Ahmed Nasser

Sakina Razvi

Zain Jameeluddin

Mohammed Aziz

**Calendar Project Manual**

**Proposal/Description:**

This project mimics the Calendly.com application. The purpose of this project is to organize an appointment schedule and ensure sure that any user creating an appointment is unable book a spot conflicting with any other appointment or appointment types.

**Test Plan:**

The Calendar Project was thoroughly tested on every aspect, creating new user, new admin, changing passwords, and being able to change emails. The database will be updated with every action. Scenario examples: if a user has entered the wrong email within the database, they will be notified that no such email exists in our database. This feature is quite helpful to users since they might not have entered the correct email, or might not have an account at all, which they can sign up for. Once users select a time block for an appointment, it automatically reflects in the database. This was tested this with many users and different document types, as well.

**Features/Objectives/Specification/Role/Framework:**

Since the start of tackling this project, we have improved upon the following features. Create a specific login for the administrator - the person who is setting up the appointment schedules. For the administrator to set up appointment types e.g., office hours, consulting hours, etc. For the administrator to specify the day availability and/or can specify the occurring events. For the administrator to set up times that he/she is available. To specify the time range for each of the appointment types. To specify the time blocks for each appointment. To specify the number of bookings for any particular appointment. For the user (the appointee) to be able to login and view the list of appointment types. Or to set up different types of users for each appointment type. For the user to be able to view the calendar and can select days of appointment availability. For the user to not be able to see or have the selection greyed out, the appointments that have satisfied the booking requirement. For each appointment to be connected to the Zoom API for the booking time and booking information. For the user to be able to delete an appointment. A future feature to be added later may be incorporating file sharing so the user and admin are able to share documents with each other that are relevant to the scheduled appointment.

The calendar is a web base implementation and would work on a web base on any platform. Any IDE that has NodeJS installed would work perfectly, such as Visual Studios and other platforms. The framework we had used is Angular which is simple, in which an admin creates appointments, and the user is able to select which type of appointment. There are many things our team has completed; Haroon is working on the database, while Ahmed, Sakina and Zain are working on the project and figuring out what is included. The four of us will work on the project as a whole group, we cannot say this person has done this or that, since our group effort is shown in each page of code, there is an equal amount of work that we are contributing into this project.

**Requirements/Installation/Implementation:**

To install the code, a person must have the code folder, in which they must go into it and delete the directory and type CMD and hit enter. By hitting enter, you will then have to install Angular and then type many other commands to get it updated, and ng serve for the project to open and work. Throughout the system, our team will have a complete specification on how to set it up completely. We also had used many implementations which are Angular, JavaScript, Database: Firebase, Design Angular Material UI. The project automatically inserts and processes data from the backend. The project creates and logins the user automatically.

**Data Model/Class Diagram:**

Diagram

Description automatically generated

Diagram

Description automatically generated**Node Graph:**

**Conclusion:**

The Calendar Projects runs perfectly without errors and has the functionality of a real-life calendar. This can be published and used for teachers in institutions to help keep students’ schedules organized or by creating office hours if they need additional help in the course. The implementation of the project mimics the Calendly.com project, however, it does not have the Zoom API due to the monthly subscription charges. Zoom API is integrated within the project, but we have not imbursed Zoom to allow us to use their living program. We were successful in being able to accomplish all that was planned. A future feature to include within this project apart from appointments would be a file sharing application for the user or admin to share files and documentation pertaining to the specific appointment. Another nice future feature to implement within this project would be a customizable theme for the interface, so the user is able to select specific themes/colors based on personal preference.