

Information

Level of Achievement	Gemini
Team Name	Anitamaxwynn
Project Name	TLC (Track.Lift.Connect)
Github	https://github.com/anitamaxwynnns/TLC

Motivation

Both of us are fitness loving individuals that enjoy going to the gym together frequently, and we would love to spread our passion to everyone else around us. However, we understand the many struggles people have in finding the drive and motivation to live a healthy lifestyle, seeing it as a chore rather than an interest. Therefore, we strive to make fitness more comprehensible and convenient to all, so that rather than seeing it as a confusing and arduous task, more people would be able to understand the enjoyment of living a healthy lifestyle.

Aim

We hope to bring fitness closer to everyone no matter what their schedule. We as university students understand what it is like to struggle to fit our fitness lifestyles around our busy and constantly changing schedules. As such, our application hopes to manage the planning of physical needs for all users to make fitness more accessible for everyone.

User Stories

1. As a university student who is new to fitness and wants to build muscle and get fitter, I want to be able to create a simple workout plan and be able to actively follow it despite my busy schedule.
2. As a housewife who wants to find a community of like minded fitness individuals to exercise together, I want to be able to connect with others that want to chase the same fitness goals.
3. I am an experienced fitness enthusiast that wants to explore different types of workouts. I want to be able to create different workout plans containing exercises that target specific muscle groups.
4. As an experienced fitness enthusiast, I want to track my workouts manually.
5. As a beginner to fitness, I want to easily follow a workout template to get started on my fitness journey.
6. As a user, I want to save my workouts so that I can use them in the future.
7. As a user, I want to plan out my workout schedule and tag a workout to each day.
8. As a user, I want to be reminded to workout so that I do not miss a session.

Technology Stack

- Expo (Frontend Development)
- React Native Paper (Frontend Styling)
- Supabase (Backend-as-a-service)
- OpenAI (AI-as-a-service)
- React Testing Library (Frontend testing)

Reasons for Technology Stack

Frontend

For the frontend, we chose Expo as our application framework as it is a popular and well-documented framework. It has many optimisations built in which will help in the speed and development of the application.

For styling, we chose react native paper as it has a great variety of components, giving us a strong foundation to build and style our application. It also provides useful react hooks that implement many functionalities.

Backend

For our backend, we chose supabase as it provides a suite of tools to build and scale our applications. Supabase uses PostgreSQL, which provides robust data storage and retrieval without abstraction. It also simplifies data abstraction operations. Its authentication module also allows us to quickly spin up email authentication, and opens up possibilities for other forms of sign up methods.

Core Features

1. Manual workout creator
 - Allows user to select exercises from a large exercise repository
 - Provides user with customisable set-rep configurations for each exercise and rest and workout timers, tailoring to their fitness goals
 - Allows user to save and edit workout for future use
 - Simple UI for easy usage
2. AI workout generator
 - Powered by OpenAI, the AI workout generator curates workouts based on users' height, weight, body type and fitness goals.
 - Allows user to save and edit workout for future use
3. Fitness Calendar
 - Has inbuilt calendar for users to tag a workout to a specific day
 - Has push notifications to remind user of their workout
4. Forum Page
 - Allow users to post their workouts for other users to see
 - Has commenting and like functionalities
 - Can export post to Instagram

Development Plan

Milestone 1: 3 June

Frontend	Backend
[Users] Login Page	Add users
[Users] Sign up Page	Authenticate users
[Feature] Add exercise repository	Exercise repository

Milestone 2: 1 July

Frontend	Backend
[Feature] Manual workout creation	Usage of Supabase to store user created workouts so that they can be referenced again in the future
[Feature] AI workout generator	Implementation of OpenAI APIs and Axios in React Native to connect bring the functionality of OpenAi's technology to allow users to create workouts based on the prompts that they enter
[Feature] Fitness forum page (Posting)	-
[Users] Profile Modifications (Profile Picture)	Update fields in supabase with the uploaded media

