Ira A. Fulton Schools of Engineering
Arizona State University

Term: Fall 2025 **Subject:** Computer Science & Engineering (CSE) **Number:** 412
Course Title: Database Management (CSE 412)

CSE 412: Project Manual

Project Title: Quiet Study Spot Finder

Team Member Names: Hemant Dua (1225688464)
Aryash Dubey (1225867879)
Kanishk Gohil (1225136744)
Ethan Ernst (1225088644)

Team Contributions:

None of the tasks and responsibilities mentioned below were exclusive to one person. All of the tasks were done by all four team members, but were led by 1, which is mentioned in table below:

Name	Role/Responsibility	% Contribution
Hemant Dua	App dev and database connection	25%
Aryash Dubey	Querying database	25%
Kanishk Gohil	Schema design	25%
Ethan Ernst	Testing and Validation	25%

Overview of the Application

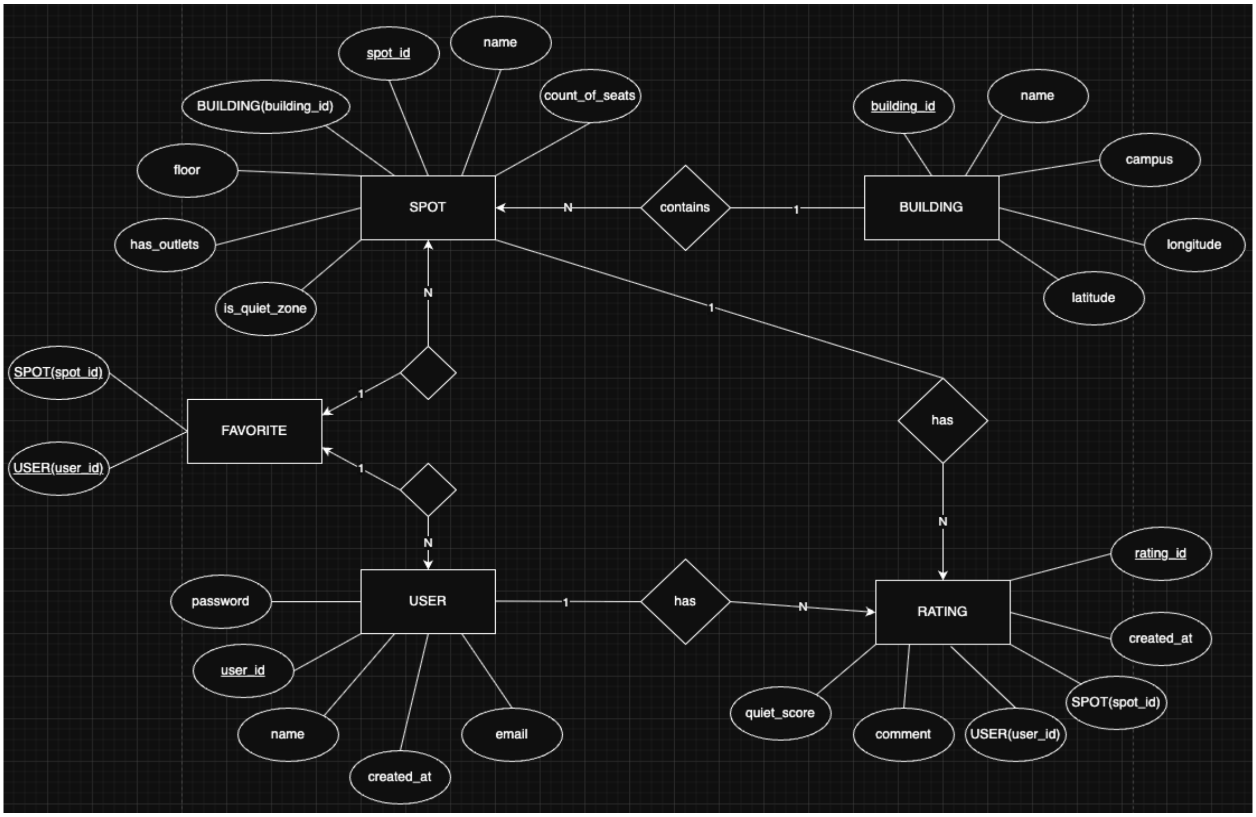
Quiet Study Spot Finder is a web application that helps students quickly find quiet, reliable and comfortable places to study around the campus. Each study spot is stored in the database with the information about its building, the floor it is on, number of seats and whether the area is designated as a quiet zone or not. It also includes whether the place has power outlets or not. Students can rate how quiet a spot feels and leave short comments based on their experiences.

