CSCI3020U – No Brand Team

Ronald Sin, Jayson Sandhu, Alexander Minz

Nathanial Armogan, Matthew Witvoet

Contents

[Topic 2](#_Toc69732653)

[Design 2](#_Toc69732654)

[Angular 2](#_Toc69732655)

[Express 2](#_Toc69732656)

[Requirement Fulfillment 2](#_Toc69732657)

[Independent Study 3](#_Toc69732658)

[Installation 3](#_Toc69732659)

[Contribution 3](#_Toc69732660)

[Citations 3](#_Toc69732661)

# Topic

For our CSCI3020U final project we decided on creating health and fitness app. It would include simple data implementations, generic CRUD features such as creating and reading data values such as weight and heart rate history. We also wanted to include a Google Map integration that would allow the user to enter and save routes for planning fitness runs.

# Design

For this project we decided on a two-server approach. The front-end is an Angular based client facing server and the back-end is a NodeJS Express based server.

## Angular

There is nothing noteworthy about the implementation of Angular. It’s version 11 Angular running with compatible typescript.

## NodeJS Express

The back-end server is a NodeJS Express framework, running over a Sequelize ORM, wrapping a PostgreSQL database.

# Requirement Fulfillment

|  |  |
| --- | --- |
| Requirement | Fulfillment |
| SVG | Multiple D3 graphs throughout bloodpressure.componets.ts, etc. |
| HTML | \*.components.html |
| CSS and CSS Frameworks | Bulma is used throughout the Angular app |
| D3 | Multiple D3 graphs throughout bloodpressure.componets.ts, etc. |
| JavaScript, jQuery | NodeJS server, \*.components.ts |
| DHTML | Use of dynamic Bulma assets |
| AJAX, web services | Google Maps integration routes.components.html/ts |
| Node.js | NodeJS back-end server |
| Client-side framework | Angular 11 |
| Database | Postgresql with Sequelize ORM |
| Sockets, multi-threading | Dropped |

# Independent Study

For our independent study we decided to explore and integrate JWT (JSON Web Tokens) into our application for server authentication and database querying. We also decided to deploy our NodeJS server on a cloud-based service, in this case we decided on Heroku due to it’s easy-to-use nature and it’s free option.

<https://docs.google.com/presentation/d/1-DSLPeA1waVdRY2TWw0uvydcXP6MTRHXIz-_HWTuqc4/edit?usp=sharing>

# Installation

## Angular

## NodeJS Express

# Contribution

* Ronald Sin,
* Jayson Sandhu
* Alexander Minz
  + JWT Implementation for Angular and NodeJS
  + Created and implemented all elements of NodeJS server.
  + Created and deployed PostgreSQL database with Sequelize ORM.
  + Deployed NodeJS server and DB onto cloud based Heroku web service.
  + Integrated back-end calls, authorization, and data piping for Angular app.
  + Setup Angular app routing.
  + Created Angular components for signin, register.
  + Documentation.
* Nathanial Armogan
* Matthew Witvoet

# Citations

* https://bezkoder.com/angular-11-jwt-auth/