Calendar+

Chad Kinnard, Bo Yang Li, Tyler Marrapese, Warren Overstreet

Dept. Of Computer Science and Information Technology

Abstract

Calendar+ is a calendar run as a dynamic web application that helps users personalize their schedule as they see fit. Through a simple account creation, a user's calendar is personalized to them. Accessed through the Google Chrome web browser, users can view their calendar from any Windows 10 device. Calendar+ makes it to where users can plan their days, weeks, months, and years according to their own needs. Through web browser notifications, users are reminded of important events that they have scheduled.

Time management is an important part of day-today- life, whether as a student, or a full-time employee, and Calendar+ acts as a place for users to keep track of their schedules and become more productive.

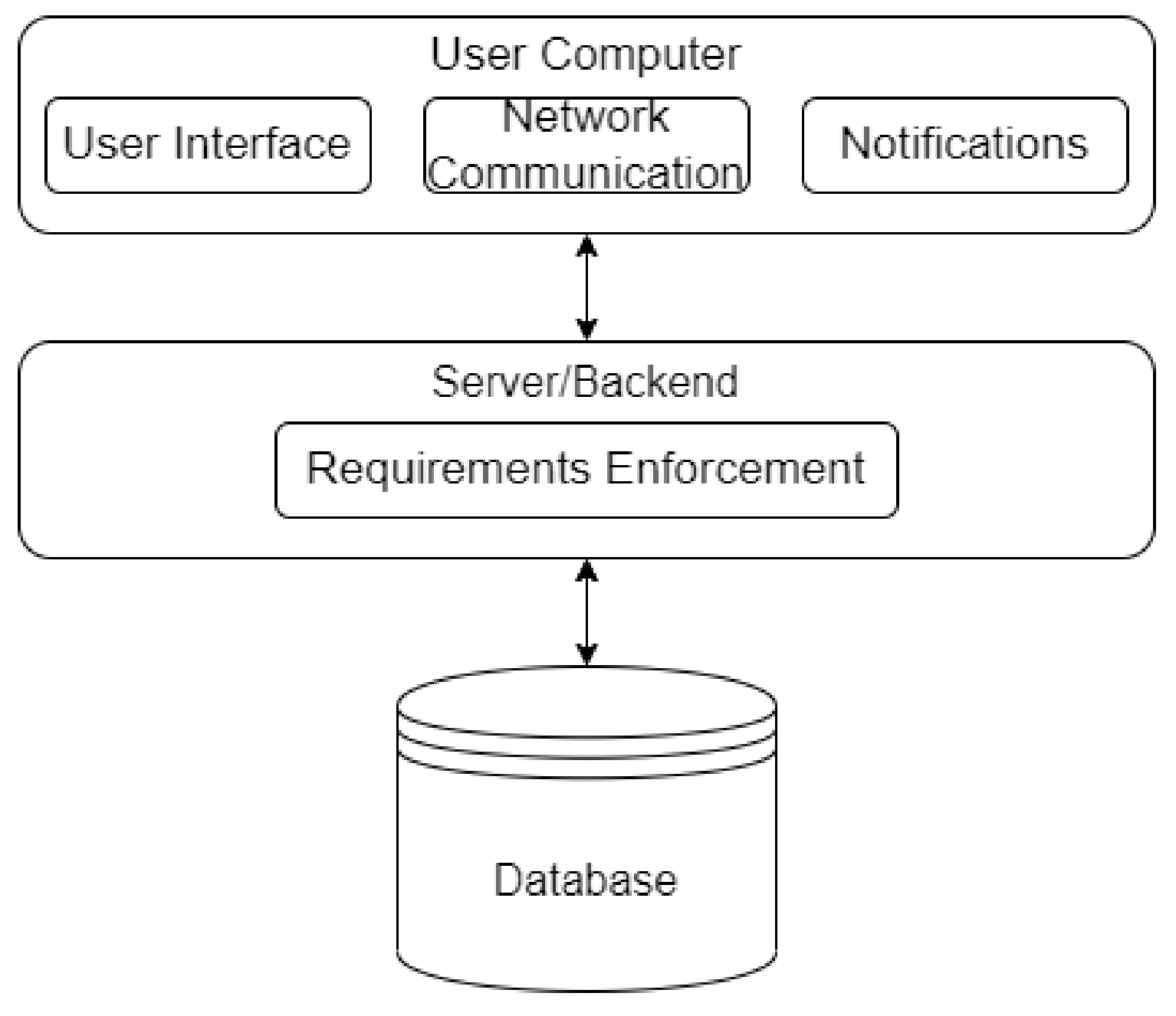


Figure 1: The block diagram for Calendar+. A three-tier layered architecture comprising of the client, the application server, and the database.

Purpose

For many of us, time is the most precious resource as it can never be returned. As such, time budgeting has become one of the most important tools in our daily lives. It is not only about knowing what one might do at what time and for how long, but also about other important details such as the specific details of planned events and the location of such events. Along the way, one might even desire to be reminded of important events some time before event start. Indeed, a calendar application would promote personal accountability, realistic event/task duration estimation, task prioritization, and event time boundary setting.

Calendar+ is designed to fulfill these needs. As a web application, it can be accessed from a Windows laptop with an Internet connection anywhere in the world.

Technology

For the web stack, Calendar+ uses Node.js and Express.js for the backend, MySQL via XAMPP for the database, Linode for cloud hosting, and HTML/CSS/JavaScript for its programming languages. We chose these technologies as they were commonly found in the private sector.

We also use git as our primary version control system, GitHub as our repository host, and Visual Studio Code as our code editor.

Our target platform is the Google Chrome browser on the Windows 10 operating system.

