# LiftWell

***Overview:***

When deciding on what app to create, our group thought about the most impactful way we could create an app that would help people in their day to day lives. We first started to think about a calendar app that could assist people with planning their everyday lives. However, after discussing it, a calendar would not be impactful enough alone. We then started to think about a workout app that would give workouts to users and allow them to keep a schedule of them. This allowed us to combine our calendar idea with the workout app in order to give users a way to track their workouts as they complete them. This will also give users flexibility about when they want to schedule their workouts.

In order to allow users to login and to track their own workout data as well as other data stored in the app, we first connected our app to firebase. We then created a login page that asked users to enter their email and password in order to log in. If a user is looking at the app for the first time, there is also a prompt at the bottom of the page that allows the user to create an account. This allows the new user to enter a user id, and their name, email, age, and gender. Once signed in, the app will take the user to the home page in which the user can navigate to the calendar, quick workouts page, exercise page, the friends page, or sign out. Signing out will take them back to the login page. Clicking on the calendar link will allow the user to navigate to the calendar page in which they can open their calendar, where the user can view the workouts they have completed in the past. Clicking on the exercise link will take the user to the exercise page. In the exercise page, there are a couple different workouts for each of the following six body parts: legs, chest, biceps, triceps, back and shoulders. There will be pictures of each individual workout as well as a recommended amount of reps and the amount of time from each exercise.

Also, clicking on each exercise will take the user to a different page that has a longer description of the specific exercise, (as well as a link to a YouTube video of the exercise. The issue that was faced with this was that there were difficulties in trying to make the YouTube video play in the app itself. It was difficult to embed the video and run it in the app. However, a work around was to open the video in the browser and play the video that way. In the future, there will be a way to embed the video in the app using some type of library online. Clicking on the quick workout link will take the user to a page that allows them to select which exercises they would like to do based on whether they are trying to complete a full body workout, an upper body workout, or a lower body workout. We also implemented a friends page which allows the user to see all the friends they have. The view consists of a text box for the user to input an email address and a button to click to add a friend. The friends will then show up on the page implemented by a profile view. The goal of this feature was to allow users to interact with each other users based on their emails. The view uses Cloud Firebase to read the friends list and show the users friends.

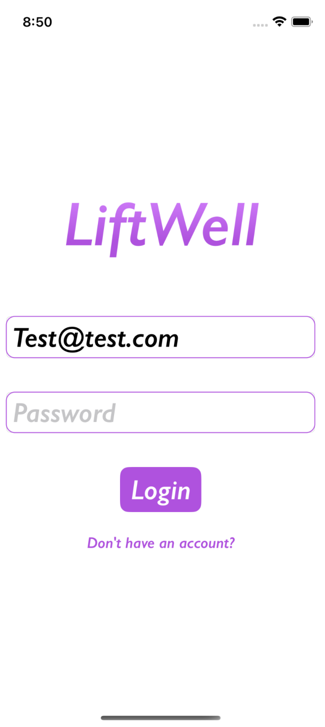
***Goals:***

Our first goal of the app was to allow the user to schedule different workouts for different days of the week. We accomplished this goal by setting up the calendar which allows the user to track their workouts throughout the days, weeks, and months, as well as schedule their own workouts in advance. Another goal of the app that we were able to implement was having a calendar functionality to plan their workouts for the week. We accomplished this by having a calendar view which let the user see their workouts they have completed. The last minimal goal that we set out to accomplish was the “Friends” functionality. In this, a user would be able to communicate their progress with other users of the app, more particularly users that they have “friended”. We were able to get this to a point where users can view their friends and their profiles, such as their physical stats and goals.

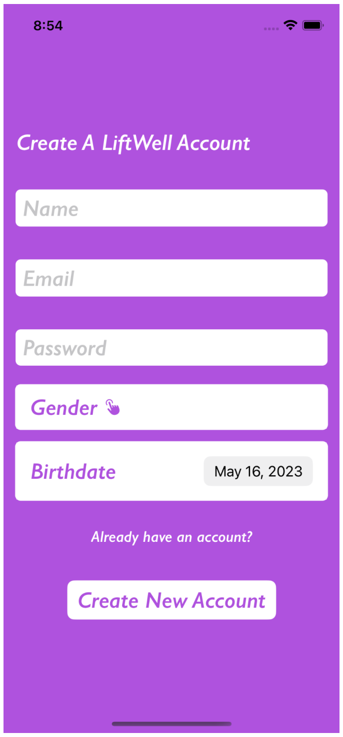
One goal of our app that we had to change as we implemented it was generating a workout plan for the user depending on the goals of the user. When it came to implementing the firestore database, we realized that it took more time than expected in terms of saving workouts to individual users, so instead we decided to have individual workouts that users could choose based on their own plans. This gave users more flexibility and allowed them to shape their workout plan exactly how they wanted. In the same vein, the idea to give the user the ability to choose between losing weight or gaining muscle relied on the fact that we could have separate workout plans, which as mentioned before did not fall within the capabilities of our app based on our current implementation. As such, we moved these to be a stretch goal. One goal of the app that we were not able to implement was the use of notifications for users. Due to having to be an Apple developer to test the use of Swift notifications, we were not able to implement this goal.

