Buff Busters

A Degree Audit

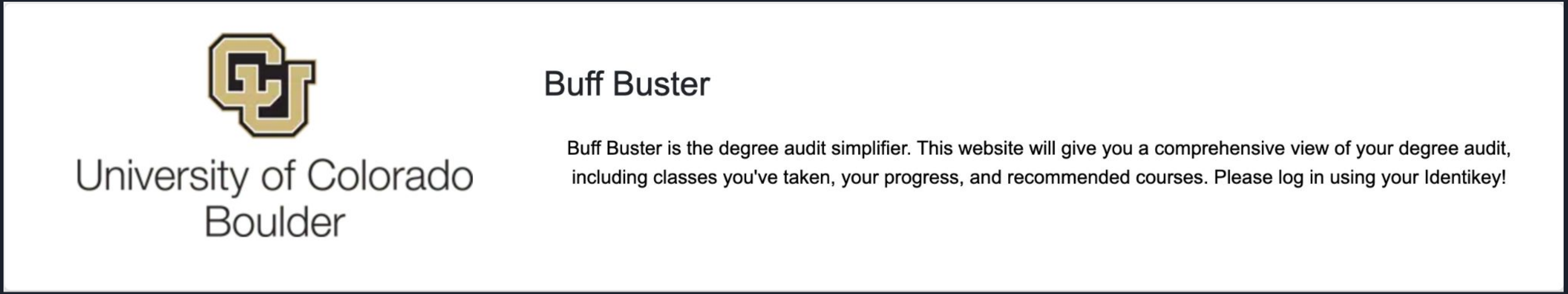
By: Sania Joshi, Ryan Brace, Sean Esslinger, & Boston Abrams

Description of the Project

It is **critical** for keeping an organized system of all of classes a student takes, making sure a student is able to graduate, and looking at past and current coursework.

While the current CU Buff Degree Audit is useful, it can get **overwhelming** and **confusing**.

To fix this, we added features which we thought would make our website easier to understand and work with. Including the ability to see what classes you have taken.



Buff Buster

Buff Buster is the degree audit simplifier. This website will give you a comprehensive view of your degree audit, including classes you've taken, your progress, and recommended courses. Please log in using your Identikey!

All the Tools Our Group Used (Pt 1)

Tool #1: Github/Git

Purpose:

- Version Control System (VCS)
- Used for committing numerous changes and collaborating on writing code with other group members
- We used github project board to manage completion of tasks (user stories)

Ranking: 5/5

- Great for branching, merging, and keeping track of changes to code made by all group members over time

All the Tools Our Group Used (Pt 2)

Tool #2: Visual Studio Code

Purpose:

- Integrated Development Environment (IDE)
- Supports various different kinds of programming languages, debugging, and integrates very well with Git and other version control systems

Ranking: 4/5

- Could sometimes be challenging to resolve merge conflicts, in general a little bit finicky when integrating via Github
- Overall worked well, especially for writing code in different languages / combining lots of different code together

All the Tools Our Group Used (Pt 3)

Tool #3: Docker

Purpose:

- Run Code in Same Type of System
- Isolates applications in containers (only what app needs to run) allowing for efficiency/consistency across different environments

Ranking: 4/5

- Some problems with windows firewall
- Overall a convenient way to have our code work on different computing environments

All the Tools Our Group Used (Pt 4)

Tool #4: Postgres

Purpose:

- To store and handle large amounts of data
  - Is a relational database management system (RDBMS)
- Really good at handling structured data (ex: tables, rows, etc.)