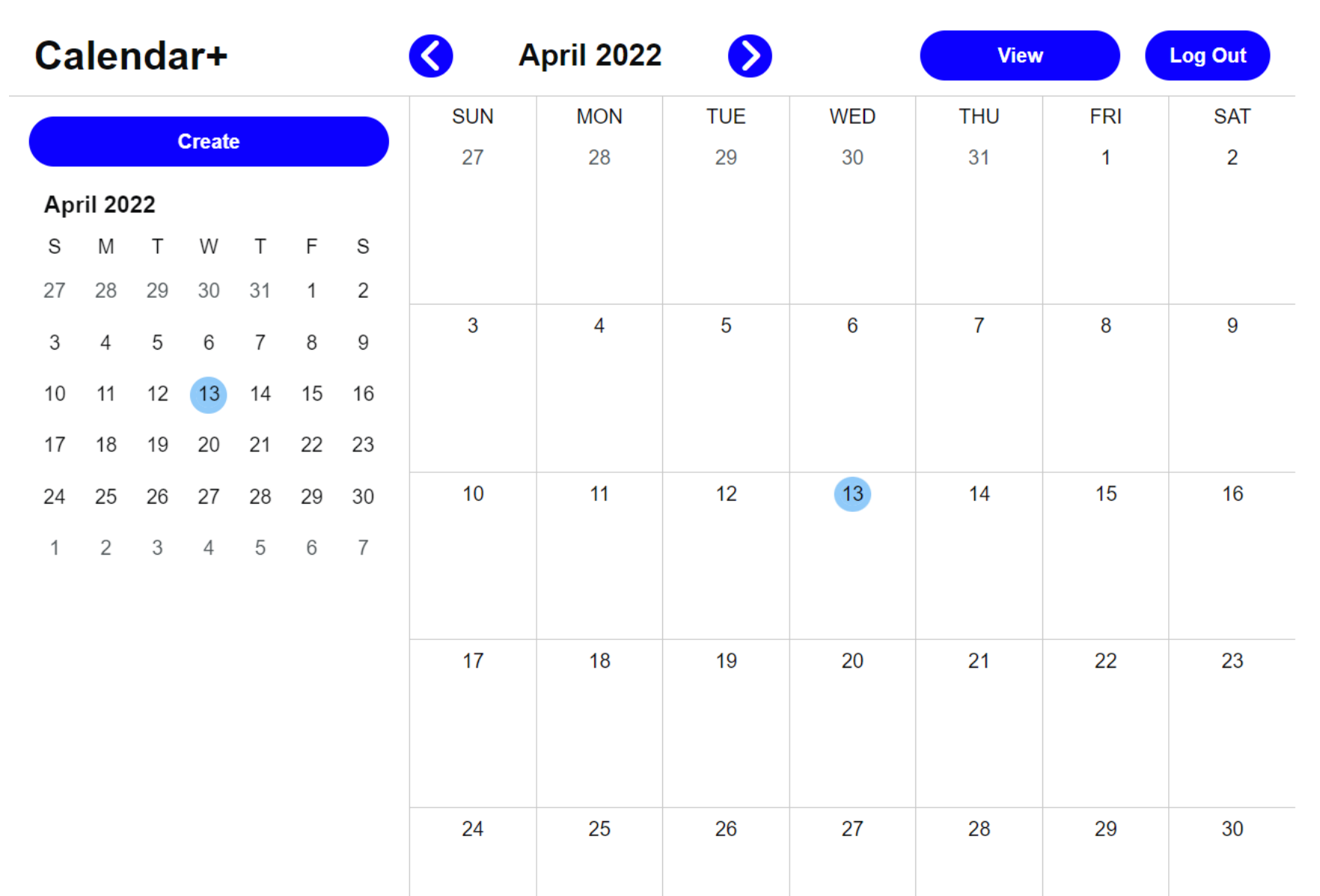


Figure 2: The month view, a 7x5 or 7x6 grid of cells containing the number of the day, and any events that fall on that day. Clicking a date on the calendar brings the user to the day view for that specific date. There is a month calendar on the sidebar which only shows the current month, for easy navigation back to the current day or month when the user is in a different view.

