Capstone Project: Fitness Tracker and Workout Logger Web Application

Luis Perez Ocana

Benedictine University

Capstone Project (CMSC_4398_XC_0785)

Dr. Hernandez

May 5, 2023

Abstract

In this proposed project we describe the process for developing a comprehensive fitness tracker and workout logger web application using the FERN stack and deploying using Google Cloud services. This web application allows users to create an account profile where they can track their workouts, caloric intake, body weight, and monitor their progress via graphs and other visualizations, track their nutrition, and have access to a feature-rich user-friendly interface.

Keywords: Fitness tracker, Web Application, FERN Stack

Capstone Project: Fitness Tracker and Workout Logger Web Application

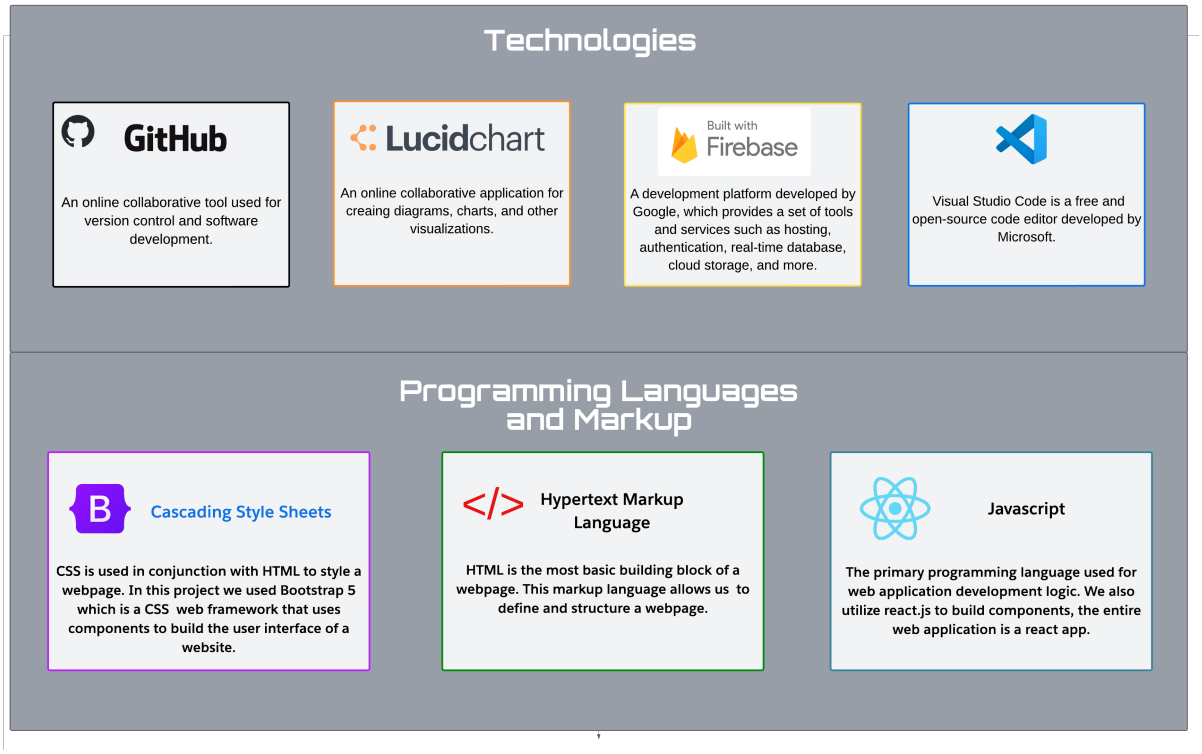
Software Development Framework

In this section, we discuss the software development methodology for developing a web application. The first step crucial step to developing any kind of software is to have a clear idea of what is to be developed. Before we start working on any project, we must ensure that we take the time to clearly define what we want to build this process involves creating the precise system requirement and we must also keep in mind what exactly we expect a software application to do since we must have realistic expectations. We must also be mindful of how we will achieve expectations and goals to ensure we develop software as was designed and intended. In this particular software project, the most relevant software development methodology for this project is agile development. Agile development emphasizes flexibility and rapid iteration which will benefit solo development. It also allows adaptation to changing or updated requirements and allows a solo developer to make progress quickly on a software project.