Ranking: 4/5

- Good way to access large amount of data conveniently, also very flexible
- A little slow, and confusing to learn at first - has a learning curve

All the Tools Our Group Used (Pt 5)

Tool #5: HTML

Purpose:

- UI Tool
- Designing web pages and giving them a visual appeal
- Structuring content on a webpage

Ranking: 5/5

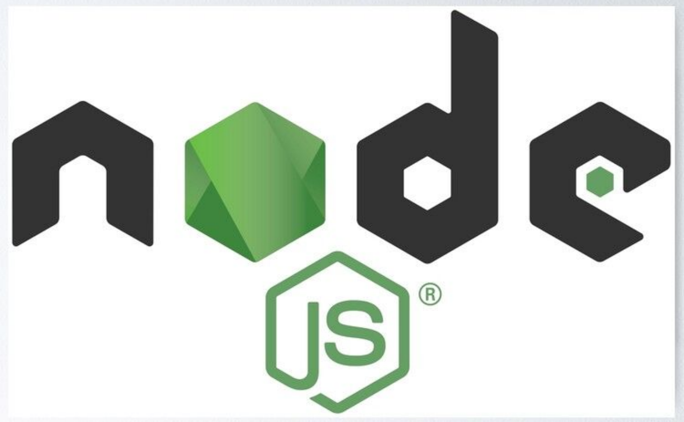
- Great way to edit look and format of a webpage
- Easy to incorporate bootstrap or other HTML templates and combine it with preexisting code

All the Tools Our Group Used (Pt 6)

Tool #6: Node JS

Purpose:

- Application Server
- Able to use Javascript on both front/back end
- Is able to handle many tasks at the same time (API's) b/c it directly moves onto another task



Ranking: 4/5

- Occasionally caused some asynchronicity errors but worked well.

All the Tools Our Group Used (Pt 7)

Tool #7: Render

Purpose:

- Deployment Information
- Allows people to easily build/deploy applications and do it efficiently
- Handles databases/containers (Docker)

Rating: 2/5

- Very Difficult to Configure with the Database
- Integrates very well with Github and easily pushes updates/changes to code on Git
- Had to restructure directory

All the Tools Our Group Used (Pt 8)

Tool #8: Mocha & Chai

Purpose:

- Mocha = Testing tool to help organize tests
- Chai = assertion library that integrates easily with mocha
- Easier to write tests for code / make sure code is working as expected
- Helps to make sure new changes don't break existing code

Ranking: 4/5

- Good for catching coding bugs early on
- Has useful features like assertions to easily detect if tests are passing or not
- Only an assertion library
  - Needs to be paired with another framework like Mocha to actually execute tests

