



Gator Room

Team 103

[Gator Room](#)

The Team



Nico Graves

Renaissance Man, Team 103 Lead.

Creates art, [learn more here](#)



Jakhongir Khusanov

Full Stack Fellow

The person who makes sure that everything works and runs smoothly. [Learn more here](#)



Michael McDonald Tran

Git Master/Full Stack Fellow

Takes care of whatever you give him. [Learn more here](#)



David Carl Hernandez

Front End Fellow

Design and documentation. [Learn more here](#)



Feona Mae Lao Garcia

Scrum Master/Front End Fellow

Makes sure that each member is growing. [Learn more here](#)



Carlos Velasco

Back End Fellow

Loves todo designs and customize the project. [Learn more here](#)



Michael Nelson

Full Stack Fellow

Likes working on backend, assembly and mobile development. [Learn more here](#)



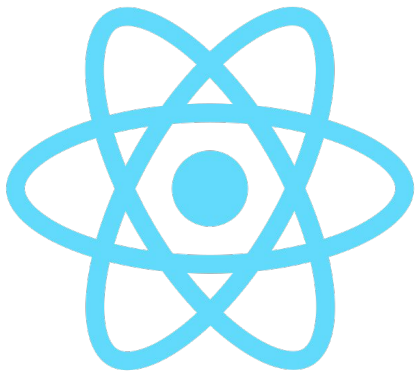
GatorRoom?



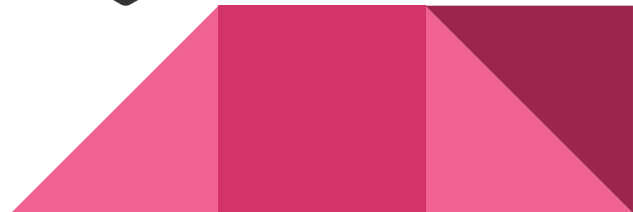
GatorRoom!



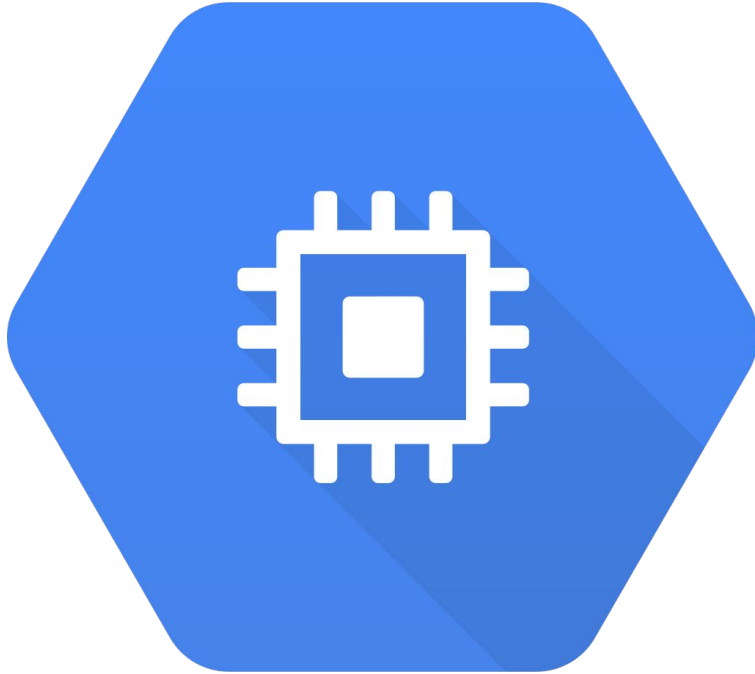
Tech Stack



express



Google Compute Engine - Our Server Choice



Server Machine Specifications

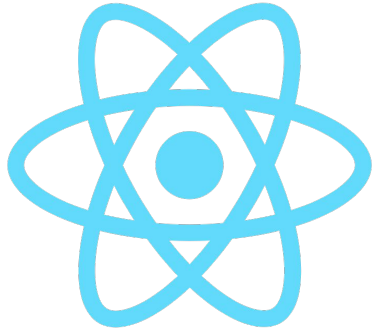
Machine Type: g1-small(1vCPU, 1.7 GB memory)

Boot Disk: Size -10 GB, Type - Standard persistent disk

Installed Programs to Host Site:

Nginx, NodeJS with Express

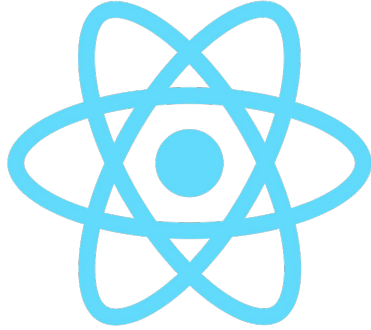
React!



