reminders, and follow-ups for meetings.

Accessibility: A responsive, mobile-friendly platform accessible on any device with internet connectivity.

Scalability: The ability to support a wide range of users, from individuals to large organizations.

1.3 PROBLEM STATEMENT :

Effective project management requires regular meetings for coordination, progress tracking, and decision-making. However, scheduling these meetings, especially in large teams or cross-functional groups, often becomes a complex and time-consuming task. Coordinating between participants' availability, managing time zones for global teams, avoiding conflicts, and ensuring agenda alignment is challenging and prone to inefficiencies. This results in:

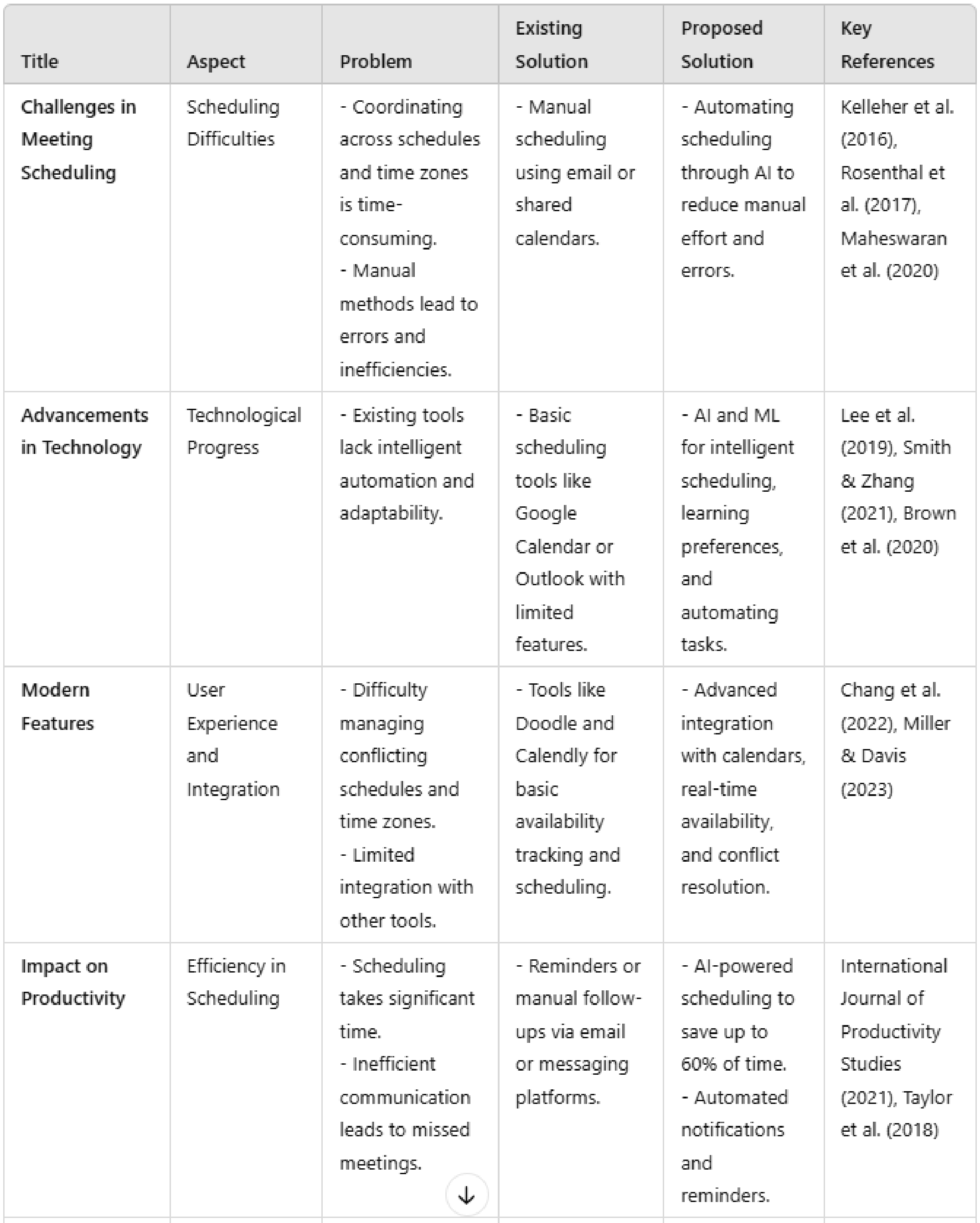
- Time wastage due to back-and-forth communication for scheduling.
- Missed meetings or delays caused by overlooked conflicts.
- Low productivity due to poorly timed or unnecessary meetings.
- Difficulty tracking decisions, action items, and meeting outcomes.
- Inability to adapt to sudden changes in availability or priorities.

1.4 SCOPE :

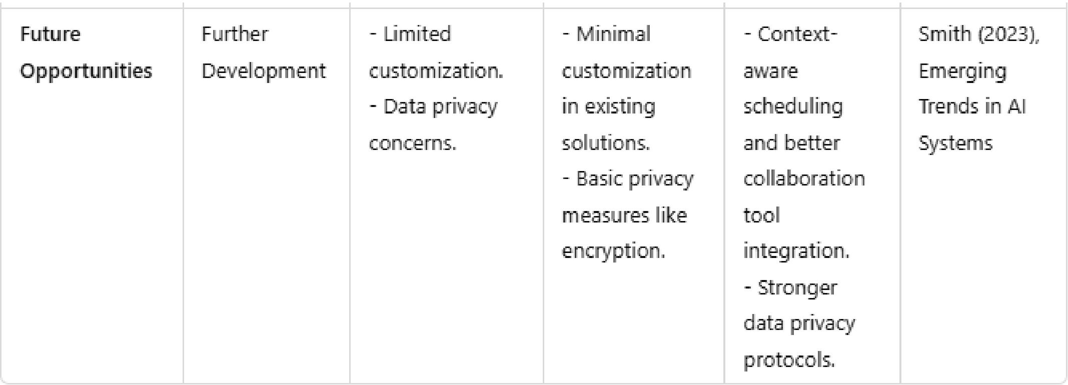
The meeting scheduling application will enable users to:

- Schedule meetings with participants based on mutual availability.
- Set meeting agendas and share necessary resources.
- Send notifications and reminders to participants.
- Integrate with calendar systems (e.g., Google Calendar, Outlook). The system will support both desktop and mobile web browsers.

CHAPTER 2 : LITERATURE REVIEW



Title	Aspect	Problem	Existing Solution	Proposed Solution	Key References
Challenges in Meeting Scheduling	Scheduling Difficulties	- Coordinating across schedules and time zones is time-consuming. - Manual methods lead to errors and inefficiencies.	- Manual scheduling using email or shared calendars.	- Automating scheduling through AI to reduce manual effort and errors.	Kelleher et al. (2016), Rosenthal et al. (2017), Maheswaran et al. (2020)
Advancements in Technology	Technological Progress	- Existing tools lack intelligent automation and adaptability.	- Basic scheduling tools like Google Calendar or Outlook with limited features.	- AI and ML for intelligent scheduling, learning preferences, and automating tasks.	Lee et al. (2019), Smith & Zhang (2021), Brown et al. (2020)
Modern Features	User Experience and Integration	- Difficulty managing conflicting schedules and time zones. - Limited integration with other tools.	- Tools like Doodle and Calendly for basic availability tracking and scheduling.	- Advanced integration with calendars, real-time availability, and conflict resolution.	Chang et al. (2022), Miller & Davis (2023)
Impact on Productivity	Efficiency in Scheduling	- Scheduling takes significant time. - Inefficient communication leads to missed meetings.	- Reminders or manual follow-ups via email or messaging platforms.	- AI-powered scheduling to save up to 60% of time. - Automated notifications and reminders.	International Journal of Productivity Studies (2021), Taylor et al. (2018)



Future Opportunities	Further Development	- Limited customization. - Data privacy concerns.	- Minimal customization in existing solutions. - Basic privacy measures like encryption.	- Context-aware scheduling and better collaboration tool integration. - Stronger data privacy protocols.	Smith (2023), Emerging Trends in AI Systems

Feature/Aspect	Description	Target Audience	Competitors
User Interface	Intuitive, easy-to-navigate design	General users, teams	Doodle
Integration	Compatibility with calendars (Google, Outlook)	Professionals, businesses	Microsoft Teams, Zoom
Availability Checking	Automatic availability checks among participants	Project managers, teams	Time Zone Ninja, When2meet
Time Zone Support	Adjusts for different time zones	Remote teams	World Time Buddy, Every Time Zone
Recurring Meetings	Support for setting up regular meetings	Businesses, organizations	Zoom, Microsoft Teams
Custom Links	Personalized scheduling links for users	Sales teams, marketers	Acuity Scheduling

Chapter 3: ANALYSIS

3.1 FEASIBILITY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

3.1.1 Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. A feasibility study evaluates the project's potential for success.

3.1.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Requirements:

- **Architecture :**
  X86 or X86-64-bit hardware architecture.
  Intel/AMD processor with compatibility Motherboards.
- **Processing Power :**
  Core2Duo 2.0-gigahertz(GHz) processor or faster.
- **Memory :**
  At least 512 megabytes(MB) of RAM(1GB is recommended).
- **Secondary Storage :**
  At least 10 gigabytes(GB)of available space on the hard disk.

3.2.2 Software Requirements:

- **Operating System:** Windows 7 or higher.
- **Front-end:** Next.js, Tailwind, JavaScript.
- **Front-end Library:** React js**.**
- **Backend:** Neon DB (Postgres DB).
- **Operating System**: Windows 7, 8, 9, 10 .

Chapter 4 :DESIGN

4.1 Design Overview

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.2 Module Description

We have separated the software into different modules and these all modules have different tasks in the system.

- Calendar integration: Syncs with Google calendar to access participants' availability.
- Automated scheduling: Suggests the best available times for all participants based on their calendar data.
- Meeting room booking: For in-person meetings, integrates with room management systems to ensure availability.
- Recurring meetings: Allows easy scheduling of repeat or recurring meetings with set intervals.

- Custom availability windows: Participants can set availability preferences for certain days or times.

4.3 FLOWCHART

A flowchart is a visual representation of a process or algorithm using various shapes, symbols, and arrows to illustrate the sequence of steps or actions. It provides a clear and concise way to depict the flow of information, decision points, and the order of operations within a system or procedure. Flowcharts are widely used in various fields such as software development, business processes, project planning, and problem-solving. They serve as a valuable tool for understanding, documenting, and communicating complex processes in a easily understandable and systematic manner.

THIS IS A FLOWCHART OF TIME ALIGN PROJECT

A meeting scheduling application website

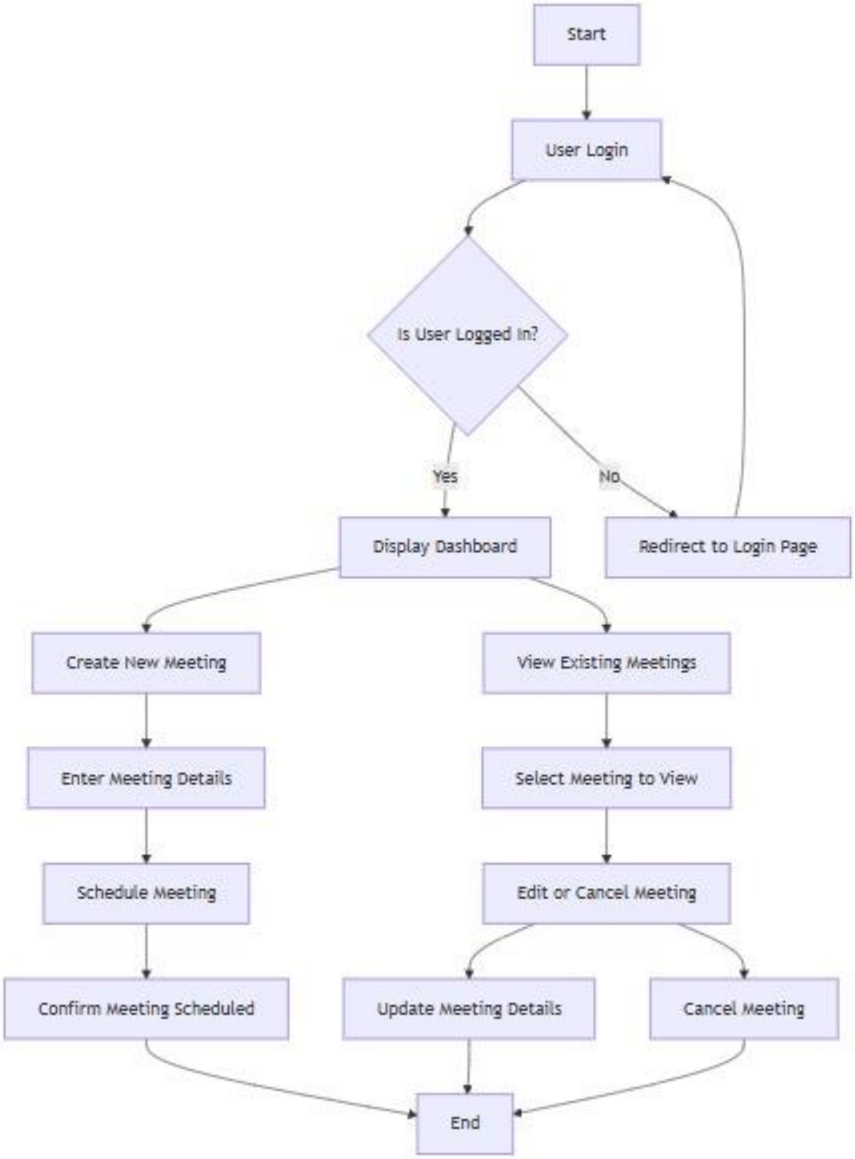


Fig 4.3: FLOW CHART

4.4 USE CASE DIAGRAM

A use case diagram is a graphical representation that illustrates how a system interacts with external entities (actors) and showcases the various ways these entities can interact with the system. The primary purpose of a use case diagram is to capture and communicate the functional requirements of a system in a clear and visual manner.

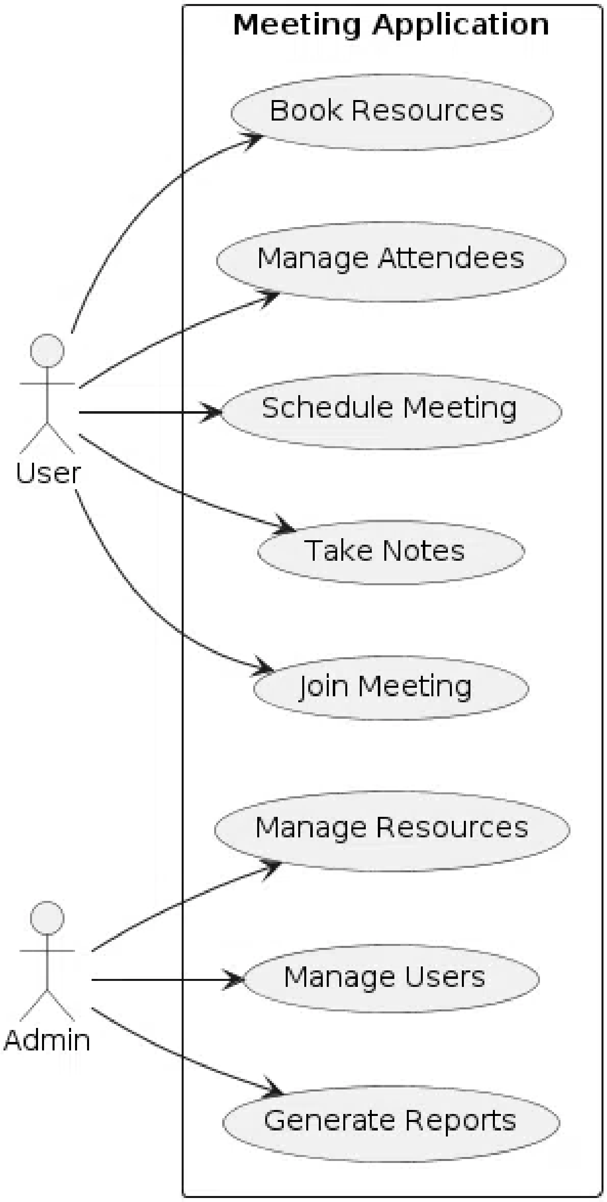


Fig 4.4: USE CASE DIAGRAM

4.5 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a diagram that describes the flow of data and the process that change data throughout a system It's a structured analysis and design tool that can be used for flow charting in place of or in association with information. The DFD reviews the current system, prepare input and output specification, specifies the implementation plan etc. Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

4.5.1 Context Level Diagram (0 Level)

A Level 0 Data Flow Diagram (DFD), also known as a Context Diagram, provides a high level view of a system by illustrating the interactions between the system and external entities. It represents the system as a single process and showcases the flow of data between the system and its external entities. The primary purpose of a Level 0 DFD is to offer a concise and easily understandable overview of the system's scope and boundaries.

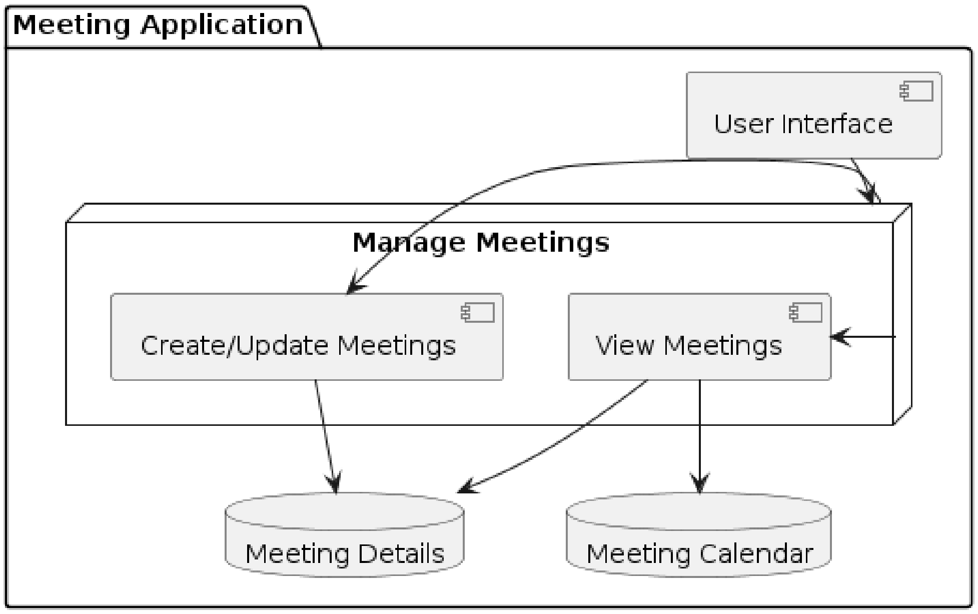


Fig 4.5.1: LEVEL 0 DFD

4.5.2 Data Flow Diagram (1 Level)

A Level 1 Data Flow Diagram (DFD) is a visual representation that provides a more detailed view of the processes and data flows within a system compared to a Level 0 DFD. At Level 1, the diagram expands on the processes identified in the Level 0 DFD and decomposes them into sub processes, offering a more granular depiction of the system's functionality.

The Level 1 DFD acts as an intermediary step between the high-level overview of the Level 0 DFD and the more detailed representations that might follow in subsequent levels. It helps in breaking down complex processes into manageable components, making it easier for analysts and stakeholders to understand the system's inner workings.

