COMP4983 Capstone Project
SimpleSport Team Management
Project Report

Ryan Murney
151802m
Dr. Amir Eaman
December 1, 2022

Abstract

This paper describes my Capstone Project and what I aimed to achieve with this course. There were multiple goals I wanted to reach including creating a web application, making something related to my experience and learning a new skill. The project technologies consisted of some fundamental web development languages such as HTML, CSS, and JavaScript. I also used a Linux development environment and multiple JavaScript libraries. I named the application SimpleSport, and the functionality aimed to create easy-to-use software for sports team administrators or managers. The modules contained within the application are the Dashboard, Schedule, Roster, Messenger, and Settings each with their own respective functions. Throughout the project I faced some problems that challenged me. Some problems required me to learn new skills or work on existing ones. Learning new web skills was the most prevalent problem which required me to think differently about my software. I was also challenged to manage the project all on my own. In retrospect, there are considerations that I have learned that will help me create better projects in my future. The Capstone Project helped me learn a lot about my skills and was a wonderful experience to end my time at Acadia University with.

Table of Contents

Introduction

My name is Ryan Murney, and I am a fifth year Computer Science student specializing in Software Development. For my Capstone Project, I wanted to accomplish a few different goals. My main interest in software is in Web Development. With this opportunity I wanted to create a web application that could highlight my abilities to develop a web-based software. As I have completed Co-operative Education and worked as a junior web developer, I thought it would be a good part of my education to display. I also wanted to relate it to another part of my school experience. During my time at Acadia University, I also played the sport of rugby. In my upper years, I became the president of the club to which I took on many managerial requirements. It was my goal to create something that would be useful to other administrative members to manage a team. My last goal for my project was to take advantage of the project to learn a new web development skill. I wanted to do this to learn something new and give me an opportunity to get a head-start on my professional career. The culmination of all my goals became SimpleSport, a Sports Team Management Application that allows a team manager to schedule team events, create a roster with player information and mass message members of the team via email. The application aims to solve the problem of having a dedicated system to store data on your sports team instead of relying on spreadsheet software or multiple resources. It is used to compile all important team information.

Technologies

The technology I used for this project consisted of some standard web development software. The development environment I decided to use was Linux Ubuntu through a Virtual Machine software called VirtualBox. Linux is a great operating system to develop in and it is the one I am most comfortable developing in. I used VirtualBox since my laptop has Windows installed. My core front-end stack included JavaScript, HTML (HyperText Markup Language), CSS (Cascading Style Sheets). I used these technologies because they are great to use to create dynamic websites. My core back-end stack consisted of MySQL, NodeJS and ExpressJS. MySQL makes it easy to handle local databases and Node executes JavaScript code on the server, where Express is a framework of Node that handles routing to the server from the client. It is a popular stack that makes front-end and back-end connections seamless. The new web development skill I decided to learn was ReactJS, a front-end library that was created and is maintained by Facebook. It is a widely used open-source technology that uses a hierarchy of user interface components to structure applications. To handle version control of my code I used Git. It helped me save and control my code throughout development of the project. Lastly, I used a few different open-source JavaScript libraries for the front-end components of the project. These libraries include FullCalendar, React-Select, MomentJS and EmailJS.

Implementation

The implementation of the SimpleSport project included four main components built up by sub-components as per the React structure. My vision for the project was for the user to do three main functions, schedule events, save team roster data and send mass-messages to their team. This turned into the three modules respectively named the Schedule, Roster, and Messenger. However, I decided to add a fourth module called the Dashboard (See Appendix B). This is the traditional homepage of the application where the user views quick, upcoming information regarding their team and can-do simple functions without needing to access the other three modules. This resulted in four sections including a weekly view of events, a quick event creation, roster list with shortened data and a quick team message that emails everyone on the roster. While it could be defined as its own component, I think it is relevant to include the navbar description in this section as well. The application also has a top and side navigation bar, where you can choose to open or close the side navigation that shows the name of the module the user is currently using.

The first main module is the Schedule (See Appendix C). In this module I used an open-source library called FullCalendar to help create a dynamic calendar to display and add events. Events by definition are objects of data that consist of a title, type, start and end time, location, and a description. From the module users can add a new event by a button or by clicking on the calendar. Events were displayed and could be edited or deleted as well by clicking on existing events. Using a color key, events would appear with a certain background color on the calendar to define each event type as well. Existing events also provided a functionality to connect with Google and Microsoft services to add the SimpleSport event to your calendars of other systems. This was accomplished by creating a URL parameter string that would open a new browser window with the Google or Outlook calendar URL.