In this application that we have built, students can search through a list of all the spots that are available and narrow it down using simple filters such as campus, building or a minimum rating. Each spot's entry shows key details of the area. So that it becomes easier for them to decide if the location fits the kind of studying place they want for that day or not.

The application analyzes and uses these ratings to rate the places so users can see which spots are consistently quiet and which ones might be busier or louder. Students can also mark certain spots as favorites so they can return to them easily when they want to come back later. Overall, the goal of this application is to reduce the time that the students might be spending searching for a good place to study and make use of real world feedback from other students to make studying around the campus a seamless and hassle-free experience.

Database Design Summary

ER Diagram:



Explanation of Tables:

The database is a PostgreSQL database and we have 5 main tables. We have implemented this on Supabase. The description of tables are as follows:

1. **users**: Stores application users.
    a. user_id (Primary Key) uniquely identifies each user.
    b. name, email, and password store basic account information.
    c. created_at tracks when the user was added.
2. **building**: represents campus buildings that contain study spots.
    a. building_id (Primary Key) uniquely identifies each building.
    b. name is the building name.
    c. campus indicates which campus the building belongs to.

   d. longitude and latitude store the building’s coordinates for location-based queries or future map features.

3. **spot**: represents individual study spots inside buildings.
   a. spot_id (Primary Key) uniquely identifies each spot.
   b. building_id (Foreign Key) links each spot to a row in building.
   c. name gives a simple label for the spot.
   d. floor stores the floor number.
   e. seat_count stores how many students can sit there.
   f. has_outlets indicates whether power outlets are available.
   g. is_quiet_zone indicates whether it is meant to be a quiet area.
4. **rating**: stores user feedback for each spot.
   a. rating_id (Primary Key) uniquely identifies each rating.
   b. user_id (Foreign Key) links the rating to a user in users.
   c. spot_id (Foreign Key) links the rating to a spot in spot.
   d. quiet_score is a small integer rating of how quiet the spot is (for example 1–5).
   e. comment allows an optional text review.
   f. created_at records when the rating was made.
5. **favorite**: stores which spots individual users have saved as favorites.
   a. user_id (Foreign Key) references users.
   b. spot_id (Foreign Key) references spot.
   c. Together, user_id and spot_id act as a composite key so the same user cannot favorite the same spot twice.
   d. created_at records when the favorite was added.

The database has relationships which have a very clear structure. Each building can have as many spots as there can be. Each spot can have many ratings and many favorites. Each user can create many ratings and many favorites but unique to per spot.

Technology Stack

Frontend and application: The user interface of this application is built using Next.js with type script. Nextjs provides routing and page structure where we have made all the screens. The screens include spots list, the details of a spot, favorites page etc. TypeScript makes everything type safe so data from the database including all the five table objects is handled safely and consistently across the entire code base and it also reduces the chances of error so makes it much much better and also makes it easier to maintain.

Database: The data is stored in a postgreSQL database which is hosted on Supabase. We chose Supabase because it helps with API integration and we don't have to run the database locally. Supabase runs on Postgres service. The application communicates using the pg library like in a node environment which sends SQL backend operations for all the queries that we have for the tables.

Integration: All the frontend pages are called Supabase using the server components, in other words, API routes in next.js to load all the data into the application. This keeps architecture simple and clear and easily understandable and manageable. There is one central database, all the operations are routed using the backend and all the features in the app are built on the same schema. This also keeps the types maintainable and makes sure the data entered in the database is safe.