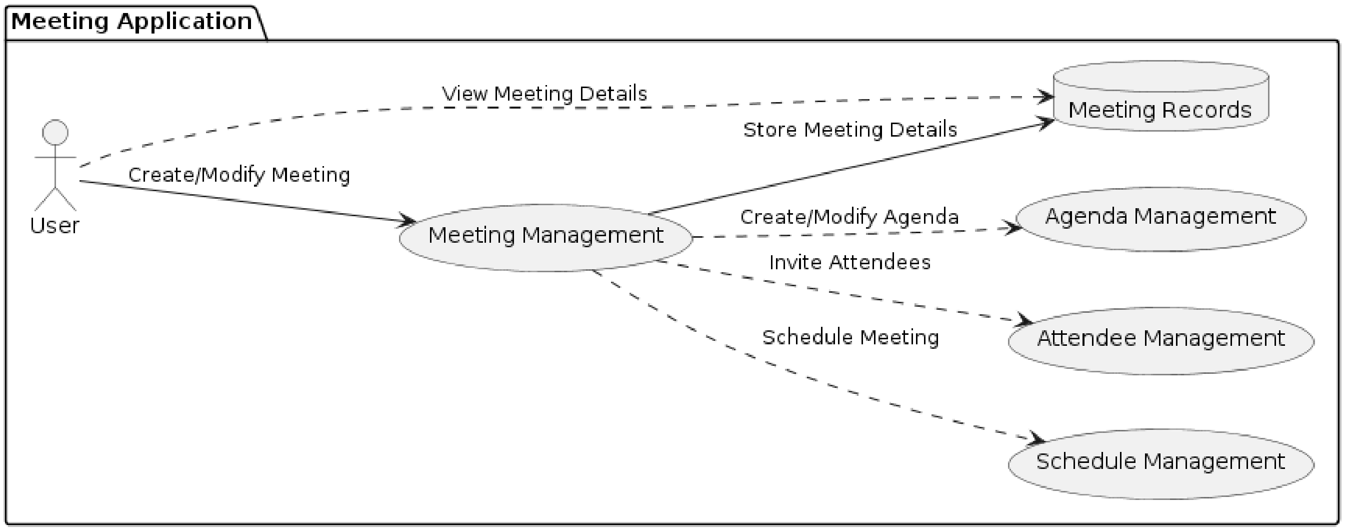
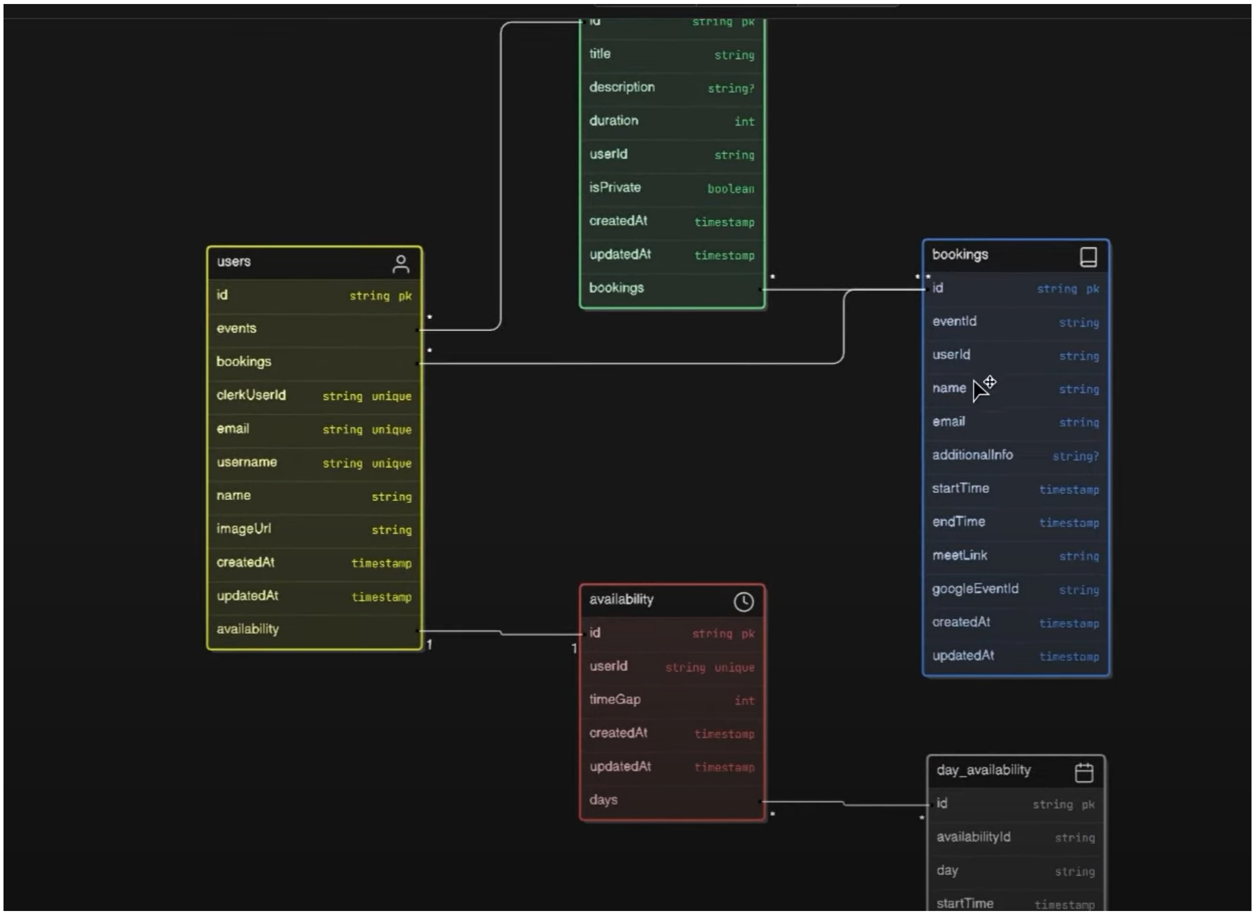


Fig 4.5.2: DFD Level 1

4.6 DATABASE DESIGN



id	string pk
title	string
description	string?
duration	int
userId	string
isPrivate	boolean
createdAt	timestamp
updatedAt	timestamp
bookings	

users	
id	string pk
events	
bookings	
clerkUserId	string unique
email	string unique
username	string unique
name	string
imageUrl	string
createdAt	timestamp
updatedAt	timestamp
availability	

bookings	
id	string pk
eventId	string
userId	string
name	string
email	string
additionalInfo	string?
startTime	timestamp
endTime	timestamp
meetLink	string
googleEventId	string
createdAt	timestamp
updatedAt	timestamp

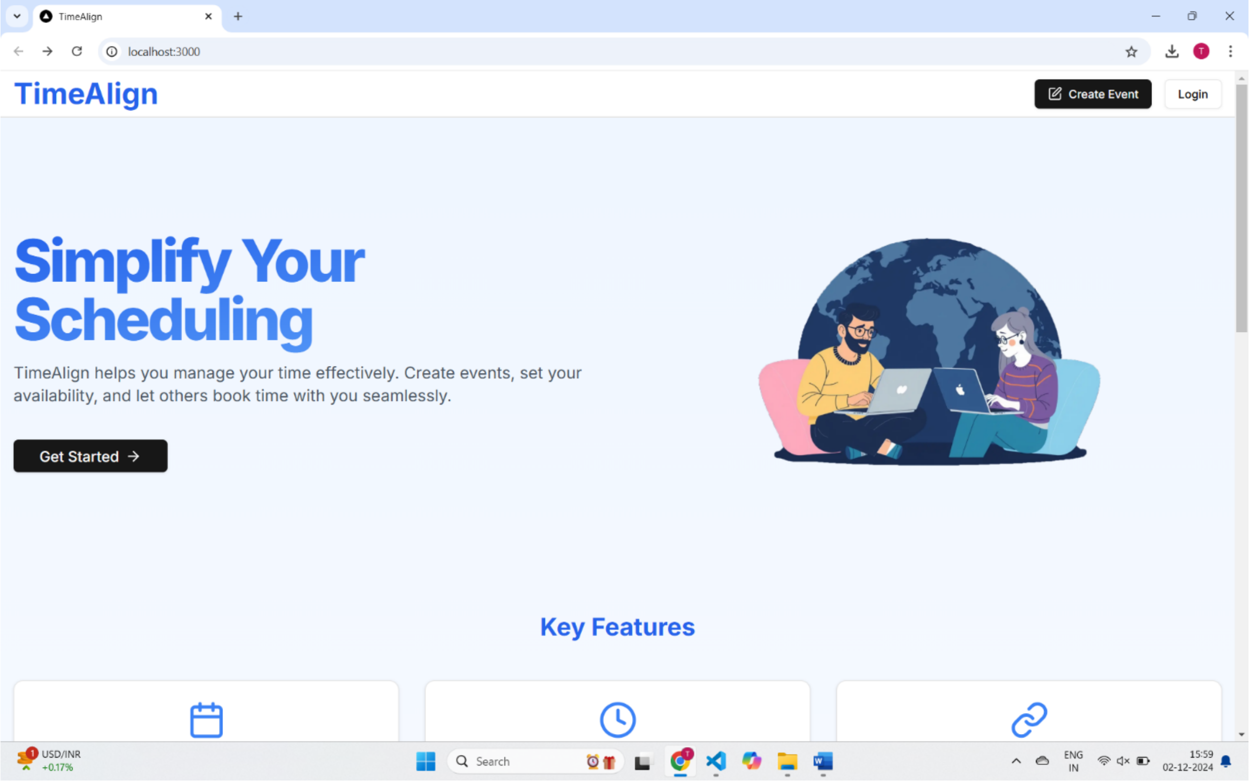
availability	
id	string pk
userId	string unique
timeGap	int
createdAt	timestamp
updatedAt	timestamp
days	

day_availability	
id	string pk
availabilityId	string
day	string
startTime	timestamp

Fig 4.6: Database design

4.7 INPUT/OUTPUT FORM DESIGN

4.7.1 USERS INTERFACE :



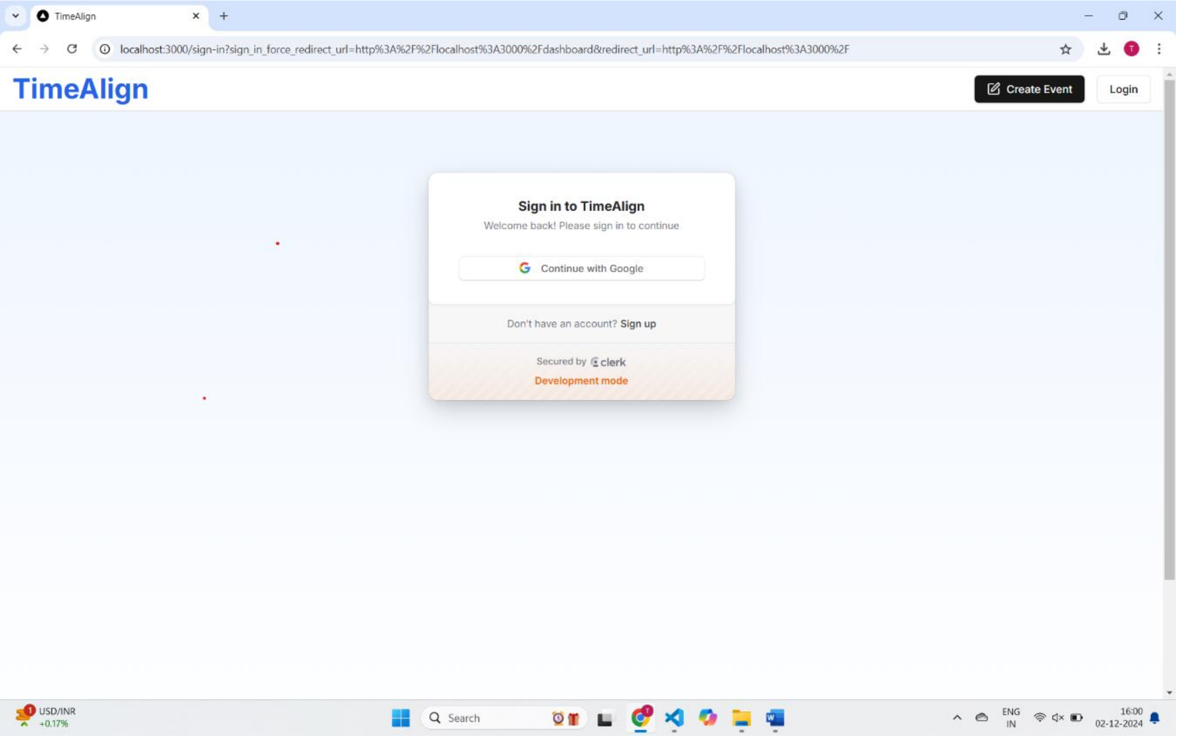
TimeAlign

Simplify Your Scheduling

TimeAlign helps you manage your time effectively. Create events, set your availability, and let others book time with you seamlessly.

Key Features

Fig4.7.1.1: Landing Page



TimeAlign
localhost:3000/sign-in?sign_in_force_redirect_url=http%3A%2F%2Flocalhost%3A3000%2Fdashboard&redirect_url=http%3A%2F%2Flocalhost%3A3000%2F
TimeAlign
Create Event
Login
USD/INR
+0.17%
Search
ENG
IN
16:00
02-12-2024

Sign in to TimeAlign

Welcome back! Please sign in to continue

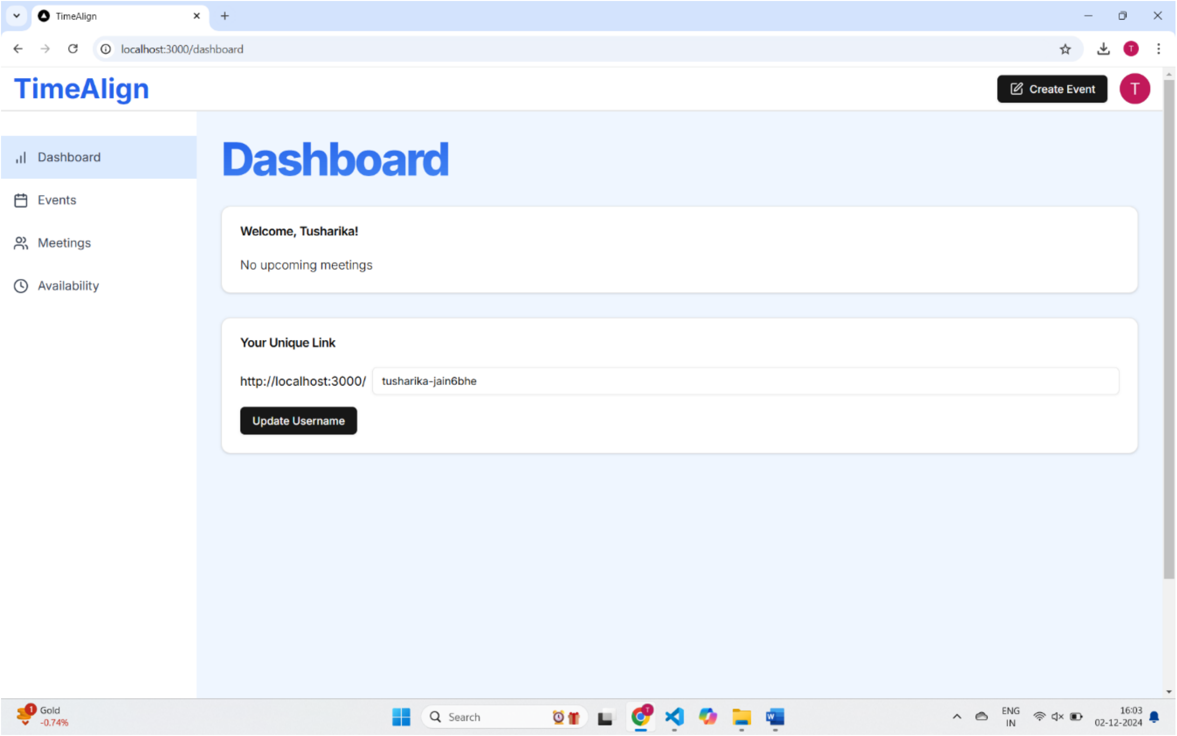
Continue with Google

Don't have an account? Sign up

Secured by clerk

Development mode

Fig4.7.1.2: Login page



Dashboard

Welcome, Tusharika!