- Modular component structure
- Javascript instead of HTML
- Easy to maintain scale
- GitHub management became much easier
- Work distribution became easier and more organized
- Very fast and smooth
- Reload on save for better workflow
- Industry standard for many companies



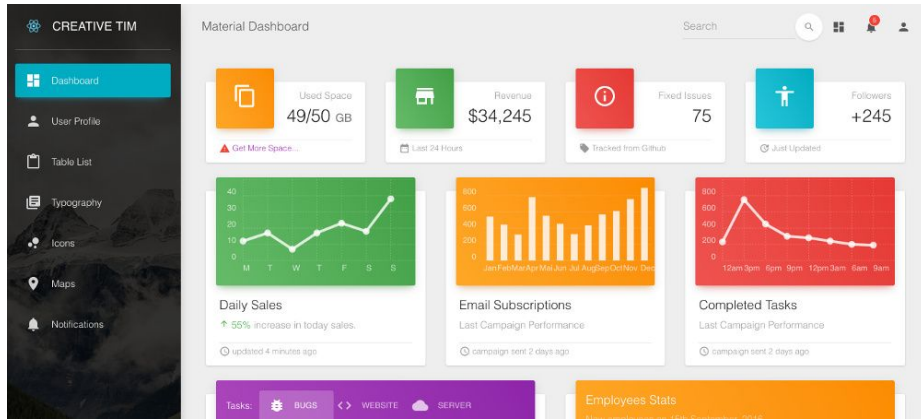
React...



- Challenging to learn initially
- If you don't know JavaScript well it's impossible to learn
- Setup can be a bit of a hassle
- Documentation is there, but it's still a fairly new framework (2017)
- 412 (Software Lab) is child's play compared to this class
- Parts of team had trouble keeping up with both working on school work AND learning a whole new web framework
 - Put us behind other teams who had experience with web frameworks



Material UI



- Quick and easy library for UI design
- Made specifically for react, fast and integrated into workflow of react
 - Match made in heaven, helped our workflow tremendously
- Dynamically resizable
 - Less styling for us to do!
- No loading screens!
- Mobile responsivity made easy
- Industry standard