We were also able to accomplish a few stretch goals as well. One stretch goal that we were able to complete was including informative videos/pictures for each exercise included for each day. To achieve this, we added pictures in our exercise page in order to show the user how to complete the workout, and we also added a link to a youtube video that showed the user how to complete the workout. Another stretch goal that we were able to complete was having multiple user accounts on the same device. To accomplish this, we used firebase to store all of our users and allowed the user to change accounts by simply logging out of one account on their device and logging into another account. The last stretch goal that we were able to implement was the ability for users to customize their workout schedule. By including a calendar with icons for each set of workouts, users are able to customize their own workout schedule however they like.

***User Interaction Walk-Through:***

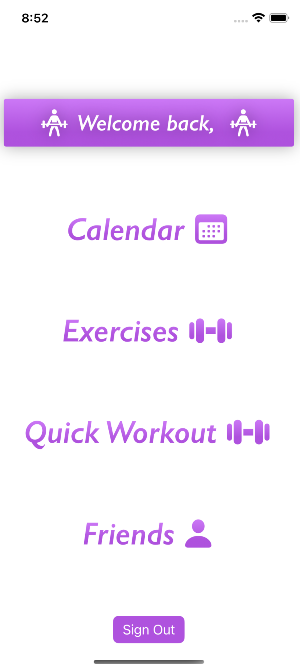


Test@test.com

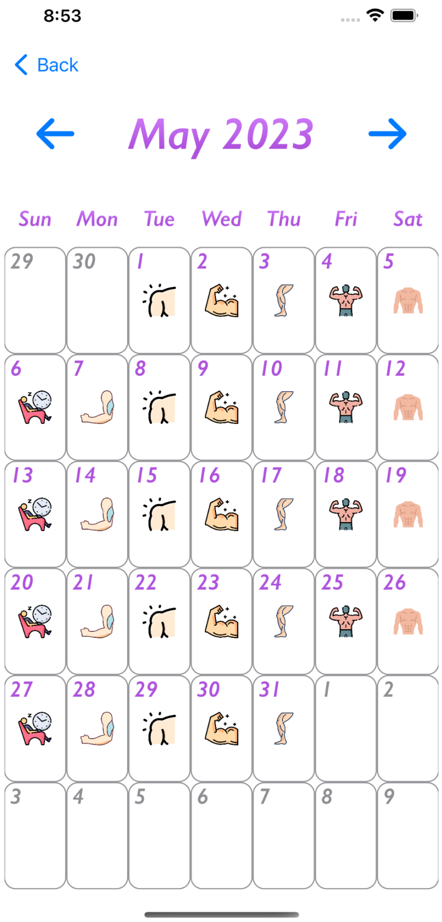


Birthdate May 16, 2023

These pictures show the login page in which users enter their email and password in order to login, and the create an account page in which users can create an account with their name, email, password, gender, and age.



This image shows the home page in which users can navigate to either the calendar, exercise page, quick workout page, or the friends page.



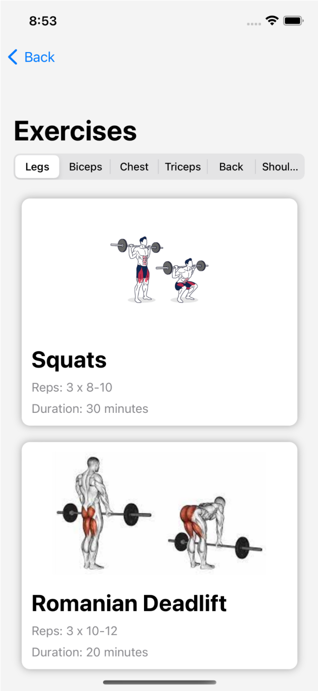
8:53

< Back

← May 2023 →

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|---|---|---|---|---|---|---|
| 29 | 30 | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

This image shows the calendar in which users can view their workouts for the month.