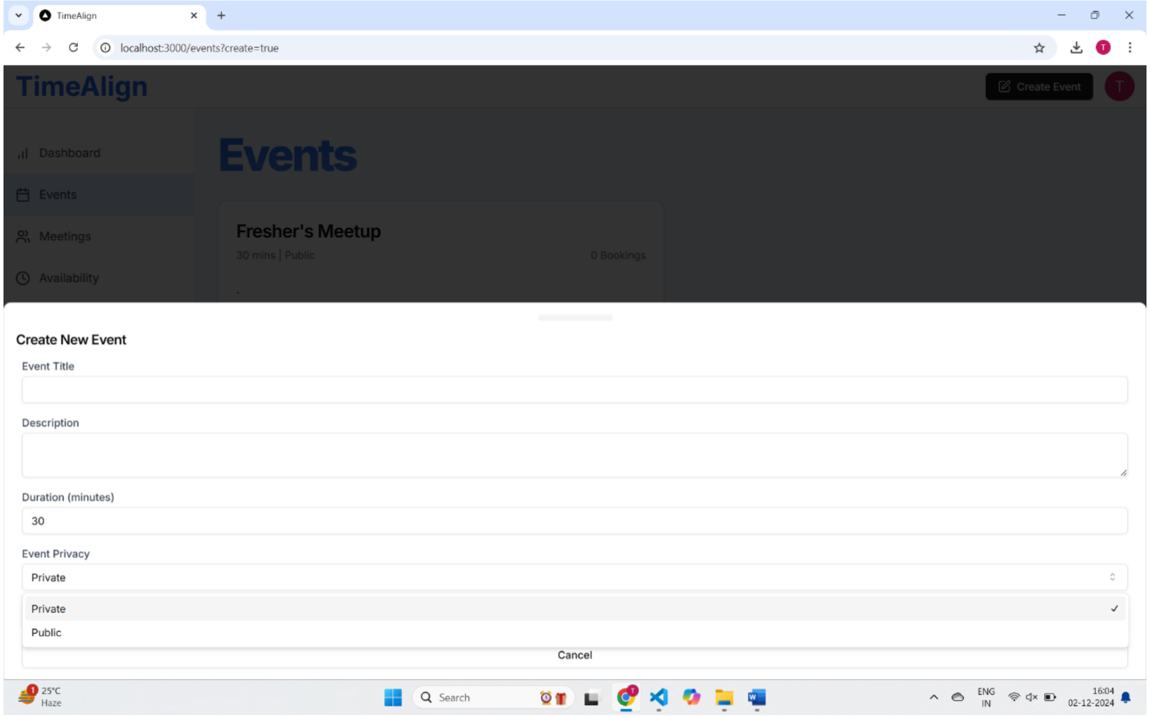
No upcoming meetings

Your Unique Link

http://localhost:3000/ tusharika-jain6bhe

Update Username

Fig4.7.1.3: Dashboard



Create New Event

Event Title

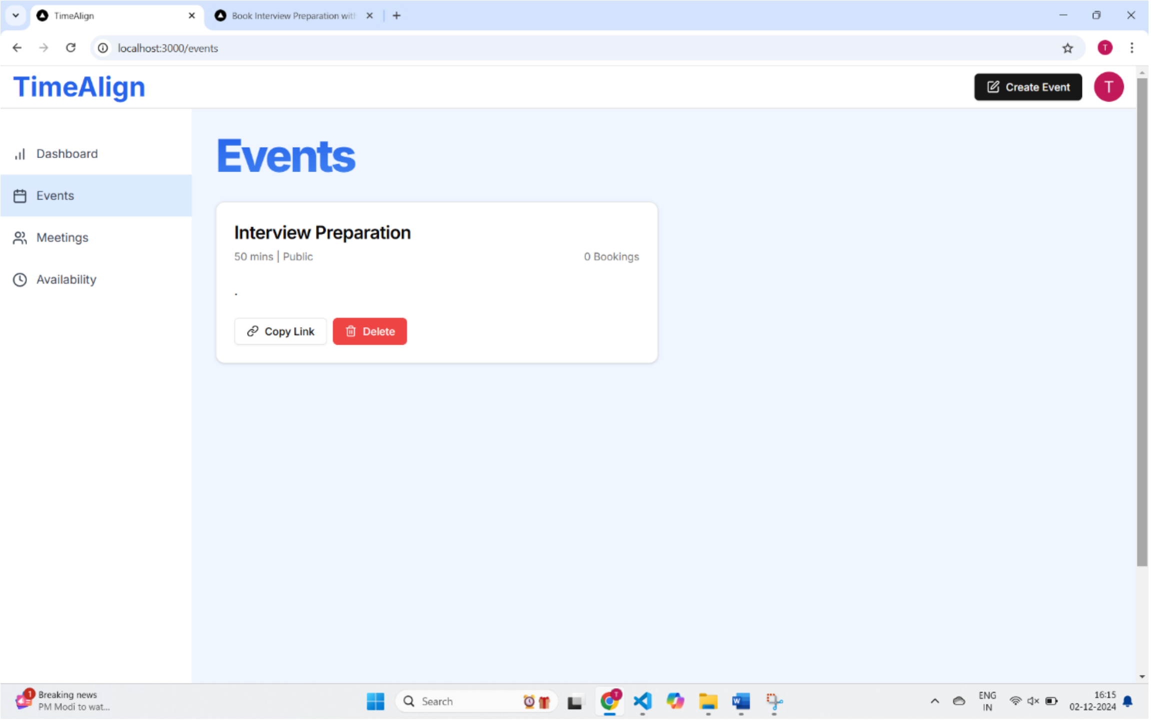
Description

Duration (minutes)
30

Event Privacy
Private

Private ✓
Public

Fig4.7.1.4: Event Creation

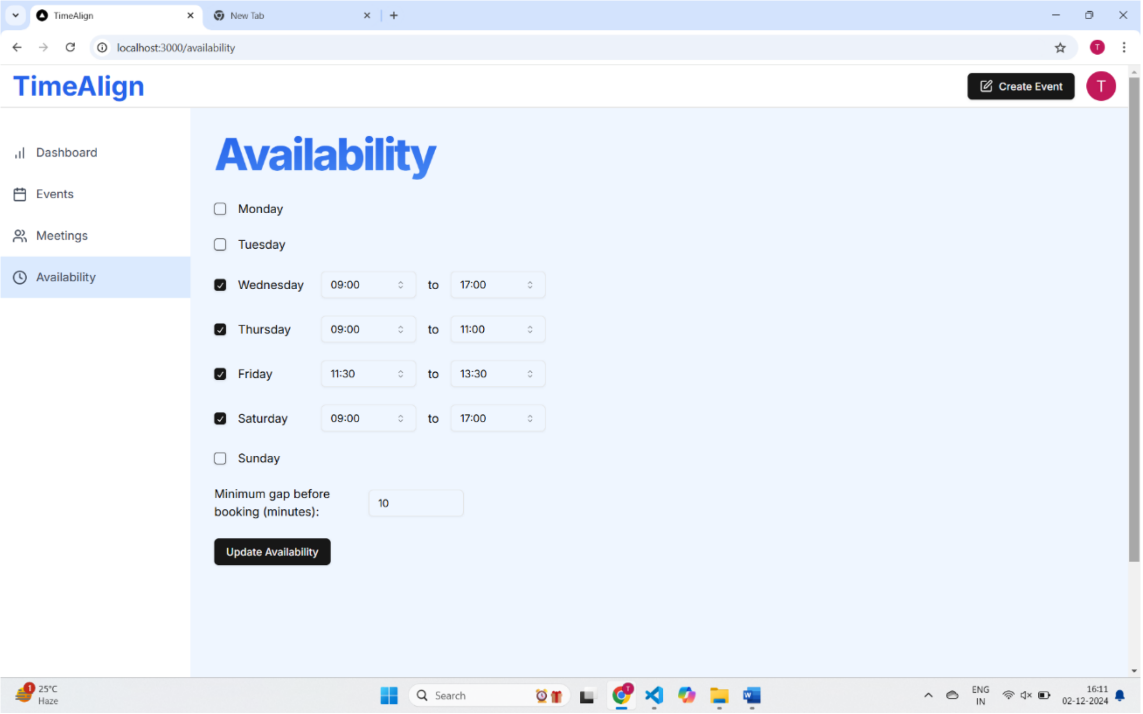


Events

Interview Preparation
50 mins | Public
0 Bookings

.

Fig4.7.1.4: Event Creation



Availability

☐ Monday

☐ Tuesday

☑ Wednesday 09:00 to 17:00

☑ Thursday 09:00 to 11:00

☑ Friday 11:30 to 13:30

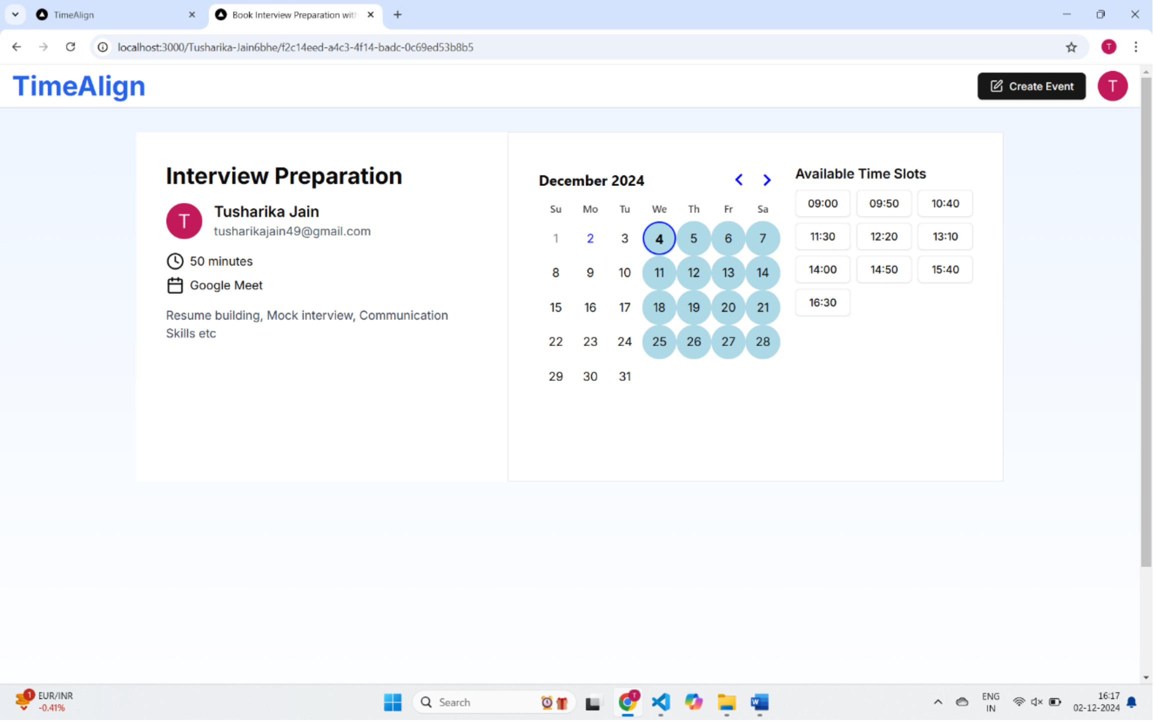
☑ Saturday 09:00 to 17:00

☐ Sunday

Minimum gap before booking (minutes): 10

Update Availability

Fig4.7.1.5: Availability


TimeAlign
Book Interview Preparation wit
localhost:3000/Tusharika-Jain6bhe/f2c14eed-a4c3-4f14-badc-0c69ed53b8b5
TimeAlign
Create Event

Interview Preparation

Tusharika Jain
tusharikajain49@gmail.com

50 minutes

Google Meet

Resume building, Mock interview, Communication Skills etc

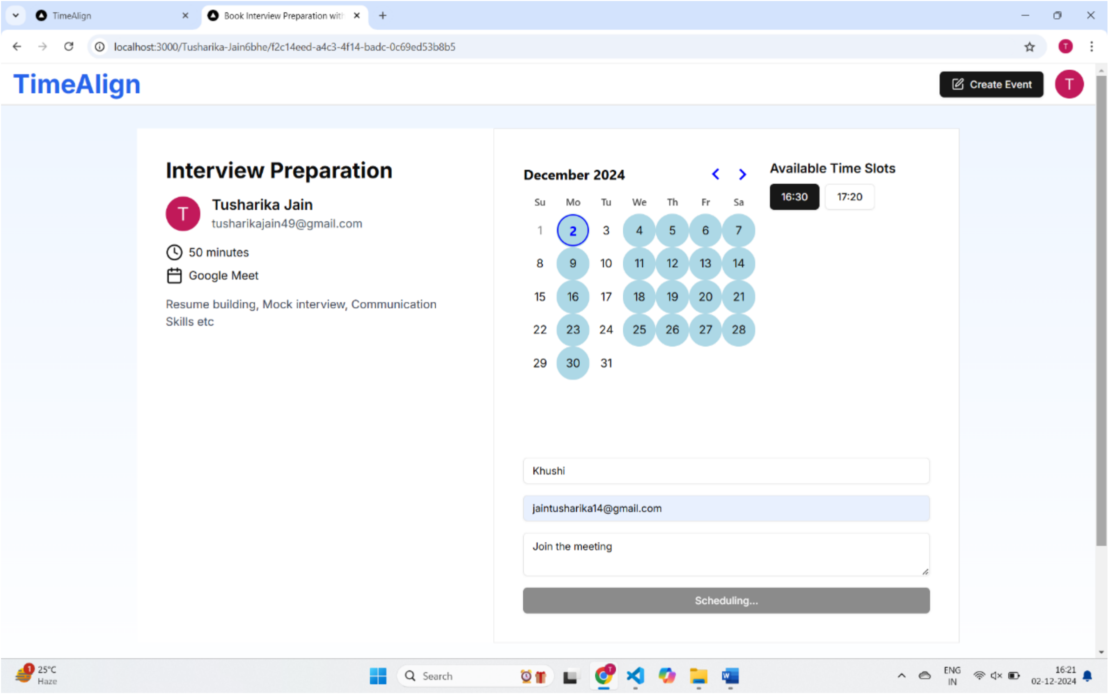
December 2024

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Available Time Slots

09:00	09:50	10:40
11:30	12:20	13:10
14:00	14:50	15:40
16:30		

Fig4.7.1.6: Choosing the date



Interview Preparation

Tusharika Jain
tusharikajain49@gmail.com

50 minutes

Google Meet

Resume building, Mock interview, Communication Skills etc

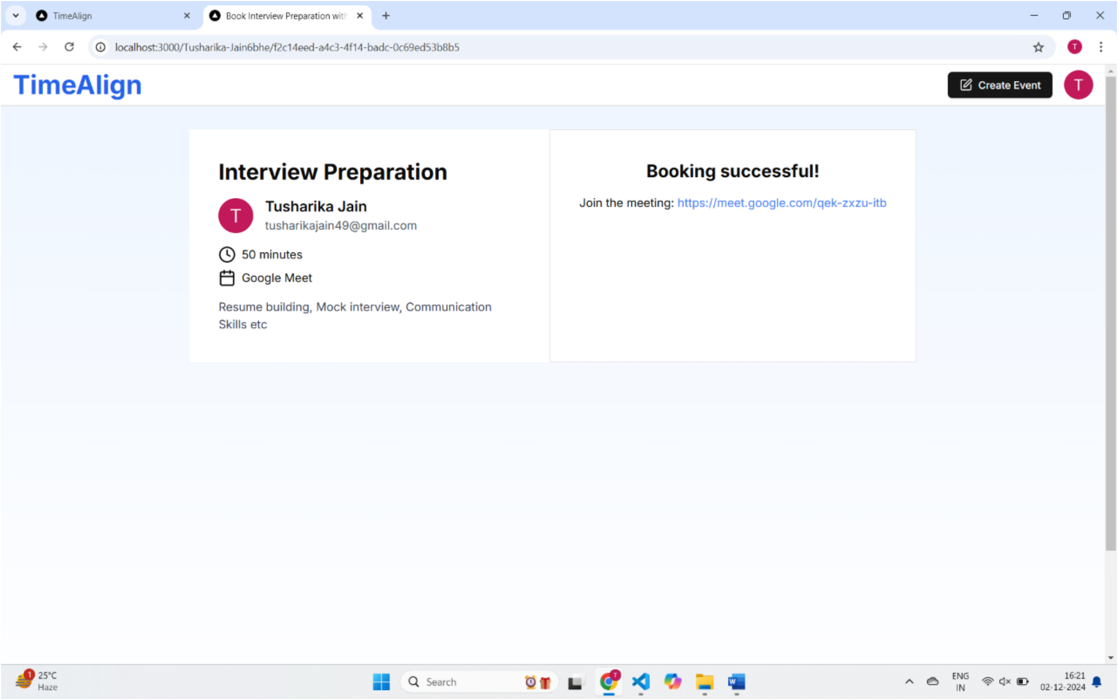
Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Khushi

jaintusharika14@gmail.com

Join the meeting

Fig4.7.1.7: Booking the Slot


TimeAlign
Book Interview Preparation wit
localhost:3000/Tusharika-Jain6bhe/f2c14eed-a4c3-4f14-badc-0c69ed53b8b5
TimeAlign
Create Event
25°C Haze
Search
ENG IN
16:21 02-12-2024

Interview Preparation

Tusharika Jain
tusharikajain49@gmail.com

50 minutes

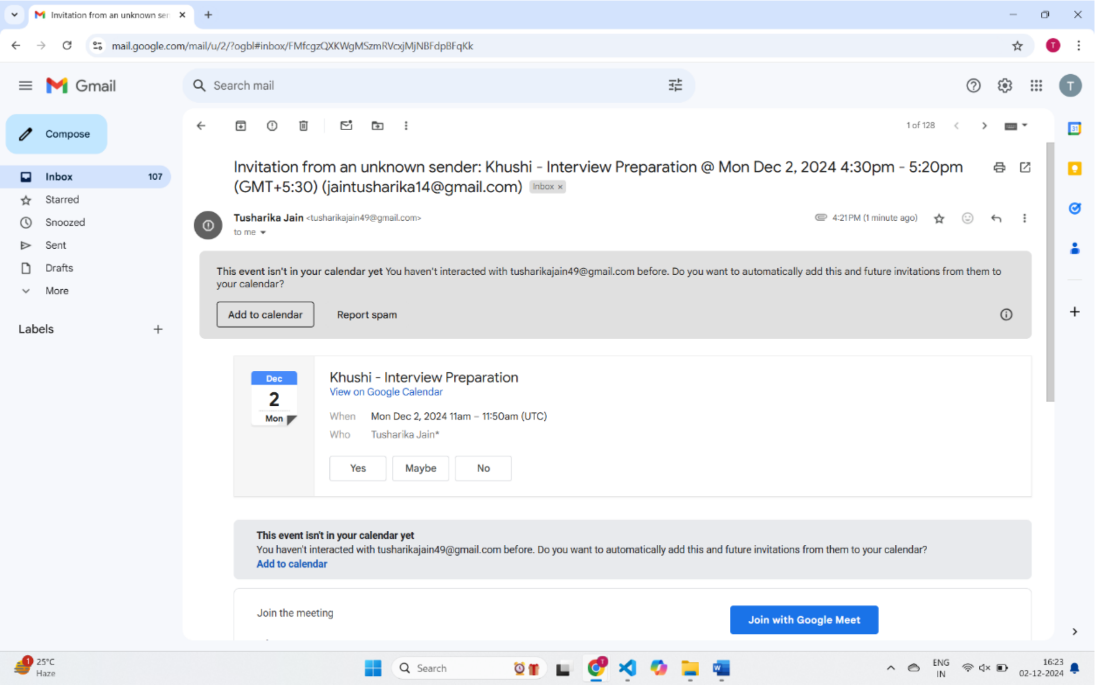
Google Meet

Resume building, Mock interview, Communication Skills etc

Booking successful!

Join the meeting: https://meet.google.com/qek-zxzu-itb

Fig4.7.1.8: Booking Successfully


Invitation from an unknown ser
mail.google.com/mail/u/2/?ogbl#inbox/FMfcgzQXKWgMSzmRVcxjMjNBFdpBFqKk
Gmail
Search mail
Compose
Inbox 107
Starred
Snoozed
Sent
Drafts
More
Labels
1 of 128
Inbox ×

Invitation from an unknown sender: Khushi - Interview Preparation @ Mon Dec 2, 2024 4:30pm - 5:20pm (GMT+5:30) (jaintusharika14@gmail.com)

Tusharika Jain <tusharikajain49@gmail.com>
to me

4:21 PM (1 minute ago)

This event isn't in your calendar yet You haven't interacted with tusharikajain49@gmail.com before. Do you want to automatically add this and future invitations from them to your calendar?

Add to calendar Report spam

Dec 2 Mon

Khushi - Interview Preparation
View on Google Calendar

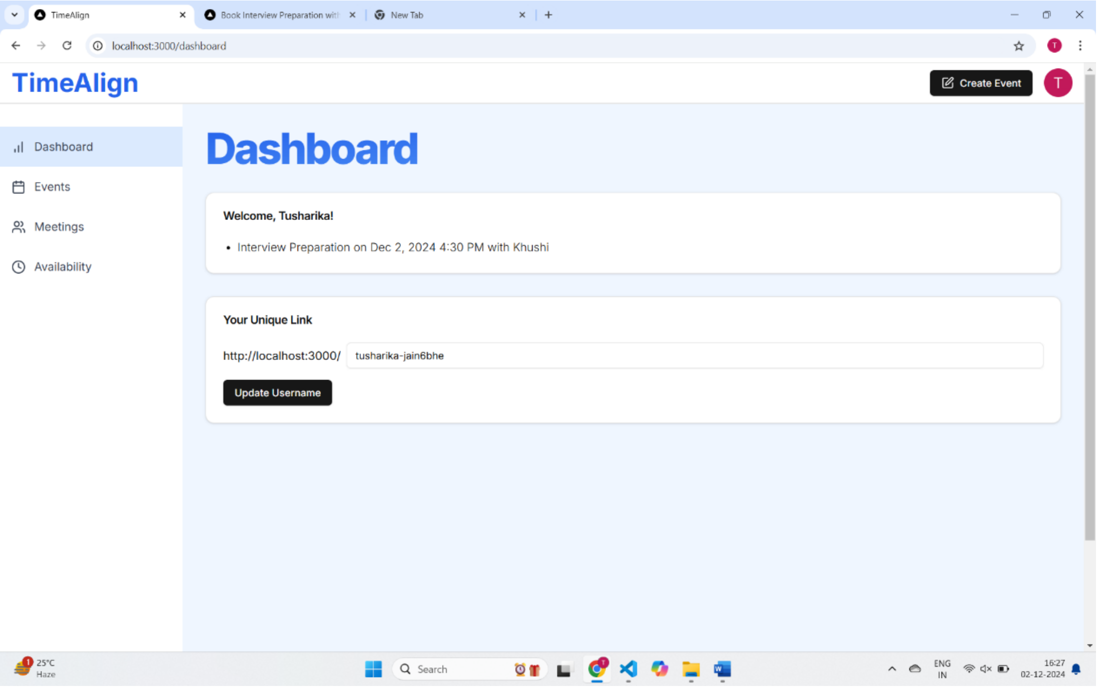
When Mon Dec 2, 2024 11am – 11:50am (UTC)
Who Tusharika Jain*

Yes Maybe No

This event isn't in your calendar yet
You haven't interacted with tusharikajain49@gmail.com before. Do you want to automatically add this and future invitations from them to your calendar?
Add to calendar

Join the meeting Join with Google Meet

Fig4.7.1.9: Meeting scheduled Mail Sent



Dashboard

Welcome, Tusharika!