Setup Instructions

Prerequisites:

- Node.js 18+ and npm
- Database is already hosted on Supabase and credentials are in env file

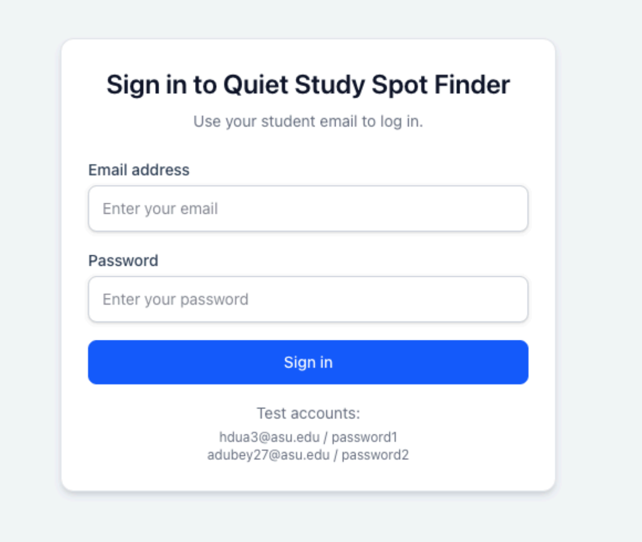
Steps:

1. Open terminal and git clone https://github.com/hemantdua30/CSE412-FInal-Project.git
2. Install dependencies using npm install
3. Start dev server using npm run dev
4. Open http://localhost:3000 or whatever url your project loads on
5. Login with test account:
    a. Username: hdua3@asu.edu, Password: password1
    b. Username: adubey27@asu.edu, Password: password2
    c. Username: kgohil1@asu.edu, Password: password3
    d. Username: eaernst@asu.edu, Password: password4

Feature List

1. Browse Study Spots: Here, all the study spots are listed from the table joined with the building table to show building name and campus names.
2. View Detailed information of a spot: For a selected Spot, it displays full information and its related building. It also shows the ratings from the rating table including the comments.
3. Rate a study spot: This allows a user to create a rating entry in the table for a given user ID and spot ID. This supports updating an existing rating where users can change the score. It also allows deleting a rating which removes that row in the rating table.
4. See user comments and ratings of a spot: Here users can add comments to the rating table. It's the same way they're adding the rating to the spot.
5. Favorite and un-favorite study spot: Users can mark a spot as their favorite. They can have multiple favorite locations which inserts a row into the favorite table using the user ID and spot ID. People can also remove a row which deletes it from the favorite table.
6. View personal favorite spots: This page provides a page that can show all the spots a user has saved as a favorite in the favorite table, which is again joined on the spot and building table so that user can see all the details instead of just random IDs that we store the data with.
7. Sort and Filter out spots: We also provide options to sort and filter out spots based on the available options. This is based on all the data in the database.

Screenshots



Sign in to Quiet Study Spot Finder

Use your student email to log in.

Email address

Enter your email

Password

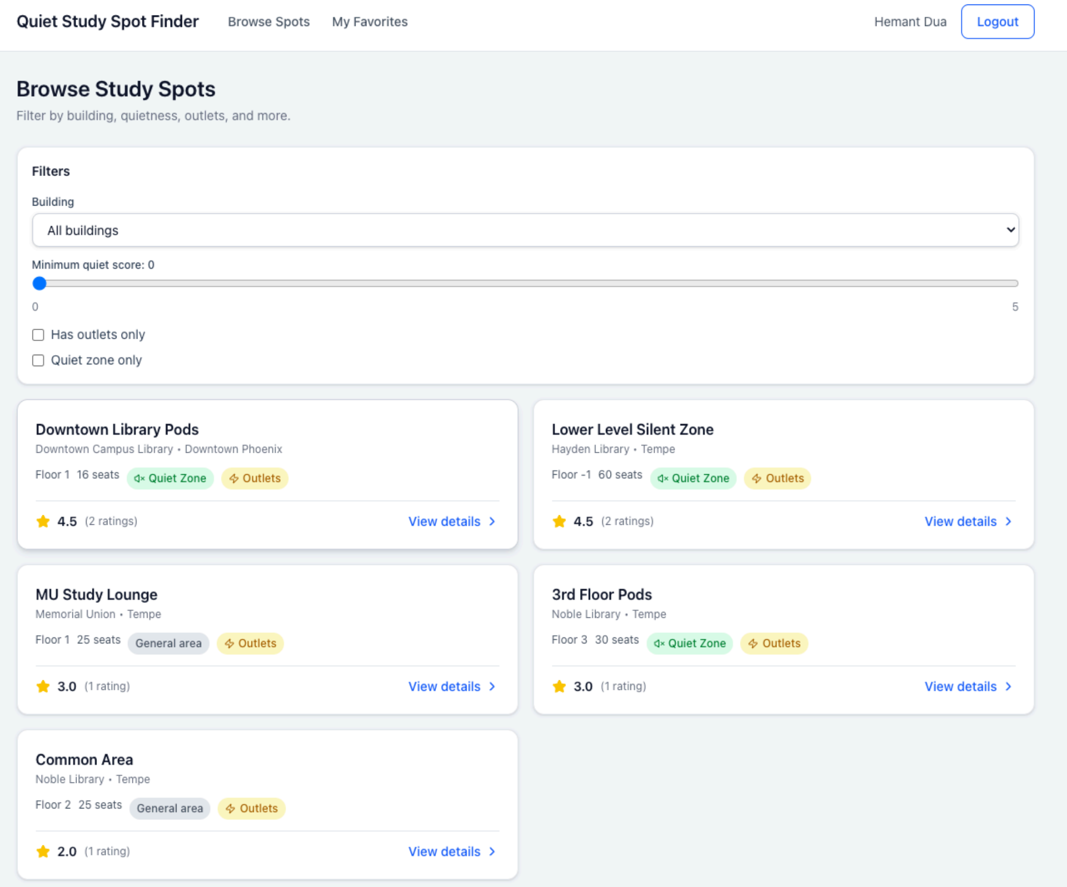
Enter your password

Sign in

Test accounts:

hdua3@asu.edu / password1
adubey27@asu.edu / password2

[Login Page]



Quiet Study Spot Finder   Browse Spots   My Favorites   Hemant Dua   Logout

Browse Study Spots

Filter by building, quietness, outlets, and more.

Filters

Building

All buildings

Minimum quiet score: 0

0 … 5

□ Has outlets only

□ Quiet zone only

Downtown Library Pods
Downtown Campus Library • Downtown Phoenix
Floor 1 16 seats Quiet Zone Outlets
★ 4.5 (2 ratings) View details ›

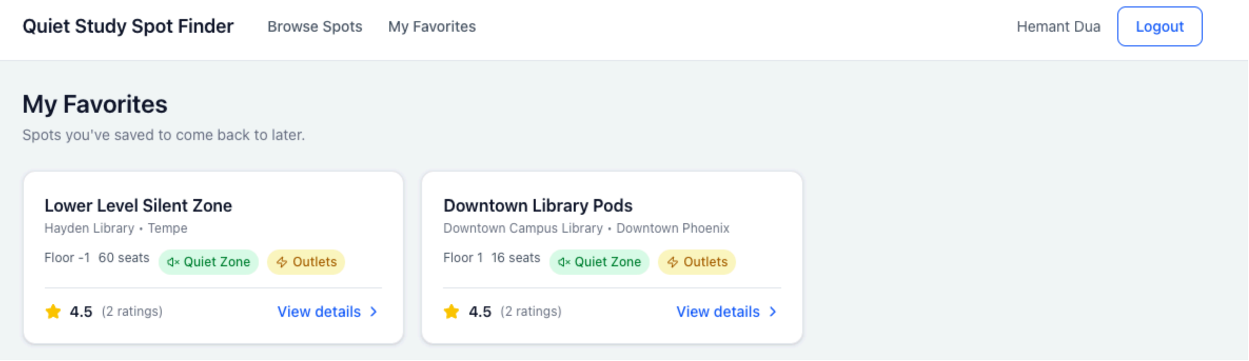
Lower Level Silent Zone
Hayden Library • Tempe
Floor -1 60 seats Quiet Zone Outlets
★ 4.5 (2 ratings) View details ›

MU Study Lounge
Memorial Union • Tempe
Floor 1 25 seats General area Outlets
★ 3.0 (1 rating) View details ›

3rd Floor Pods
Noble Library • Tempe
Floor 3 30 seats Quiet Zone Outlets
★ 3.0 (1 rating) View details ›

Common Area
Noble Library • Tempe
Floor 2 25 seats General area Outlets
★ 2.0 (1 rating) View details ›

[Browse Page]



