Milestone 3
Group 013-01: Profanity
App Name: Here And Now

Features:
Location/Google Maps api and javascript
Front and back end login page/user registration database
Back end for location database and Comments Database
Front end of home page

Demo Notes:
Shared demo of working location overlays and user location
Shared demo of working login and registration form connected to database
Shared front end web page framework
No time to share but have functional back end for locations and comments

Issues:
Set up server with nodejs and npm to get containsLocation() function working from maps API to validate that the user is inside the geofence for the area of interest (development issue)

Front end website needs more functionality, decide on a style to use

Passwords are not encrypted

Login verification and change of state need to be implemented

Issues with get and post requests when navigating from homepage.ejs to loginpage.ejs

Suggestions From TA:
Search bar query opens new page with that query
Finalizing Locations database with drawing layer
Track login status of user to validate posting ability
Encrypt Passwords!

Individual Contributions:
Even- set up google cloud console and generated javascript maps api, created functional location demo with overlays. Created database architecture diagram and finalized our application architecture.

Owen- Created loginpage.html and loginpage.ejs. Added appropriate files (package.json, docker-compose.yaml, etc.) for Node.js integration to connect backend user db and login page. Managed to get a local docker server running for demo (like in lab 7) that supports login page connection to backend SQL table.
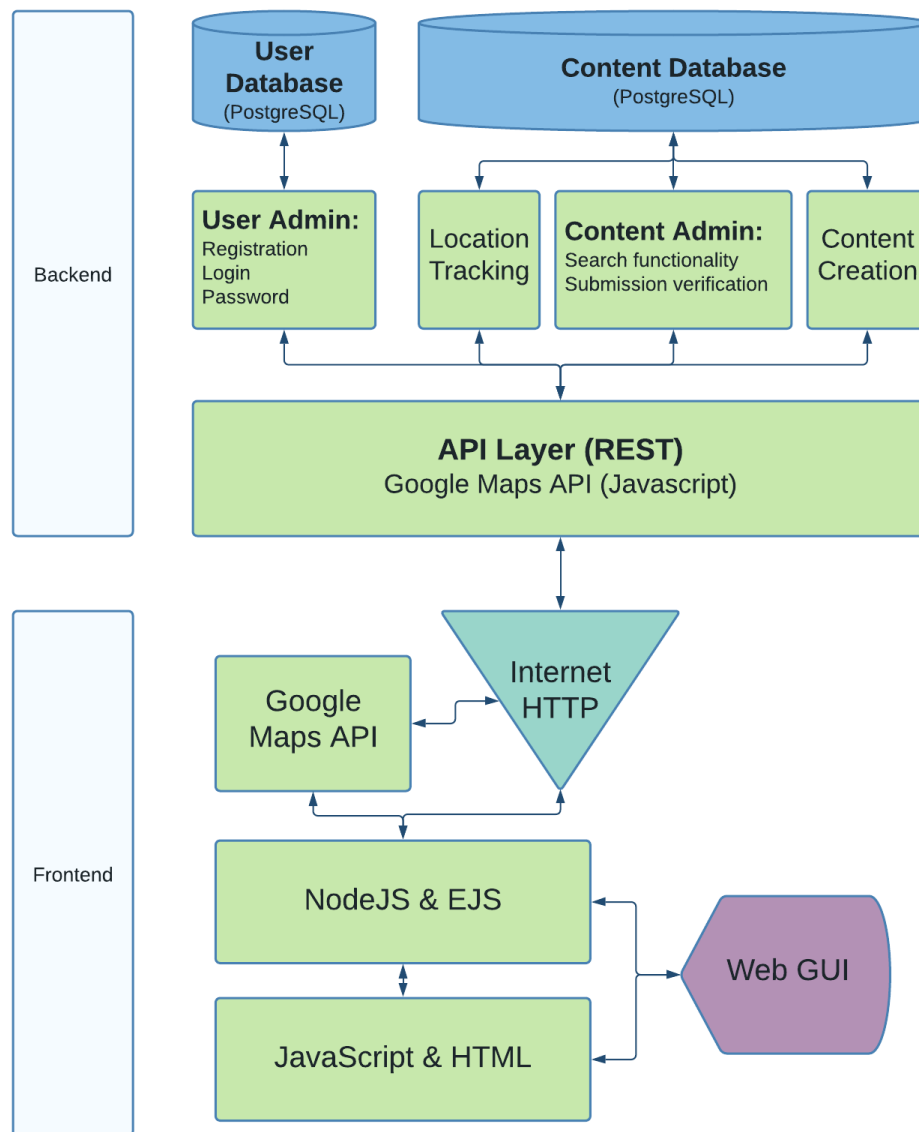
Nolan- Wrote SQL queries for user database to create users table.

Kyle- Created database for locations and comments

Yefan- Created front end landing page with search bar (homepage.html). The theme used on this page is likely one we will use for the entire site.

Architecture Diagram
Team Profanity 013-01

Database Diagram:

Users
- *userID
- username
- email
- password
- role
- favorites

Locations
- *locationID
- location_gps
- location_type
- location_tags
- likes
- dislikes

Posts
- *postID
- userID
- title
- locationID
- location_type
- description
- picture
- likes
- dislikes
- reports

Comments
- *commentID
- postID
- userID
- comment
- likes
- dislikes
- reports