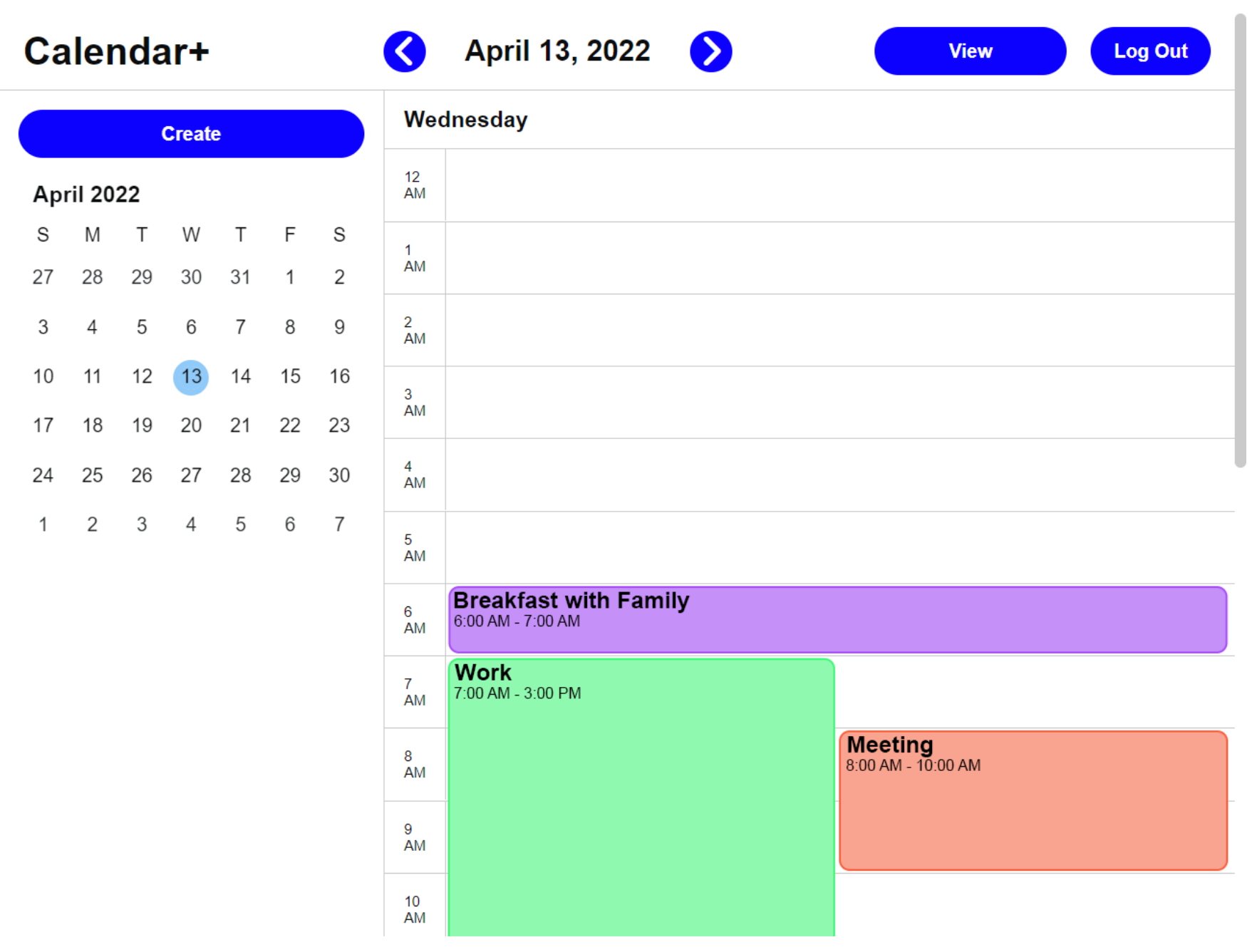


Figure 3: The day view, a row-based breakdown of events by hour. Events span from the row containing the hour they begin in, to the row containing the hour they end in. If multiple events occur at the same time, the width of each event is modified so they don't overlap.

Design

There are a total of seven pages and layouts, including: Login, Registration, Month view, Week view, Day view, Year view, and Schedule view. The page uses responsive design. Whenever someone visits the site and is not logged in, they will be greeted by the Login page.

The actual calendar pages all consist of three main elements: the top bar, the sidebar – which has multiple possible layouts and functions – and the main "calendar" component.

Events can span multiple cells, e.g., days or hours, depending on the view. On the day and week views, if there are multiple events that occur at the same time, each event occupies a proportional width of that column.

Within the database there are two tables, Users and Events. The Users table contains the log-in data of all user accounts. The Events table contains all events data across all users. Each event can only be linked to one user, but each user can be linked to many events.

