Group E

CSC667

Milestone 2: Deployment and Migrations

Calvin Ip, Jerry AuYueng, Jonah Manarang, Steven Huynh

Git Repo: https://github.com/sfsu-csc-667-fall-2017/term-project-csc667_auyeung_huynh_ip_manarang

Heroku Url: https://love-letter-csc-667.herokuapp.com/tests

During this milestone we experienced a decent amount of problems that led to this milestone taking longer than anticipated. One of the problems that everyone on the team encountered was setting up the DATABASE_URL correctly. In the documentation for Milestone 2 the instructions say "`echo DATABASE_URL=postgres://`whoami`@localhost:5432/DATABASE_NAME >> .env`" however this led to a problem with authentication. A problem that some of our members had was that their user for postgres wasn't authorized or had the correct permissions so we had to make sure that their role got upgrated to superuser. Then we ran into the problem with the echo instruction listed above. When we tried to run the "db:migrate" command it said that our user was not authorized. So we found out that the syntax for the command should be something along the lines of "`echo DATABASE_URL=postgres://`whoami`:password@localhost:5432/DATABASE_NAME >> .env`" where we put our password to the database next to our user. This ended up working and allowing us to migrate successfully. Another major problem we had was actually getting access to the database using index.js. We tried following the examples you gave us and made a test.js file in the routes with the contents you gave us but that did not seem to work. We were getting errors such as "db.any is not a function" and some connection errors. I was also getting errors with my insert command but that was syntax related and had to do with the single quotes and double quotes as well as the ` symbol. We fixed this problem by adding the get("/tests") function directly into the index.js file. We also connected to the database in the index.js file by a different means, by creating a variable that created a new client with pg and connecting to it. This also means we had to use db.query as opposed to db.any since db.any didn't seem to work. Another problem we had was once we deployed it. We were having trouble finding the user and password values for the heroku specific database but after replacing these things it seems to work.

For team development we are planning on weekly team meetings and using scrum. Scrum will be used at these weekly meetings for us to keep track of what pieces of the project we need to get done and need to get prioritized. We currently have a slack channel and a group text where we update others in our group when we have made changes and when we have problems while we are not able to meet. For programming we are planning on utilizing our Github to all mutually work on the project together, each of us having a separate branch while only pushing to master branch after the team has agreed on something. Also only one person will be in charge of updating and deploying to heroku so that we can update and deploy in an orderly fashion.