Material UI



146 Bright St # A2, San Francisco, 94132

\$3420 · 3 Bedrooms · 2 Bathrooms · 1 Kitchen



Laundry

Yes



Parking

Yes



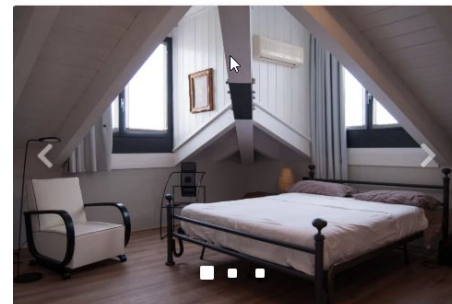
Pets

No



Smoking allowed

No



1905 Laguna St, San Francisco, 94115

\$3750 · 3 Bedrooms · 1 Bathroom · 1 Kitchen



Laundry

Yes



Parking

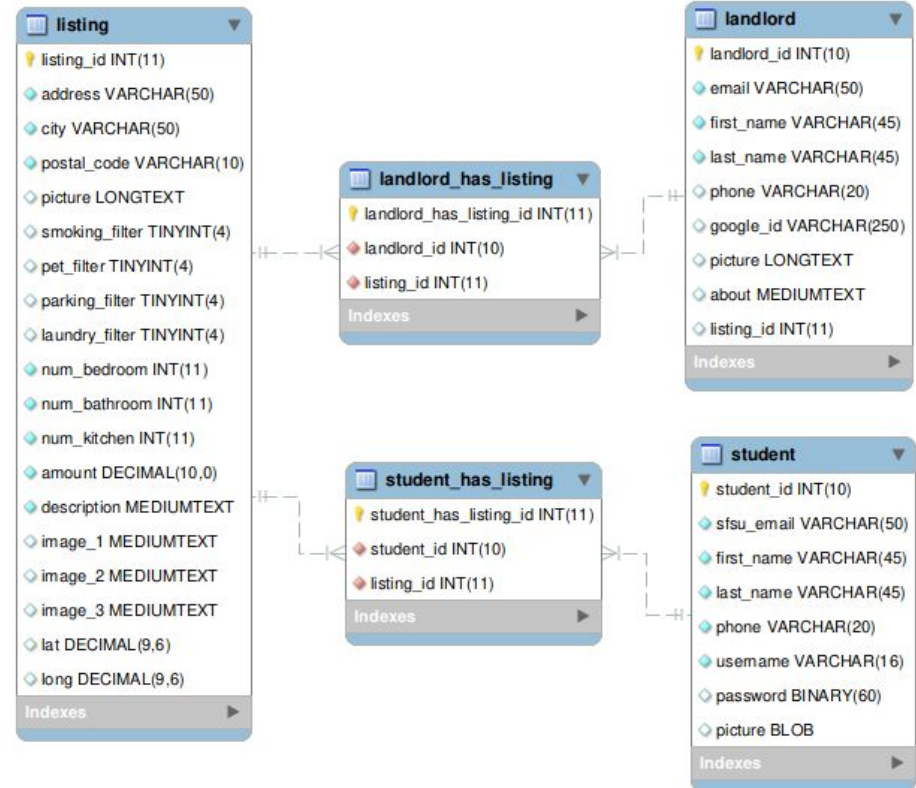
Yes





MySQL - EER Diagram

- Every table is set to **auto increment** to dynamically grow the data of the site
- Coordinates of each listing are generated with the help of **Google's Geocoding API**
- Google Sign In/Up is implemented for Landlords
- Students have to sign up with a **valid SFSU email**



GitHub Management



- Thanks to react we had to deal with very few merge conflicts
- Enforced master and development branch rules
 - No pushing to develop or master, only modifiable via pull request that must be reviewed by other team member and approved
 - Code review on every pull to develop
- CircleCI Linter Tests on every push
 - Enforced coding style
- Weekly(ish) spring cleaning of redundant branches
- We still had a couple hiccups/learning experiences

- Permissions keep any whoopsies from happening
- Enforces code reviews and testing before any major changes are pulled onto develop and master branches

Branch protection rule

Branch name pattern

master

Applies to 1 branch

master

Rule settings

Protect matching branches

Disables force-pushes to all matching branches and prevents them from being deleted.

☒ **Require pull request reviews before merging**

When enabled, all commits must be made to a non-protected branch and submitted via a pull request with the required number of approving reviews and no changes requested before it can be merged into a branch that matches this rule.

Required approving reviews: 1 ▼

☐ **Dismiss stale pull request approvals when new commits are pushed**

New reviewable commits pushed to a matching branch will dismiss pull request review approvals.

☐ **Require review from Code Owners**

Require an approved review in pull requests including files with a designated code owner.

☐ **Restrict who can dismiss pull request reviews**

Specify people or teams allowed to dismiss pull request reviews.

Coding Style

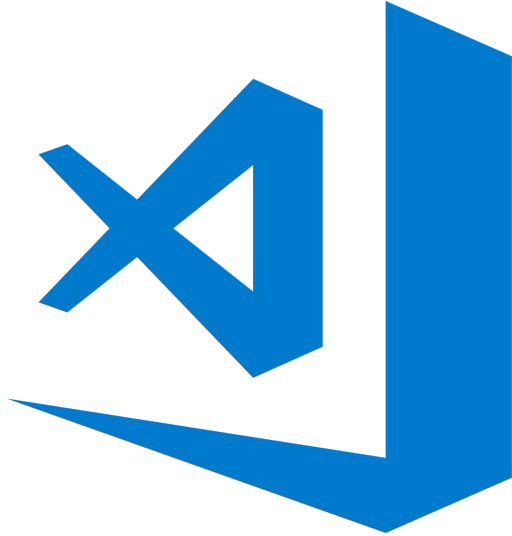


- Used AirBnB's linter rules
- Enforced via ESLint in each of our editors
 - Continuous refactoring of code as per the linter rules
- Comments were not needed in large quantities
- Feature based github commits
- Prettier VS Code extension
- React forces a certain coding style
- Format on save kept code consistent and clean throughout all of development
 - No huge refactoring sessions required