Technologies and Stack

In this section, we discuss the technologies and stack that will be utilized for developing and deploying the described web application. As for browser-based web applications, the front-end technologies utilized will include HTML, and CSS utilizing the framework Bootstrap 5, including JavaScript with React.js. The front end of the web application is a crucial component of the overall application since this will be the primary part the user will interact with, thus JavaScript will perform client-side logic, while React.js will be utilized for reusable interface components, such as workout logger, calorie tracker, etc. Another essential part of the application is the back-end of the application, this portion is responsible for managing the server-side processes of the application such as user requests or retrieving user data, handling API processes, and more. In essence, this provides data storage and manipulation, which will be necessary for creating users and storing data such as weight, height, workouts, etc. The technologies used for the back end

include Node.js and express.js which handles the server-side logic and can be integrated with other technologies like databases. However, in our case, we will utilize Firebase and the Firebase API so we don't directly work with these tools. Firebase is a comprehensive application development platform by Google. It provides software developers with a toolset to build, improve and grow their apps, apps made in Firebase are easily scalable. It provides developers with a variety of services such as authentication, databases, analytics, file storage, push messaging, etc, which would be difficult to build from scratch. Firebase is hosted in Google Cloud infrastructure which allows it to scale effortlessly with minimal work from developers. No web application is usable if it cannot be accessed via the Internet, thus we require a method to host our application so users can access the services we provide. As we mentioned hosting the application can be done with Google Cloud hosting. Using Firebase cloud hosting we are provided with scalable resources on demand, with only needing to pay for the resource utilized, meaning we can handle dozens to millions of users without any difficulty. Deploying the web application will involve copying the software and resources of the web application to the hosting platform and configuring the technologies for successful deployment. Finally, it is important to also list the tools we will use for development. It would be difficult to keep track of changes to our software which is why it is important to use GitHub to store all of our codebases. Another important tool is Visual Studio Code which will function as our integrated development environment. Lucidcharts will provide us with the capability to create visualizations that will be used for documentation, and Overleaf provides a platform for creating the documents necessary for our documentation.



System Requirements

In this section, we describe and discuss the system requirement of the web application, although we want to follow the requirements as stringently as possible some of the requirements could be subject to updates, modification, or deletion.

1. User profile: An application user should be able to create an account profile using their Google account. Their account will be linked to the Firestore database that will store personal information, including the individual user's height, weight, age, workouts, calories per day, etc.

2. Workout tracking: The application should allow users to track and log their workouts, including the time to complete their workouts. Users should be able to track strength training exercises under a progress section. In strength training, users should be able to input the exercise name, the weight used and the measurement system (kg, lbs), the number of repetitions performed for each exercise, and the amount of sets performed.

3. Goals and progress: Users should be able to track their progress. The application should provide a dashboard that displays the user's progress over time using the Chart.js Javascript library.

4. Nutritional tracking: Users should be able to track their nutrition. This includes their daily caloric intake based on their input, along with their macronutrient intake.

5. User-friendly interface: The application should have a user-friendly interface that is intuitive and easy to navigate.

6. Data security: The application should have robust data security measures to protect the privacy of users' data, users should be allowed to delete all their data from the Firestore database or retrieve all their data as a CSV file.