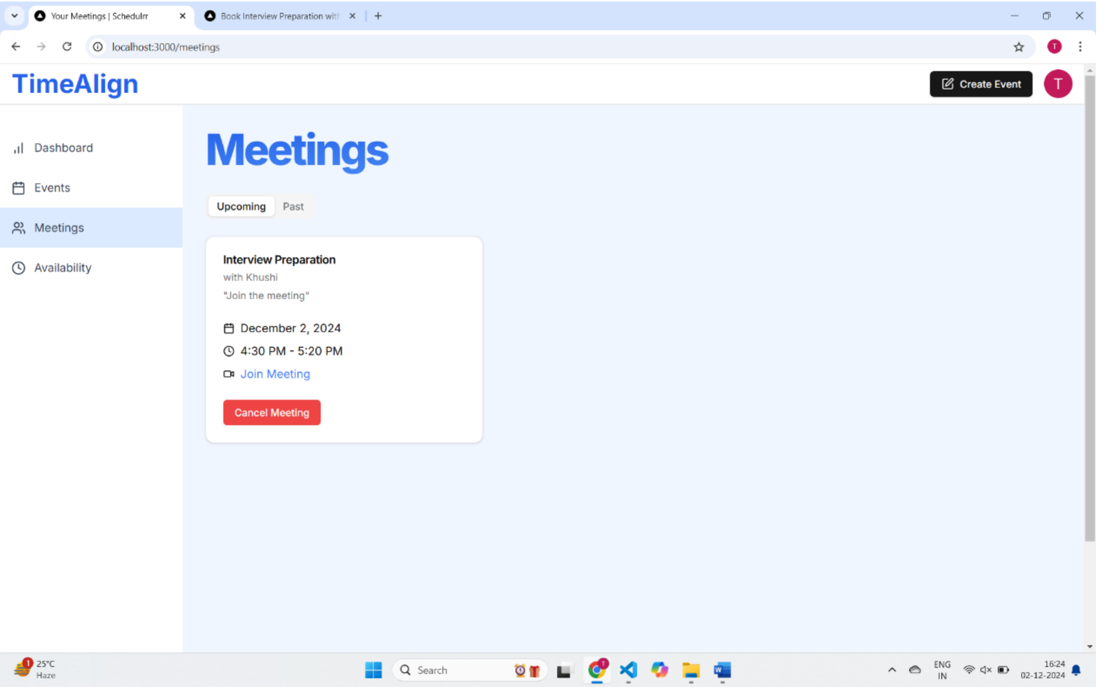
- Interview Preparation on Dec 2, 2024 4:30 PM with Khushi

Your Unique Link

http://localhost:3000/ tusharika-jain6bhe

Update Username

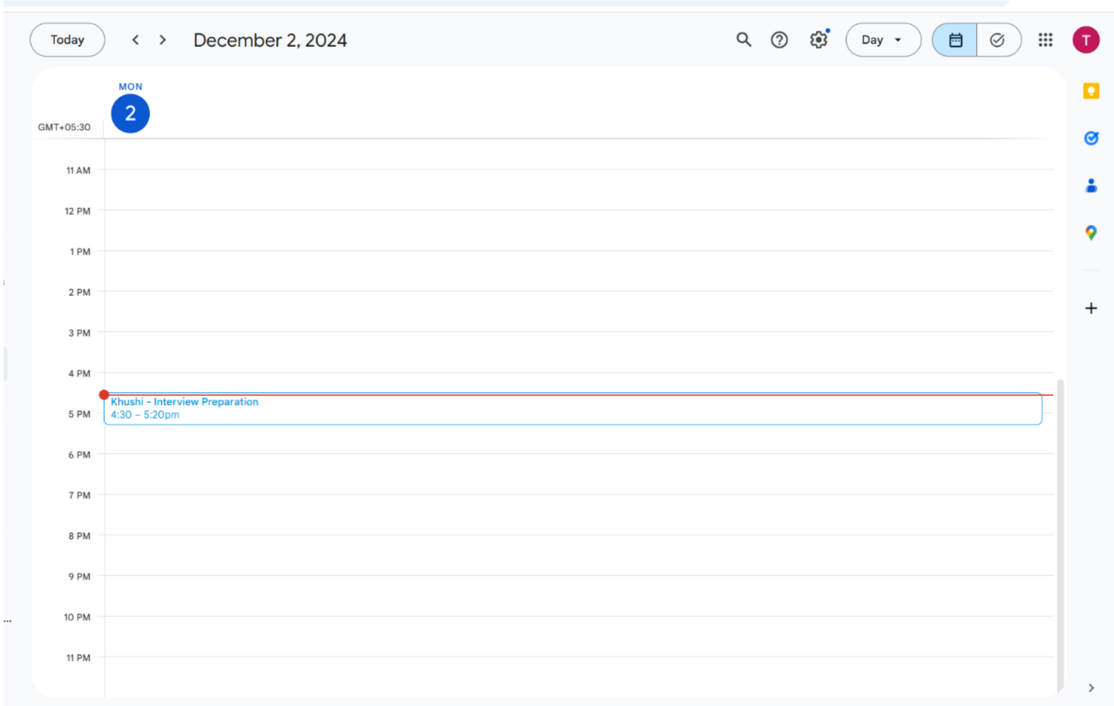
Fig4.7.1.10: Updated on Dashboard



Interview Preparation
with Khushi
"Join the meeting"

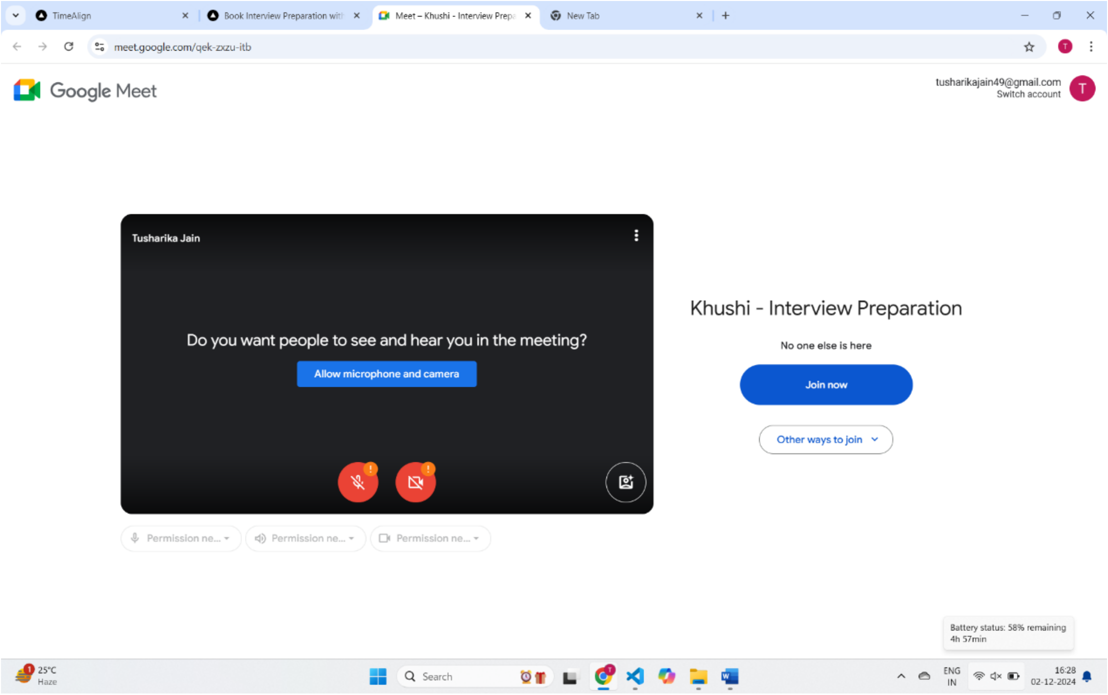
December 2, 2024
4:30 PM - 5:20 PM
Join Meeting

Fig4.7.1.11: Updated on Upcoming Meetings



Khushi - Interview Preparation
4:30 – 5:20pm

Fig4.7.1.12: Added on Calendar



TimeAlign
Book Interview Preparation wit
Meet – Khushi - Interview Prep
New Tab
meet.google.com/qek-zxzu-itb
Google Meet
tusharikajain49@gmail.com
Switch account
Tusharika Jain

Do you want people to see and hear you in the meeting?

Khushi - Interview Preparation

No one else is here

Fig4.7.1.13: Google Meet

Chapter 5: IMPLEMENTATION

5.1 Implementation Strategy

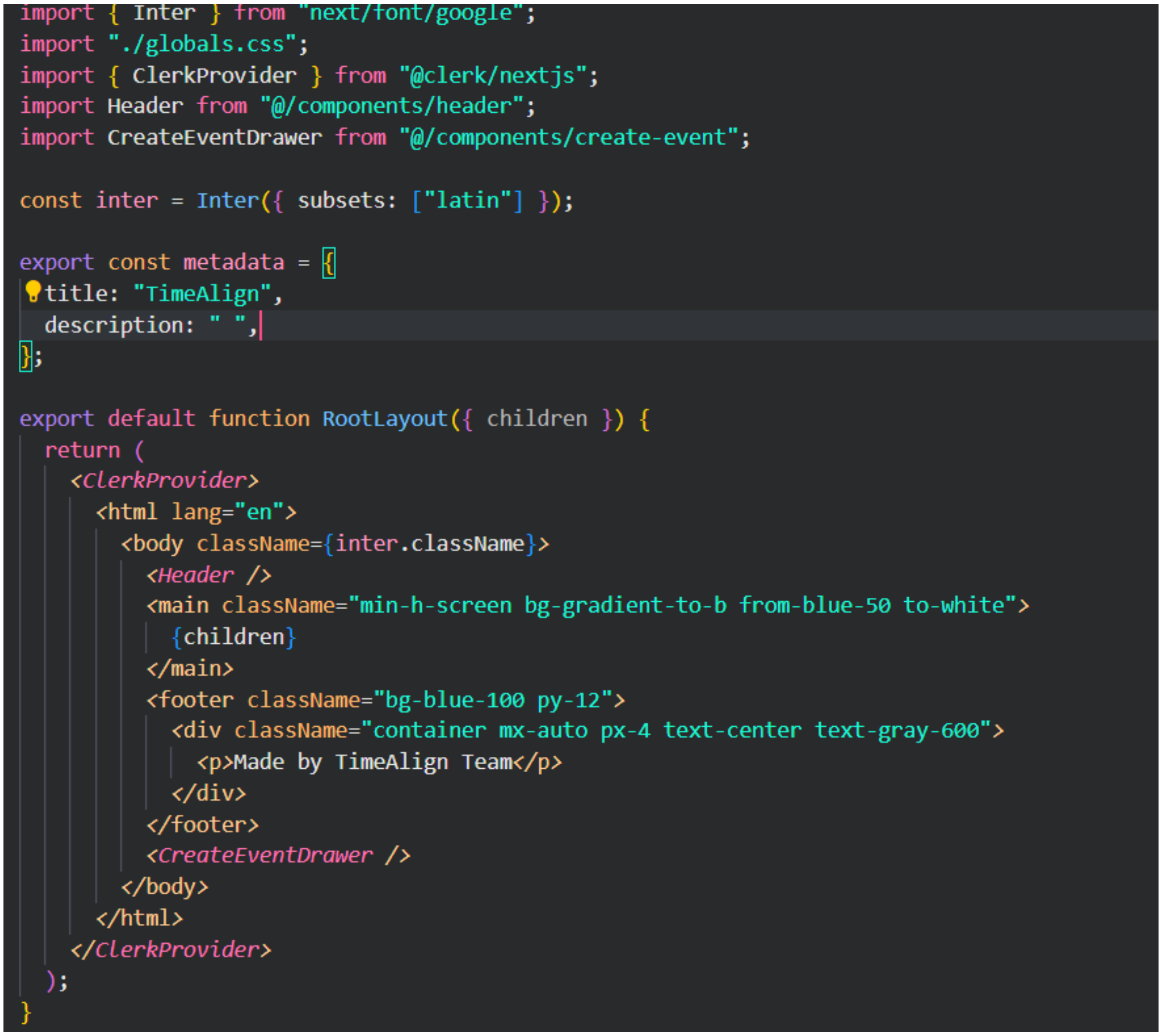
After completing the design phase, implementation phase is carried out according to the output of the previous phase using the selected design tools and techniques. This is a very important phase in SDLC because the output of this phase is the system itself. As per the design architecture, the system is divided into modules and developed individually. After that one by one, each module is integrated accordingly. In this chapter, major code segments of the system, tools, and techniques used for development and the implementation environment are discussed.

This chapter is the part that puts a planned system into action and examine in details the analysis and design of the Time Align. The present chapter discusses the implementation of the system, highlighting the testing exercise and describing some of the main components of the system's Graphical User Interface. It will give an output from programming language and other tools used to develop our system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

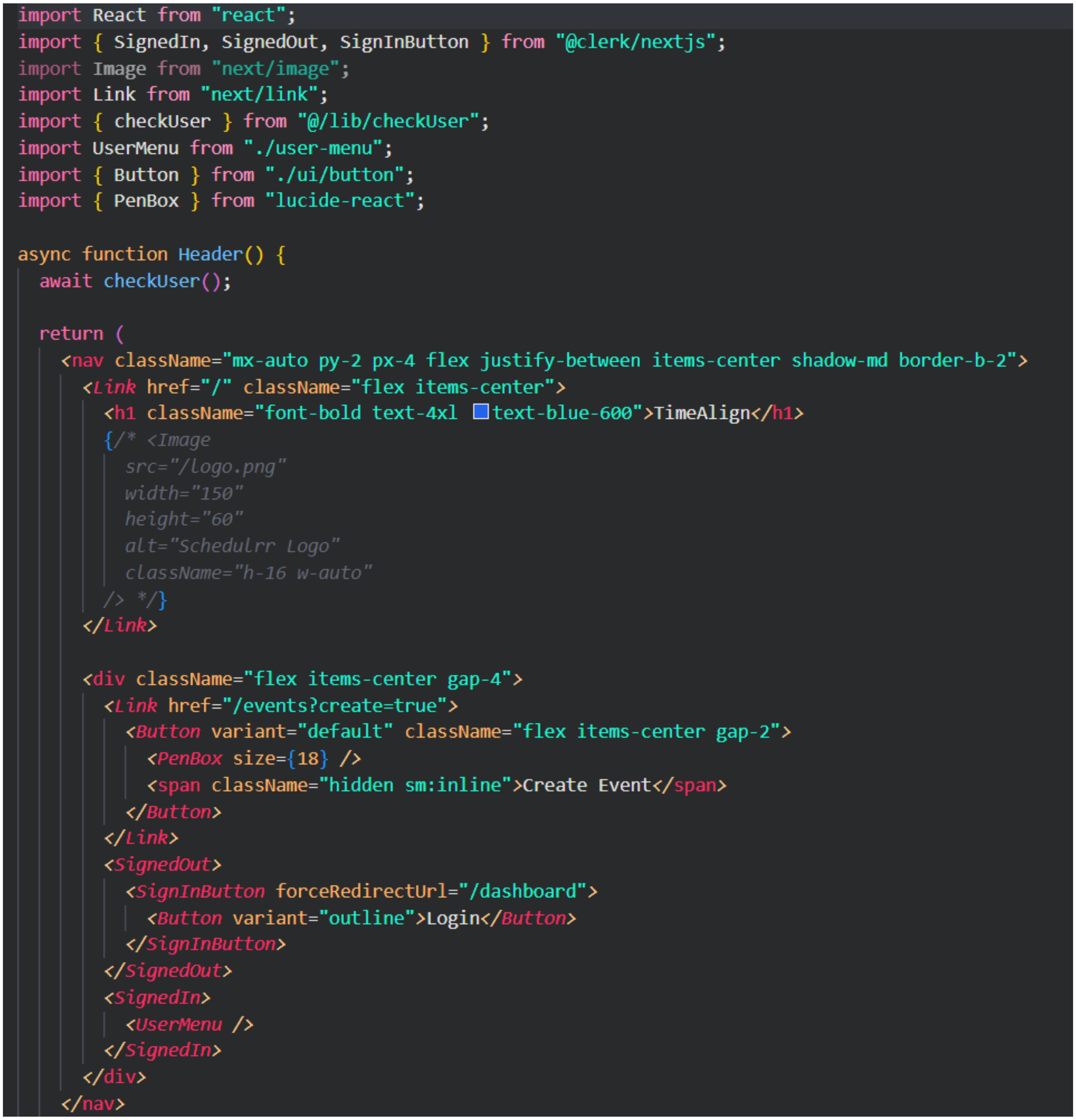
5.2 CODING

USER:

