Department of Computer Science and Engineering
University of Notre Dame

CSE 40746 - Advanced Database Projects
Spring 2023

Final Project

Group 4 - Project Abstract

Benjamin Egloff, Thomas Deiser, Jason Brammer

GainzMaster is a fitness assistant application that will change the fitness game. Although the world of working out is extensive and multifaceted and can be extremely rewarding, getting started is often considered the hardest part. For years new members of the fitness industry (also known as “gymigrants”) struggled to find their footing, as the gym can be a very intimidating place at first. Often times the biggest issue is lack of direction, having no plan once they get to the gym. At GainzMaster we plan to attack this issue with our “workout catalog” feature, which is a database where we have a catalog of pre-built workouts stored. These workouts are sourced from knowledgeable sources, just the guidance people trying to get into fitness need. Furthermore, another feature that would be useful to this group would be GainzMaster’s “muscle catalog”. One piece of data individual workouts in the database will have is what muscle it works. If the user wants to learn more information about the particular muscle, they can find it in this catalog. Some of this information would be its muscle group (i.e. triceps, quads, etc.) as well as other exercises that work this muscle. Finally, similar to the muscle catalog, this application will provide an “exercise catalog” functionality. This catalog will allow users to search for exercises by muscle or muscle group. All-in-all, GainzMaster is an begapplication that would be of great interest to any beginner!

What makes this application special is the fact that although it is helpful for people with little to no experience in the gym, it’s even more useful for those with a background in fitness. For instance, the “muscle catalog” feature would turn any intermediate’s head as they reach for any edge they can get on their progress path. In addition to this most members of the fitness community would agree that one of the least enjoyable aspects of fitness is the occasional burnout, getting bored with your fitness routine. As one can imagine, although working out is rewarding, doing the same exercises in the same order on the same day every week can definitely deter someone from being their best in the gym. This is where GainzMaster’s “randomize-a-workout” feature fits into the equation. This feature, given a series of details about what the user wants from their workout, returns a semi-random sequence of exercises that achieve this goal. For example, if a user asks for a high-volume (higher amount of reps at a lower weight) push (tricep, chest, shoulder targeted) workout our application would return a random series of exercises that achieve these requests. Therefore on the first request, the user might get something along the lines of “Bench Press 3x12; Tricep Pushdown 3x15; Shoulder Press 2x20”, and on the second perhaps “Tricep Kickbacks 3x10, Shoulder Flies 2x15, Incline Bench Press 3x10”. For reference, these two workouts work for the same three muscle groups through different exercises, rep ranges, and order. Although GainzMaster is certainly tailored more for lifting-style fitness, the two features mentioned in this paragraph would greatly improve anyone’s fitness lifestyle.

The main functional requirements for this application consist of curating queries, fetching the queries from the database, and presenting these queries to the user. The first three functionalities, the muscle, workout, and exercise catalogs, will be very similar.

These functionalities will simply take inputs from the user, and find rows that match these details. As for the workout catalog, this is just about the end of the line. This feature will simply print the workouts in each workout. However, the muscle and exercise titles will print out the titles of the muscle/exercise, at which point the user will select which muscle/exercise they want to see more information for. At this point, the program will take this input and print the information that goes with it. As for the workout generator, this program will take inputs, select a user-defined number of random exercises that match these inputs, and print each exercise out alongside the number of reps and sets for the particular exercise. Overall, we don't expect the functional requirements to be too difficult, as much of it is simply querying the database for data the user requests.

Considering our application is relatively small-scale for now, we have a fairly short list of non-functional requirements that we would like to achieve. First, we would like each of the catalog accesses to be completed within a second to cater to the user. The exercise and muscle catalogs should be capped at about a few thousand entries with the data we have available, so we believe this won't be too much of a challenge to complete for accessing those two catalogs. As for the random workout generator, selecting 3-7 or so workouts should also not be too time intensive. The only issue which we can foresee with timing is if the workout catalog grows too big from too many users, but it would likely require massive amounts of data (in the millions) to make a query take more than 1 second, which we don't foresee being an issue.

The second non-functional requirement is protecting user information. We plan on creating a user login system, meaning we will have to store encrypted passwords or outsource our user creation to a third-party system like Firebase. We are still unsure of which direction we will like to go through with, but ensuring user passwords are not accessible is extremely important, and we will make sure we fulfill this requirement. If we do use sqlplus to build our user base we will likely need to create triggers to encrypt passwords upon user creation.

The support requirements for our project require very little work on our end to achieve by way of the technologies we will use to create our project. We would like to build a system that is constantly available, easy to use, and reliable. Fortunately, using sqlplus as our backend will make queries reliable and fast, while running the project on a vm will allow for the server to be constantly available. As for making the website easy to use, that will essentially boil down to our frontend skills and our ability to leverage html and css properly. To achieve this goal, first, we will need to limit how much the user interacts with the database. Using forms and buttons to simplify how much the user can manipulate will ensure that users do not have too much free reign on the database, while also ensuring they can complete tasks easily. In all, we intend on creating a reliable product that will be open to user requests constantly.

Users of our project will be able to access a vast amount of muscle and exercise information in a highly digestible way. First, users will be able to select specific muscle groups and learn some preliminary information regarding the muscle like its scientific

name, how it contracts, etc. Second, users can view individual exercises to learn information and perhaps a video or picture on how to execute the exercise safely and properly. Third, as mentioned earlier, users will be able to create pseudo-random workout plans based on instructions that they provide to the system like muscle group, mechanics (push, pull, legs), or utility (bodyweight, machine, free weight). Fourth, in order to keep consistent with our goal of novelty in workouts, we hope to provide a random exercise button that will give users either a completely random exercise or one consistent with their desires. Fifth, and finally, we would like to make a workout log that allows users to see their previous workout plans for future use.