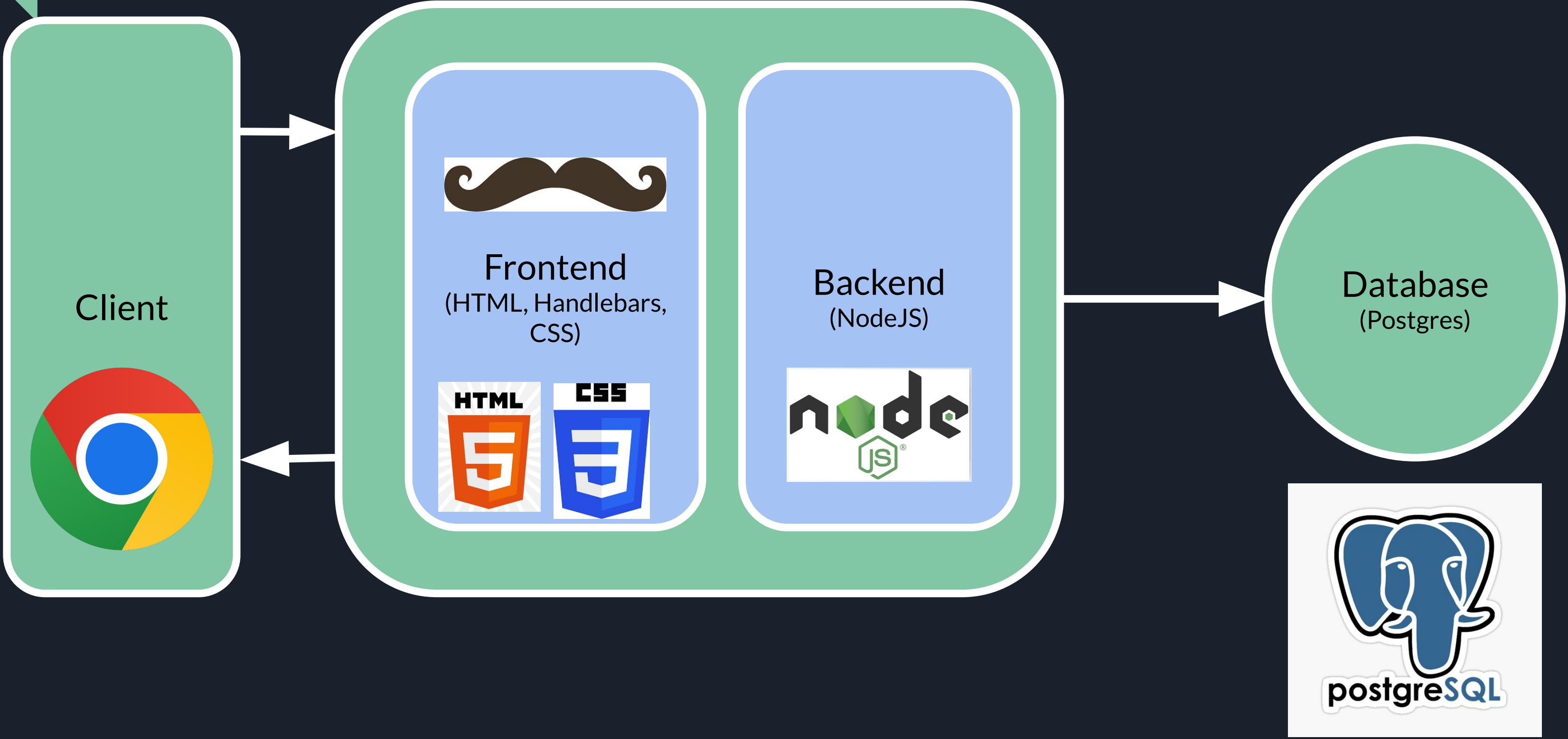
Methodologies

- Agile: (for all of these together)
- **Iterative** (Docker & VS Code)
  - Debugging Code
    - Setting breakpoints with smaller chucks/commits in code - allowed group to easily step through code
- **Pair Programming** (VS Code)
  - Used live share on status bar of VS code or Zoom
- **Continuous Integration/Continuous Development** (CI/CD)

  (Docker)

  - Do not need to rebuilt/test/deploy code every single time we make want to change our code
  - Allowed us to make a lot more commits in a shorter span of time

Architecture Diagram



Challenges

Challenge #1: Committing to main on Github - resolving merge conflicts

- Accidentally made a mistake and resolved merge conflicts incorrectly, which caused out registration / login page to stop working for some people on our team

Challenge #2: Getting Carousel on Stats Page to Dynamically Update Info

- A little bit challenging to work though dynamically updating the stats page (home page as well) to update a specific users info
  - Especially getting it to work for core, foundation, and math classes

Challenge #3: Bug While Completing Render

- Had some issues with connecting render to our database
- Had to do a lot of debugging to fix this issue

Future Scope/Enhancements

Enhancement #1: Catering Website Toward All Majors

- Could potentially also increase the scope of our degree audit to help all majors, not just computer science.
- Useful b/c all majors will be able to have an organized way to view class schedule, and better plan their schedules

Enhancement #2: Adding More Visual features / Class Schedule

- Useful for better visual of what classes student is taking
- Useful for user - easier to create a schedule for next semester (maybe drag and drop feature)
- Add a list of all classes you still need or could take.
- Separate out GPA

Enhancement #3: Future Classes/schedule Suggestions Page

- Current Buff Buster has dynamic carousel for past and current classes, not future classes
- A page that would dynamically add classes from the search classes and build you a schedule based off of times and needs to graduate.

Demo of Our Project

Backup Link to Video of Our Demo:

Questions: