A blue circle with a person running

Description automatically generated

Extrasize

Get more from every workout.

Christopher Shannon

Brian Del Carpio

Melvin Wong

Ian Wu

Simon Fraser University

CMPT 276

Contents

[Project Overview 3](#_Toc150026030)

[Kanban Methodology 3](#_Toc150026031)

[User Stories 4](#_Toc150026032)

[Technology Stack 4](#_Toc150026033)

[JavaScript/React 4](#_Toc150026034)

[HTML 4](#_Toc150026035)

[CSS 4](#_Toc150026036)

[APIs 4](#_Toc150026037)

[Google Calendar 4](#_Toc150026038)

[Strava 4](#_Toc150026039)

[Project Features 4](#_Toc150026040)

[Work Breakdown Structure 4](#_Toc150026041)

[Project Timeline 4](#_Toc150026042)

[Wireframes/Prototype 4](#_Toc150026043)

[Data Flow Diagrams 4](#_Toc150026044)

# Project Overview

Our project is centered on developing an innovative running workout application that integrates seamlessly with APIs like Google Calendar and Strava to enhance the user experience in planning and tracking fitness activities. Utilizing a dynamic Kanban methodology to manage our workflow, we are able to address the challenges posed by our team's relative inexperience with large-scale projects by allowing for flexible task management and prioritization. The application is being built with a robust technology stack, leveraging the interactive capabilities of React for the frontend, coupled with the foundational web technologies HTML and CSS for structure and style. This combination provides users with a responsive and intuitive interface to schedule workouts, track their progress over time, and analyze their performance with visually engaging maps and statistics. As we progress, our focus on effective communication within the team and a rigorous review process ensures that we maintain high standards of quality and functionality in each phase of the project, from initial planning through to deployment and closure. This way, we can make sure our app is easy to use and works well from the time we start building it until we're ready to share it with everyone.

# Kanban Methodology

Why Kanban? One of the problems we face in this project is the lack of experience we have in a project of this scale. It would be difficult to use a waterfall method for example as we do not know all the steps that would be required before we get started.

With Kanban, we can dynamically add tasks to the list of to-do things. Allowing us to be flexible and solve problems we do not know are coming. With such a small team the biggest advantage we have is the ease of communication. Kanban requires excellent communication to be pulled off.

A concern we have with using Kanban is the complexity, with the ability to always add additional tasks to the to-do list we face the problem of a never-ending list.

For our Kanban billboard, we will have 4 lists as follows:

1. To-Do - list of tasks that need to be actioned.

2. In Progress **-** tasks that are currently being worked on.

3. Review - tasks that are completed but should be reviewed by another member.

4. Completed - tasks that have passed the review process

# User Stories

# Technology Stack

## JavaScript/React

- Why Chose It: React is a powerful and popular frontend library for building interactive user interfaces. Its component-based architecture promotes reusability and maintainability, making it easier to develop and expand upon the application over time.

   - Benefits: It offers a virtual DOM, which ensures efficient updates and rendering. This is especially beneficial for applications like ours where users might interact frequently, like checking available slots or booking workouts.

## HTML

 - Why Chose It: HTML is the standard markup language for creating web pages. It structures the content and forms the backbone of any web application.

 - Benefits: Ensures content is organized, accessible, and can be rendered on any web browser.

## CSS

 - Why Chose It: CSS is essential for styling the web pages, making them visually appealing and user-friendly.

   - Benefits: Allows for responsive design, ensuring the app looks good on all devices, and provides a customizable user experience.

 In wrapping up our technology stack choices for the frontend, React, HTML, and CSS emerge as foundational tools ideal for modern web applications. React empowers us with its capability to produce dynamic and efficient user interfaces, while HTML provides the structural framework essential for web content. Complementing this, CSS grants the application a polished and adaptable aesthetic. Together, these technologies form a cohesive combination, ensuring the web application is both functionally robust and visually engaging. With such a stack, users can anticipate a seamless and enjoyable experience when booking their daily running workouts.

# APIs

## Google Calendar

## Strava

# Project Features

Api's

1. Google Calendar Api

Allows to plan future events and look at past events

https://developers.google.com/calendar/api/v3/reference

2. Strava Api

Allows for creation of running workout stores time,distance

https://developers.strava.com/docs/reference/#api-Activities-getActivityById

features

see past workouts

    -using google calendar api we can see past workouts.

    -using the id number stored in google calendar we can get the stats of each workout(maybe have link to strava workout?)

plan future workouts

    -using google calendar api set time and date for future workout

Input Workout

    -using strava api to set distance and time then sent to store in strava

    -using google api put this workout into google calendar time and date

See up coming workouts

    -using google calendar api give a list of date of time of workouts that have being schedulled

(Maybe feature)Having Map that shows run

    -using strava get polymap to show run

# Work Breakdown Structure

A screenshot of a diagram

Description automatically generated

For this running workout app, the project is split into eight big steps, like Project Planning, Design, Presentation, and others. Each step has smaller, detailed tasks to make sure every part of the project is covered. This includes choosing how to develop the software, putting the app out for people to use, and writing down how it was made. This careful organization helps keep track of everything and makes the work go smoothly.

# Project Timeline

# Wireframes/Prototype

A screenshot of a computer

Description automatically generated

The design indicates functionality that combines activity tracking with calendar management to help users plan and track their running sessions and related activities. The interface employs a clean, inspired by ChatGPT, and structured layout with clear visual markers for ease of use.

# Data Flow Diagrams

Data Flow Diagram Level 0

A diagram of a company

Description automatically generated

Data Flow Diagram level 1

A diagram of a company

Description automatically generated

The Data Flow Diagram (DFD) provides a visual representation of the data processing within the application. It illustrates how data flows through the system, from the initial page loading to the final storage of workout details in the database. The DFD for this application indicates the logical sequence of steps, such as fetching current location data, accessing workout history, and interacting with Google Calendar and Strava APIs.