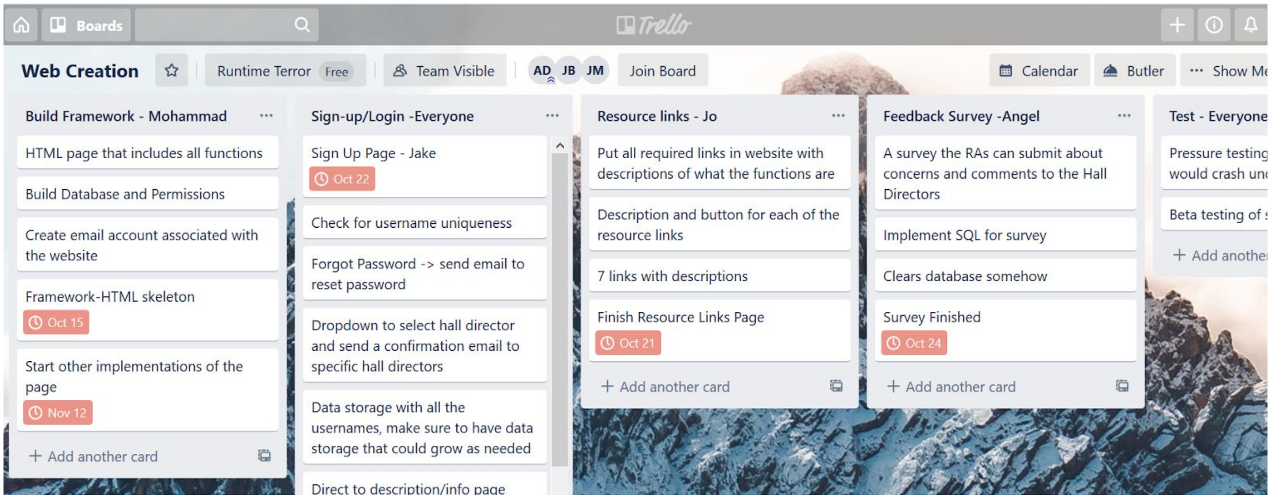
Residence Hall Staff Portal

Jake Bonner, Angel Dong, Josephine Martin, Stefan Primelles, Mohammad Ozaslan

Project Tracker (Trello)

- https://trello.com/b/gPjM5Z8Y/web-creation



Build Framework - Mohammad

- HTML page that includes all functions
- Build Database and Permissions
- Create email account associated with the website
- Framework-HTML skeleton Oct 15
- Start other implementations of the page Nov 12

Sign-up/Login -Everyone

- Sign Up Page - Jake Oct 22
- Check for username uniqueness
- Forgot Password -> send email to reset password
- Dropdown to select hall director and send a confirmation email to specific hall directors
- Data storage with all the usernames, make sure to have data storage that could grow as needed
- Direct to description/info page

Resource links - Jo

- Put all required links in website with descriptions of what the functions are
- Description and button for each of the resource links
- 7 links with descriptions
- Finish Resource Links Page Oct 21