The second main module is the Roster, where you can add members to your team (See Appendix D). The definition of a member object included member name, role, phone number and email. This was the data shown on the Dashboard however in the Roster, there is more information shown as well. The other definition values included an Emergency Contact name, phone number and email as well. This was an important piece to the roster as in a sports environment injuries can happen. Especially with youth, it is important to have emergency contacts. On the Roster module there is a data table that shows each member and their metadata using columns and rows. You could add a new member via a button or click on individual rows of the table to edit an existing member. A unique feature of Roster is the role filter, where in a select dropdown, you can filter what members you are viewing by role type. This is handy when you might have a huge team and only want to see certain roles.

The third main module is the Messenger, where you can message your entire team or specific members or by member role type (See Appendix E). The message protocol is by email only. I accomplished this by using EmailJS. It is a mailing software to send an email with parameters to people using an account. A user would select members and or a role type and fill out a subject line, name of who they are sending it to, who the email is from and the message of the email. You can also select to carbon copy (CC) a member's emergency contact if you want them on the email as well!

The last module created for the project was the Settings module (See Appendix F). To add a little customization, the module breaks down into three pieces. A user can alter site settings which includes changing the name of their team and selecting which module they want to see upon loading the application. The other two pieces are similar as they are event and role type creation. By passing a label, a user can add custom types they can use to define their events or members!

Problem Description and Solutions

There were a few problems I encountered while creating the project. Most of the problems are related to technical development however there were some connected to project management. The biggest problem I faced was understanding React. It was a steep learning curve and challenged me in the beginning. As I had plenty of web development experience, I discovered part-way through that I was not looking at the project as a React project. I have been treating it like one of the web applications I had worked on before. At this point, I started using more of the React functionality such as state and hooks. Once I realized how they worked I figured out that I could improve a lot of components of my application.

Setting up a working back-end connection was also a technical problem. Previously I had not done this before but by referring to online resources I found examples that helped show me the ropes. Using Node and Express helped significantly as I could use JavaScript to make my connection to MySQL. There was a lot of trial and error with this process as I had to call a route, or an endpoint where my Node server could receive and use Express to get database information. I spend a lot of time making queries and network calls but each time I get closer to solving my problem. This part of the project was invaluable as I had no experience with that process.

As we had meetings throughout the semester, I realized that I was behind on my timeline each time. A problem I faced was keeping on track of my project and deadlines. In retrospect, my original timeline was slightly too tight. My comprehension of how long certain tasks would take

was not accurate. I would also lose track of my timeline when I decided to add extra functionality in certain modules. An example of this would be the roster filter that was not originally mocked in the proposal. Towards the halfway mark I started putting more hours in and deciding what was going to make the application and what was not. In the end, I completed the project with the functions I wanted. However, in my professional career I may not always have enough time and need to be more aware.

Possible Extensions

Going forward there are a few extensions I would like to include in the project. As mobile technology is heavily prevalent in today's age, the project should be mobile responsive. Screen size functionality is a huge part of the design process in most modern technologies. The project can currently work on some smaller desktop screens but nothing too small.

Another extension I would love to accomplish is a full-fledged league system. This would require user authentication and some identification of what teams belong to leagues, players that belong to teams etc. SimpleSport as it is currently the system a team admin would use but it could be a huge project if it was integrated with more teams. These teams could interact with each other to confirm scheduling, message each other internally through SimpleSport and so much more.

Reflection

Upon reviewing what I have accomplished with this project, I have reflected on what are the parts that I could have done better. It is easy to understand in retrospect why certain things were done wrong, but it is important to improve by reflecting on certain aspects of a task or project. In my proposal I referenced a few strategies I had to manage the project. One of them included using a project management software. Ultimately, I never used one which could have been the cause of being behind in my timeline. For my next project, I plan to use something to manage it. Projects can be easily disorganised depending on the scale and team size. It is a good practice to be well managed. Similar to project management, I could have used Git better to control my code. I did use it to save and commit my progress to a remote server in order to not lose my project due to a system failure. But I did not use it to manage new implementations. In doing so I could have forced a system breaking change with nothing to fall back on. My last piece of reflection would be to have some peer review work done on my project. While I am confident in my code, no one is perfect. There are surely plenty of logic errors and bugs that can be fixed. Sometimes just having that extra set of eyes on it can identify lots of issues!