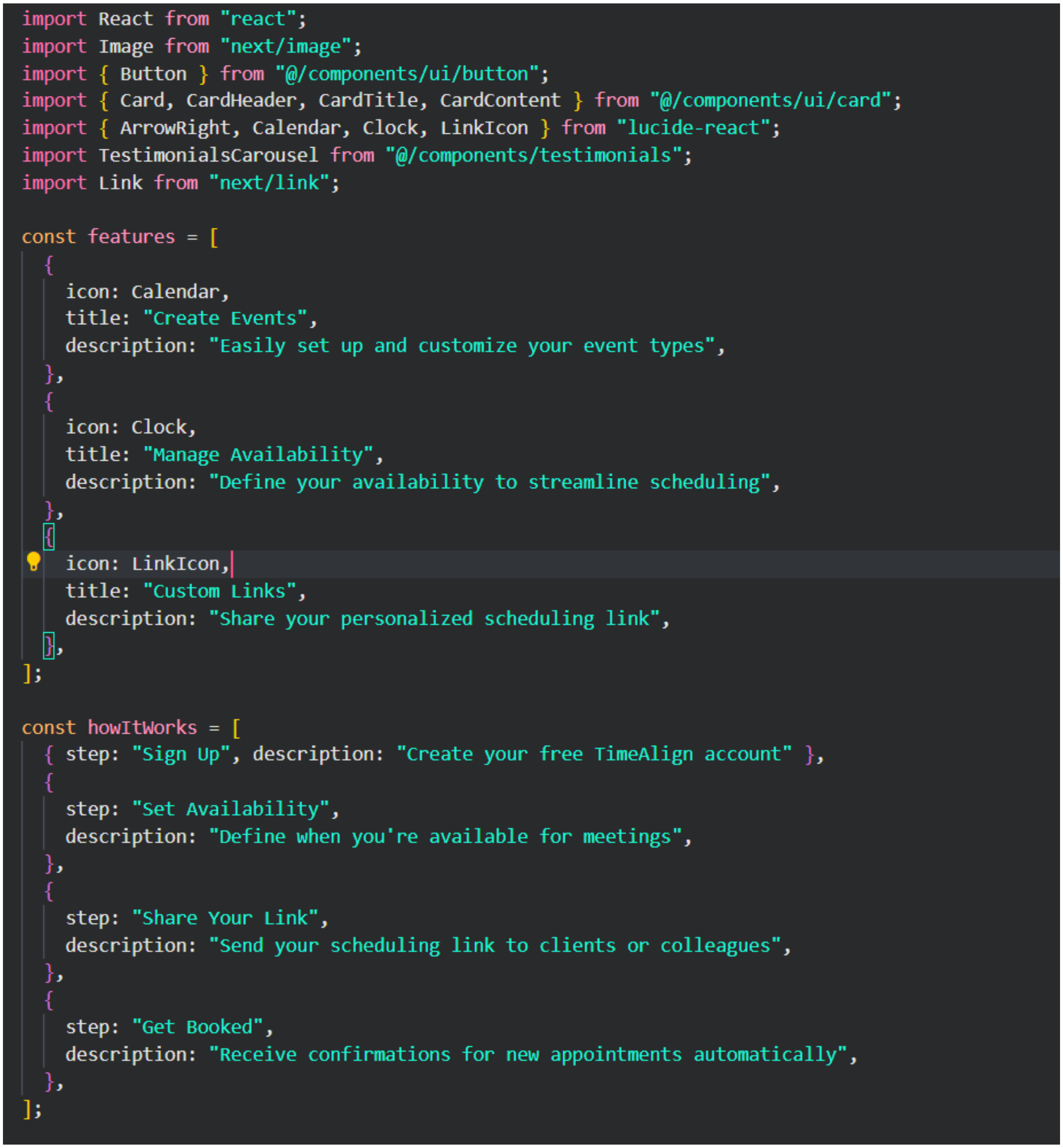
HOME PAGE



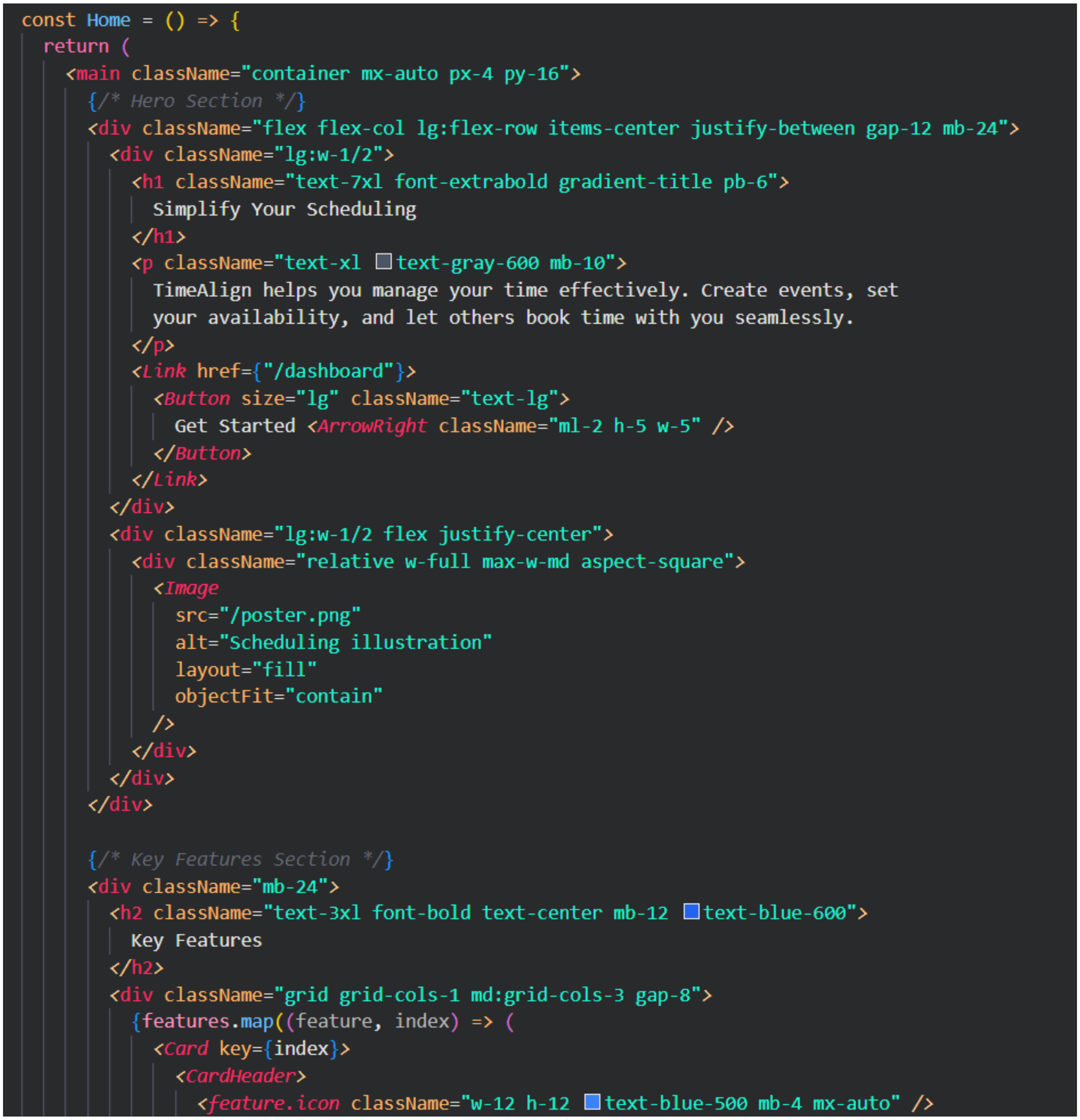
```
import { Inter } from "next/font/google";
import "./globals.css";
import { ClerkProvider } from "@clerk/nextjs";
import Header from "@/components/header";
import CreateEventDrawer from "@/components/create-event";

const inter = Inter({ subsets: ["latin"] });

export const metadata = {
  title: "TimeAlign",
  description: " ",
};

export default function RootLayout({ children }) {
  return (
    <ClerkProvider>
      <html lang="en">
        <body className={inter.className}>
          <Header />
          <main className="min-h-screen bg-gradient-to-b from-blue-50 to-white">
            {children}
          </main>
          <footer className="bg-blue-100 py-12">
            <div className="container mx-auto px-4 text-center text-gray-600">
              <p>Made by TimeAlign Team</p>
            </div>
          </footer>
          <CreateEventDrawer />
        </body>
      </html>
    </ClerkProvider>
  );
}
```



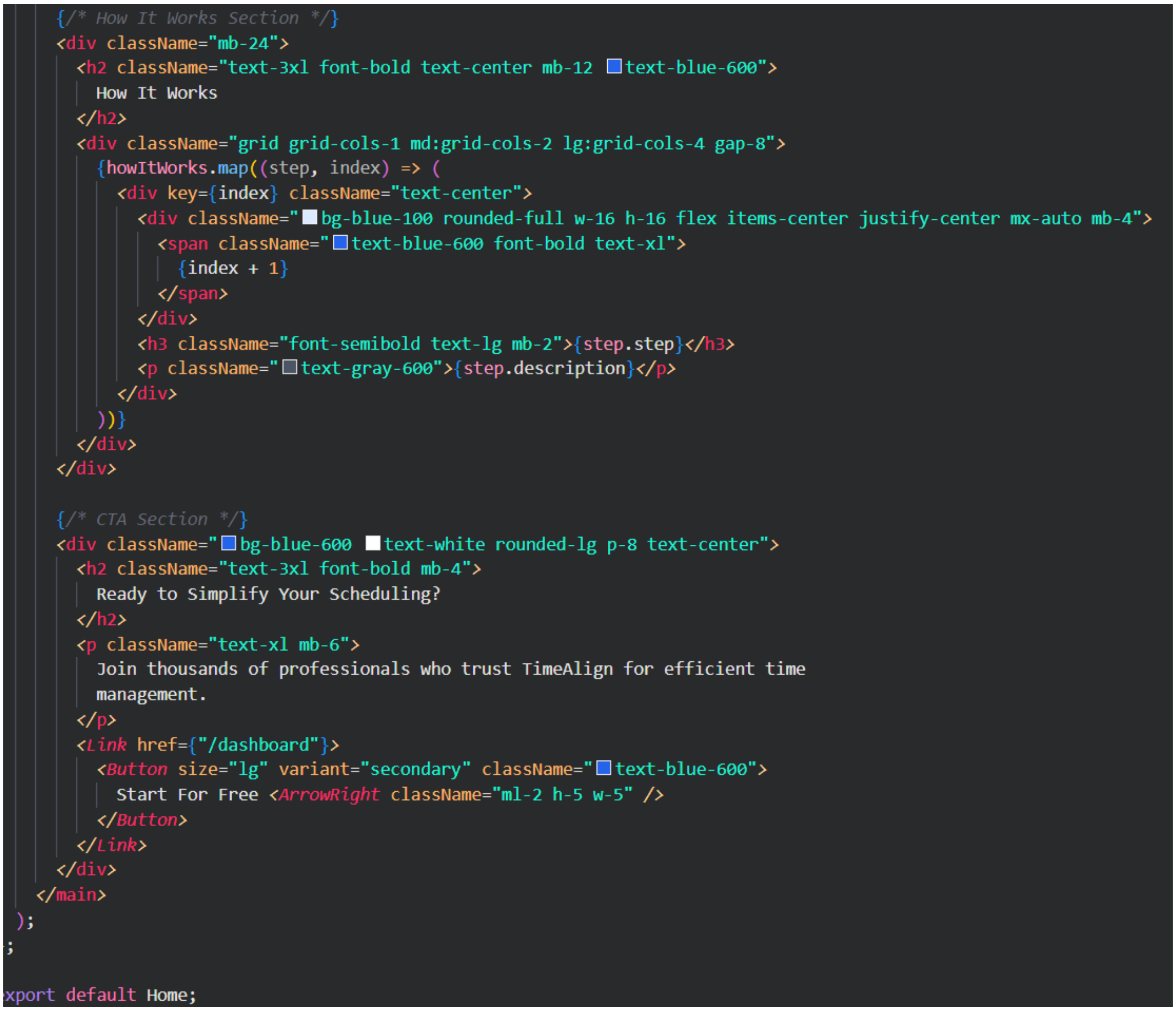
```
import React from "react";
import { SignedIn, SignedOut, SignInButton } from "@clerk/nextjs";
import Image from "next/image";
import Link from "next/link";
import { checkUser } from "@/lib/checkUser";
import UserMenu from "./user-menu";
import { Button } from "./ui/button";
import { PenBox } from "lucide-react";

async function Header() {
  await checkUser();

  return (
    <nav className="mx-auto py-2 px-4 flex justify-between items-center shadow-md border-b-2">
      <Link href="/" className="flex items-center">
        <h1 className="font-bold text-4xl text-blue-600">TimeAlign</h1>
        {/* <Image
          src="/logo.png"
          width="150"
          height="60"
          alt="Schedulrr Logo"
          className="h-16 w-auto"
        /> */}
      </Link>

      <div className="flex items-center gap-4">
        <Link href="/events?create=true">
          <Button variant="default" className="flex items-center gap-2">
            <PenBox size={18} />
            <span className="hidden sm:inline">Create Event</span>
          </Button>
        </Link>
        <SignedOut>
          <SignInButton forceRedirectUrl="/dashboard">
            <Button variant="outline">Login</Button>
          </SignInButton>
        </SignedOut>
        <SignedIn>
          <UserMenu />
        </SignedIn>
      </div>
    </nav>
```



```
import React from "react";
import Image from "next/image";
import { Button } from "@/components/ui/button";
import { Card, CardHeader, CardTitle, CardContent } from "@/components/ui/card";
import { ArrowRight, Calendar, Clock, LinkIcon } from "lucide-react";
import TestimonialsCarousel from "@/components/testimonials";
import Link from "next/link";

const features = [
  {
    icon: Calendar,
    title: "Create Events",
    description: "Easily set up and customize your event types",
  },
  {
    icon: Clock,
    title: "Manage Availability",
    description: "Define your availability to streamline scheduling",
  },
  {
    icon: LinkIcon,
    title: "Custom Links",
    description: "Share your personalized scheduling link",
  },
];

const howItWorks = [
  { step: "Sign Up", description: "Create your free TimeAlign account" },
  {
    step: "Set Availability",
    description: "Define when you're available for meetings",
  },
  {
    step: "Share Your Link",
    description: "Send your scheduling link to clients or colleagues",
  },
  {
    step: "Get Booked",
    description: "Receive confirmations for new appointments automatically",
  },
];
```

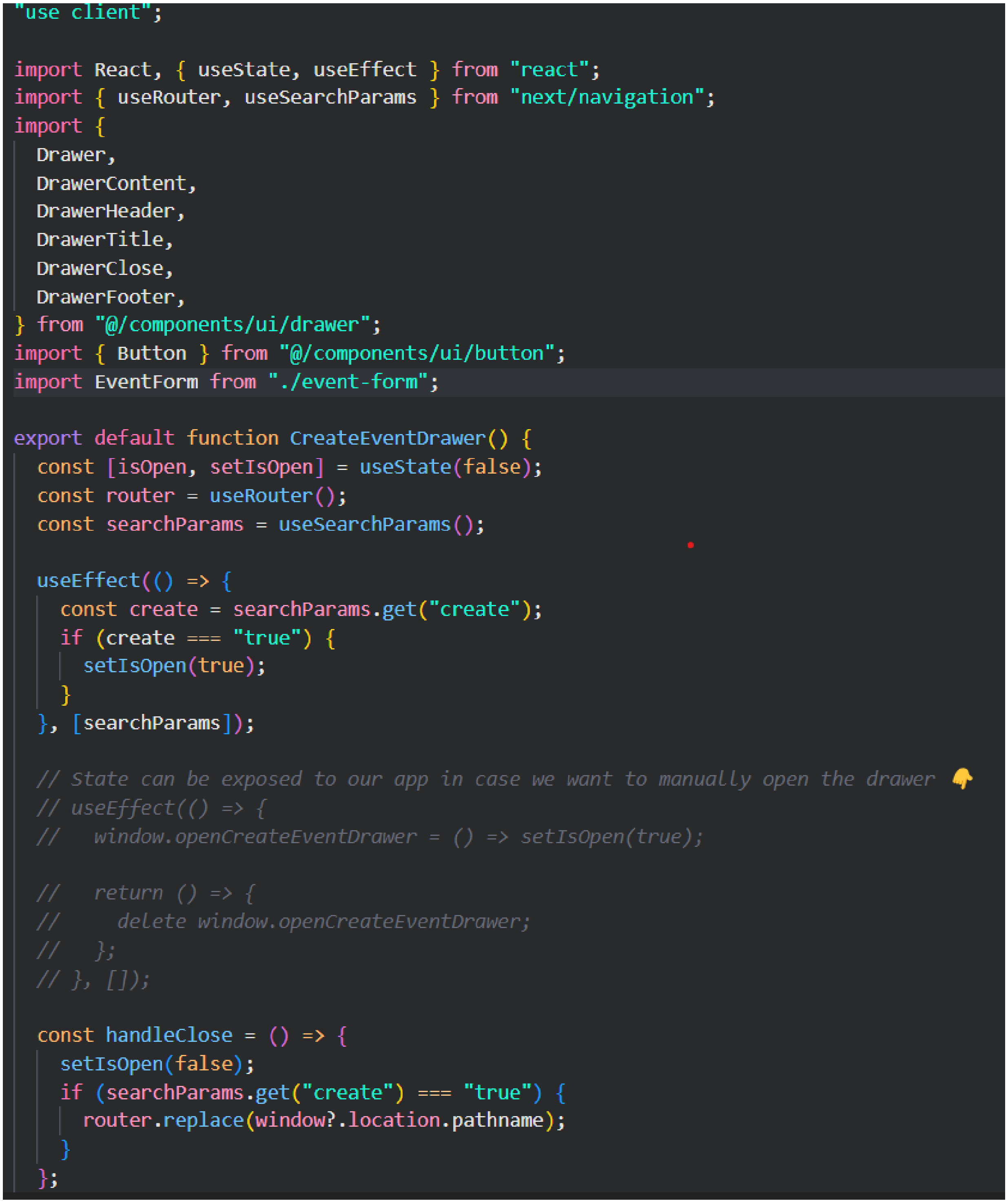


```jsx
const Home = () => {
  return (
    <main className="container mx-auto px-4 py-16">
      {/* Hero Section */}
      <div className="flex flex-col lg:flex-row items-center justify-between gap-12 mb-24">
        <div className="lg:w-1/2">
          <h1 className="text-7xl font-extrabold gradient-title pb-6">
            Simplify Your Scheduling
          </h1>
          <p className="text-xl text-gray-600 mb-10">
            TimeAlign helps you manage your time effectively. Create events, set
            your availability, and let others book time with you seamlessly.
          </p>
          <Link href={"/dashboard"}>
            <Button size="lg" className="text-lg">
              Get Started <ArrowRight className="ml-2 h-5 w-5" />
            </Button>
          </Link>
        </div>
        <div className="lg:w-1/2 flex justify-center">
          <div className="relative w-full max-w-md aspect-square">
            <Image
              src="/poster.png"
              alt="Scheduling illustration"
              layout="fill"
              objectFit="contain"
            />
          </div>
        </div>
      </div>

      {/* Key Features Section */}
      <div className="mb-24">
        <h2 className="text-3xl font-bold text-center mb-12 text-blue-600">
          Key Features
        </h2>
        <div className="grid grid-cols-1 md:grid-cols-3 gap-8">
          {features.map((feature, index) => (
            <Card key={index}>
              <CardHeader>
                <feature.icon className="w-12 h-12 text-blue-500 mb-4 mx-auto" />
```

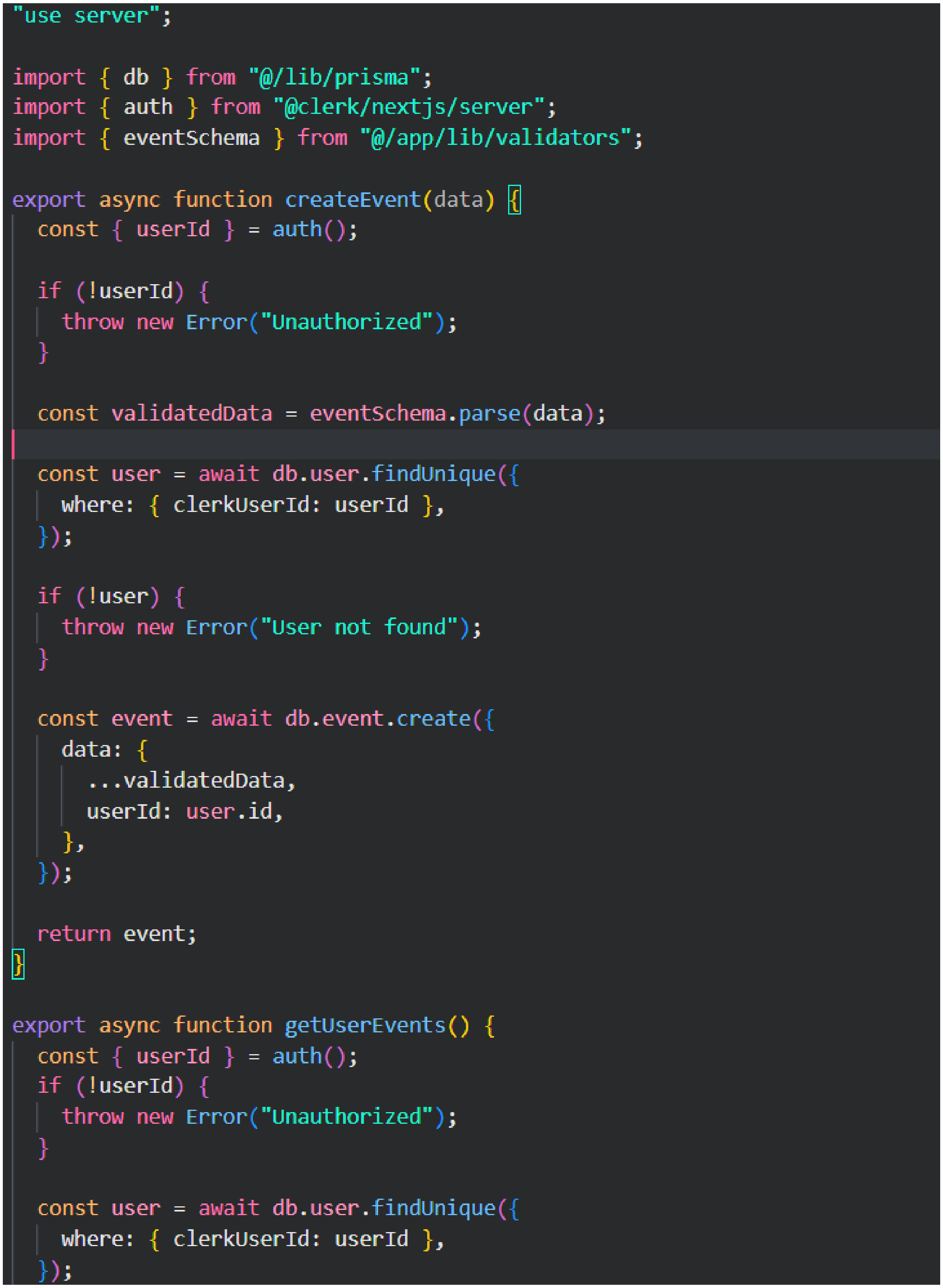


```
     {/* How It Works Section */}
     <div className="mb-24">
       <h2 className="text-3xl font-bold text-center mb-12 text-blue-600">
         How It Works
       </h2>
       <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-8">
         {howItWorks.map((step, index) => (
           <div key={index} className="text-center">
             <div className="bg-blue-100 rounded-full w-16 h-16 flex items-center justify-center mx-auto mb-4">
               <span className="text-blue-600 font-bold text-xl">
                 {index + 1}
               </span>
             </div>
             <h3 className="font-semibold text-lg mb-2">{step.step}</h3>
             <p className="text-gray-600">{step.description}</p>
           </div>
         ))}
       </div>
     </div>

     {/* CTA Section */}
     <div className="bg-blue-600 text-white rounded-lg p-8 text-center">
       <h2 className="text-3xl font-bold mb-4">
         Ready to Simplify Your Scheduling?
       </h2>
       <p className="text-xl mb-6">
         Join thousands of professionals who trust TimeAlign for efficient time
         management.
       </p>
       <Link href={"/dashboard"}>
         <Button size="lg" variant="secondary" className="text-blue-600">
           Start For Free <ArrowRight className="ml-2 h-5 w-5" />
         </Button>
       </Link>
     </div>
   </main>
 );
;

xport default Home;
```