7. Open and transparent: The software itself must be open-source and free to use.

The first requirement essentially describes what should happen when a user creates an account once they have accessed the web application. Users will create a user account with their Google account or email, this is important as Firebase comes with tools for authentication. The second requirement focuses on describing the functionality of the web app, we aim to provide a feature-rich experience for users. The third requirement emphasizes the main characteristic of the web application, for a user to continuously use the application they must receive something important and significant from the services provided by the web app. This requirement ensures that users are able to set fitness goals and track their progress, this is especially important because this will motivate users to continue their fitness journey alongside the web application. The fourth requirement is probably an overlooked feature but fitness involves exercise and appropriate nutrition. This feature of the web application will probably be limited to just storing calories consumed, and macronutrient intake per day. The fifth requirement is self-explanatory, the user interface (UI) should be clean and easy to use, and we can use Bootstrap 5 to accomplish

this task. The sixth requirement is essential and which is why we discussed using Googles Firebase services to protect users, Firebase comes with user authentication so we don't need to worry about developing these features. The final requirement follows the principle that the technologies used for developing the application are open-source thus it would be beneficial to other developers and users to open-source the software used in the creation of the web app for transparency and future community involvement and collaboration.

Development Timeline

In this section we discuss the development timeline, note this is an estimate of the length it would take to complete each task, in general, the tasks must be completed before the deadline. The figure below is an overview of the typical sprints that will be followed during the development of this project.

Task to complete	Sprints
•Creative Design •Basic user interface design • Create requirements	Sprint 1 : February 6 - 19
•Set up all technlogies, Node.js, React.js, Firebase. •Basic web design and navigation.	Sprint 2 : February 26 - March 5
•Workout tracking component. •Add weight, sets, repetitions, and exercise •Calorie tracking componet, add macronutirent traking. Add wieght tracking component.	Sprint 3 : March 6 - March 19
•Create style sheets, improve website desing. Implement connection to database. Allow option to delete all data from the database or retrieve data as a CSV or excel document. •Test the application for bugs and fix them.	Sprint 4 : March 20 - April 16
•Finalize the user interface •Deploy and launch the application	Sprint 5 : April 17 - April 30

First Sprint Objectives

Create the application name, create the logo, and come up with a theme color. We go more in-depth later on under the design overview.

Create visuals of how the application will look, reference the design overview section.

- Define the requirements for workout tracking: Determine the information that users need to input for tracking their workouts, such as the type of workout, weight, measurement system, and the number of repetitions.

    Users must be able to input the type of exercise in their strength training routine.

    Users must be able to input the weight lifted, options of kg and lbs will be given.

    Users must be able to input the number of repetitions completed per each exercise.

    Users must be able to save their workout logs.

    Users must be able to add and remove exercises.

    Users must be able to view their workout history and track their progress over time.

    Users must be able to delete their workout history and export all data to a CSV file.

Second Sprint Objective

- Set up the development environment: Install the necessary tools, such as Node package manager, react.js, react-chart.js, and other app dependencies.
- Define the project structure: Decide on the directory structure and file organization for the project.
- Implement user account sign with Firebase API.
- Implement user interface design, use Bootstrap 5 to stylize the web app.

- Use a front-end framework, such as React.js, to build the user interface, by this we mean to develop the main web pages, such as a home, about, features, and login, each page will be a component.

Third Sprint Objective

- Create all components, the workout logger component, the calorie tracking component, and the weight tracking component. Create the client-side logic, for example in the workout component a user can add and remove exercises and modify the repetitions, sets, weight, and measurement system (kg, lbs). Connect these components to the Firestore database, we program the logic that allows the user to submit their data to the Firestore database.

Fourth Sprint Objective

- Write some CSS and use Bootstrap 5 classes to stylize the web pages. Create the progress component and program the logic that connects to the database to retrieve or delete user data. Test some inputs and verify that the inputs are registered in the Firestore database.

Fifth Sprint Objective

- Complete the user interface and fix any final issues. We create a final repository to store our production build application. We then deploy the application using Firebase services.

Submission cases

Here is an example of how a user might use the workout tracking feature:
John is a strength training enthusiast who regularly lifts weights at his local gym. He logs into FitloggerPro and navigates to the workout tracking feature. He inputs the weight he lifted and the number of repetitions he completed for each exercise he did, including the number of sets he performed. The system automatically uses the current date of

submitting the workout and stores this information in the Firestore database. Over time, John is able to view his workout history and track his progress, allowing him to adjust his workouts as needed to meet his fitness goals.

