Department of Computer Science and Engineering
University of Notre Dame

CSE 40746 - Advanced Database Projects
Spring 2023

Final Project

Group 4 - Progress Report

Benjamin Egloff, Thomas Deiser, Jason Brammer

Our project is moving along at a good pace to ensure our ability to finish on time and submit a satisfactory deliverable. The main time sink of our project so far has been securing an appropriate amount of exercise and muscle data to fill our sqlplus database. We created a scraper to collect detailed muscle information and an extensive list of over 2400 exercises complete with mechanics, force, and targeted muscle for each individual exercise. Our scraper design took html and parsed it with regex and the python BeautifulSoup module to create three csv files, general_muscles.csv, detail_muscles.csv, and exercises.csv. Following the completion of the web scraper, and the creation of our three main csv files, we quickly made .ctl files to insert our data into a sqlplus database using sql loader.

Once we had secured our data and made sure it could be inserted into sqlplus, we focused on developing our backend REST API interface. Since we wanted to take advantage of the simplicity and power of a vue.js app, we decided to forego using php as our backend interaction software and opted to receive data through GET requests and modify the database using POST requests. This problem was quite difficult to solve unfortunately, as it required a lot of manipulation of the httpd.conf file to allow the apache server to run a python file that imports the external cx_Oracle module. Nevertheless, after some struggles with installing pip3, globally installing oracle client libraries, and looking for solutions on stack overflow, our group machine provides a REST API endpoint for communicating with the sqlplus database over network communication via GET and POST requests.

Following the completion of our backend-focused tasks, we created a blank vue project to start our front-end development. To ensure guarded and quick navigation of our webpage, we decided to use the vue-router with some appropriate navigation-guards to eliminate unauthorized access. In addition, we created a vuex store to limit the amount of database queries and limit the amount of waiting time between page switches. As for the individual pages of our web app, we've made a considerable amount of progress quite quickly. At this point we have three of our seven pages completely functional, two others almost complete, with two pages still to be created. The fully operational pages are our sign-in page, create-an-account page, and our muscle catalog page. The sign-in and create-an-account pages are very similar, so it didn't take too much work to complete the other once the first was finished. To ensure secure authentication, we opted to outsource user creation and login to firebase**.** Using firebase as our authentication endpoint, we were easily able to create users and login as such users with few lines of code, while also ensuring the security of a google based service.

As for the muscle catalog page, we were able to aesthetically display each muscle name, muscle group the muscle belongs to, an image of the muscle, other names for the muscle, and related muscles. In addition to this, clicking on any entry in the table brings the user to an endpoint showing a more readable version this information, as well as a brief description. This allows users to gain a better understanding of the muscles they are exercising and trying to improve on. The two pages we have in progress are the home landing page and the exercise catalog. The home landing page serves as a welcome page for users after signing in, greeting the user and displaying an inspiring quote, giving the user a random workout, and a log of their last

workout. The last two features can still use some tuning from a middle-end standpoint, querying data from the database. The workout catalog is nearly finished, missing only search feature. We believe this page is very close to being completed. The final two pages left to be created would be the create-a-workout page, as well as our log page. We expect these pages to be very manageable, as these functionalities will already be ironed out to an extent on the home page.

Despite the substantial amount of progress that has been made so far, we want to use this momentum to complete the remaining significant tasks. These include adding finishing touches to the muscle catalog and individual pages. As mentioned, we also want to touch up the design of the home page and exercise catalog a little more and fix a few database dependencies and fields that did not get inserted correctly before the final design deadline. In addition to another catalog, we will continue working on creating a workout log and an automated suggested workout for the user. We wish to include a preferability scale on every exercise so the workouts that are more favorable to the user will get suggested more often. Another feature we are planning to add in is a page where the user can log their workout for the day. After these features are implemented, the only thing left we would need to focus on is fixing any bugs or overall issues and protecting against SQL injection attacks.