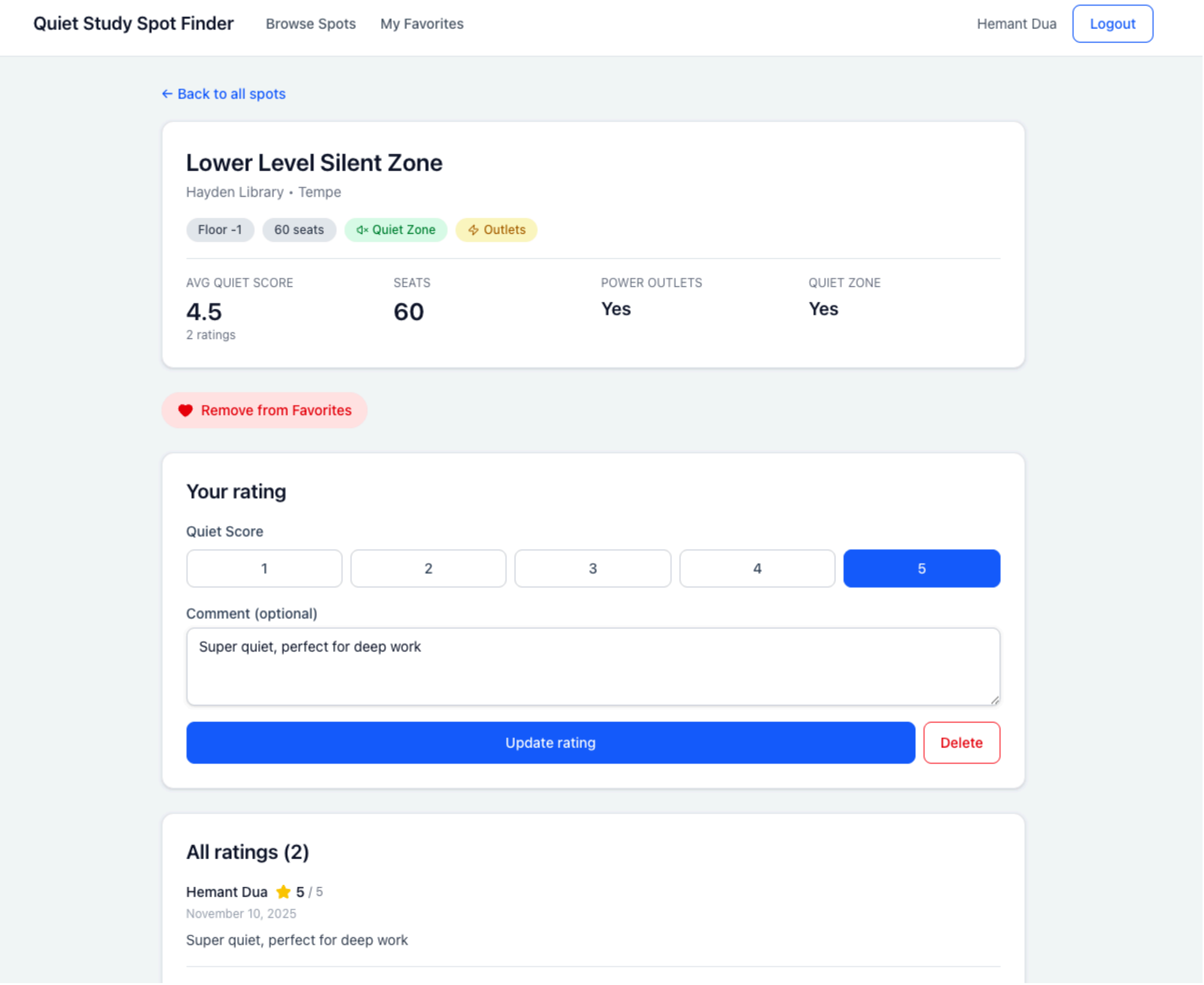
My Favorites

Spots you've saved to come back to later.

Lower Level Silent Zone
Hayden Library • Tempe
Floor -1 60 seats Quiet Zone Outlets
★ 4.5 (2 ratings)

Downtown Library Pods
Downtown Campus Library • Downtown Phoenix
Floor 1 16 seats Quiet Zone Outlets
★ 4.5 (2 ratings)

[Favorites Page]