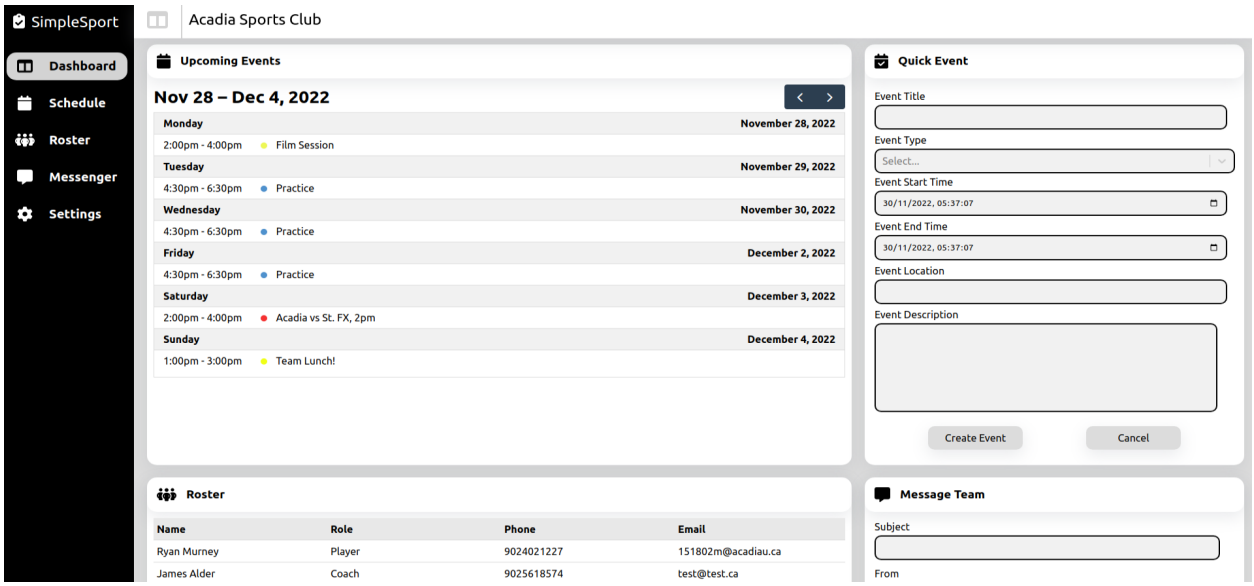
Conclusion

In conclusion, my capstone project was a fantastic learning experience in multiple diverse ways. I was able to improve my existing skills while learning new valuable skills in my field. My experience taught me a lot about my project management skills and things I should look to be aware of the next time I take on a project personally or professionally. The best part might also be that I was able to create something entirely of my own that has plenty of opportunity to build upon in the future. Overall, it was a success in terms of my project goals. I created a web application that connected my education and extracurricular experience of my time at Acadia University. With React I learned a popular skill that may help me in my professional career. The project taught me a lot about my own skills and abilities. It was an amazing experience that culminated my entire Computer Science degree at Acadia!

Appendix A: List of Files

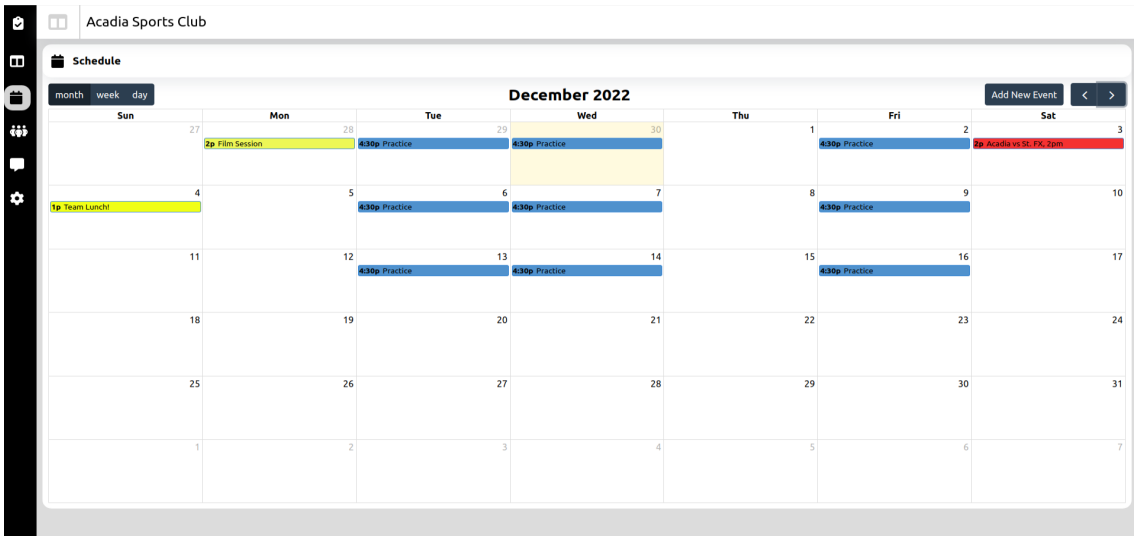
In the source code directory, there are two main directories, client and server. They are respectively named after what they contain. The client directory contains all the source code for the front-end of the application. This concerns the React library, functionality of the application and the visual content. The most important section in the client is the src folder. This contains all the React components, CSS and JavaScript files that run our application. The server directory contains the source code for the back-end of the application. In there, there are multiple code files that connect the Node server to our database and run the server routing. The most important files in the server are the JavaScript files database.js and index.js. The master.sql file is a history of my local SQL queries in MySQL to create my database. All files are viewable at https://github.com/Murnster/SimpleSport

Appendix B: Dashboard



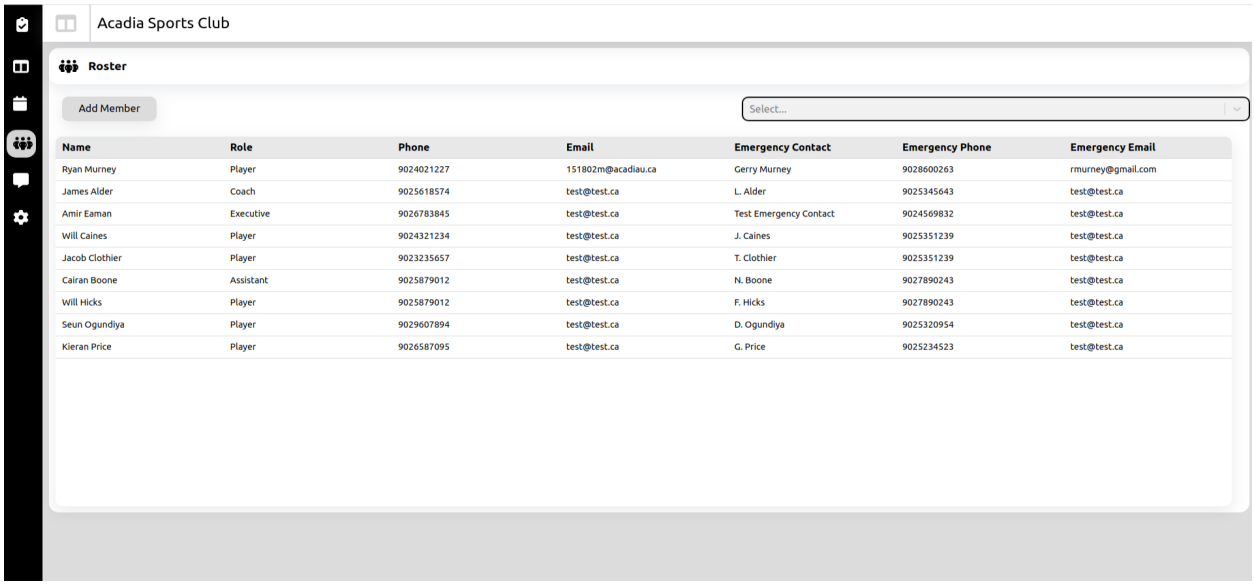
Dashboard: Traditional home screen of SimpleSport

