

design doc - CruzMaps

assignees

everyone can jump on any role they see fit ^.^

jason - full stack

ryan - back

tommy - fullstack

andre - back (second day)

stack

nextjs, typescript, mongo(--> dynamo), vercel

ext: daisyui, mui, heroicons, axios

local development

1. to prepare, install yarn from [this link](#)
2. when cloning the repository, add remote to your local and then yarn install to get all required dependencies
3. yarn run dev to run a local development, we will auto deploy on vercel so you can see live changes per branch

good practices to keep (for everyone's sanity)

1. when making changes, create a new branch by yourself
 - a. (ex. bugfix/web-auth) if making a branch to fix web authentication
2. push to your branch, send a PR and tell everyone u sent a PR (or me, or anyone related to your role) and they'll merge if it looks good and doesn't break on build
3. make sure your indentation is nice (if you can't handle this, or just forget, try configuring prettify everytime you save your code)
4. delete your branch after merging with main
5. git fetch git pull everytime someone says they're pushing to main
6. avoid non-contextual variable naming
 - a. it will eventually cause a bug, name your variables no matter how long it is!!
7. components should be TitleCased, serverless/apis should be lowercase, and clients should be camelCased

problem & solution

ucsc has shitty maps, the best one is [this one](#), there are many complaints on how it's very disorganized, primitive, and most of all very unorganized. there is no context to a lot of places on campus, and it's just overall underwhelming. this causes new students to have difficulty finding their class, current students having difficulty finding a certain place on campus, and tourists having a different time finding anything on campus.

we can solve this problem by creating a more contextual map that is still lightweight, and much more helpful to new& current students, and tourists. this is intended to replace <https://maps.ucsc.edu>. in addition, users can add their own touch to the maps by exploring the beauty of UCSC and also finding friends near them with the same music taste

features

- P0 – the site is down and all work stops, until this issue is resolved
- P1 – finish this task to unblock smoe else, req to be done before other things
- P2 – not required
- P3 – nice to have

~~slide in pop up for general use case - p1 (daniel)~~

- left for user related (menu, saved, userProfile) right for location related (waypoint, event details)
- don't make separate components (like userProfile.tsx) our intention is not to

~~mongodb instance methods - p1 (ryan)~~

ui menu adding instance - p1 (jason)

grabbing event data and display on card using daisyui - p1 (tommy)

filtering - p1

menu - p1

pin img needs to be replaced - p2

grid on map when editing - p2

hackathon history

~~mapbox implementation - p0 [jason]~~

~~pin (saving a location feature) - p0 [ryan, tommy]~~

~~travel (heading to a specific point coordinate on the map) - p0 [ryan]~~
~~search bar ui (similar to Apple's spotlight search in terms of design) - p0 [tommy]~~
~~— icon on bottom p1~~
- -> spotlight search ui change p1 [tommy]

no deployment, localhost deployment instead

slide-in pop-up for specific locations - p1 [jason]
- ui - urgent
~~get geolocation - p1 [ryan]~~

QA assurance testing

styling - p2 [jason, tommy]
- add diff color for nav bar, prof color, match all colors together, add new fonts
- too rough when switching, new colors
- outer bounds need to extend to the city

~~theme light switch~~ (dark/light) - p2 [jason]

social layer - p2 [ryan, andre]

- School events will pop up on map letting students easily discover events
- Groups feature
 - For friend groups, clubs, etc
 - Allows users in group to view group pins and events

bus feature - p3

post purchase

search bar ui using gpt-3.5-turbo to infer user's next answer - p4

error handling - p4

spotify layer (seeing the culmination of spotify data in specific colleges) - p4

hidden map layer (exploring UCSC, where you initialized the app is where your starting point - p4

architecture

hosting

will be hosted on github @ this link (not implemented)

brackets should not be in command

all commit messages should be concise and descriptive

every feature should be on their own branch, but the boilerplate should all be on one branch to

avoid git branching issues

everyone should have gitkraken on their pc

cloning repo

git clone [https]
git remote add origin
git fetch
git pull

creating new branch

git checkout -b [branch-name]

checking out new branch

git checkout [branch-name]

performing push

git status
git add [files / add all by .]
git commit -m "message goes here"
git status (to check again)
git push

performing pull

git fetch
git pull

more cheatsheet: <https://education.github.com/git-cheat-sheet-education.pdf>
<https://docs.github.com/en/get-started/getting-started-with-git/git-cheatsheet>

preset components

subject to change

- globals.css

```
/* globals.css */
:root {
  /* color schema */
  --color-primary: #000000; /* pure black */
  --color-secondary: #FFFFFF; /* white */
  --color-accent: #333333; /* darker shade of black */

  /* font interfaces */
  --font-ui: 'SF UI', sans-serif;
```

```
--font-text: 'Inter', sans-serif;  
/* reference as var(--font-ui) or var(--font-text)*/  
}
```

component pipeline

1. create a component in src/components/ (export default)
2. import @ top into a page in views/map/index.tsx
3. should automatically render in pages/index.tsx