Exercises

Squats

Reps: 3 x 8-10

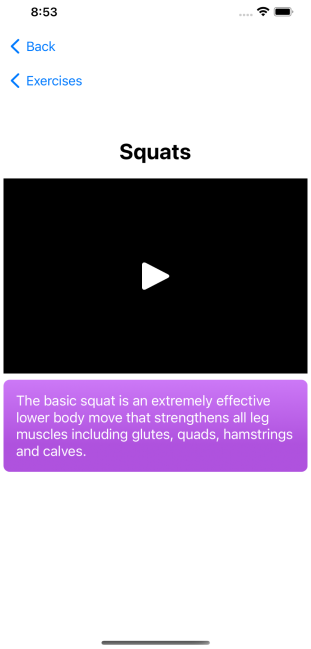
Duration: 30 minutes

Romanian Deadlift

Reps: 3 x 10-12

Duration: 20 minutes

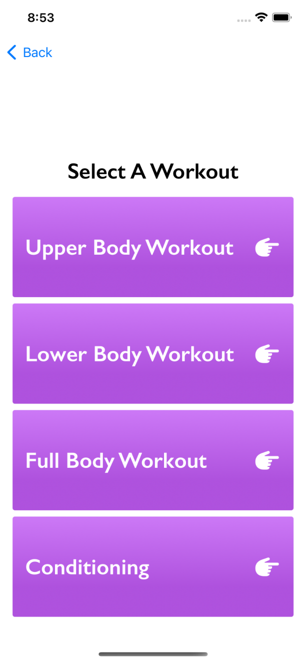
This image shows the exercise page in which users can view workouts based on different body parts, with recommended rep counts and duration.



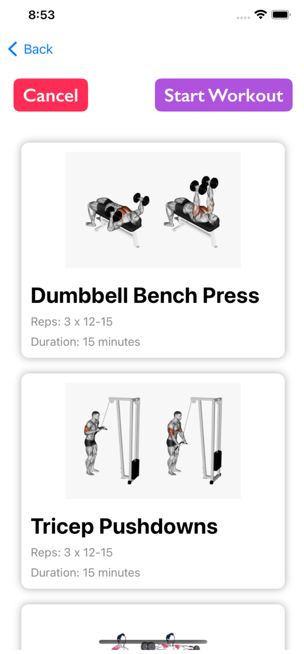
Squats

The basic squat is an extremely effective lower body move that strengthens all leg muscles including glutes, quads, hamstrings and calves.

When clicking on a workout, it will open this page which gives a video and description of the workout.



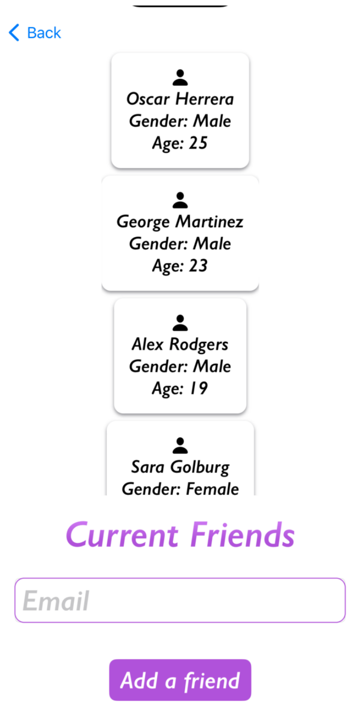
This image shows the quick workout page in which users can select workouts based on upper body, lower body, full body, or conditioning.



Dumbbell Bench Press
Reps: 3 x 12-15
Duration: 15 minutes

Tricep Pushdowns
Reps: 3 x 12-15
Duration: 15 minutes

When clicking on a specific quick workout page, this page will show up showing the specific workouts from the quick workout page.



Oscar Herrera
Gender: Male
Age: 25

George Martinez
Gender: Male
Age: 23

Alex Rodgers
Gender: Male
Age: 19

Sara Golburg
Gender: Female

This is the profile view of the friends page. You can add a friend if their account exists in the database.

***Development Process:***

For the development of the workout page, we initially tried to set up a picker that would change the view everytime the user selected a new workout. However, when attempting to set this up, we realized that this would now allow us to have much flexibility in the way we add specific workouts to each body part. To fix this shortcoming, we used a tab view in order to show each body part and a couple of workouts for each body part. To make it more informative for the viewer, we also added the duration and a recommended amount of reps. We also wanted to add images of each workout in order to show users how to complete the different workouts. To do this, we found images online that aligned to our workouts and added them to our assets in xcode. We then used these images in our exercise view using Image(imagename). To give our users a full description of the workout and the exact muscles that they work, we also added implementation in order to allow users to see the full description when they clicked on the specific workout. To accomplish this, we created another struct that would open when a user clicked on the workout and it would only show the description of the specific workout.

One of the essential functionalities of the app was to establish a database that would persist user information. We used Firebase's Firestore database and authentication packages to complete this task, allowing users to log in and out of our app, and retrieve information about their account, personal information, and their previous workouts. During the process of creating this dependency, we created two main login authentication views - one to allow users with existing accounts to log in, and the other to allow users to create a new account. One of the major setbacks of this aspect of the project was dealing with how to write and read data from the firestore database. We were able to circumvent this issue by making all of the data we wrote to the database codable and easy to understand and retrieve. By persisting the user data, we could read information about a user after they log in. For example, we made the app more user-friendly by having their name displayed on top of the app after they log in. In addition, users can access all of their previous workouts in a calendar so that they are able to track their own individual progress.

