

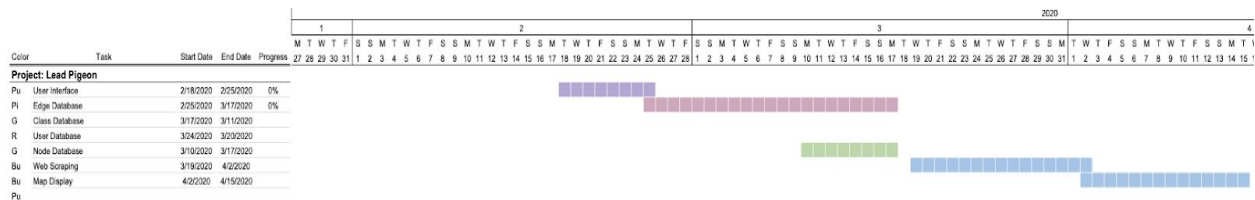
Buff Guide

Andrew McQueary, Christen Malloy, Daniel Vignanker, Justin Yara, Michael Donovan, & Rachel Williams

Project Description:

Buff Guide was designed for freshmen at CU Boulder, new transfer students, and faculty. The webpage is designed using several units we were taught during CSCI 3308. The SQL Database has every single course offered at CU Boulder. Accompanying the classes, is the coordinates, room number, building location, and more. The webpage allows you to input your schedule, and save it when you create an account and log in. After you save your schedule, you can find directions to your classes, and go from class to class. The interactive map also allows you to use many Google Maps APIs; such as, Directions, Marker Clustering, and an Interactive Map. The first API we use is to display a map of the entire campus, and it's surrounding areas. The map updates in real time, and will display closures of buildings. As of now, it displays the Rec, Norlin, and the C4C temporarily closed due to COVID-19. Another API we use is Marker Clustering. This allows us to display all of the locations, and the user can go into street view and stand on the sidewalks between buildings. This feature is great for people new to campus; for example, they can interact with the map to put them on the same sidewalk they are on, and the markers will show up in different colors to display the building.

Project Tracker: Gantt Chart




VCS:



<https://github.com/mido3801/BuffGuide>

Contributions:

Commits on Apr 17, 2020


login and add class working



 mido3801 committed 10 days ago

 1533c7f 


Commits on Apr 13, 2020



Merge branch 'master' of https://github.com/mido3801/buffguide

 mido3801 committed 14 days ago


 81ff0a2 



test

 mido3801 committed 14 days ago

 12e18c2 


working site version 0.1



 mido3801 committed 14 days ago

 bc4f5f2 


Commits on Apr 23, 2020



with db now

 mido3801 committed 4 days ago


 504a38d 



test

 mido3801 committed 4 days ago


 792cfab 



test

 mido3801 committed 4 days ago


 5a0f24a 



test

 mido3801 committed 4 days ago


 724bdcd 



test

 mido3801 committed 4 days ago


 b0d782f 



test

 mido3801 committed 4 days ago


 ff1514a 



test

 mido3801 committed 4 days ago

 73b3108 


test



 mido3801 committed 4 days ago

 117ca7f 


Commits on Apr 21, 2020



fixed row ids

 mido3801 committed 6 days ago

 f850629 


working



 mido3801 committed 6 days ago

 34d70a9 

Commits on Apr 17, 2020

login and add class working

 mido3801 committed 10 days ago

 1533c7f 

Mike: I created the site architecture/routing with Flask, set up the DB with sqlite and facilitated scraping class data into the database. I also wrote the jquery and ajax code to allow for interacting with the page, inputting user classes, and getting directions between 2 inputted classes. Wrote routes in Flask app to allow for returning of class/location info via ajax queries with minimal page reloading. Also wrote code for user login functionality and saving class schedules.

Justin: I worked on the basic framework of the webpage and created the nav bar and design. We went to office hours and got Client IDs/API keys and instructions on how to use all of Google's resources. I set up the interactive map and the marker clustering API with Christen. I also set up Directions API, but my implementation did not work on my database, so Michael did the bulk of the Directions. Christen and I also went to office hours numerous times to further our Google API mastery. I also helped in the database, but I was mostly learning from Andrew and Mike as we progressed.

Andrew: I worked on assisting everyone with many parts of the project. I worked on the POST/GET for the server side of the projects. Additionally, I assisted in making and figuring out the database and how we want it to interact with our project. I was a big part in many of the brainstorming and finding general solutions to the developmental needs of the project. I also spent time working on the general logic of the web application itself and figuring out the processes we want to get our project to perform the way we envisioned it. I also did a lot of the work around writing the milestones. We spent the whole of the project doing group coding. Michael was the most proficient by far so as a result he has all of the commits.

Christen: I developed the original idea for the project in our brainstorming session; I have been keeping meeting notes for every project meeting as well as identifying action items to focus on as the project developed. I helped with deploying the marker clustering with Justin by sourcing code and links to implementation guides. I helped Michael debug database code (Javascript / Python), and helped with implementing Google API's and the Heroku deployment. I also brought several logistical issues to office hours with Justin and Michael (Oauth error / Google API deployment issues). Designed page features to add to the home screen (nav bar / links / background). Designed page wireframe on paper. Helped with powerpoints and submitting milestone documents.

Rachel: I helped out with ideas and brainstorming many parts of the project. I worked with everyone on the initial structure and setup of the project. I learned a lot about javascript and python throughout the project and my group was very helpful in allowing us all to learn from each other. I also learned how to and got to help implement the API keys with Justin and Daniel.

Daniel: I missed the original brainstorming meeting but since then I've tried to be as much help as possible. I worked with the rest of the group to do some of the initial structure/set up as well as work with Justin and Rachel on implementing the Google

APIs into the site. I also worked a little with Michael on the scripts for parsing our class data. I learned a lot from the entire process of working with a large group on a project of this caliber, from implementing logins and Google APIs to producing the overall design of our app.

Deployment:

API Keys were removed from the GitHub repository due to security reasons. The APIs are in a separate document that you can paste into the code to run.

```
mike@mike-VirtualBox:~/PycharmProjects/BufTest$ git push heroku master
Counting objects: 2, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 229 bytes | 229.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Python app detected
remote: -----> No change in requirements detected, installing from cache
remote: -----> Installing SQLite3
remote: -----> Installing requirements with pip
remote: -----> Discovering process types
remote:          Procfile declares types -> web
remote:
remote: -----> Compressing...
remote:          Done: 99M
remote: -----> Launching...
remote:          Released v8
remote:          https://buffguide.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/buffguide.git
   5a0f24a..792cfab master -> master
```

While we were able to push code to the heroku platform, we ran into issues with environment variables and were unable to resolve them. As such we do not currently have it deployed. However if you visit the github there are instructions for launching the app on your system.