

CSC 667 - Term Project

Milestone Two

Team name: JJRV

Team Members

Vern Saeteurn

Jessica Serrano

Jair Gonzalez

Rodrigo Gallardo

GitHub Repository Link

<https://github.com/sfsu-csc-667-spring-2021-roberts/term-project-team-jjrv>

Application Link

<https://project667.herokuapp.com/>

Application with Test Route

<https://project667.herokuapp.com/test/>

Complications & Solutions

When setting up the database, we realized that our machine did not have all of the applications required to get the database running properly. During our team meeting, we took the time to make sure we installed them; applications such as Node.js, Postgres, etc. Creating the database took a little researching, as we were not sure what the correct command syntax was. Found correct syntax after looking at postgres documentation.

Some of our team members ran into some minor trouble setting correct credentials to access the database in our .env file. Found solutions online to correctly format the URL so that we can input user and password within the file.

Our team ran into some trouble while trying to deploy our application to Heroku. When we tried to run the app, we were notified that the build was successful but it also gave us a wall of errors. For example, when we tried to push our code to Heroku, it gave us an error saying the SSL was off. To remedy this, we tried a solution that was posted on Discord by another classmate. However, this gave us another error that said we were using a “self-signed certificate”. We managed to get Heroku to build successfully, but it was still not able to connect to our database.

After Several attempts on trying different things from Stack Overflow, we reached out to some colleagues for solutions. We found that several steps had to be done prior to successfully configuring access to Heroku postgres.

The first thing we had to do was to make a modification to our config/config.js. We added a dialect option for SSL, specifically setting rejectUnauthorized to false.

The next thing we had to modify was our database connection parameters inside of our db/index.js. What we did was include the line: `process.env.NODE_TLS_REJECT_UNAUTHORIZED = "0"`.

Finally, we had to modify a setting in the Heroku configurations. We used the command “heroku config:set PGSSLMODE=require”. After following all of these steps, our Heroku postgres was correctly connected to our project.

Team Development Setup

All members of our group have access to the same Github repository which we will be using for deployment on Heroku. Members also have access to the project deployed on Heroku and are free to make any changes. For Team development, we have created 3 additional branches to our github repository labeled frontend, backend and migrations. Each one for its individual purpose on the project. All users have access to these branches and are able to create additional branches as needed. Development in certain portions of the project will be made in the appropriate branch in which that portion of work will be done, then one team member will be able to merge into our main branch which will hold all the code that will be deployed on Heroku.