A possible workout log that gets stored in the database


```
  {
"user_id": "0wK82IO7B#GiK4f"
"date": "2023-04-15",
"duration": "65",
"exercises": [
  {
    "exercise": "Bench Press",
    "weight": 235,
    "reps": 10,
    "sets": 5
  },
  {
    "exercise": "Squat",
    "weight": 185,
    "reps": 8,
    "sets": 4
  },
  {
    "exercise": "Deadlift",
    "weight": 325,
    "reps": 6,
    "sets": 5
  }
```

```
  ]
}
```

A possible caloric tracking submission that gets stored in the database

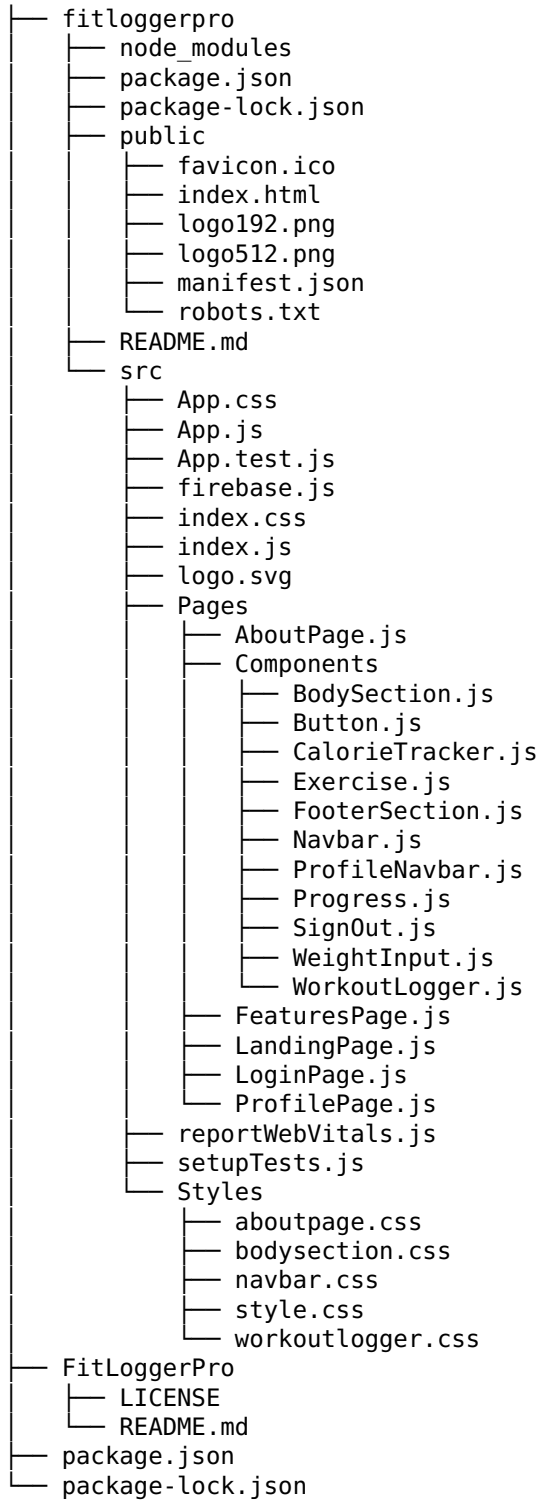
```
{
  "user_id": "0wK82IO7B#GiK4f"
  "date": "2023-04-15",
  "calories": "2265",
  "macronutrients": [
      "carbohydrate": "100",
      "protein": "120",
      "fat": "60"
    ]
}
```

A possible weight that is tracked by the user submission that will get stored in the database

```
{
  "user_id": "0wK82IO7B#GiK4f"
  "date": "2023-04-15",
  "weight": "185",
  "unit": "lbs"
}
```