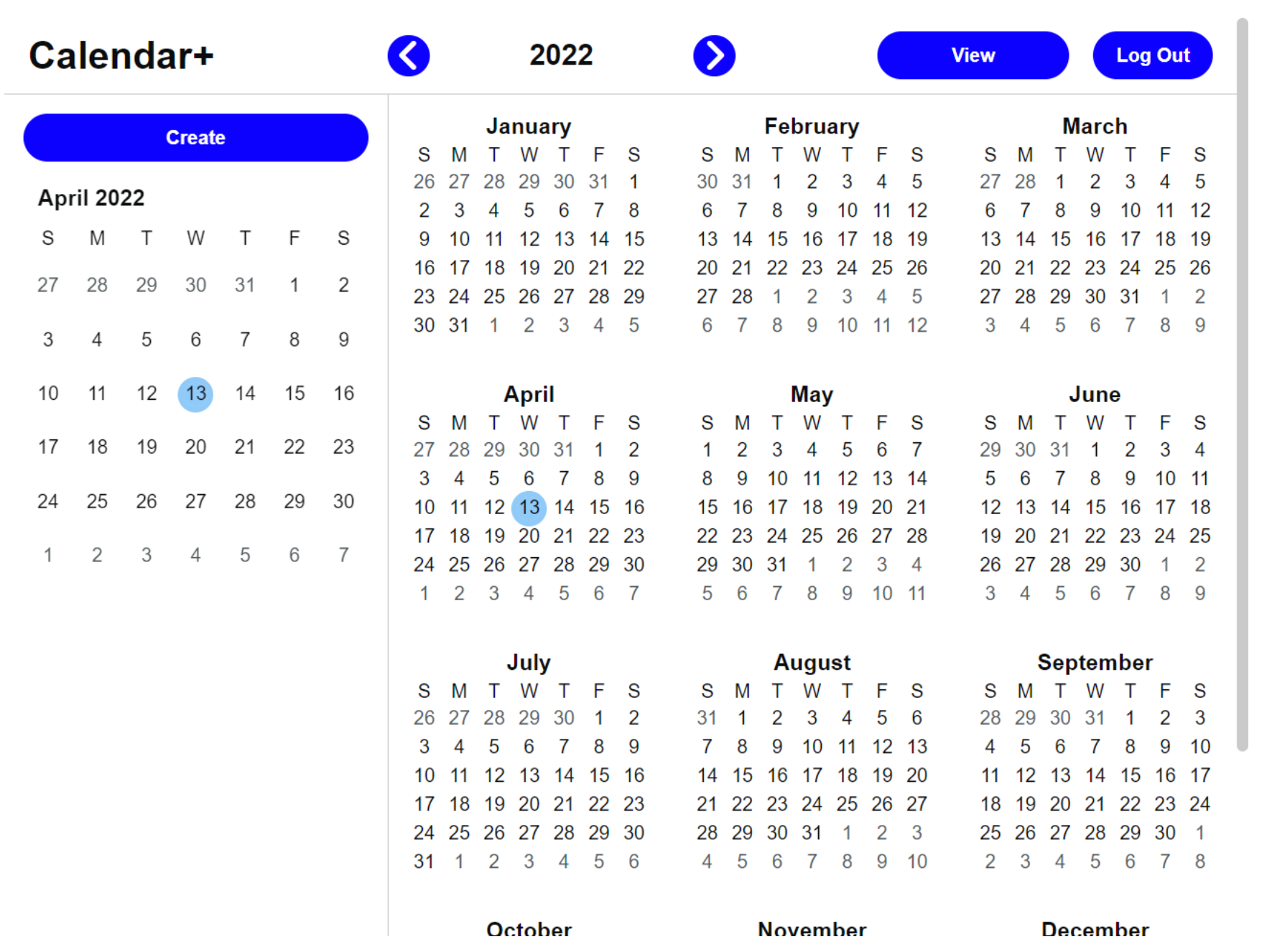
