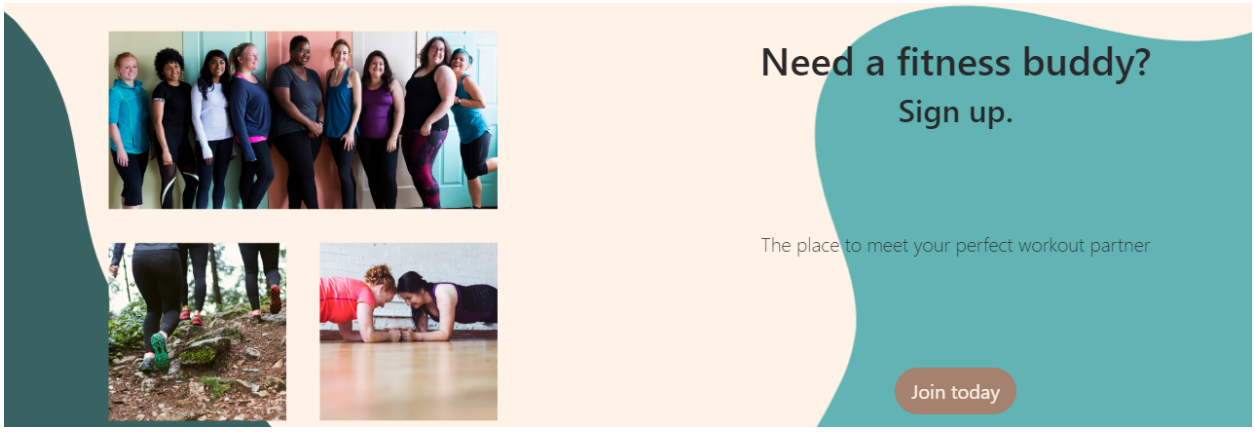



# Workout Buddy

CFG Nanodegree Project

—

Lynsay Dunne
Sydney Humayun
Maya Scott
Chekira Willis

# Overview

Our project is called Workout Buddy. It is a social media app/website built using Flask, Python, and MySQL to manage our user databases. HTML and CSS using Bootstrap were also used to integrate the front end, as the team felt it was important to have a fully operational application.

The aim is to give any person access to the home page, the about page and the disclaimer page. Other pages that everyone has access to are the workout ideas/exercises and Smoothie Recipes page using the respective api connections. We also created a SQL database to have motivational quotes randomly generated on every page visited. The user can also choose to join Workout Buddy, or try to find a match, which will automatically bring them to the sign in page, with a join up option.

Once the user has joined and signed in, they then have access to the restricted users only pages. These pages include the Find a Buddy matching page, where they can get a random match with a fellow user, check the matched users public profile page and send them a message if they want to connect. The users only section also has a personal profile page for the user to update their profile information, a messages page where the user can check past messages and send new ones and a sign out page.

# Background

In today's busy world, it can be difficult for women and non-binary people to find motivation to work out, keep fit and healthy, and to find like-minded friends who are willing to help you reach those goals.  A study by Kean University found that when less active individuals interacted with others who exercised regularly, they felt encouraged and exercised more. This also had positive improvement on their mental health, finding that many of the participants found group encouragement helpful and held a level of accountability that they lacked when exercising alone [1].

We wanted specifically focus on women and AFAB non-binary people due to safety and security concerns in regards to potentially meeting strangers via social media. The focus on women and non-binary was also with an intent to aid anxiety or self-esteem issues that many women have, to create a supportive, safe environment without the fear of judgment. A recent systematic review found that there is an increase of social media use in those with anxiety and self-esteem issues to find the connection they find difficult in the physical world [2]  The idea is that users do not necessarily have to meet in-person (although obviously they can if they choose), but can also use it as a virtual support and motivational tool.

The main aim of our web application is to make fitness more accessible regardless of your current fitness level or experience. It will be called 'Workout Buddy', and inspired by other applications such as Meetup[3]. We have chosen neutral colours in calming tones; the intention is to project a calming, approachable feeling. Drawing on our own experiences, we wanted to include motivational quotes, randomly generated exercises, a way to connect with other users to motivate each other.

# Specifications and Design

## Technical and non-technical requirements:

**Landing page:**

We decided that our landing page should be a simple visual introduction to what our web application is about. A photograph would show women exercising together, and a very brief tagline encouraging people to join and sign up.