Feedback Survey -Angel

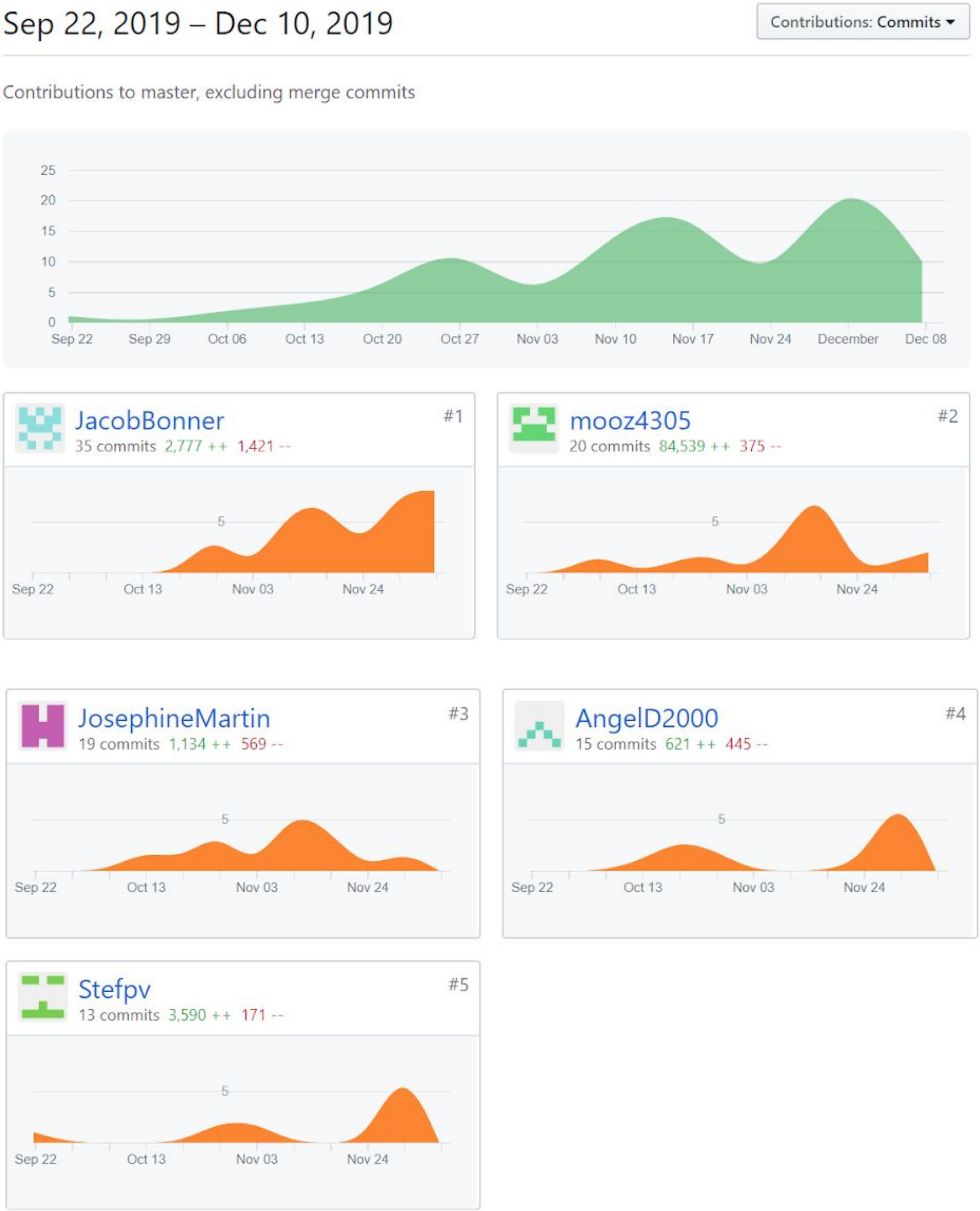
- A survey the RAs can submit about concerns and comments to the Hall Directors
- Implement SQL for survey
- Clears database somehow
- Survey Finished Oct 24

Test - Everyone

- Pressure testing would crash un
- Beta testing of s

VCS (GitHub)

- https://github.com/Stefpv/SoftDev_ProjectCode
- https://github.com/JacobBonner/csci3308-Team-Meeting-Logs
- https://github.com/mooz4305/csci3308-Milestone-Submissions



Sep 22, 2019 – Dec 10, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

JacobBonner #1
35 commits 2,777 ++ 1,421 --

mooz4305 #2
20 commits 84,539 ++ 375 --

JosephineMartin #3
19 commits 1,134 ++ 569 --

AngelD2000 #4
15 commits 621 ++ 445 --

Stefpv #5
13 commits 3,590 ++ 171 --

Deployment

Our app was deployed on local host. We decided to do this for simplicity and due to security concerns and time constraints. For the app to be deployed the user would need to:

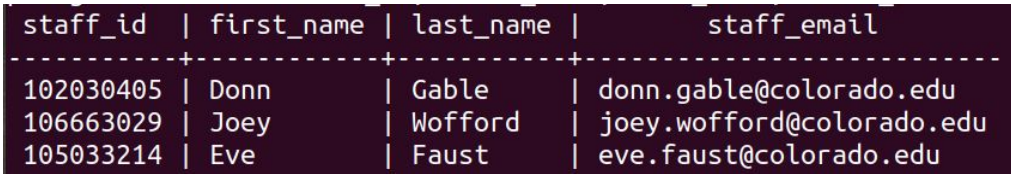
- download the entire repository from:
  - https://github.com/Stefpv/SoftDev_ProjectCode
- install PostgreSQL (if not already installed),
- initialize database using fullDataBase.pgsql file, and start database server:
  - sudo -u postgres service postgresql start
- update database name and password in node.js to match database server,
- run NodeJs on server.js, and finally open a browser at:
  - http://localhost:3000/homepage.ejs

  to access the website.