Appendix C: Schedule



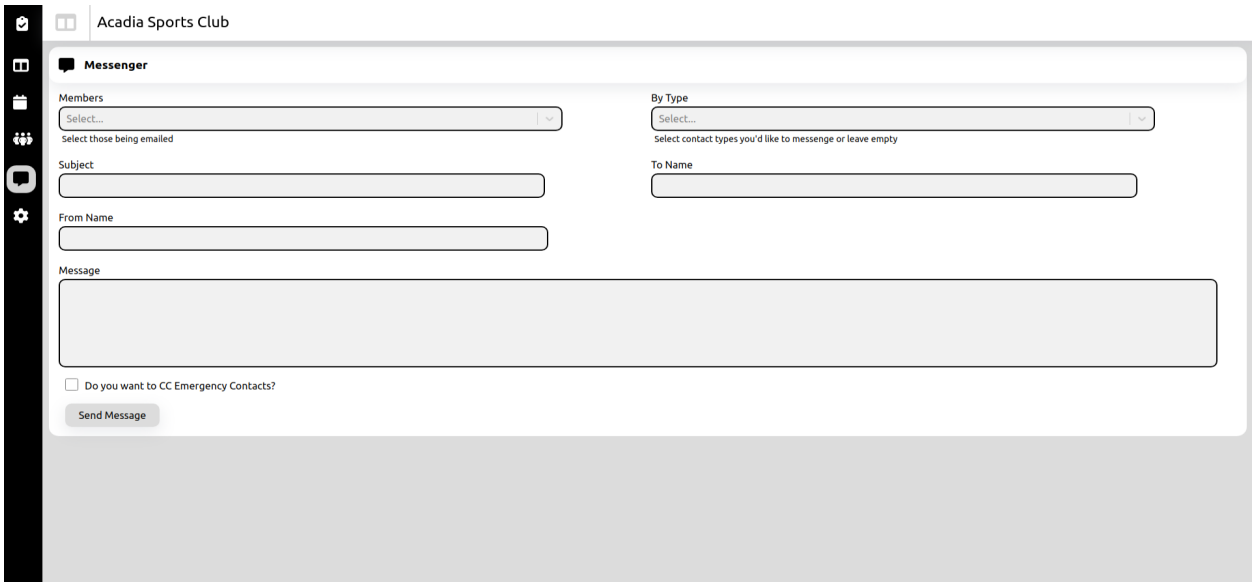
Schedule: Calendar view where users can create and delete events

Appendix D: Roster



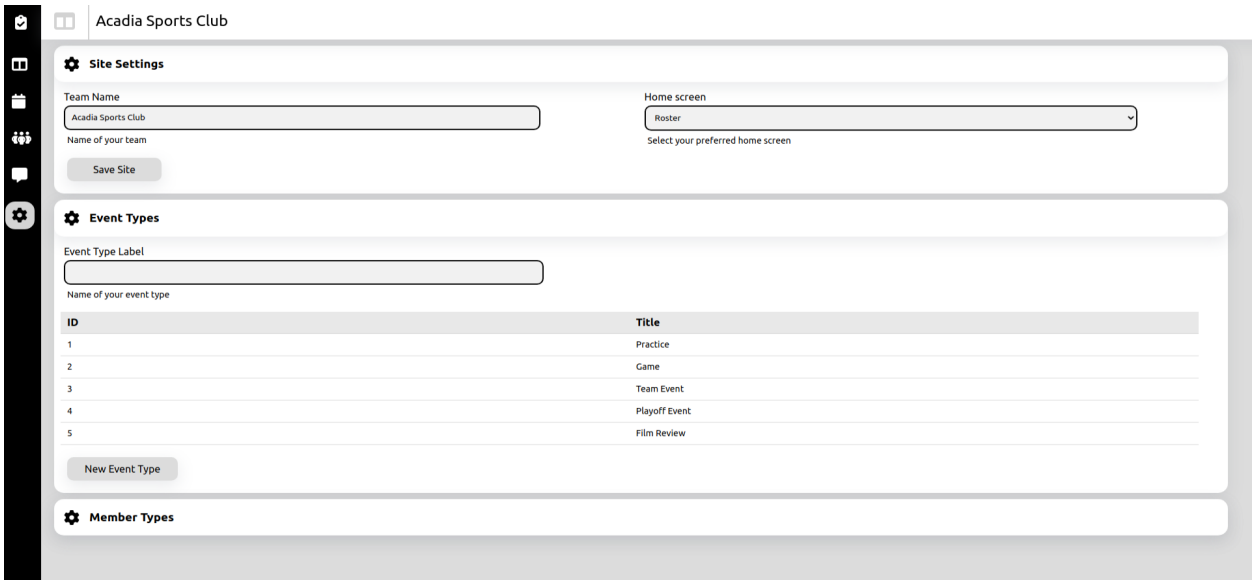
Roster: Roster table where user can see data on team members and create and delete members

Appendix E: Messenger



Messenger: Message service where user can email team based on different selections

Appendix F: Settings



Settings: System settings where user can update different components of the application