**User registration, and login:**

The login and registration pages would have text fields for user input. Registration would require a name, Email address, and date of birth. The submit button needs to initiate the SQL query to check whether the user exists (by checking for a matching Email address), and simultaneously set off an SQL trigger to copy the new user into our user data table. An age verification also happens at the point of registration to make sure the user is over 18 based on the date of birth entered, otherwise access is denied. The login page provides two input fields for Email address and password, and uses SQL commands to check for matched passwords.

**User profiles:**

We wanted to provide users a way to update their public profile information, using get/post to both view and submit the update. We wanted user profiles to be viewable to other users, and to allow for users to match each other, as well as have the capacity to send each other messages.

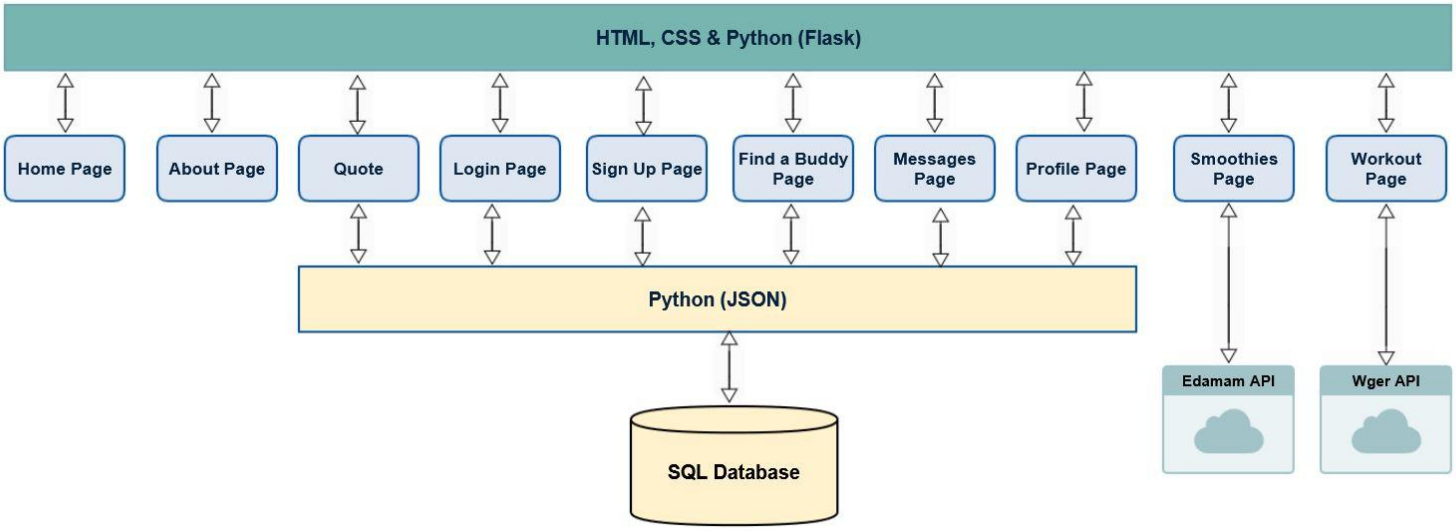
**APIs:**

We also wanted to include exercise examples for users who may not have much experience with exercising; users would be able to search and find a selection of exercises to practise. We used wger.de API to provide a search to integrate into our website. We also used edamam.com to find a smoothie endpoint API in order to allow users to find a quick recipe for a smoothie.

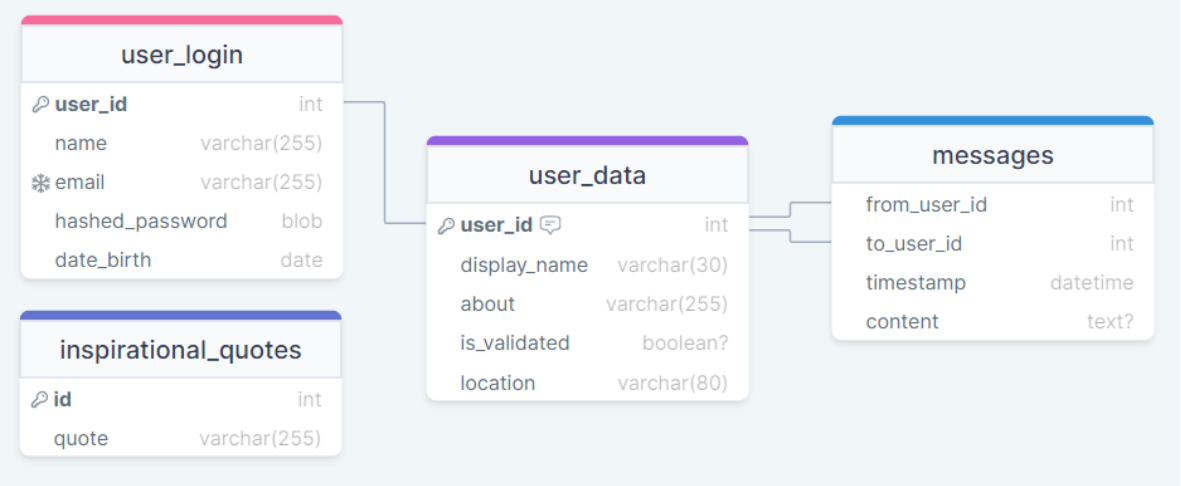
## Database:

A SQL database would be used to store all of our user data, as well as integrating our inspirational quotes database. We decided to have user login information and public user information in separate tables to provide an extra layer of protection for user details. User passwords would be stored in the login table, hashed, using bcrypt. Our user_data table would store public information and would be easily updated by the user. As we wanted to include an inspirational quote on every page, we built our own table of quotes and used Python and MySQL connector to draw on it. This allowed us to be in full control of the quotes that appeared to make sure they were appropriate for the target audience.

## System Architecture Diagram:




## Database Diagram:



# Implementation and Execution:

## Development approach/Team member roles:

Our development approach attempted to follow the Agile approach. We had a once a week sprint ceremony where we discussed the aims for the week and what had been completed so far. We also included daily written communication about our tasks, asking for input from team members as needed. Most evenings during the week, we held a daily scrum to more reliably inform each other of what we were attempting to accomplish and what challenges we may face in trying to get tasks done. While we understand that an average scrum sprint cycle is longer than a week, it made more logical sense to our group to view each week as its own cycle due to the limitations of the project. Another limitation we identified is that due to the size, skill, and criteria of the project all members had to at times step into each scrum role (product owner, developer, and business stakeholder). We used Trello Board as our project management tool https://trello.com/b/bWNxisWA/cfg-project-task-board. Below is a description of each team member's contribution to the project as a whole:

Maya created the original database design. They also did the front-end HTML/CSS code, using prior experience and Bootstrap. Using SQL, Maya wrote the code for the user_login and user_data tables, and a trigger to enable automatic insertion of user details into the data table upon user registration. They wrote the base of the app file for viewing pages that the team would be adding their functions to, the config, and the SQL connection. Using Python and Flask, Maya wrote the code for: user management, login, registration, updating profiles, a date of birth check, a smoothie API search function, and user management.

Sydney's contribution to this project included: organisational tasks such as upkeep of google docs, notes, and trello boards. This also involved clearing of Trello board backlog. Participated in various group sprints/meetings (daily scrums) on topics including initial project design and implementation of decided features. Writing of code for app's matching feature which was integrated in various portions of the final code as well as contribution of html code to front end.

Lynsay created the quotes database using SQL and associated code in python. This was then moved to the front end to display on every page with the team's help. There was also a lot of research on the best way to integrate the chat/messages feature. Final decision was to create a SQL server message database and use my-sql connector in python to integrate. The team helped with the front end integration, and adding the option to message to the matching page. Lynsay also finalised the git ReadMe file with contributions from the whole team and worked on testing.

Chekira's contribution to the project included the design, implementation and testing phases. Chekira worked in an agile methodology and made changes along the way to tackle

problems. The initial idea was based on research and personal experience, Chekira then designed the website through Canva, she made her choices based on current trends in society and what would keep the design innovative. For the implementation phase she worked on the API that would generate random workouts, this was important for the idea of the program in order to cater to people from various fitness abilities. The testing phase was to ensure Chekira was getting the right output from the API.

**Tools and libraries:**

The main tools we used for our project were MySQL Workbench and Python. Python is ideal as it comes with so many built-in libraries and compatible tools. The ones we decided to use were mysql-connector, Flask, timedate, functools, requests, random, and bcrypt. For our front-end work, we decided to use HTML, CSS and Bootstrap.

## Implementation Challenges:

During our decision and design phase we discussed many ideas and avenues that this app could travel down. We considered pushing the idea of users meeting physically and the challenges that we would face using a model like that. Some concerns were liability (if a meet-up went wrong, who takes responsibility?) and necessity. After debating and completing a costs-benefits analysis we ultimately decided to focus on motivation and electronic means of communication as we have access to scientific research documenting how effective online friendships can be in combating stress and feelings of loneliness[3,4], and felt that this was justification enough to not push users to meet in person. It allows users to meet more freely with people from all walks of life. These changes are reflected in several of our Trello board cards as we had fully discussed the ramifications of each decision.