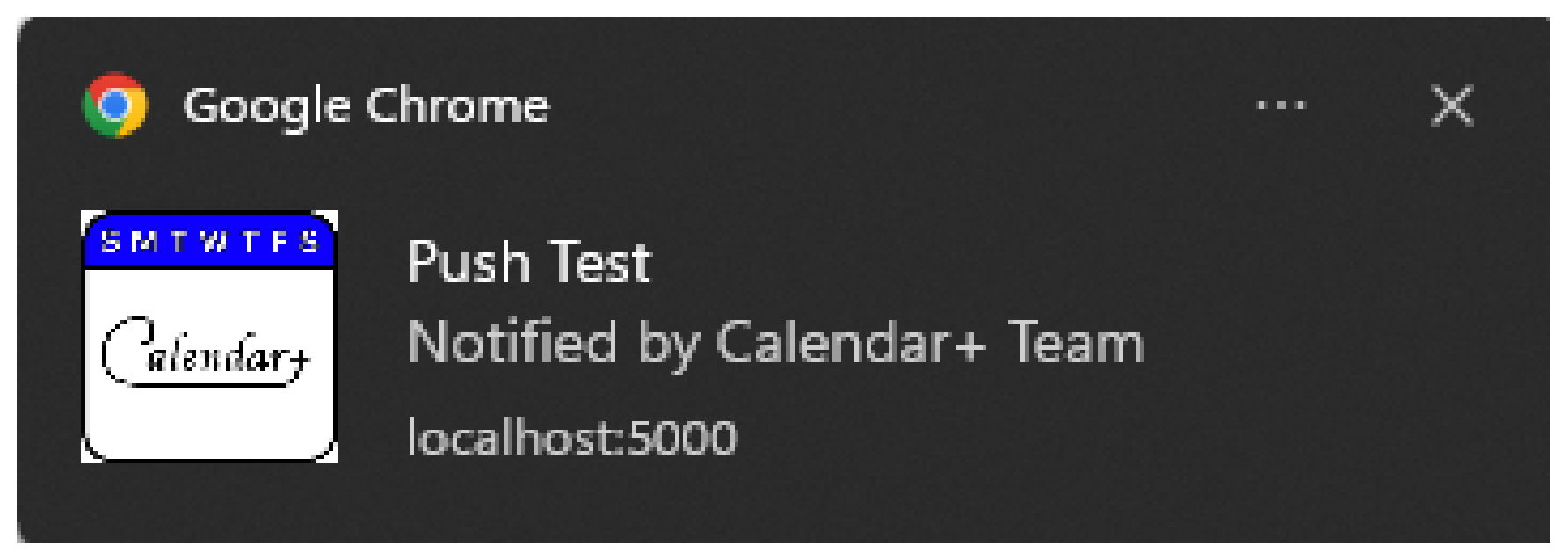