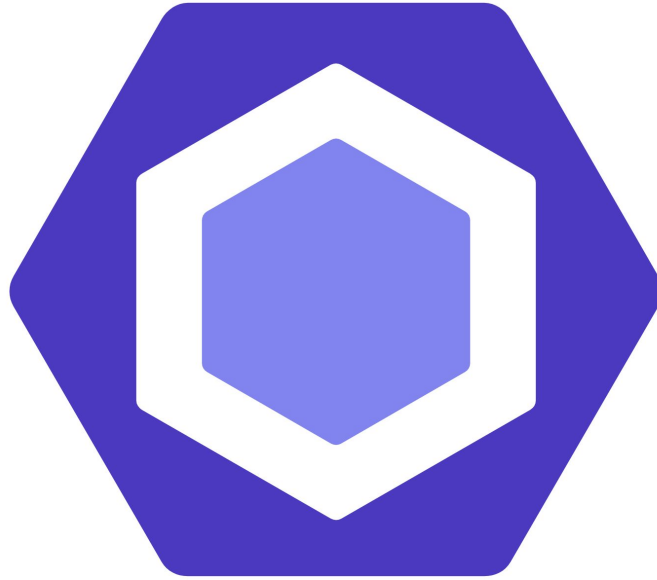
```
</Stepper>
<div>
  <div className={classes.stepperContent}>
    {activeStep === 0 ? <ListingLocationForm /> : null}
    {activeStep === 1 ? <ListingFiltersForm /> : null}
    {activeStep === 2 ? <ListingImagesForm /> : null}
  </div>
  {activeStep === steps.length ? (
    <div>
      <Typography className={classes.instructions}>
        All steps completed - you're finished
      </Typography>
      <Button onClick={this.handleReset} className={classes.button}>
        Add more
      </Button>
    </div>
  ) : (
    <div>
      <Typography className={classes.instructions}>{getStepContent(activeStep)}</Typography>

      <div className={classes.backNextButtons}>
        <Button
          disabled={activeStep === 0 || activeStep === 1}
          onClick={this.handleBack}
          className={classes.button}>
            </div>
    </div>
  )
}
```


Tools - installed on each team member's local machine



VS Code

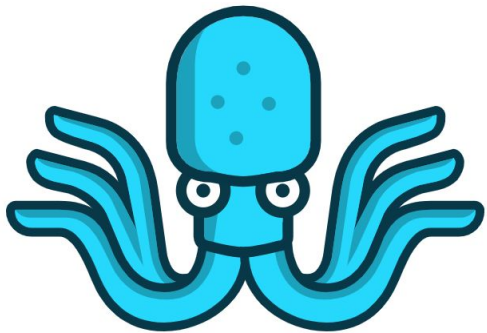


ESLint



Prettier

Documentation tools



React Styleguidist

<https://react-styleguidist.js.org/>



<https://react-styleguidist.js.org/>

<https://medium.com/trabe/document-your-react-components-using-docz-1b8bd888ebe3>

Testing

- Unit Testing and Integration testing done by hand, and by use of CircleCI
unit testing implemented in github
- Since we had features added so frequently rewriting testing scripts or trying to maintain automated tools become more work than it was worth sometimes
- Several team members were designated as testers, spent most of development running unit and integration tests
- Continuous testing

Unit Testing

Test #	1	2	Result
Operating System	Mac	Linux Mint	n/a
Browser Type	Chrome	Mozilla Firefox	n/a
Feature #1: Search	Free Text Search & Additional Features	Free Text Search & Additional Features	PASS
Feature #2: Account Sign Up	Database Updates & Successful Portal Creation	Database Updates & Successful Portal Creation	FAIL

Integration Testing

*TABLE 1.0 : Using Chrome on MacOS

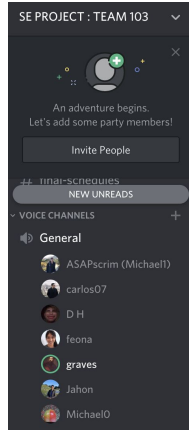
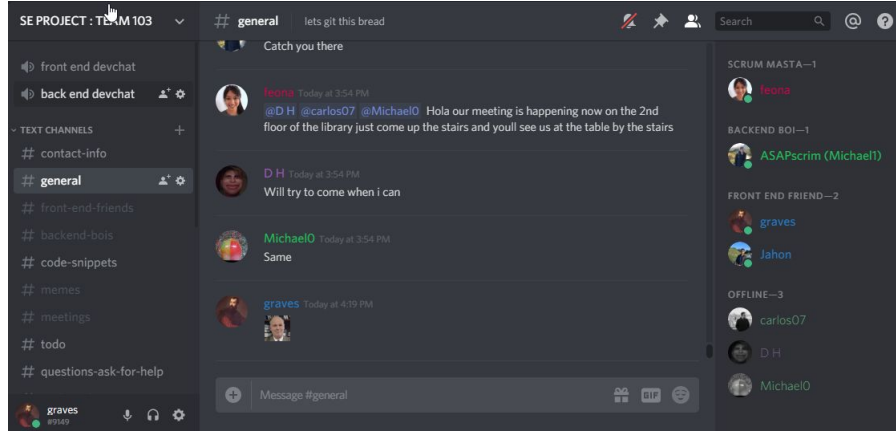
Feature	Case Type	Case Desc.	Dates	Scenario	Prereqs.	Data	Results
Login	Smoke & Unit	Build verification ready for user portal	05/05, 05/06	Validation responsive; Portal successfully populates in a timely manner.	A valid account pre-registered in the database.	Harry's (login info): email & password	FAIL
Home Page	Smoke & Unit	Build verification for general visitors/guest users.	05/05, 05/06	No dead links; Responsive. Harry wants to live close to SFSU.	n/a	n/a	PASS
Search	Smoke & Unit	Build verification for further pages, as well for filters (ie; details page). Build verification for maps.	05/05, 05/06	Shows search results close to campus, scalable, no dead links; Map and listings populate providing visual distance.	Desirable areas for residency, SFSU relative zip codes.	San Francisco, San Fran, SF, San Francisco State University, Daly City, 94132, 94114	PASS
Details Page	Smoke & Unit	Build verification for contacts, filters and routing.	05/05, 05/06	Corresponds with map feature. Details page with active contact link.	Desired filters/amenities in mind.	Laundry, gym, close to public transit system, etc.	FAIL

Beta Testing



- Limited Beta test extended to friends and family
- Due to small size of our beta test it was difficult to get any real feedback from users
- Biggest feedback we got was involving the user portals
 - Somewhat confusing to users
 - We had placeholders still there that were not connected to backend so they thought the website was broken

Teamwork Practices



- Discord was our main base of operations
- Provided a hub for collaboration, resource sharing, communication, and socializing
- Channels for various tasks
- Great alternative to slack, has many of the same features in an (arguably) easier interface

Teamwork Practices



- Important to keep team spirit high
- Conflict isn't good for anyone, the projects quality suffers
- Non-Project related social outings helped keep everyone happy and not trying to kill eachother
- Discord server was a healthy mix of discussion related to both the project and other classes work
 - Helped eachother with other classwork

A brief summary of our dev cycle

Denial

Anger

Bargaining

Depression

Acceptance

A brief summary of the dev cycle

- Our dev cycle was not the smoothest but we learned alot from it
- Major issue was workload distribution and improper delivery of sprint functionalities
 - Very often we were going back over old work and having to redo it
- Key Takeaway : make sure all team members understand the tools they're working with

Things we did right!



- Consistent enforcement of coding style using linters and VS code extensions
- No conflicts, communication in terms of what features we wanted added was great
- GitHub was clean, well managed, and the important branches were protected via permissions and code review enforcement
- Good choice of tech stack, React and Material-UI are a match made in heaven
- Development direction always focused, no 'now what do we do moments'
 - Always a next goal
- Whole team was kept informed regardless of their role in the project
 - Agile methodology followed
- Testing was done correctly, and continuously
 - Continuous integration

Things we did wrong...



- Workload distribution was awful
- React was a great tool to use, but some team members had trouble picking it up due to a limited javascript knowledge coming into the class
 - Partially due to the 412 software lab being a joke
 - Once the codebase was sufficiently complicated it became near impossible for someone to jump into it without having worked on it alot
 - Bottlenecked our feature development as only a couple developers could deliver functioning components
- No enforcement of local testing environments until later into the project
 - We were helping some group members set up testing environments very late into the projects development
- Failed to properly deliver some features we had planned in the beginning

Things we did wrong...



- Slacking off between milestone 2 and 3 put us behind quite a bit
 - Workload spiked exponentially after the second milestone
 - By the time we had features done for milestone 3, milestone 4 was only a week around the corner
- Did not follow agile principle of maintaining a sustainable pace
 - Many late nights working before milestone submissions
 - Days between commits sometimes
- Near the end of the project (crunch time) only a couple team members had the know-how and comfort with the system to implement needed features
 - Delivery of functionalities was bottlenecked resulting in some features having to be canned
- But we learned from all these mistakes!



Things we learned

- How to modify our delivery plans based on what the development team realistically thinks will be deliverable at deadlines
- Working on a team of different specialties, and how to work with people's strengths and weaknesses
- Effective distribution of work requires a constant sustainable pace
- Working with somewhat vague requirements
 - Normally as students we're used to very concrete requirements, this class is a little more freeform in what you're asked to do
- LEARNING WEB DEVELOPMENT from very little experience
 - The hardest thing about this class was learning software engineering practices AND having to learn full stack web development at the same time
- Understanding and working in a team
 - We're students and we're learning, we're not industry professionals (yet)



Gator Room

Thank You!