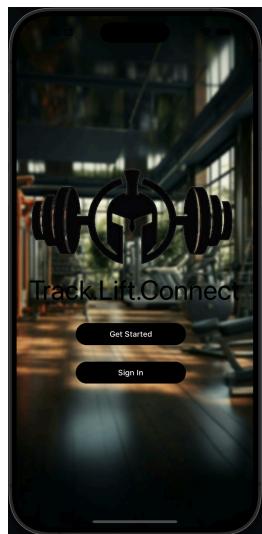
Milestone 3: 29 July

Frontend	Backend
[Feature] Post notifications for workout	-
[Feature] Create workout history	Store workout history in database
[Feature] Fitness forum page (Like and Comment)	-
[Users] Verify email page	Enable functionalities on supabase
[Users] Forget password page	Enable functionalities on supabase
[Feature] Workout scheduler (tagging workouts to a day)	Use workout history in supabase to tag workouts to calendar

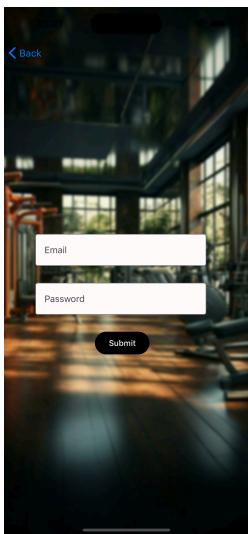
Good to have:

- Other sign in options: Google, Apple, Facebook etc.
- 2-FA authentication for additional security

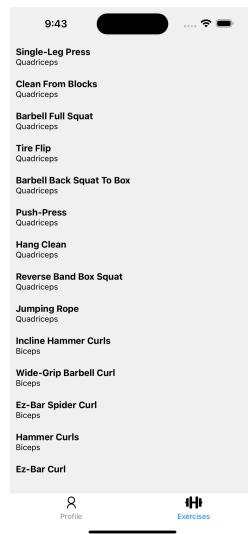
Technical Proof of Concept



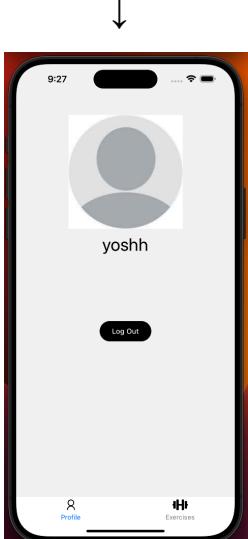
Start Screen



Sign in/up screen



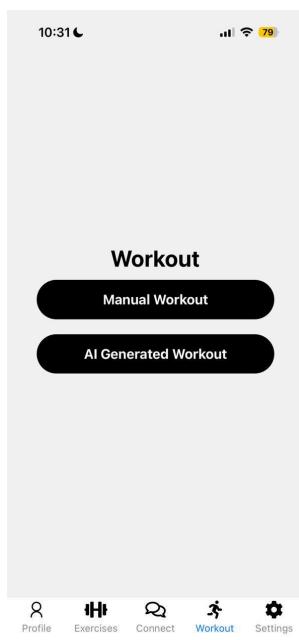
Exercise Page



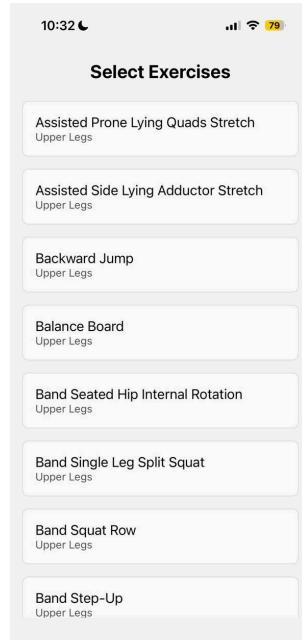
Profile Screen

Features

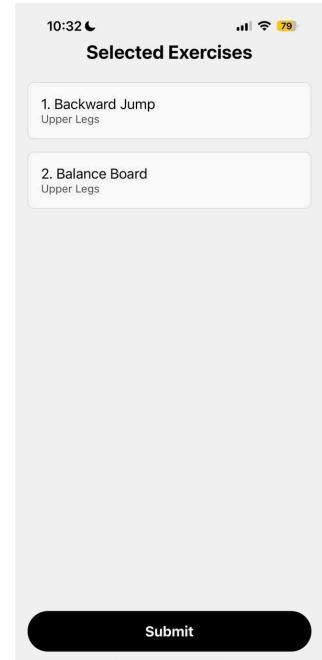
Name	Manual Workout Generator



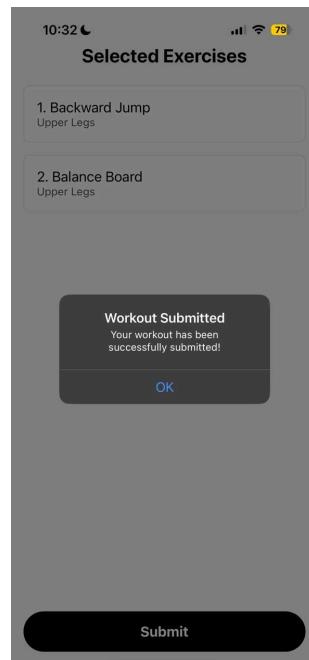
Select Workout Type



Select Exercises

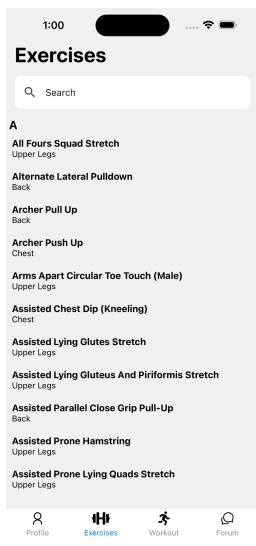


Exercises Selected

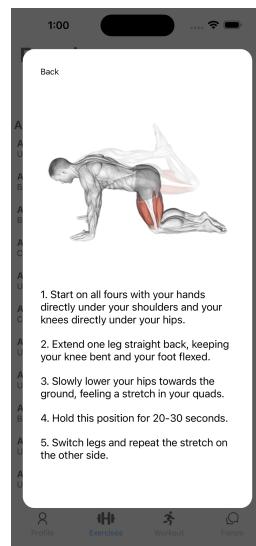


Workout Submitted

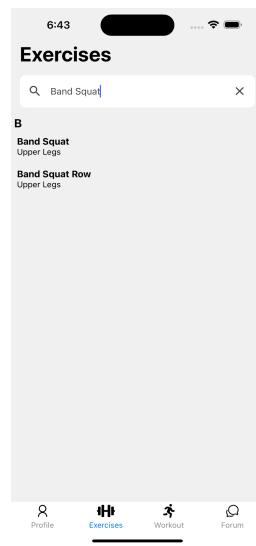
1. Exercise Repository



SectionList of Exercises

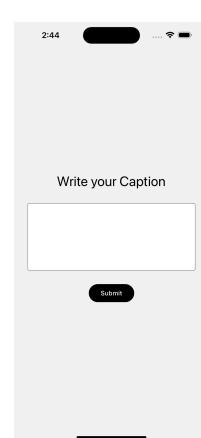
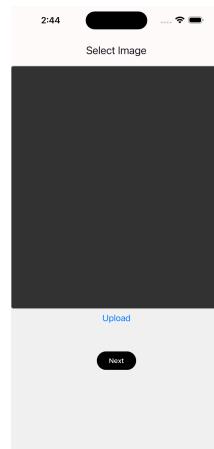