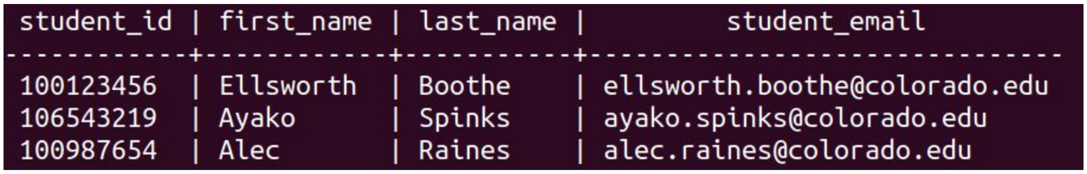
Using the App

In order to use the application, the user needs to login on the home page with their account credentials. However, if the user does not have an account, they would need to sign up first.

All users would need to be an Hall Director/RA and their information should match the Hall Director or RA database stored locally. Running fullDataBase.pgsql creates the following tables:



```
 staff_id  | first_name | last_name |       staff_email
-----------+------------+-----------+--------------------------
 102030405 | Donn       | Gable     | donn.gable@colorado.edu
 106663029 | Joey       | Wofford   | joey.wofford@colorado.edu
 105033214 | Eve        | Faust     | eve.faust@colorado.edu
```



```
 student_id | first_name | last_name |         student_email
------------+------------+-----------+-------------------------------
 100123456  | Ellsworth  | Boothe    | ellsworth.boothe@colorado.edu
 106543219  | Ayako      | Spinks    | ayako.spinks@colorado.edu
 100987654  | Alec       | Raines    | alec.raines@colorado.edu
```

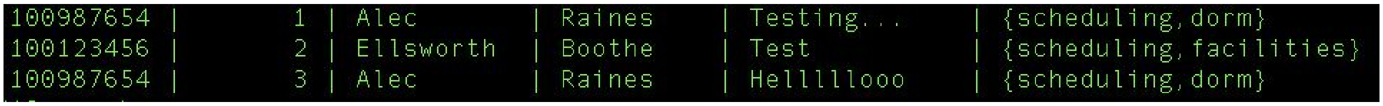
Once the user is in the system, they can use the navigation bar to access all pages in the application, it contains buttons to the Staff, Profile, Resources, and Survey pages. For example, the user can click the Resources button in the navigation bar and go to the resources page, where the links to the websites of interest are stored. On the resources page the resources are dynamically created using the database. This allows the user to add resources that might become useful in the future to the site by filling out the add resource form with the name, links and description of the site.

The user can go on to the survey page and fill in a feedback form. The profile page contains a form that the user can fill in to add some details about them, such as a bio, a picture, their preferred name, and gender identity. The user can navigate the staff page to see the other RAs in their hall or in the housing system. This page displays each RA's profile with nicely formatted cards.

Testing

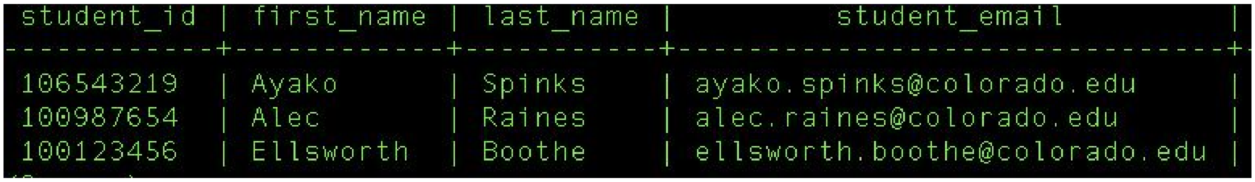
To test the program we inputted sample student id and university emails to the data table and signed them up through sign up page. After logging into our website we then tested the staff profile and staff page to see if updating in staff profile would appear on the staff page. This also tests whether we've correctly implemented our database and server.js that links staff profile and staff page together. The user could also go into the resources page and add a new resource that updates both the database and the Html form to correctly display the added resource. Finally, the user could go into the survey tab and submit comments that update the database and later is able to display using the database.

Survey:



```
100987654 |        1 | Alec       | Raines     | Testing...     | {scheduling,dorm}
100123456 |        2 | Ellsworth  | Boothe     | Test           | {scheduling,facilities}
100987654 |        3 | Alec       | Raines     | Hellllllooo    | {scheduling,dorm}
```

Signup:



```
 student_id | first_name | last_name |        student_email         |
------------+------------+-----------+------------------------------+-
 106543219  | Ayako      | Spinks    | ayako.spinks@colorado.edu    |
 100987654  | Alec       | Raines    | alec.raines@colorado.edu     |
 100123456  | Ellsworth  | Boothe    | ellsworth.boothe@colorado.edu |
```