Lower Level Silent Zone

Hayden Library • Tempe

Floor -1 60 seats Quiet Zone Outlets

AVG QUIET SCORE	SEATS	POWER OUTLETS	QUIET ZONE
4.5 2 ratings	60	Yes	Yes

Your rating

Quiet Score: 5

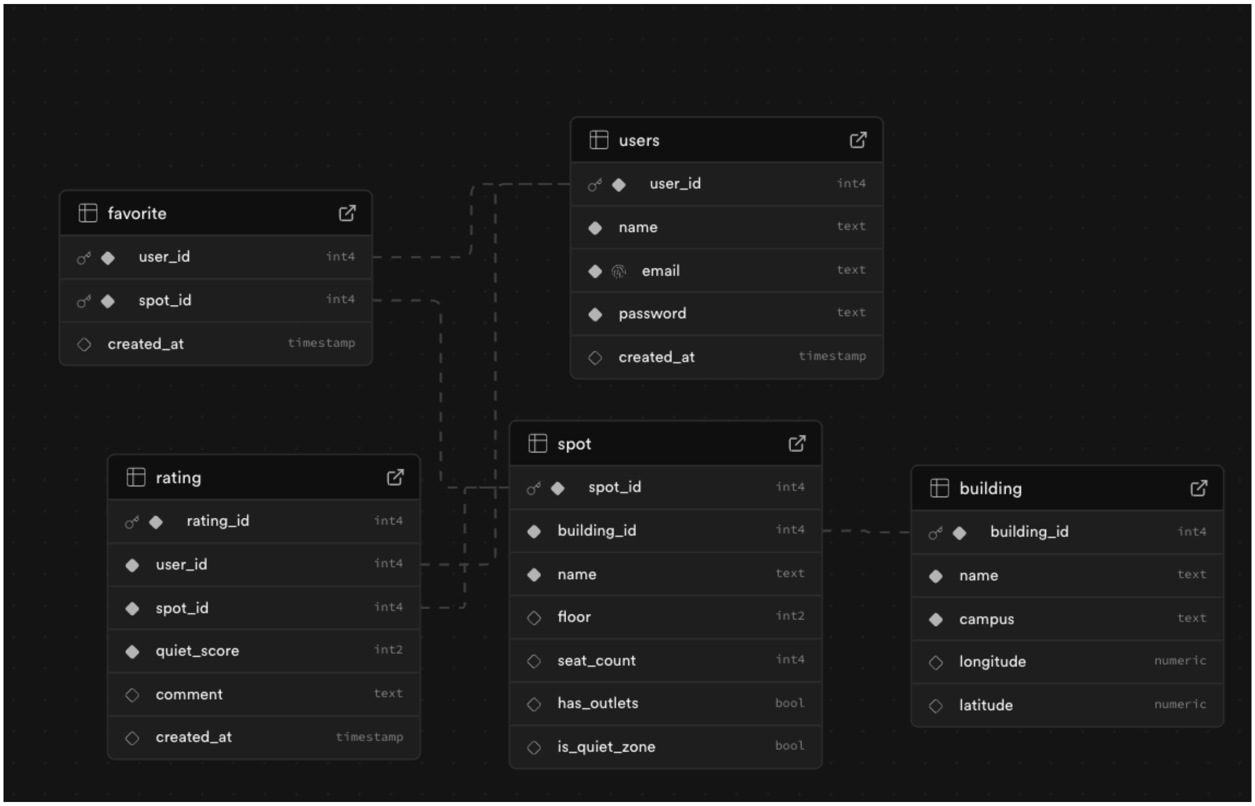
Comment (optional)

Super quiet, perfect for deep work

All ratings (2)

Hemant Dua ★ 5 / 5
November 10, 2025
Super quiet, perfect for deep work

[Spot Detail Page]



users	
◆ user_id	int4
◆ name	text
◆ email	text
◆ password	text
◇ created_at	timestamp

favorite	
◆ user_id	int4
◆ spot_id	int4
◇ created_at	timestamp

rating	
◆ rating_id	int4
◆ user_id	int4
◆ spot_id	int4
◆ quiet_score	int2
◇ comment	text
◇ created_at	timestamp

spot	
◆ spot_id	int4
◆ building_id	int4
◆ name	text
◇ floor	int2
◇ seat_count	int4
◇ has_outlets	bool
◇ is_quiet_zone	bool

building	
◆ building_id	int4
◆ name	text
◆ campus	text
◇ longitude	numeric
◇ latitude	numeric

[Database Structure]

Video Demonstration Link:

https://youtu.be/GPwKgFB_xsU

GitHub Repository Link:

https://github.com/hemantdua30/CSE412-FInal-Project

The sql files in this one are exports of the database from Supabase.

References:

https://www.postgresql.org/docs/

https://nextjs.org/docs

https://supabase.com/docs