Figure 4: The year view, displays a grid of all months, clicking on a date brings the user to the day view for the specific date which was clicked on. Clicking on a month's name brings the user to the month view for that specific month.


Google Chrome

Push Test

Notified by Calendar+ Team

localhost:5000

Figure 5: An example of push notifications from the website displaying in the Windows notification tray. The user is notified at a chosen time before or at the start of the event, and the event of the title is displayed.

Future Work

There are many areas for future improvement of Calendar+. One future improvement is the implementation of multiple calendars which would allow users to toggle on/off a set of events. Such an improvement would allow users to better organize their events. Another area for future work is to implement event search by event title, allowing users to find specific events in their busy schedules. It would also be natural to implement Google Maps integration for those events, allowing users to obtain directions to destinations. To-do lists and tasks implementation would also add considerable utility to Calendar+ as would a mobile app port.

Backend improvements could also be made. For example, user session data could be stored in MySQL instead of the Express default.

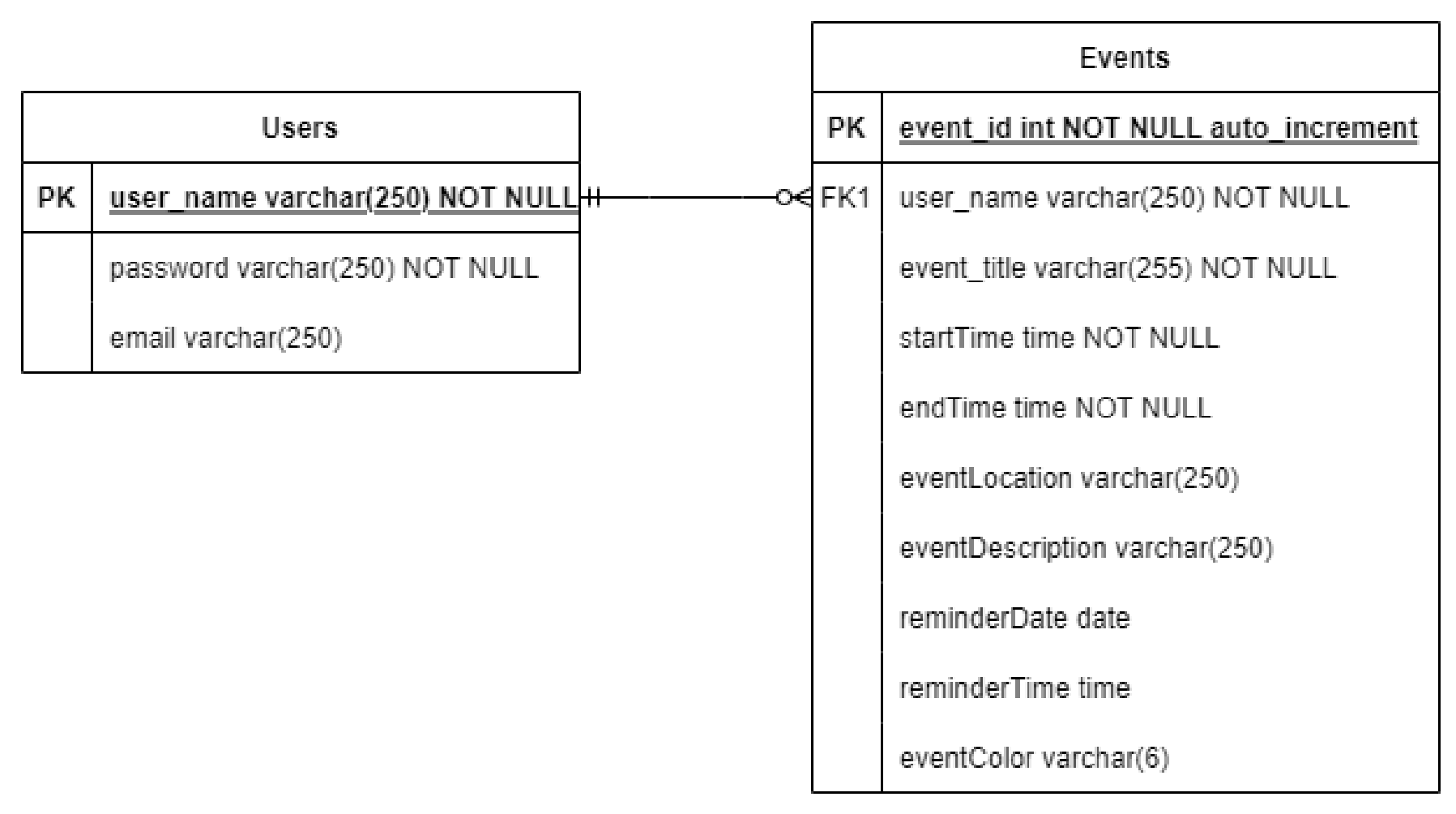

Users
PK
user_name varchar(250) NOT NULL
password varchar(250) NOT NULL
email varchar(250)
Events
PK
event_id int NOT NULL auto_increment
FK1
user_name varchar(250) NOT NULL
event_title varchar(255) NOT NULL
startTime time NOT NULL
endTime time NOT NULL
eventLocation varchar(250)
eventDescription varchar(250)
reminderDate date
reminderTime time
eventColor varchar(6)

Figure 6: The Entity Relationship Diagram (ERD) for Calendar+. Depicts a one-to-many relationship: one user may have many events.

Resources

1. Node.js - https://nodejs.org/
2. Express.js - http://expressjs.com/
3. MySQL - https://www.mysql.com/
4. XAMPP - https://www.apachefriends.org/download.html
5. Visual Studio Code - https://code.visualstudio.com/
6. Google Chrome - https://www.google.com/chrome/index.html

Acknowledgements

We would like to thank Dr. Karen Meisch for her support of students in the College of Science, Technology, Engineering & Mathematics, Dr. Leong Lee for his support of students in the Department of Computer Science and Information Technology, and Dr. John Nicholson for his support and guidance over the course of this project.