**Project Summary**

Our project goal is to build a web app for students to rate, review, and rank buildings across campus, specifically for water fountains, bathrooms, and vending machines. The application will be an interactive platform for ranking these amenities,helping people make decisions about where to go.

The problem we are trying to solve is to help students be satisfied with the basic amenities of campus buildings. For example with Grainger Library, the bathrooms are unclean, there aren't any vending machines on main floors, and the water fountains are poorly maintained. However, right next to Grainger is CIF with clean and new bathrooms and water fountains worth taking a quick commute for.. This kind of student data in theory could also provide UIUC cleaning staff with data to improve maintenance.

**Detailed Description**

We will develop a web application where users can rank campus amenities. The three categories of interest are:

Water Fountains – ranked by taste, water pressure, and temperature.

Bathrooms – ranked by cleanliness.

Vending Machines – ranked by variety of snacks, pricing, availability, and working conditions.

The problem we solve: Students often encounter bad quality amenities without warning. By aggregating ratings, we can improve student convenience and experience at various buildings throughout the campus.

**Creative Component**

To stand out, our project will include:

Interactive Maps: A searchable, interactive campus map where amenities are color-coded by average rating. This will use the Google Maps or OpenStreetMap API with custom overlays.

Gamification & Engagement System: Students earn points/badges for contributing reviews (“Top Bathroom Rater of Grainger Library”). This also includes a game like ranking system

These components are challenging and need multiple APIs, going beyond simple web forms or static data.

**Usefulness**:

The application is highly useful to the UIUC community. Amenities are things everyone uses on a regular basis no matter what, with high volume of usage. By having quality reporting on the state of amenities at a given building, students can make more accurate decisions about the buildings they frequent, and staff can use this information to determine which buildings need improvement in this regard.:

Basic Features: View ratings,gith

Advanced Features: Interactive map navigation

**Comparison with Existing Tools**

If you are familiar with the cs ranking website that was somewhat famous last semester, we are trying to do something like that but with uiuc buildings.

**Realness (Data Sources)**

We will use multiple real data sources to build and enrich the application:

Campus Map Data: UIUC building maps and amenity location data, available in CSV or shapefile formats from the UIUC Facilities & Services GIS datasets.

Student Crowd-Sourced Data: User-submitted ratings stored in a SQL database (PostgreSQL or MySQL). This will grow dynamically as users contribute reviews.

External Sources: Integration with Google Maps API or OpenStreetMap for geolocation and navigation.

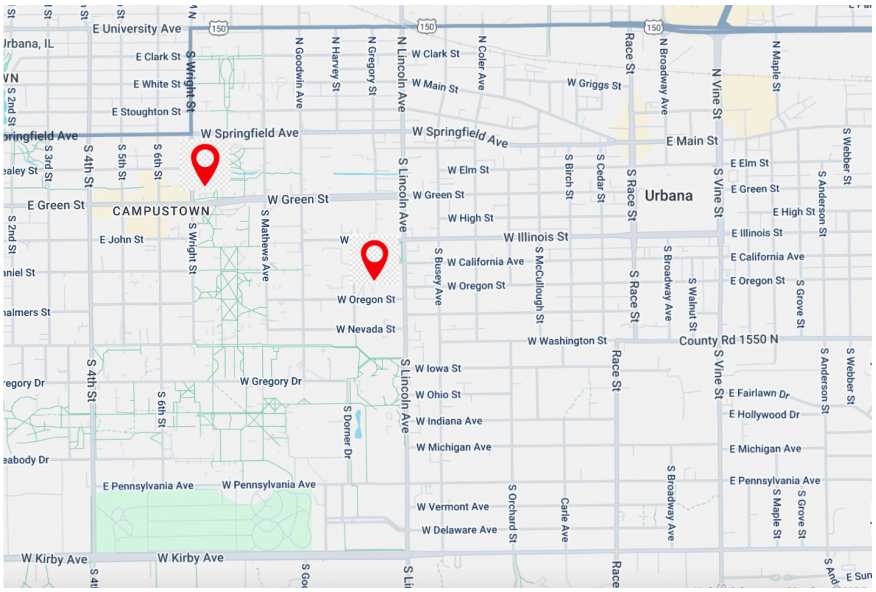
Size: UIUC has 200+ buildings; we anticipate ~100 amenities in the database initially, growing as users add more.

**Functionality**

Our web app will offer CRUD features, which allow users to add new reviews, update their past reviews, delete them, and view information about campus amenities. Each review will include qualities specific to each amenity type: bathrooms (cleanliness), water fountains (taste, pressure, and temperature), and vending machines (snack variety, pricing, availability, and working condition).

The app will also provide search and filter features so students can locate amenities based on building or type. The amenities will be ranked by scores that combine multiple reviews, and each location will display its overall rating. These results will be available in two formats, a list view that shows the ratings and reviews in a ranking list, and an interactive campus map view with ratings layered.

**UI Overview:**


**Work Overview:**

rhand7: Frontend Components and Map UI

samuel67: Database management and data collection

Sgiri6 & siyer35: Middleware and FastApi backend