Another one of our goals was to include the aforementioned calendar. At first, we ran into issues with how to implement this, given that a calendar can be dynamic in the sense that not every month has the same number of days and pairings. After conducting some research, we were able to find that Swift has a useful library called Calendar, which we were able to use to handle some of the backend calendar calculations. For example, attempting to calculate the days in a month could be an intensive task without hard-coding certain values. By using the Calendar library we were able to utilize built-in functions to help us determine these values. From there, individual cells could be made with the corresponding day. After designing the initial cell view for the calendar days, we got to work on including icons to represent the different workouts that a user could choose from. We followed a typical 7 day split, with exercises occurring for 3-4 days straight, followed by a rest day for recovery.

For our app, we wanted to stay consistent with how the user interface was presented - in font, color, style, etc. This was not always a simple task to implement as there are many parts to this project in the model and in the views, and managing a consistent UI was difficult at times working in a group of our size. We overcame this by frequently communicating about style and constantly revising each other's work to maintain consistency.

We had to make a few changes from the initial plan. Initially, we wanted to generate a workout plan based on the user's personally defined goals. However, with time constraints and not wanting to limit users to a restricted workout schedule, we allowed for more freedom and flexibility in how a user could select a workout in the Quick Workouts view. This allows a user to choose a workout for a given day based on their own personal preference, and will ultimately write this selected workout to their stored database to be displayed in the calendar view. As of now there are only four preset workout options, but in the future we hope to add more and perhaps implement a custom workout feature that allows users to build their own workout, providing even more user freedom than they have now. Another change we had to make was in regards to sharing workouts and messages. We were able to implement a friend functionality, but unable to establish a system that allowed communication between users. Although we were not able to implement this by the submission date, in the future this is a functionality that absolutely could be implemented and would enhance the quality of the app.

When developing the ExerciseView, there were a few parts of the process that we were unsure on. How would we store the workouts? How would we display the workouts? How would we allow the user to find the workout they want? These questions were solved as we iterated through the view. The approach we chose was a “card view”. This allowed users to see a preview of each exercise and the time it took to complete it. This can be beneficial to the user as it allows them to pick workouts based on how much time they have to spend.

***Potential Future Directions:***

In order to fully develop our app, there are many different directions in which we can explore. One potential development for our app would be to create meal planning that is based on the user's info as well as their workout plan in order to assist the user in keeping up with their workouts as well as getting the most out of their workouts. In order to implement this, we would have to create an algorithm that will allow us to take a user's weekly workout as well as their age, gender, and weight, in order to output a recommended meal plan that will help users make the most of their exercises. We can also leverage this into a business opportunity by partnering with a meal service planner such as Weight Watchers in order to allow them to advertise their meal planning while helping users in our app.

Another way that we can help users by developing our app is by creating challenges that users can complete in order to earn points or prizes. A similar idea of this is used by Sweatcoin, which keeps track of a user's daily steps and allows them to turn it into sweatcoins which the user can use to buy prizes in the Sweatcoin shop. In order to make our users more likely to keep up with their workout schedule, we can incentivize them by using a similar approach, and tracking their workout data over a long period of time. We can also use this as a business opportunity by creating our own shop in our app and selling items from different companies.

In order to help our users workout correctly, we can also develop our app by adding a coaching tab in which users can schedule workouts that can be completed on zoom with instructors that will motivate them as well as prompt them how to correctly complete the workout. We can also have video coaching as well for users who like to workout late at night or early in the morning. If we wanted to make money off of this as well, we could create a paid version of this app and charge for coaching services.

Another yet unforgotten aspect of app making that we would focus more on is accessibilities. As the app becomes more polished in the future, the focus would instead be "Does it work", to "Who does it work for?". Later on in the app development cycle we would focus on implementing features for those with disabilities and accommodations that would not be able to use the app. As a group we talked about having a feature for red/green color blindness, visual impairment, dark mode, etc. These features would help alleviate those issues that our currently faces (regarding how accessible it is).

**Image Sources:**

Chest calendar icon - "Icon made by Rabit Jes from www.flaticon.com"
Back calendar icon - "Icon made by Freepik from www.flaticon.com"
Legs calendar icon - "Icon made by Icongeek26 from www.flaticon.com"
Bicep calendar icon - "Icon made by justicon from www.flaticon.com"
Shoulders calendar icon - "Icon made by Vitaly Gorbachev from www.flaticon.com"
Triceps calendar icon - "Icon made by Anggre Tionanda from www.flaticon.com"
Rest calendar icon - "Icon made by Freepik from www.flaticon.com"