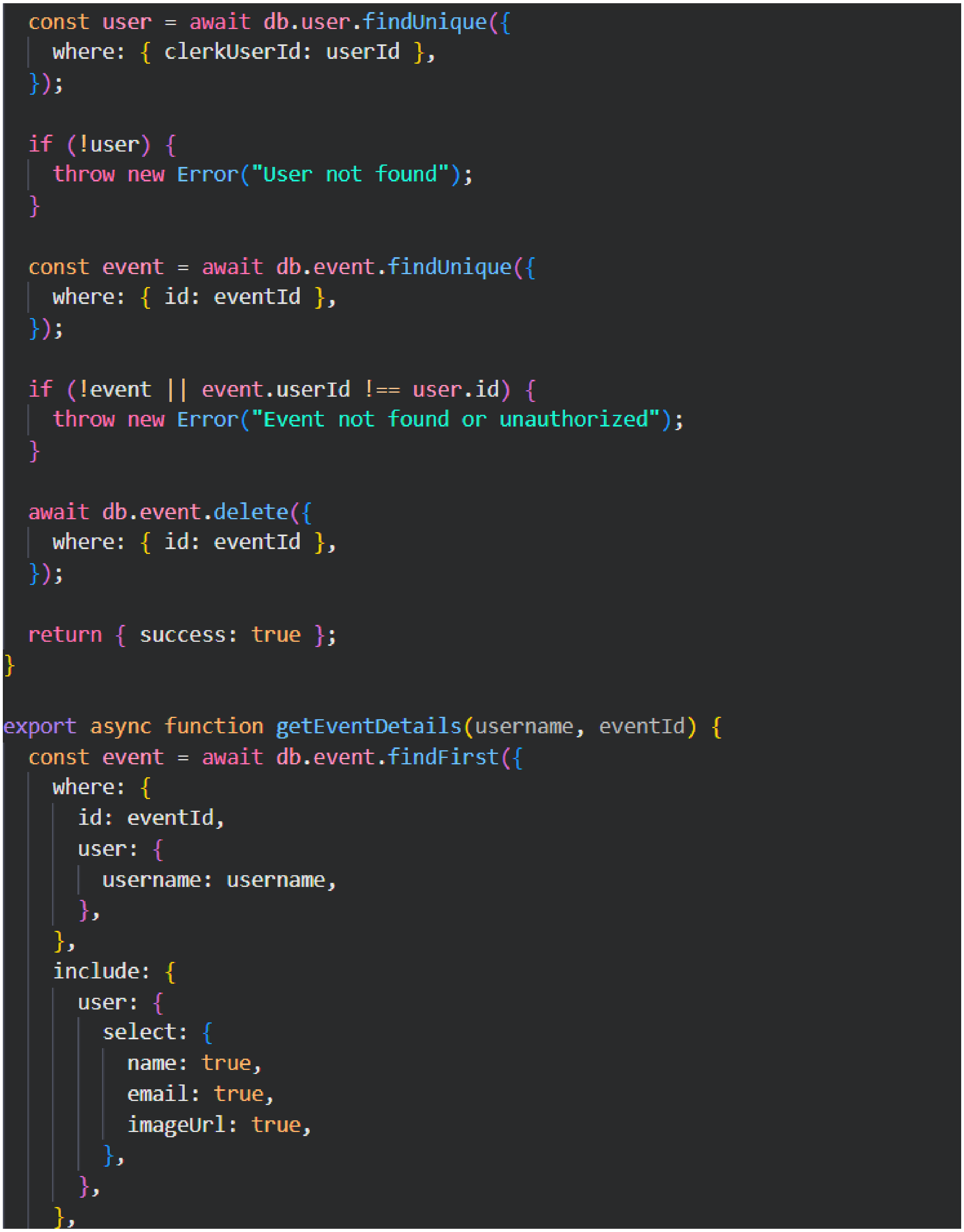
EVENT PAGE:



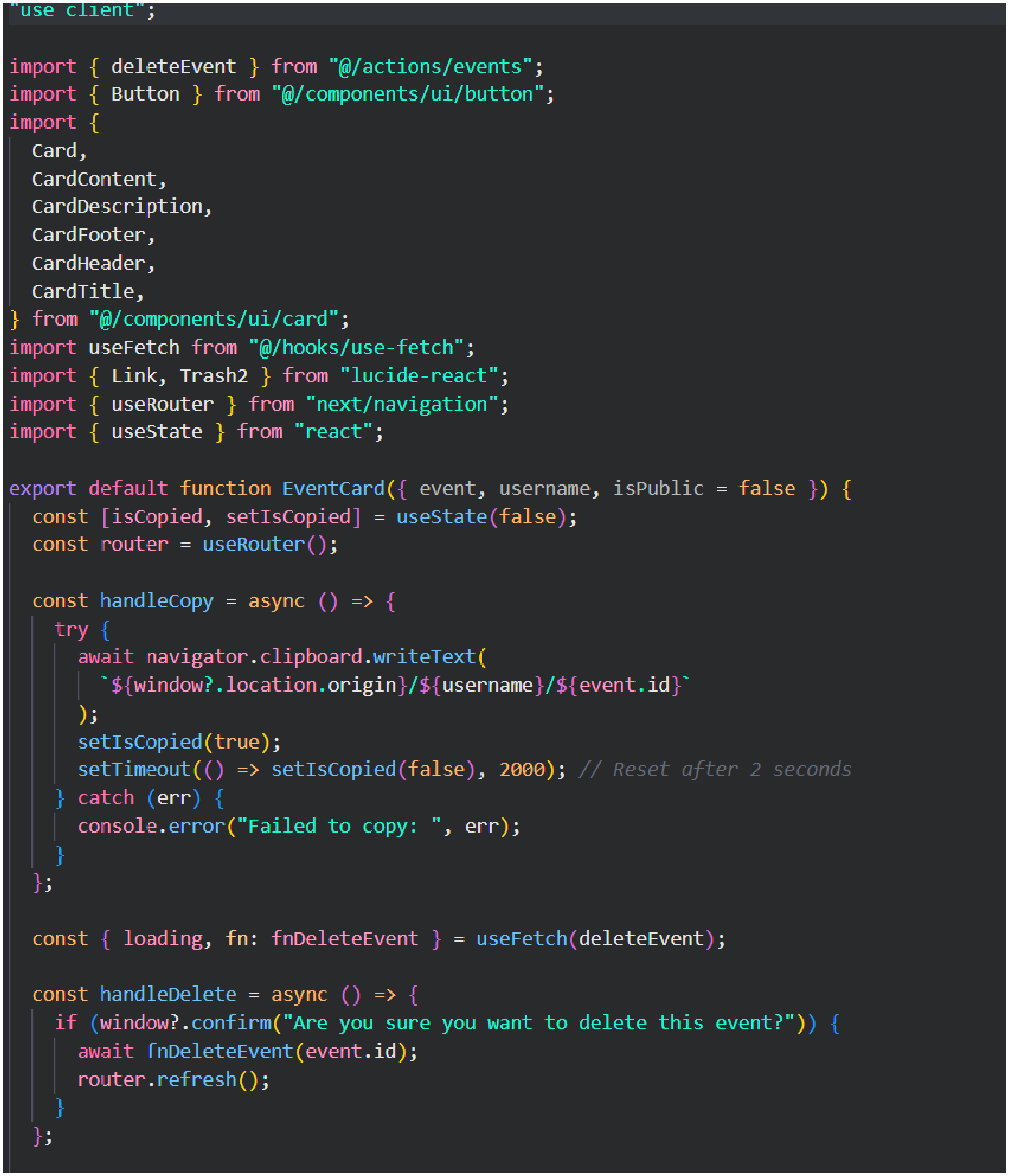
```
"use client";

import React, { useState, useEffect } from "react";
import { useRouter, useSearchParams } from "next/navigation";
import {
  Drawer,
  DrawerContent,
  DrawerHeader,
  DrawerTitle,
  DrawerClose,
  DrawerFooter,
} from "@/components/ui/drawer";
import { Button } from "@/components/ui/button";
import EventForm from "./event-form";

export default function CreateEventDrawer() {
  const [isOpen, setIsOpen] = useState(false);
  const router = useRouter();
  const searchParams = useSearchParams();

  useEffect(() => {
    const create = searchParams.get("create");
    if (create === "true") {
      setIsOpen(true);
    }
  }, [searchParams]);

  // State can be exposed to our app in case we want to manually open the drawer 👇
  // useEffect(() => {
  //   window.openCreateEventDrawer = () => setIsOpen(true);

  //   return () => {
  //     delete window.openCreateEventDrawer;
  //   };
  // }, []);

  const handleClose = () => {
    setIsOpen(false);
    if (searchParams.get("create") === "true") {
      router.replace(window?.location.pathname);
    }
  };
```



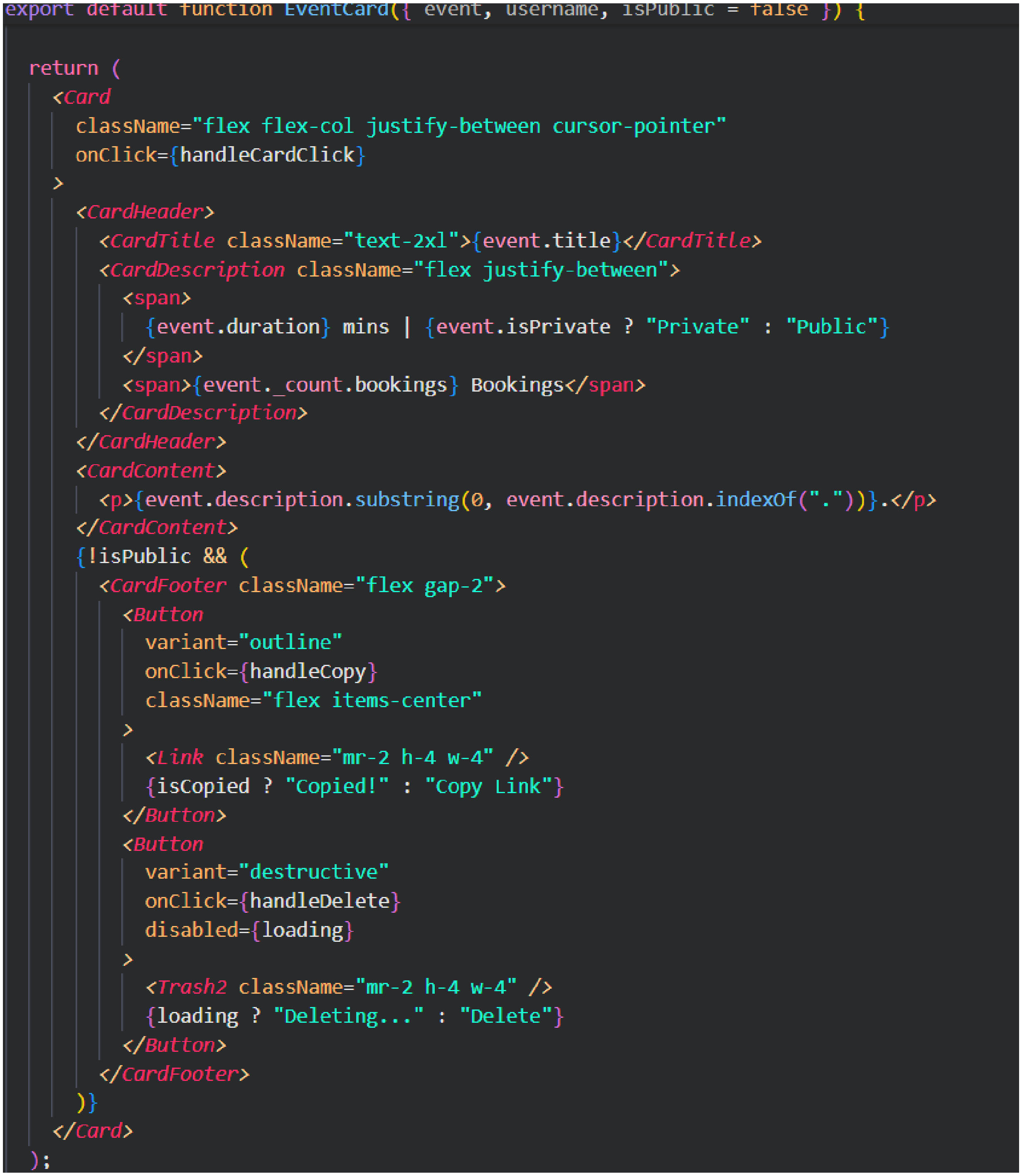
```
"use server";

import { db } from "@/lib/prisma";
import { auth } from "@clerk/nextjs/server";
import { eventSchema } from "@/app/lib/validators";

export async function createEvent(data) {
  const { userId } = auth();

  if (!userId) {
    throw new Error("Unauthorized");
  }

  const validatedData = eventSchema.parse(data);

  const user = await db.user.findUnique({
    where: { clerkUserId: userId },
  });

  if (!user) {
    throw new Error("User not found");
  }

  const event = await db.event.create({
    data: {
      ...validatedData,
      userId: user.id,
    },
  });

  return event;
}

export async function getUserEvents() {
  const { userId } = auth();
  if (!userId) {
    throw new Error("Unauthorized");
  }

  const user = await db.user.findUnique({
    where: { clerkUserId: userId },
  });
```



```
  const user = await db.user.findUnique({
    where: { clerkUserId: userId },
  });

  if (!user) {
    throw new Error("User not found");
  }

  const event = await db.event.findUnique({
    where: { id: eventId },
  });

  if (!event || event.userId !== user.id) {
    throw new Error("Event not found or unauthorized");
  }

  await db.event.delete({
    where: { id: eventId },
  });

  return { success: true };
}

export async function getEventDetails(username, eventId) {
  const event = await db.event.findFirst({
    where: {
      id: eventId,
      user: {
        username: username,
      },
    },
    include: {
      user: {
        select: {
          name: true,
          email: true,
          imageUrl: true,
        },
      },
    },
```

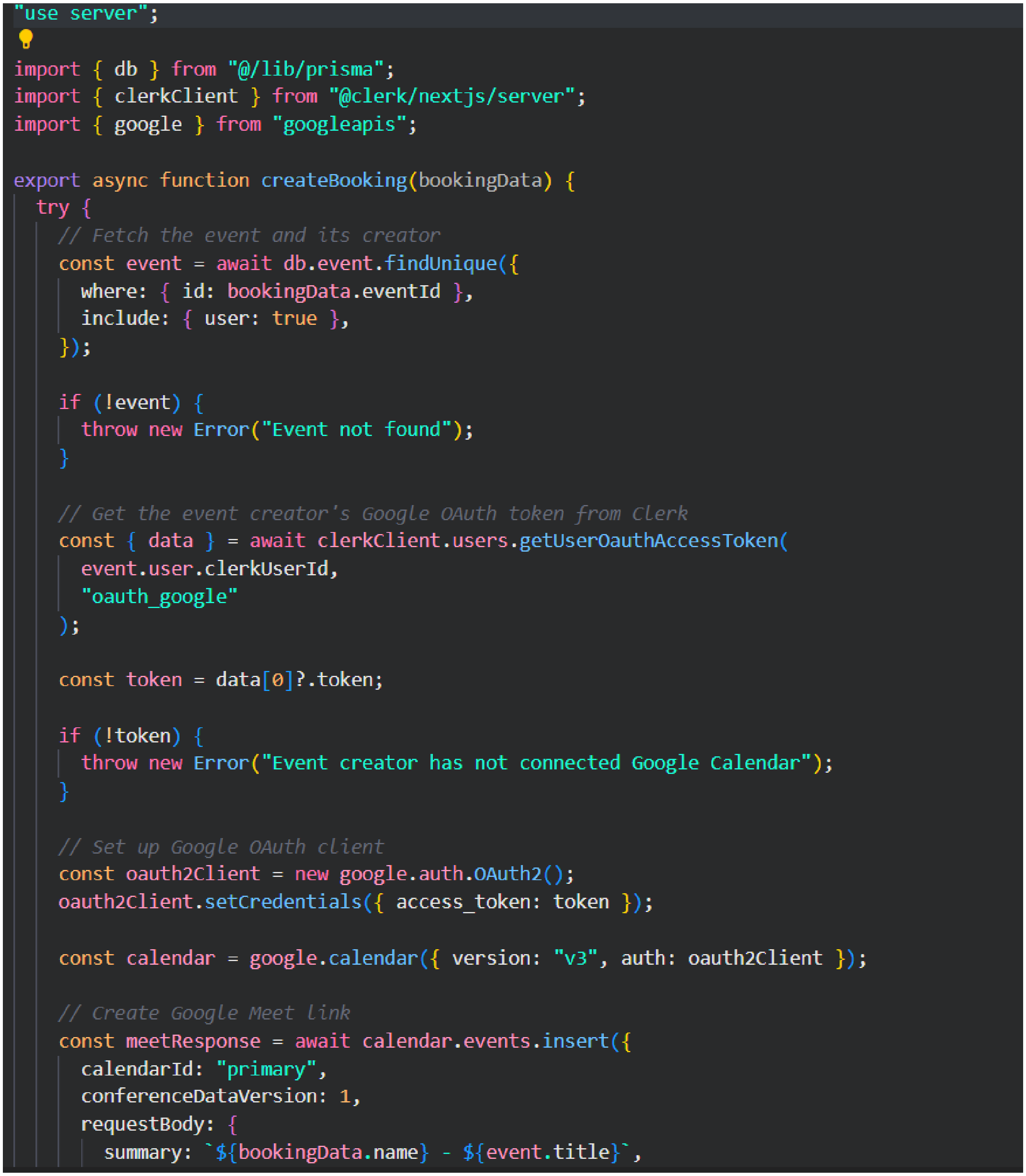


```
"use client";

import { deleteEvent } from "@/actions/events";
import { Button } from "@/components/ui/button";
import {
  Card,
  CardContent,
  CardDescription,
  CardFooter,
  CardHeader,
  CardTitle,
} from "@/components/ui/card";
import useFetch from "@/hooks/use-fetch";
import { Link, Trash2 } from "lucide-react";
import { useRouter } from "next/navigation";
import { useState } from "react";

export default function EventCard({ event, username, isPublic = false }) {
  const [isCopied, setIsCopied] = useState(false);
  const router = useRouter();

  const handleCopy = async () => {
    try {
      await navigator.clipboard.writeText(
        `${window?.location.origin}/${username}/${event.id}`
      );
      setIsCopied(true);
      setTimeout(() => setIsCopied(false), 2000); // Reset after 2 seconds
    } catch (err) {
      console.error("Failed to copy: ", err);
    }
  };

  const { loading, fn: fnDeleteEvent } = useFetch(deleteEvent);

  const handleDelete = async () => {
    if (window?.confirm("Are you sure you want to delete this event?")) {
      await fnDeleteEvent(event.id);
      router.refresh();
    }
  };
```