Modal for each exercise



Functional Search Bar

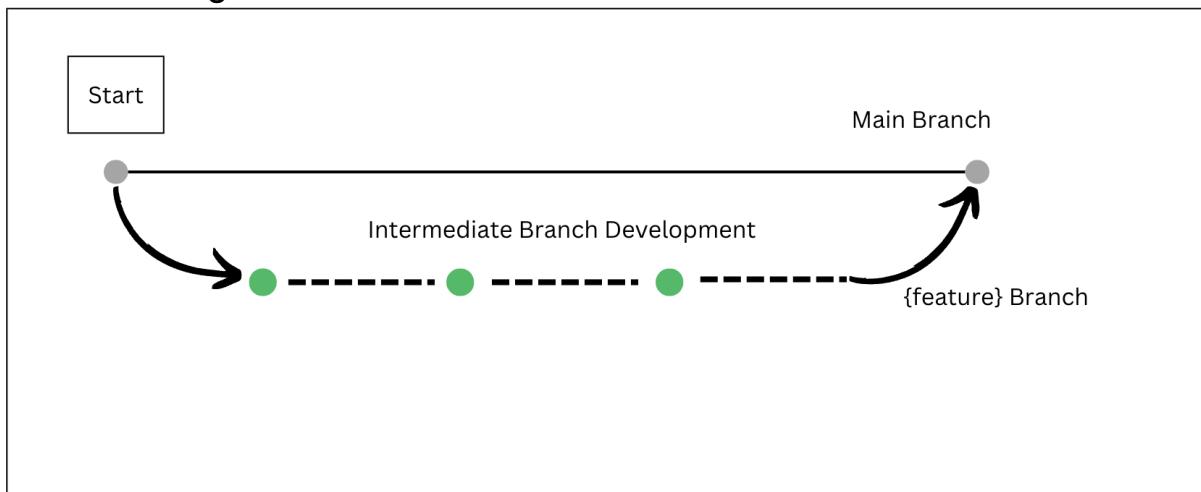
2. Forum Page



Software Engineering Practices

Git Workflow

Git Branching



For our git workflow, initially we have 1 branch, main. Our workflow starts when we want to implement a new feature:

- Create new feature branch from main
- Implement feature and commit changes
- Pull changes from main and resolve merge conflicts
- Create pull request to merge feature branch into main
- Test the changes and fix bugs
- A tag is created for each milestone

Issues

When we encounter a bug during testing, we file an issue under the related pull request. All bugs should be addressed and fixed before merging pull requests.

Software Engineering Principles

Abstraction

In our codebase, we wrote functions to encapsulate code often reused throughout our project such as Authentication functions as well as React Native components, keeping Abstraction and reusability of code in mind.

Incremental Development

Throughout the development process of our mobile application, we ensure that there is never a situation where multiple functionalities are added in a single pull request. Instead, as explained in our branching workflow, we branch off the main branch for each functionality to be added, ensuring that we can test these functionalities independently in later stages, without having to worry about bugs being difficult to trace.

Single Responsibility Principle

We have designed our codebase such that every function only has responsibility over its intended functionality. For example, the functions which are called for writing to Supabase only write the corresponding data without modifying anything beyond the scope of the function's responsibility.

Anticipation of Change

In accordance with the Single Responsibility Principle and Abstraction Principle, the codebase is modularised and abstractions are written in a way to allow for changes to be easily implemented without breaking other functionalities to fit specific needs if the needs arises, without having to worry about having to rewrite the function.

Use Cases

Description	Done
Authentication	
Sign up for account	T
Sign in to account	T
Select Profile Picture	T
Add custom name	T
View other profiles	T
Sign out of account	T
Exercise Repository	
View SectionList of Exercises	T
Modal for each exercises	T
Gifs and exercise info in each modal	T
Search bar to filter exercises	T
Create workout	T
Forum Page	
Add image to post	T
Add captions to post	T
View Flatlist of posts	T
Comment on posts	T
Like posts	T

Frontend Testing

Unit Tests

Description	Test Case	Expected	Passed
Authentication			
Renders login button if logged in	User logged in	Renders login button	T
Renders logout button if logged out	User logged out	Renders logout button	T
Error message renders correctly	test	test	T
Password input			
Password is hidden upon input	Password hidden	type= “password”	T
Password length at least 8 characters	Password length	Error: password too short	T

Manual Tests

Description	Test Case	Expected	Passed
User Account Forms			

Email Validation	Empty string	Missing fields error	T
	test	Invalid email error	T
	test@email	Invalid email error	T
	gmail.com	Invalid email error	T
	.com	Invalid email error	T
	test@email.com	Successful	T
Username Validation	Empty String	Invalid username	T
	A	Successful	T
	a	Successful	T
	A test	Successful	T
	A test testing	Successful	T
Password Validation	Empty String	Missing fields error	T
	String of length <=8	Too short	T
	String of length >=8	Successful	T
Avatar	No file	Missing fields error	T
	Have file	Successful	T
Email	Duplicate email	Email already used	T
Exercises			
Search Bar	Empty string	Display every	T

		exercise	
	String with key letters	Display exercise with key letters	T
	Random string	Display nothing	T