Our team also faced challenges during the execution process. Our most difficult to implement feature became the chat function in our app. Initial research pointed us towards having to create a separate client/server option and this was heavily researched, along with chat API's, but there was an incredible complexity in the client/server option that was beyond our skillset and significant cost involved with the chat API's. The final decision was to incorporate a much more understandable messaging system using a SQL server message database and python to integrate and assure that messages remained between two users only and users had the ability to recall prior messages and to send and receive new messages.

Another challenge we faced was the integration of code between team members, and between front end and back end. As a group we decided that we wanted as close to a fully operational application as we could get in the designated time. Fortunately, one of our team members had previous experience in working with html and CSS, so our frontend was

created with relative ease, but we lacked the experience in connecting the two sides. Through some trial and error we were able to get everything communicating properly. In regards to the integration of code between users, it took some time to discover that the messaging system and much of the front end needed to use parts of the initial matching functions individually. Once we got the messaging feature to work, some of the matching code was retired, however in further releases we would like to expand on the matching feature to integrate some of the unused potential.

Further challenges our team faced were organisational in nature: schedules, assigning of tasks, and real-life difficulties unrelated to the direct task. As a group, we discussed each roadblock as it emerged and solved most of them through constant, clear, and honest communication. These discussions primarily took place within our daily scrums and weekly sprint ceremonies.

# Testing and Evaluation:

## Testing Process:

Testing is a vital part of software development to assure function, reliability and that as many use scenarios are covered to ensure a satisfactory user experience. We created a specific directory named 'tests' and the testing incorporated the users portion and the api. Unittest was used for all the testing.

The api testing called on the api connections to the api endpoints within the api directory. On the first test a specific workout that was available was used to check connection and return. On the subsequent test, a category was chosen to ensure appropriate return with a different search parameter.

For the users section testing we had to set up a mock database connection and a mock database cursor. This was used as we didn't have a test database to use and we had to use this mock process to initiate the database connection and cursor function. We completed 7 tests in all, each checking measures put in place to avoid missing data, email availability and profile accessibility.

## System Limitations:

Our project's main limitation is that as our project is running from a local network and reliant on Flask there are security risks inherent to both of these structures. There is little we can do to mitigate these security concerns at the current level we are working at. We have included a 'validated' column in our user data table, as our idea was to have a way of app moderators verifying users were who they said they were, but the implementation of this, at this stage in development, we were not able to complete.

Further limitations on the project are related to time constraints. We would like to include several user safety sections to our app, as well as sections to further motivate and help users on their health and fitness journey. Some examples of safety features include: email verification, chat monitoring for inflammatory, demeaning, or rude behaviour, bot exclusion, reporting form,  and reminders to keep personal data safe from strangers before chats initialise. Possible options for furthering users in their health and fitness include mindfulness approaches, calorie calculators, or forums and communities.

## Conclusion:

Our original aim was to create a web application that allowed users to register, search recipe and workout APIs, and connect with other people to motivate and support each other on their fitness journey, regardless of their experience level.  We believe that we achieved these aims through the use of several API's, user databases, HTML in our frontend development and Python in our backend development. Further development may include the addition of more functionality to our app through the inclusion of communities, chat monitoring and further overall fitness and health help.

[1] Rackman, A: 19-Oct-2022 – "Friends could be key to finding fitness motivation" [Online]

Available From: https://www.bbc.co.uk/news/health-63321248 - [Accessed 19-Oct-2022]

[2] Emily B. O'Day, Richard G. Heimberg, 2021. Social media use, social anxiety, and loneliness: A systematic review. Computers in Human Behavior Reports, 3

[3]  Henderson, A. and Bowley, R., 2010. Authentic dialogue? The role of "friendship" in a social media recruitment campaign. Journal of Communication Management, 14(3), pp.237-257.

[4] Vallor, S., 2011. Flourishing on facebook: virtue friendship & new social media. Ethics and Information Technology, 14(3), pp.185-199.

[5] Winstead, B., Derlega, V., Lewis, R., Sanchez-Hucles, J. and Clarke, E., 1992. Friendship, Social Interaction, and Coping With Stress. Communication Research, 19(2), pp.193-211.