Structure

File Structure



Design Overview

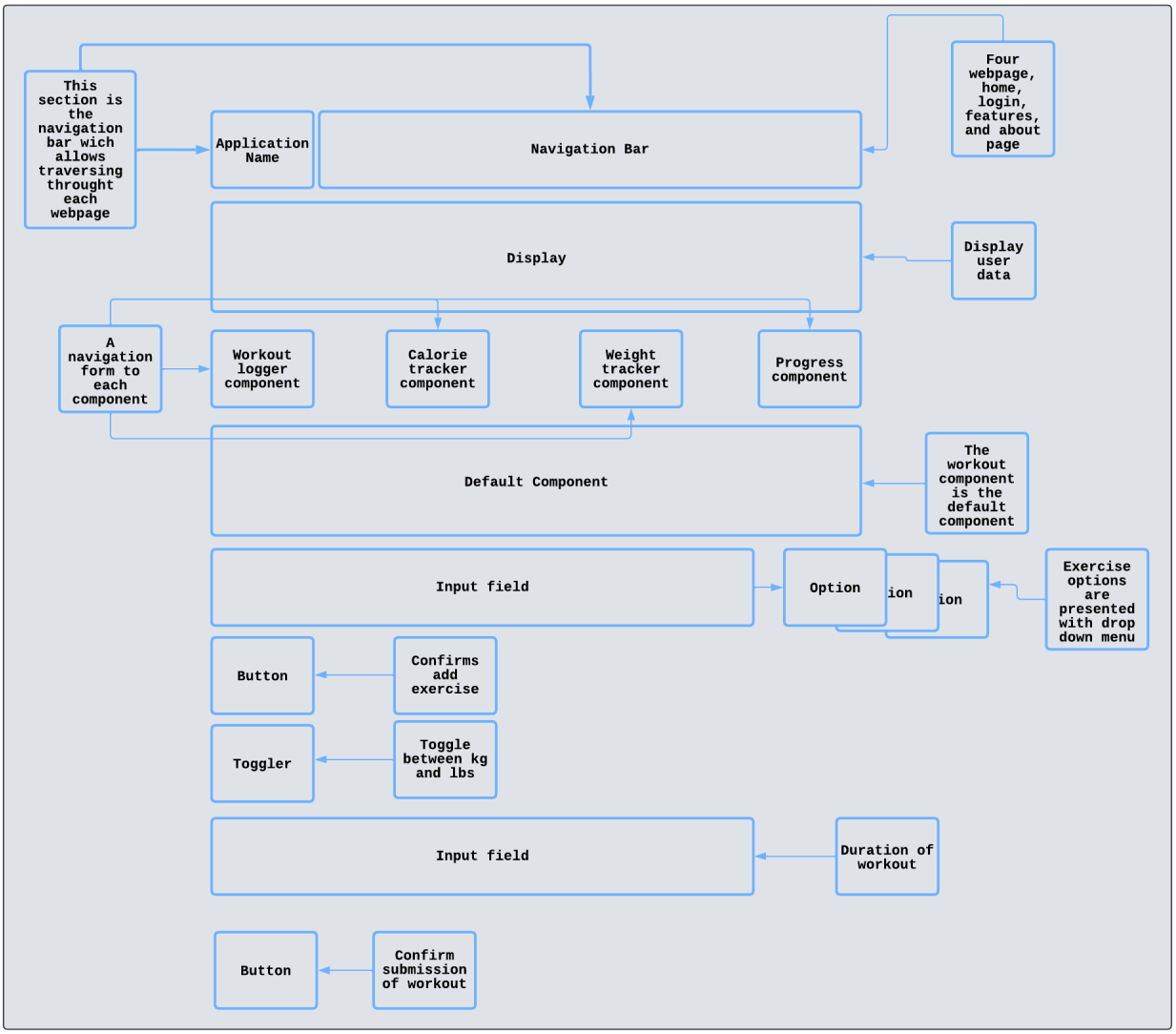
Dashboard: The user's dashboard should display four key components, a workout logger, calorie tracker, weight logger, and progress which will include charts or graphs that visually show how they've progressed over time. This can help motivate the user to continue working out and achieving their goals. Easy Navigation: The web application should have clear and easy-to-use navigation, making it simple for users to find the information they need. Input Fields: Users should be able to input a variety of information such as the type of workout, weight, number of repetitions, and date of the workout. These fields can be arranged in a form format and can be made more user-friendly with the use of drop-downs or auto-complete fields. Progress Tracking: The application should allow users to track their progress over time, including tracking their weight, total weight lifted, and other fitness-related metrics.

Logo and Web Application Name

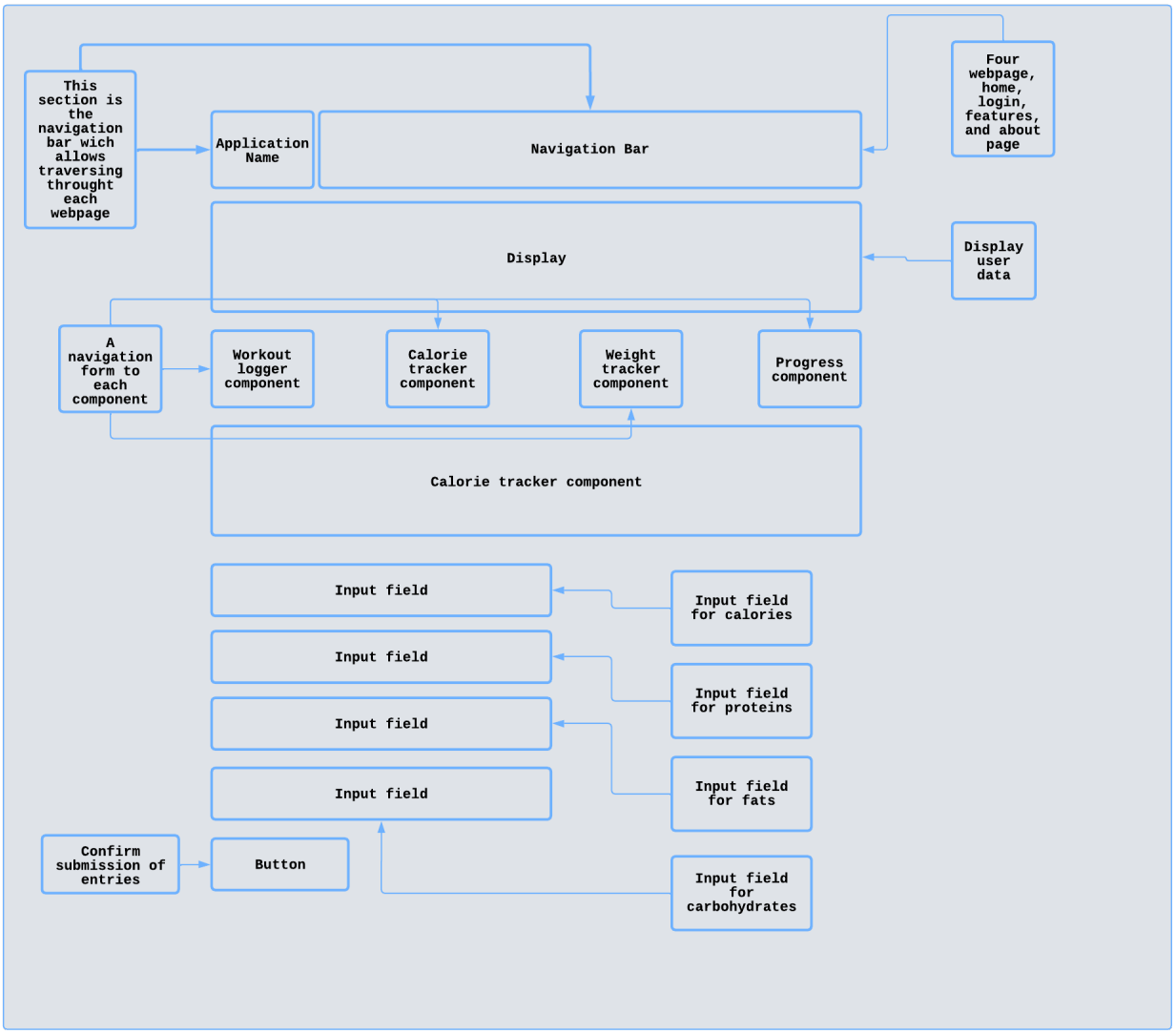
The app "FitloggerPro" boasts a name that is both engaging and easy to remember for those interested in tracking their fitness progress. With "Fit" in the name, it instantly conveys the app's focus on health and exercise. Additionally, "logger" implies that the app keeps track of one's fitness journey. To top it off, "Pro" hints at a high-quality and dependable experience. As a result, the name "FitloggerPro" effectively captures the essence of the app. Below is the logo for the web application

Component Mock Ups

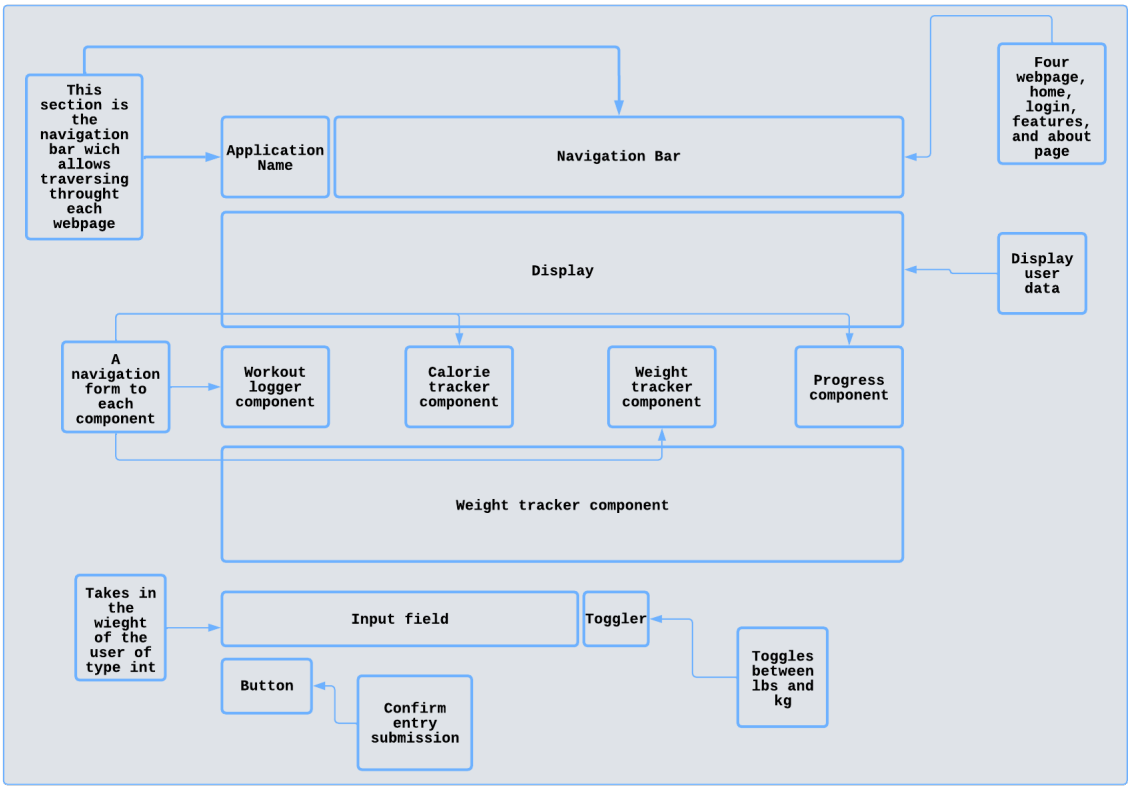
Workout component



Calorie tracker component



Weight tracker component



Implementation

We configure Firebase, first, we install all dependencies sudo npm i-g firebase-tools, npm i react-firebase-hooks, npm i react-charts-2, npm i papaparse, npm i react-router-dom, all other dependencies are installed automatically for react.js and node.js.

```
dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "chart.js": "^2.9.4",
    "firebase": "^9.21.0",
    "papaparse": "^5.4.1",
    "react": "^18.2.0",
    "react-chartjs-2": "^2.11.2",
```

```
    "react-dom": "^18.2.0",
    "react-firebase-hooks": "^5.1.1",
    "react-router-dom": "^6.11.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  }
```

```
import firebase from 'firebase/compat/app';
import 'firebase/compat/auth';
import 'firebase/compat/firestore';
const firebaseConfig = {
  apiKey: process.env.REACT_APP_API_KEY,
  authDomain: process.env.REACT_APP_AUTH_DOMAIN,
  projectId: process.env.REACT_APP_PROJECT_ID,
  storageBucket: process.env.REACT_APP_STORAGE_BUCKET,
  messagingSenderId: process.env.REACT_APP_MESSAGING_SENDER_ID,
  appId: process.env.REACT_APP_APP_ID
};

// Initialize Firebase
firebase.initializeApp(firebaseConfig);

export const auth = firebase.auth();
export const firestore = firebase.firestore();
```

This configuration allows us to read, write, create, and delete to the database. In essence, it provides the necessary settings and credentials to connect to our Firebase project and use various Firebase services like Firestore (database), Authentication, and Storage. This allows us to skip the development of the backend and focus on the front-end user

experience. The apiKey is the project's API key, which is used for authorization. The authDomain is the project's authentication domain, which is used for user authentication via there Google account. The projectId is the project's unique ID, which is used to identify your Firebase project for when we want to deploy to Google Cloud services. The storageBucket is the project's storage bucket, in case we want to use Firebase Storage. The messagingSenderId is the project's sender ID, used for Firebase Cloud Messaging. Finally, the appId is the project's unique app ID that identifies the app within Firebase. We have the following .env file, this will be added to our .gitignore file since the contents of this file should not be made public.

```
REACT_APP_API_KEY=
REACT_APP_AUTH_DOMAIN=fitloggerpro.firebaseapp.com
REACT_APP_PROJECT_ID=fitloggerpro
REACT_APP_STORAGE_BUCKET=fitloggerpro.appspot.com
REACT_APP_MESSAGING_SENDER_ID=
REACT_APP_APP_ID=
```

Deployment

