

# Buff's Bulletin

Mia Ray, Jessie Hsu, Jules Novoa,  
Julia Aronow, & Makaela Fauber

**Team 12-09**



# Buff's Bulletin Vision



Buff's Bulletin is for CU students who need a better way to advertise events and find events to attend on campus. This CU Boulder specific map uses the Mapbox API for a seamless experience. Unlike the campus map or Google maps, campus events can be shown all in one place. Users can simply pull up the map and see upcoming events going in a list and on a map. Students can also view events by club and follow clubs to prioritize their events. Students can create, view, comment on, and RSVP to events on campus. They can also add other users as friends.

# Tools



## GitHub Project Board

Purpose: Project tracking  
Rating: 3



## PostgreSQL

Purpose: Database  
Rating: 4



## MapBox

Purpose: External API  
Rating: 5



## CSS

Purpose: Stylization  
Rating: 5



## GitHub

Purpose: VCS  
Rating: 5



## Docker

Purpose: Containerization  
Rating: 5



## HTML

Purpose: UI Tools  
Rating: 5



## Chai

Purpose: Assertion library  
Rating: 4



## Discord

Purpose: Communication  
Rating: 5



## NodeJS

Purpose: Application server  
Rating: 5



## Handlebars

Purpose: UI Tools  
Rating: 3



## Mocha

Purpose: Testing framework  
Rating: 4



## VSCode

Purpose: IDE  
Rating: 5



## Express in Node

Purpose: Framework  
Rating: 5



## Bootstrap

Purpose: Stylization  
Rating: 4



## Render

Purpose: Deployment  
Rating: 5

# Agile Methodology

We used **1-week sprints** to structure our workflow. We held **weekly scrum meetings** on Tuesdays to organize sprint goals. The **GitHub project board was used as a Kanban** to visualize progress.

**Week 1 - Project Setup:** environment configuration, documentation, partials, initial database schema

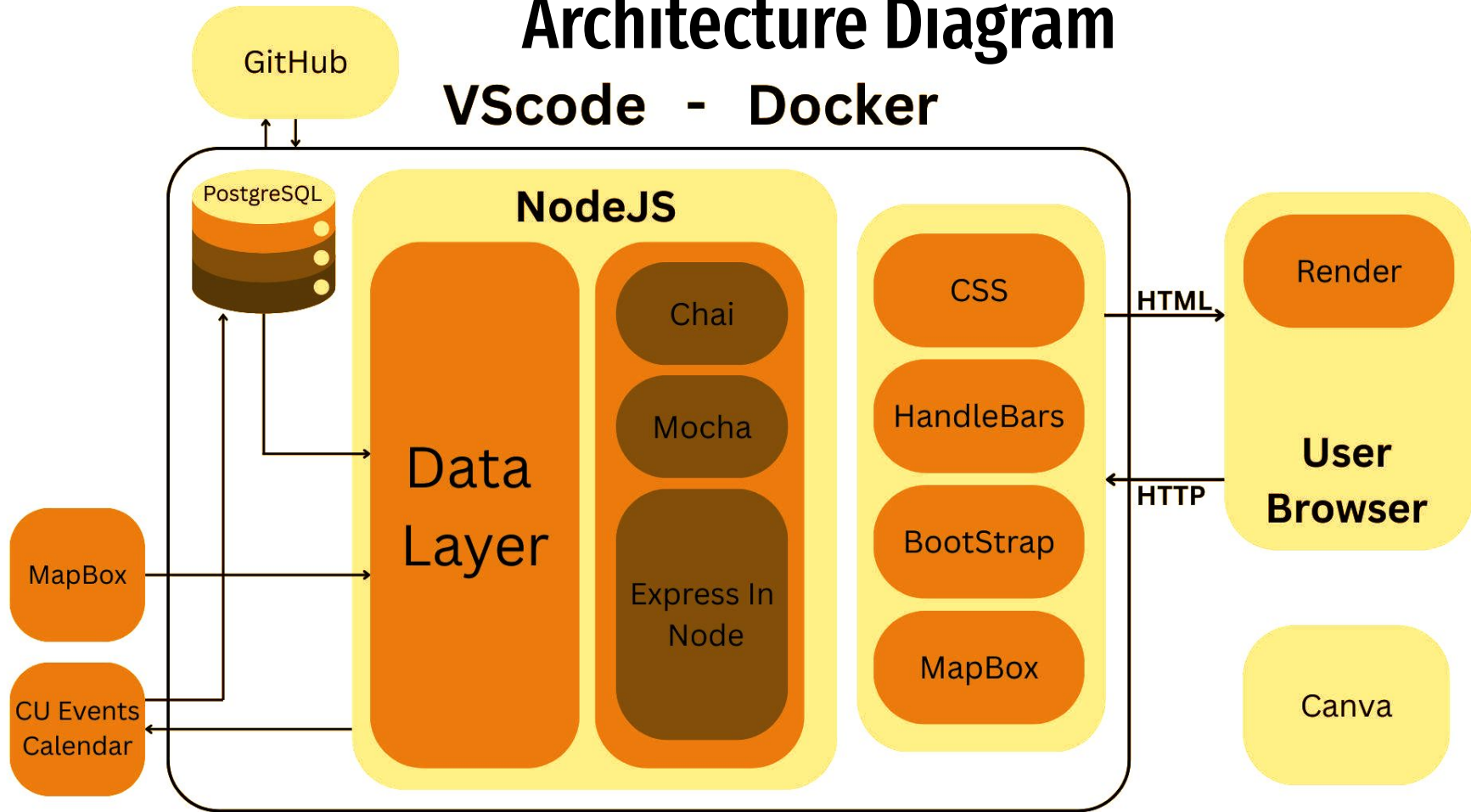
**Week 2 - Backend:** API routes, map with pins, key page layouts (Home, Events, Profile, etc.)

**Week 3 - Functionality:** web scraping, data parsing, session authentication

**Week 4 - Final Touches:** UI refinement, user testing, cloud hosting

# Architecture Diagram

## VScode - Docker



# Challenges

- **Mapping and Pins**

Manually establishing coordinates for buildings, syncing pins with sidebar events and reflecting correct category color

- **Event Filter**

Sidebar reformatting for handling event data

- **Event Web Scraping**

Parsing/sorting through limited ICS information and obtain meaningful data

- **Database**

Database issues when implementing unit testing and deploying our application using Render

- **Developer Roles**

Primarily frontend/backend roles -> full stack feature-based roles

- **API Routing**

Profile picture posting, time formatting, linking events to event categories



# Future Enhancements



## Social

Allowing users to view the events their friends are attending and store user commenting information to add friend functionality

## Database

Updating the database to remove events that have already occurred along with regularly adding new events

## Web Scraping

Implementing systems for pulling more events from sites like Instagram, since not all clubs post consistently on BuffConnect

# Demo