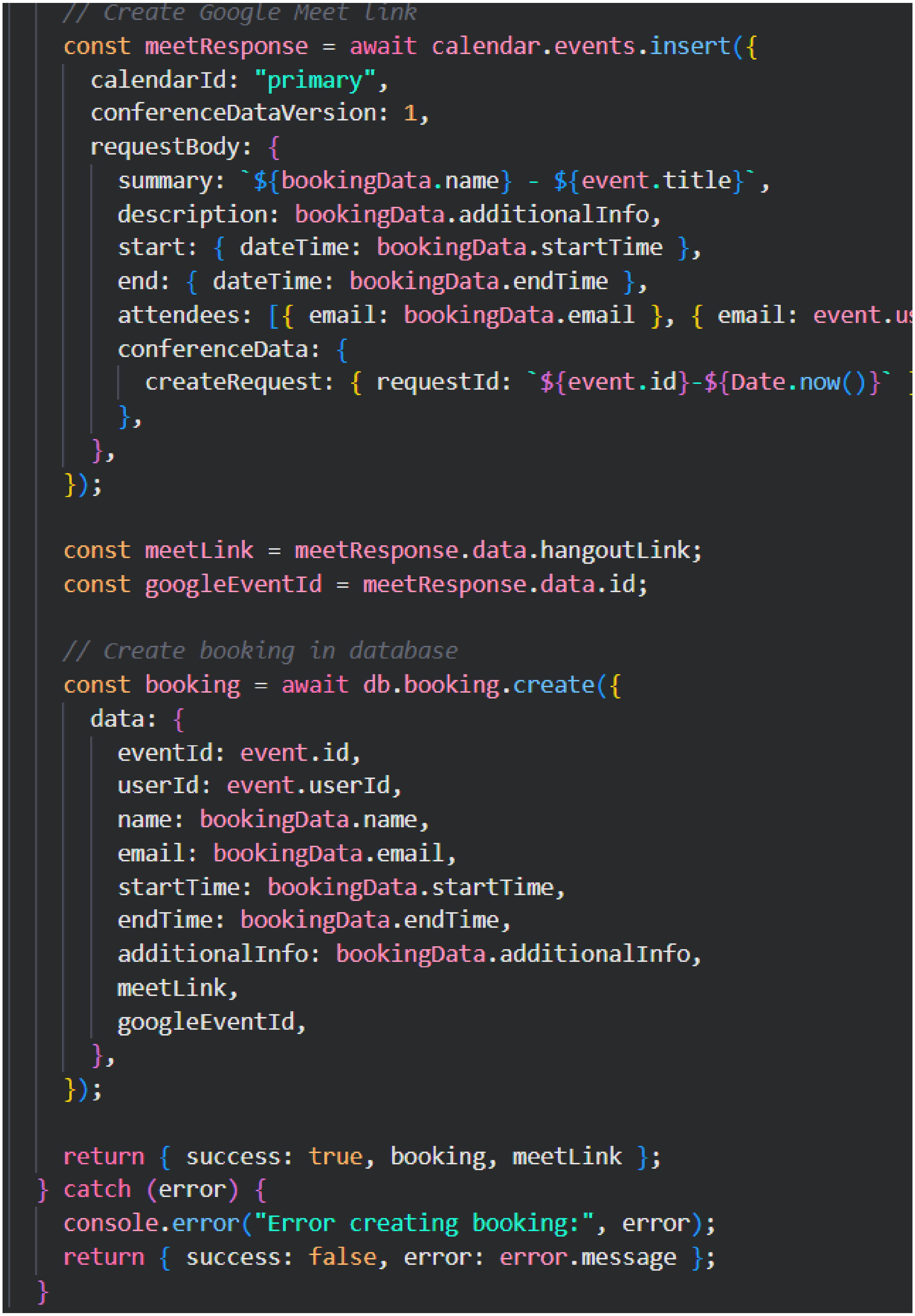


```
export default function EventCard({ event, username, isPublic = false }) {

  return (
    <Card
      className="flex flex-col justify-between cursor-pointer"
      onClick={handleCardClick}
    >
      <CardHeader>
        <CardTitle className="text-2xl">{event.title}</CardTitle>
        <CardDescription className="flex justify-between">
          <span>
            {event.duration} mins | {event.isPrivate ? "Private" : "Public"}
          </span>
          <span>{event._count.bookings} Bookings</span>
        </CardDescription>
      </CardHeader>
      <CardContent>
        <p>{event.description.substring(0, event.description.indexOf("."))}.</p>
      </CardContent>
      {!isPublic && (
        <CardFooter className="flex gap-2">
          <Button
            variant="outline"
            onClick={handleCopy}
            className="flex items-center"
          >
            <Link className="mr-2 h-4 w-4" />
            {isCopied ? "Copied!" : "Copy Link"}
          </Button>
          <Button
            variant="destructive"
            onClick={handleDelete}
            disabled={loading}
          >
            <Trash2 className="mr-2 h-4 w-4" />
            {loading ? "Deleting..." : "Delete"}
          </Button>
        </CardFooter>
      )}
    </Card>
  );
```

BOOKING CODE:

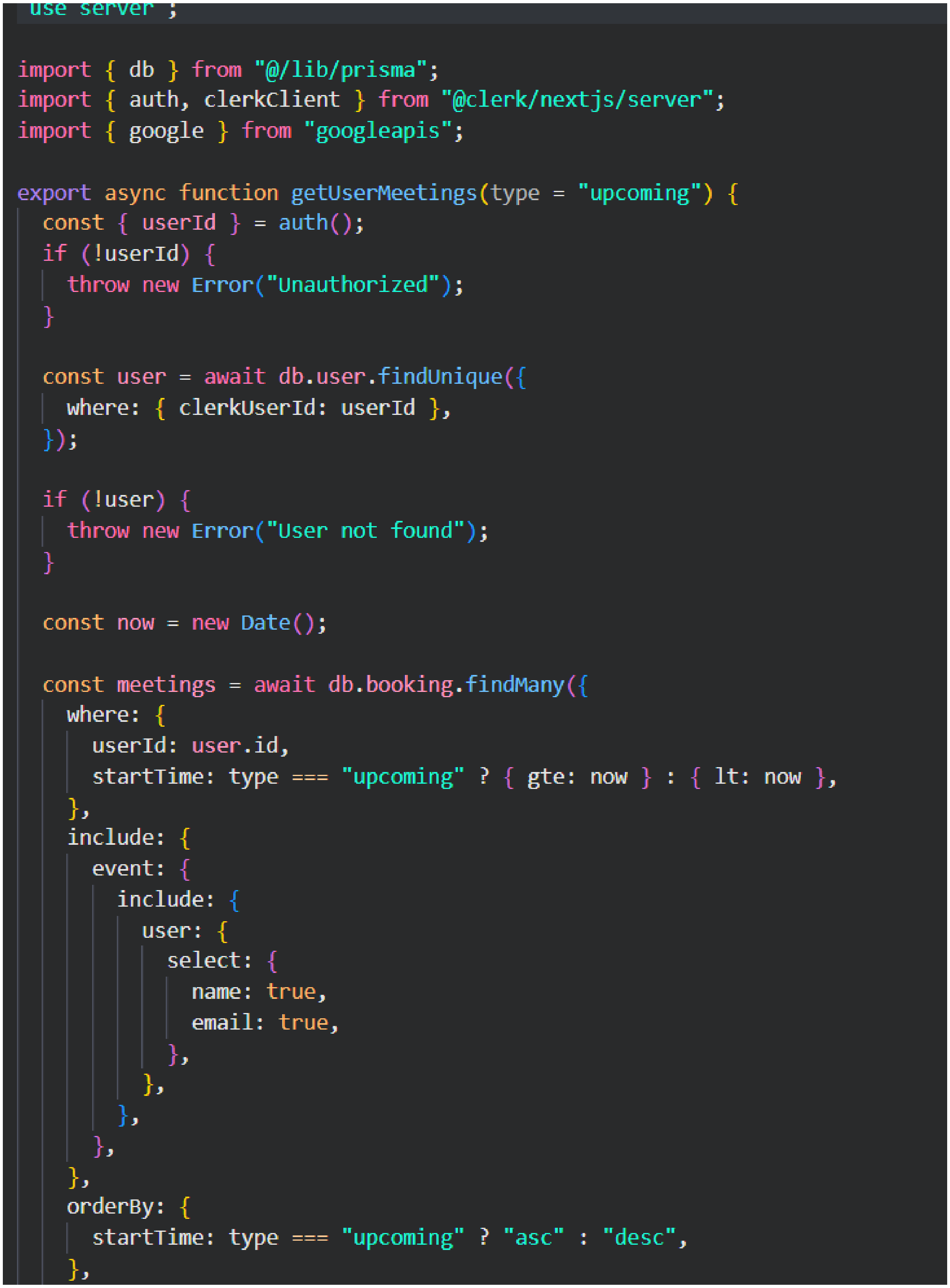


```
"use server";

import { db } from "@/lib/prisma";
import { clerkClient } from "@clerk/nextjs/server";
import { google } from "googleapis";

export async function createBooking(bookingData) {
  try {
    // Fetch the event and its creator
    const event = await db.event.findUnique({
      where: { id: bookingData.eventId },
      include: { user: true },
    });

    if (!event) {
      throw new Error("Event not found");
    }

    // Get the event creator's Google OAuth token from Clerk
    const { data } = await clerkClient.users.getUserOauthAccessToken(
      event.user.clerkUserId,
      "oauth_google"
    );

    const token = data[0]?.token;

    if (!token) {
      throw new Error("Event creator has not connected Google Calendar");
    }

    // Set up Google OAuth client
    const oauth2Client = new google.auth.OAuth2();
    oauth2Client.setCredentials({ access_token: token });

    const calendar = google.calendar({ version: "v3", auth: oauth2Client });

    // Create Google Meet link
    const meetResponse = await calendar.events.insert({
      calendarId: "primary",
      conferenceDataVersion: 1,
      requestBody: {
        summary: `${bookingData.name} - ${event.title}`,
```

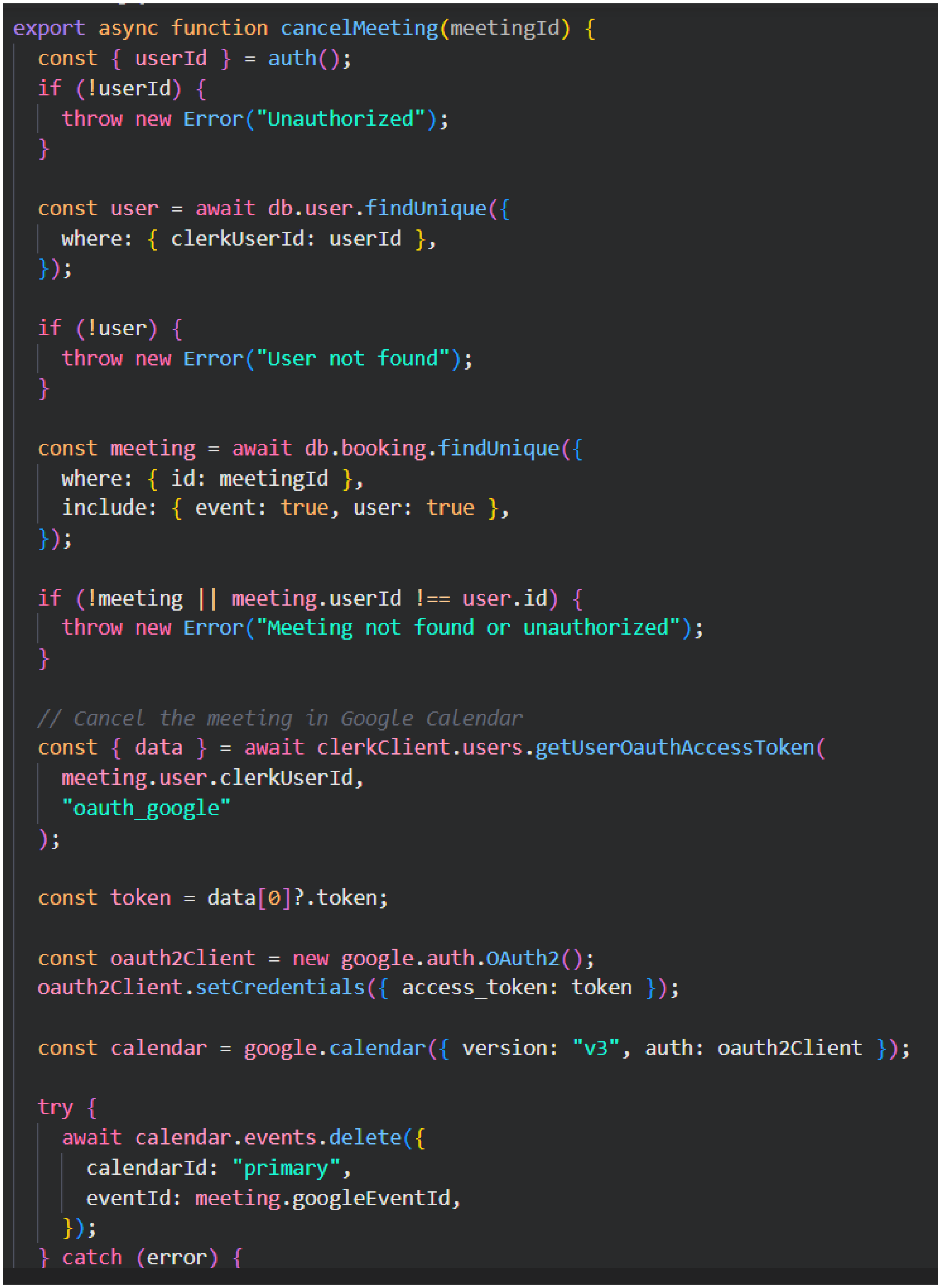


```
    // Create Google Meet link
    const meetResponse = await calendar.events.insert({
      calendarId: "primary",
      conferenceDataVersion: 1,
      requestBody: {
        summary: `${bookingData.name} - ${event.title}`,
        description: bookingData.additionalInfo,
        start: { dateTime: bookingData.startTime },
        end: { dateTime: bookingData.endTime },
        attendees: [{ email: bookingData.email }, { email: event.us
        conferenceData: {
          createRequest: { requestId: `${event.id}-${Date.now()}`
        },
      },
    });

    const meetLink = meetResponse.data.hangoutLink;
    const googleEventId = meetResponse.data.id;

    // Create booking in database
    const booking = await db.booking.create({
      data: {
        eventId: event.id,
        userId: event.userId,
        name: bookingData.name,
        email: bookingData.email,
        startTime: bookingData.startTime,
        endTime: bookingData.endTime,
        additionalInfo: bookingData.additionalInfo,
        meetLink,
        googleEventId,
      },
    });

    return { success: true, booking, meetLink };
  } catch (error) {
    console.error("Error creating booking:", error);
    return { success: false, error: error.message };
  }
```

MEETING CODE:

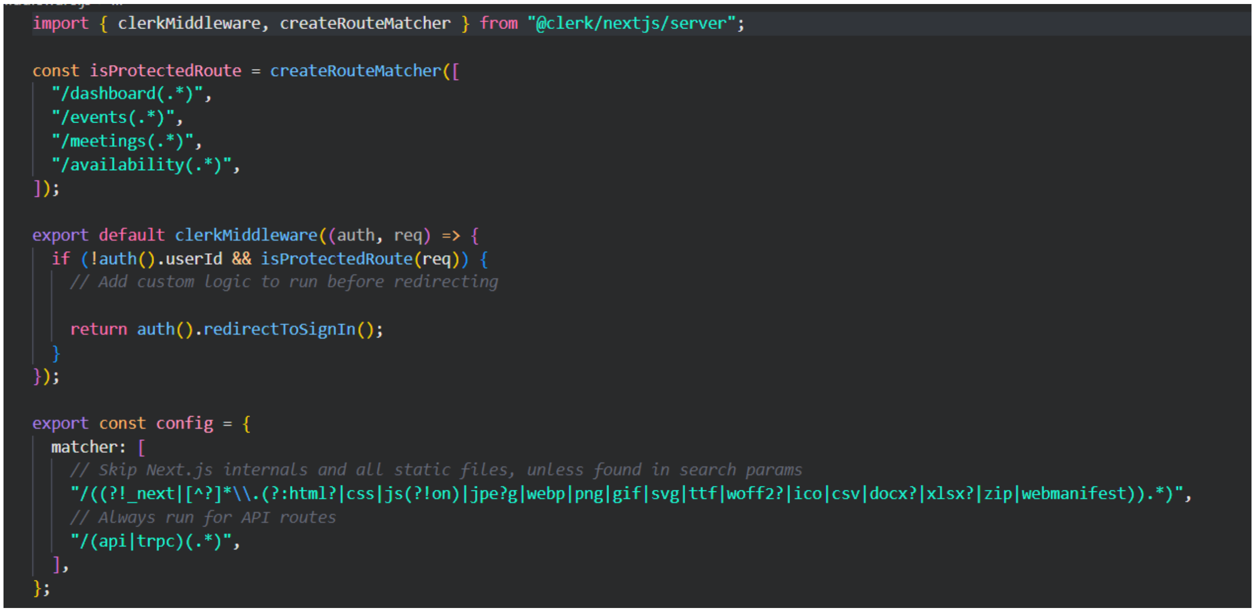


```
"use server";

import { db } from "@/lib/prisma";
import { auth, clerkClient } from "@clerk/nextjs/server";
import { google } from "googleapis";

export async function getUserMeetings(type = "upcoming") {
  const { userId } = auth();
  if (!userId) {
    throw new Error("Unauthorized");
  }

  const user = await db.user.findUnique({
    where: { clerkUserId: userId },
  });

  if (!user) {
    throw new Error("User not found");
  }

  const now = new Date();

  const meetings = await db.booking.findMany({
    where: {
      userId: user.id,
      startTime: type === "upcoming" ? { gte: now } : { lt: now },
    },
    include: {
      event: {
        include: {
          user: {
            select: {
              name: true,
              email: true,
            },
          },
        },
      },
    },
    orderBy: {
      startTime: type === "upcoming" ? "asc" : "desc",
    },
```



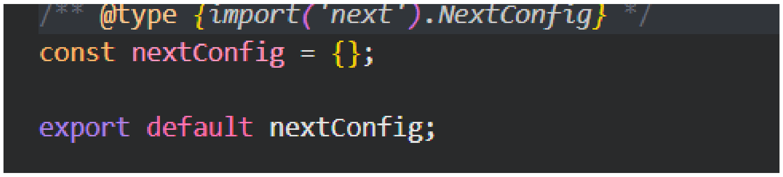
```
export async function cancelMeeting(meetingId) {
  const { userId } = auth();
  if (!userId) {
    throw new Error("Unauthorized");
  }

  const user = await db.user.findUnique({
    where: { clerkUserId: userId },
  });

  if (!user) {
    throw new Error("User not found");
  }

  const meeting = await db.booking.findUnique({
    where: { id: meetingId },
    include: { event: true, user: true },
  });

  if (!meeting || meeting.userId !== user.id) {
    throw new Error("Meeting not found or unauthorized");
  }

  // Cancel the meeting in Google Calendar
  const { data } = await clerkClient.users.getUserOauthAccessToken(
    meeting.user.clerkUserId,
    "oauth_google"
  );

  const token = data[0]?.token;

  const oauth2Client = new google.auth.OAuth2();
  oauth2Client.setCredentials({ access_token: token });

  const calendar = google.calendar({ version: "v3", auth: oauth2Client });

  try {
    await calendar.events.delete({
      calendarId: "primary",
      eventId: meeting.googleEventId,
    });
  } catch (error) {
```

MIDDLEWARE:



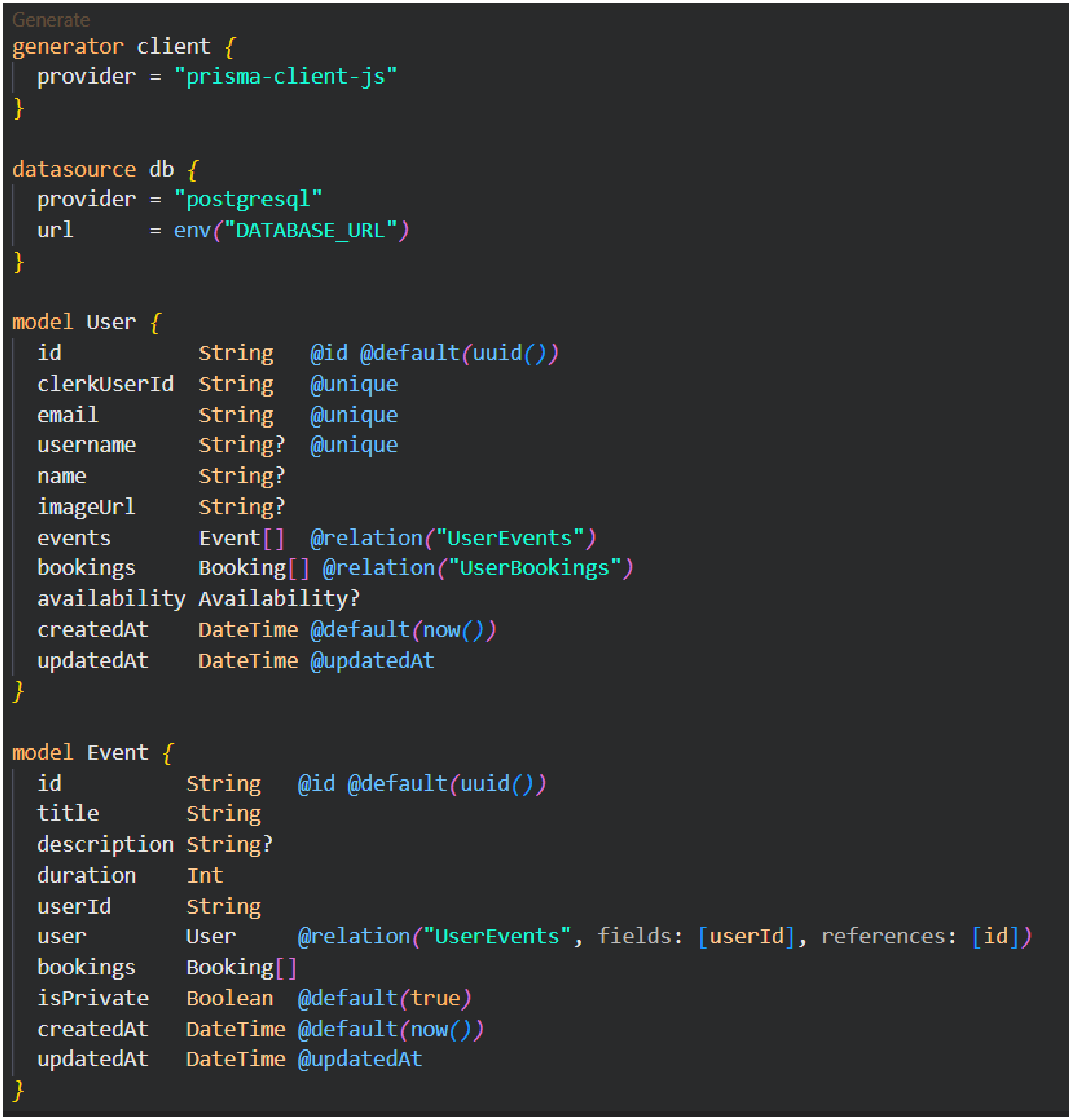
```
import { clerkMiddleware, createRouteMatcher } from "@clerk/nextjs/server";

const isProtectedRoute = createRouteMatcher([
  "/dashboard(.*)",
  "/events(.*)",
  "/meetings(.*)",
  "/availability(.*)",
]);

export default clerkMiddleware((auth, req) => {
  if (!auth().userId && isProtectedRoute(req)) {
    // Add custom logic to run before redirecting

    return auth().redirectToSignIn();
  }
});

export const config = {
  matcher: [
    // Skip Next.js internals and all static files, unless found in search params
    "/((?!_next|[^?]*\\.(?:html?|css|js(?!on)|jpe?g|webp|png|gif|svg|ttf|woff2?|ico|csv|docx?|xlsx?|zip|webmanifest)).*)",
    // Always run for API routes
    "/(api|trpc)(.*)",
  ],
};
```

NEXT CONFIG:

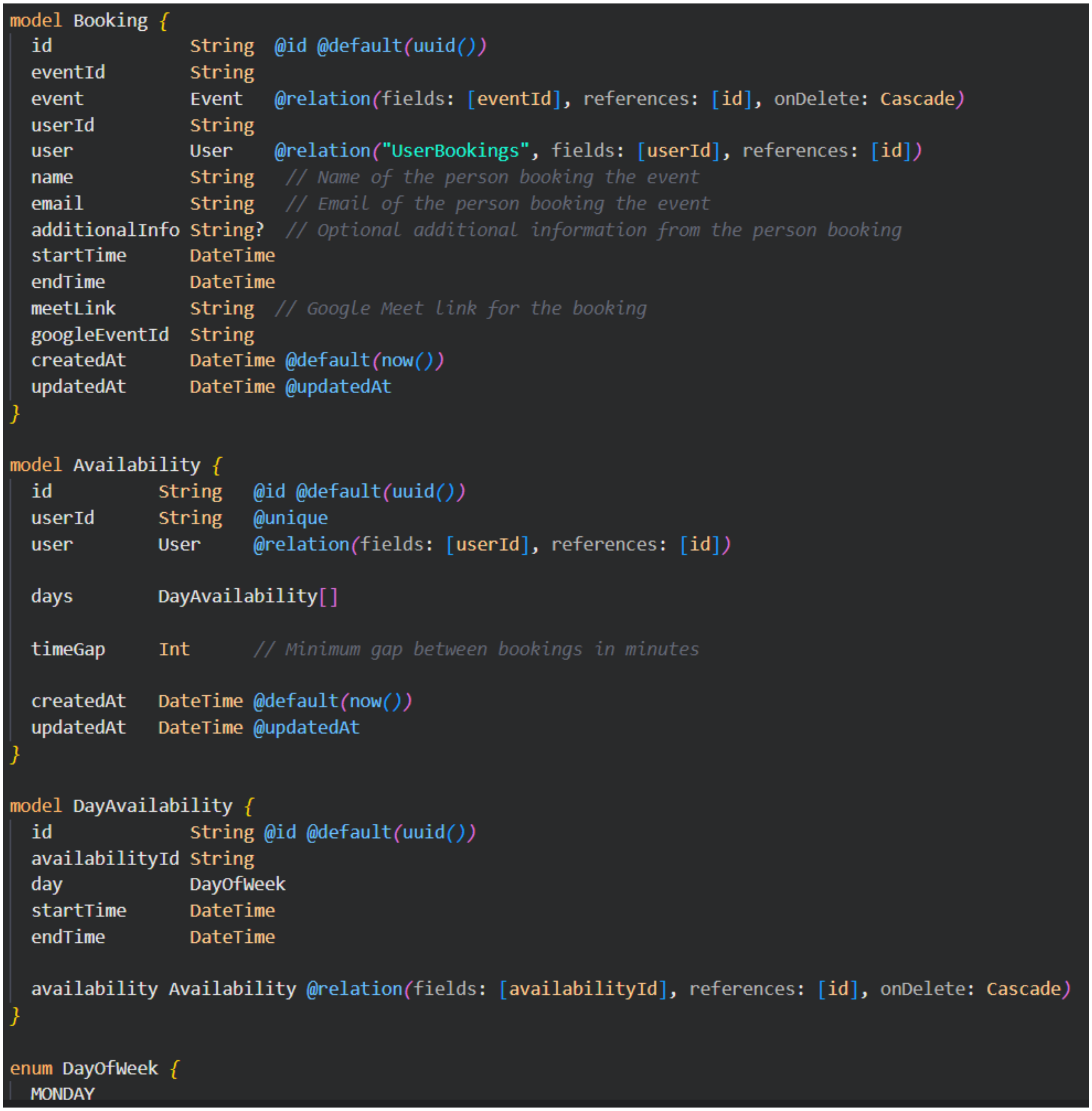


```
/** @type {import('next').NextConfig} */
const nextConfig = {};

export default nextConfig;
```

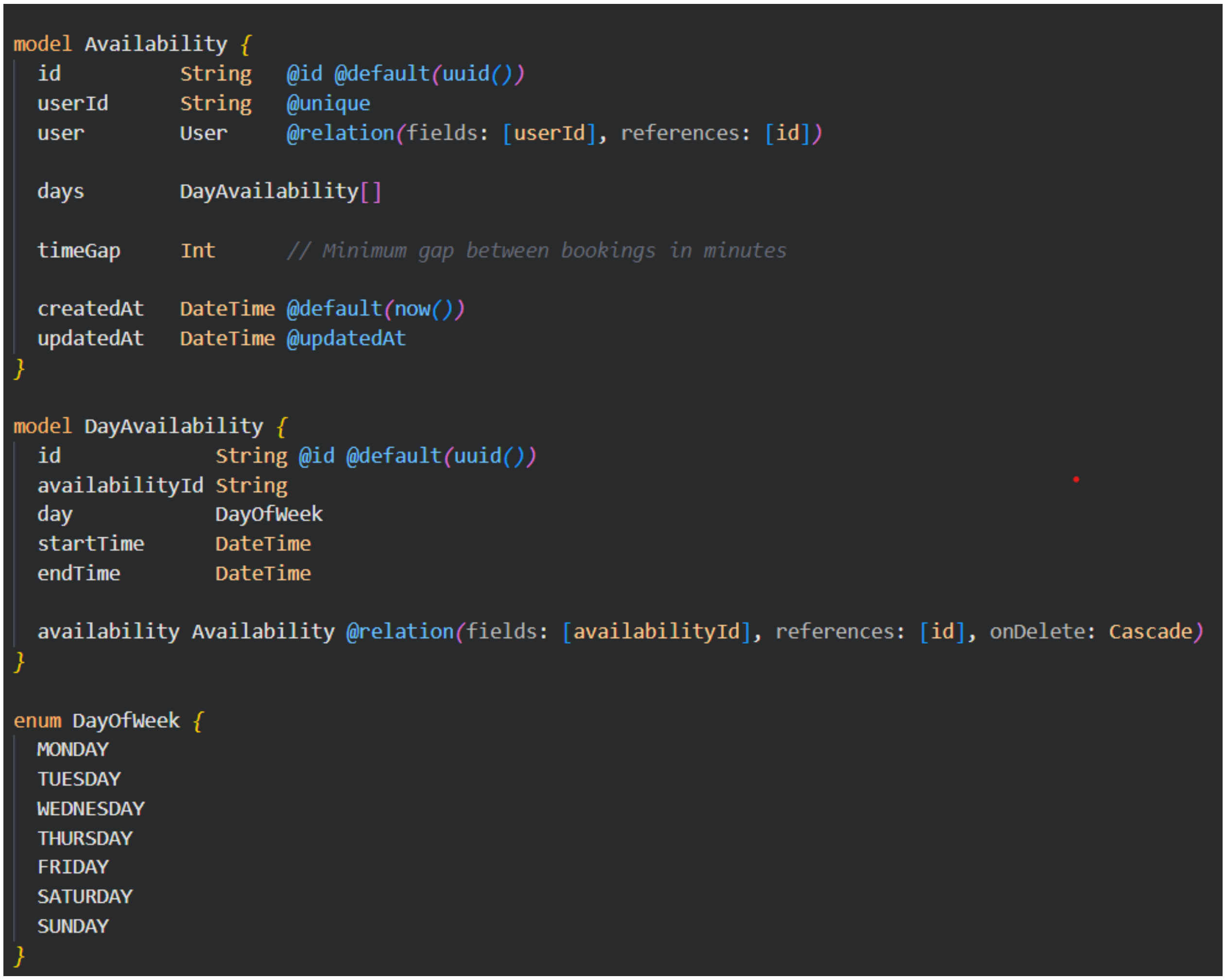
DATABASE CODE:



```
Generate
generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id           String   @id @default(uuid())
  clerkUserId  String   @unique
  email        String   @unique
  username     String?  @unique
  name         String?
  imageUrl     String?
  events       Event[]  @relation("UserEvents")
  bookings     Booking[] @relation("UserBookings")
  availability Availability?
  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt
}

model Event {
  id          String   @id @default(uuid())
  title       String
  description String?
  duration    Int
  userId      String
  user        User     @relation("UserEvents", fields: [userId], references: [id])
  bookings    Booking[]
  isPrivate   Boolean  @default(true)
  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}
```



```
model Booking {
  id             String  @id @default(uuid())
  eventId        String
  event          Event   @relation(fields: [eventId], references: [id], onDelete: Cascade)
  userId         String
  user           User    @relation("UserBookings", fields: [userId], references: [id])
  name           String   // Name of the person booking the event
  email          String   // Email of the person booking the event
  additionalInfo String?  // Optional additional information from the person booking
  startTime      DateTime
  endTime        DateTime
  meetLink       String  // Google Meet link for the booking
  googleEventId  String
  createdAt      DateTime @default(now())
  updatedAt      DateTime @updatedAt
}

model Availability {
  id         String   @id @default(uuid())
  userId     String   @unique
  user       User     @relation(fields: [userId], references: [id])

  days       DayAvailability[]

  timeGap    Int      // Minimum gap between bookings in minutes

  createdAt  DateTime @default(now())
  updatedAt  DateTime @updatedAt
}

model DayAvailability {
  id             String @id @default(uuid())
  availabilityId String
  day            DayOfWeek
  startTime      DateTime
  endTime        DateTime

  availability Availability @relation(fields: [availabilityId], references: [id], onDelete: Cascade)
}

enum DayOfWeek {
  MONDAY
```



```
model Availability {
  id          String   @id @default(uuid())
  userId      String   @unique
  user        User     @relation(fields: [userId], references: [id])

  days        DayAvailability[]

  timeGap     Int      // Minimum gap between bookings in minutes

  createdAt   DateTime @default(now())
  updatedAt   DateTime @updatedAt
}

model DayAvailability {
  id             String @id @default(uuid())
  availabilityId String
  day            DayOfWeek
  startTime      DateTime
  endTime        DateTime

  availability Availability @relation(fields: [availabilityId], references: [id], onDelete: Cascade)
}

enum DayOfWeek {
  MONDAY
  TUESDAY
  WEDNESDAY
  THURSDAY
  FRIDAY
  SATURDAY
  SUNDAY
}
```

5.3 TESTING

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that the software product is Defect free). It is a process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include

- Preventing bugs
- Reducing development costs
- Improve customer satisfaction
- Improve product quality
- Improve security
- Improving performance etc.

Testing is an important phase in SDLC that need to carry out during the implementation phase. The testing phase can be divided into two sub-segments. Testing phase and evaluation phase. Testing is carried out during the implementation phase and evaluation can be conducted after implementing the system or a prototype of the system.

1. CREATE EVENT

No	Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid email and event name is entered	Email: abc@gmail.com Event name : Test 1	Event create successful and send mail	Event created and mail sent	Pass
2	Check response when valid email and event name is entered	Email: abc@gmail.com Event name : null	Event create successful and send mail	Event created fail & event name required	Fail
3	Check response when valid email and event name is entered	Email: ABC Event name : Test 1	Event create successful and send mail	Email not valid	Fail

Table 5.3.1: EVENT CREATION

2. SCHEDULE EVENT

No	Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when valid Schedule Date	Select Date From Past month	Allow to Select Time Slots	Not Valid Date	Fail
2	Check response when valid Schedule Date	Select Date for Next Year	Allow to Select Time Slots	Valid	Pass
3	Check response when valid Schedule Date	Select Date for current month (upcoming)	Allow to Select Time Slots	Valid	Pass

Table 5.3.2: FOR SET DATE

No	Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Check response when send invite	Email : abc@gmail.com Event Desc : TEST event Event Loction : XYZ	Allow to Send Mail	Valid	Pass
2	Check response when send invite	Email : abc@gmail.com , xyz@gmail.com Event Desc : TEST event Event Loction : XYZ	Allow to Send Mail	Valid	Pass
3	Check response when send invite	Email : abc@xyz Event Desc : TEST event Event Loction : XYZ	Allow to Send Mail	Not Valid Email	Fail

Table 5.3.3: FOR SEND INVITES

Chapter 6: Conclusion

In conclusion, the web-based meeting scheduling application effectively addresses the challenges of organizing and managing meetings by providing a streamlined and user-friendly platform. Through features such as real-time availability checks, automated conflict detection, and seamless time zone adjustments, the application simplifies the scheduling process, saving users significant time and effort. Its intuitive interface ensures accessibility for users of varying technical expertise, while mobile responsiveness enhances its usability across devices. Integration with popular calendar systems and third-party tools allows the application to fit seamlessly into existing workflows, improving overall productivity and coordination.

The inclusion of automated notifications and reminders reduces the likelihood of missed meetings, while comprehensive analytics provide valuable insights into meeting trends and attendance patterns. Robust security measures and compliance with data privacy regulations, such as GDPR, ensure that sensitive user information is well-protected. Furthermore, the application's scalability and adaptability make it suitable for diverse user needs, from individuals to large organizations. By reducing administrative overhead, enhancing collaboration, and fostering efficient time management, this application emerges as a valuable solution for modern meeting management challenges.

Chapter 7: FUTURE SCOPE

The future scope of the web-based meeting scheduling application lies in enhancing its functionality, expanding integration capabilities, and leveraging emerging technologies to improve user experience. Incorporating advanced artificial intelligence and machine learning algorithms can enable the application to provide intelligent suggestions for optimal meeting times based on user behavior, organizational priorities, and historical patterns. Integration with more third-party tools, such as project management platforms and video conferencing systems, can further streamline workflows and improve collaboration. Additionally, supporting voice-activated scheduling via virtual assistants and extending compatibility to wearables could enhance accessibility and convenience.

To cater to a global audience, the application can include features such as multi-language support, localized time zone adjustments, and compliance with regional data privacy laws beyond GDPR. Scalability can be further enhanced to handle increasing user demands, making the application suitable for enterprise-level deployment. By incorporating advanced analytics, the application can provide deeper insights into productivity metrics, meeting effectiveness, and resource allocation. With these enhancements, the application has the potential to evolve into a comprehensive solution for scheduling, resource management, and organizational planning in diverse industries.

REFERENCES

1) Reddy, P. R., & Suresh, M. (2018). A novel approach to meeting scheduling applications using a mobile platform. *International Journal of Computer Applications, 179*(3), 25-29.

2) "Meeting scheduling software." (2023, July 15). In *Wikipedia*.

3) "Groupware." (2023, June 10). In *Wikipedia*.

4) Smith, A., & Johnson, L. (2021). Enhancing productivity with AI-powered meeting scheduling tools. *Journal of Information Technology Research, 16*(4), 54-63.

5) Survey on Web-based Meeting Scheduling Application TAST Thalawattha1#, DU Vidanagama,General Sir John Kotelawala Defence University, Sri Lanka