When deploying the application, there are several steps to follow in order to ensure a successful deployment. The process is as follows, first we prepare the code for deployment. By this we check that all dependencies are set and up to date, this involves testing the code in our local development environment using npm start a live server running on default port 3000. Before deploying, we build the application for production. This typically involved running a build command npm run build, which optimizes the code and assets for production used in our deployment. The npm run build process creates a folder called build containing the production-ready files, which will be used in Firebase for deployment. We sign in to Fireabse and run the command firebase deploy which prompts us to sign in using Google authentication. After a successful sign in, our application is then deployed, and the platform will provide a unique URL for accessing the application. After

deployment, we can monitor the application for any issues and we can view our Firestore database for any user entries.

Discussion and Conclusion

Throughout the entire development of this project, we have worked on creating a comprehensive fitness application that allows users to log their workouts, track their weight, and monitor their calorie and macronutrient intake. We have utilized the react.js framework for building the entire front end using components. We used Firebase for the backend services to diminish the development time and load. We utilized various additional libraries and tools such as papaparse which creates a CSV file of the user's data and chart.js with react.js to enhance the functionality and user experience of the application. Throughout the development process, we encountered many challenges, primarily getting the backend to work and overall, understanding Firebase services for properly integrating this technology within our react.js components. Overall, this project demonstrates the power of modern web development technologies such as react.js, node.js, npm, and Firebase to develop a web application that is secure and easy to use. As we progress and make changes to our fitness app we can explore additional features and enhancements to implement such as artificial intelligence tools using LLM's that can guide and coach users to further improve their experience and achieve their fitness goals. We also continue our mission to be open and transparent, open-source and free to use, empowering everyone to reach their fitness goals.

References

[1] https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0

[2] https://firebase.google.com/docs/web/setup

[3] https://medium.com/nerd-for-tech/how-to-add-firebase-to-your-javascript-project-1cb998b51856