Project Title: Rate My Apartment

Project Summary:

Our project is “Rate My Apartment.” Oftentimes, students have difficulty selecting off-campus housing options that are right for them due to a lack of information available to them. However, being able to see other residents’ and students’ prior experiences will help students make an informed decision and find an apartment perfect for their needs. Our app provides a centralized location for all of the reviews and opinions about various apartments to be congregated at. This application is useful because it is personal to the area around it and will be more comprehensive than Google Reviews on locations as it is specifically geared towards features unique to an apartment. It will be useful because it will have an easy-to-use and interactive interface that will make it easy for students to filter by their preferences for an apartment and put into consideration student needs. They can filter and rank apartments by noise level, location (how close it is to their classes), and if there is a bus stop near them as many college students use public transportation.

Through this app, students will be able to learn about the apartment’s management, noise levels, wifi strength and many more factors essential to a student's success. Similar sites and applications include ApartmentRatings. However, the difference between ours and theirs is that ours is geared towards students and accommodating various study habits and academic lifestyles. As for the data realness, to make sure that it is validated, we will compile apartments from Zillow and other housing sources and insert them into our database manually. If time permits, we will add functionalities so that users will also have the ability to suggest new locations for us to approve so that they can review them.

Data Stored

<u>Apartment Table</u>

Apartment ID: VARCHAR(50) PRIMARY KEY
Name: VARCHAR(50)
Management: VARCHAR(20)
State: ENUM(state abbreviations)
City: VARCHAR(20)
Zip Code: VARCHAR(15)
Address: VARCHAR(50)
1br: BOOLEAN
2br: BOOLEAN
3br: BOOLEAN
4br: BOOLEAN
MinPrice: INT
MaxPrice: INT
BusStop: BOOLEAN
Gym: BOOLEAN
6MonthLease: BOOLEAN
12MonthLease: BOOLEAN

Reviews Table
ReviewID: VARCHAR(50) PRIMARY KEY
ApartmentID: VARCHAR(50) FOREIGN KEY
OverallAvgScore: REAL
NoiseAvgScore: REAL
LocationAvgScore: REAL
WifiAvgScore: REAL

Comments Table
CommentID: VARCHAR(50) PRIMARY KEY
ApartmentID: VARCHAR(50) FOREIGN KEY
NumLikes: INT
DatePosted: DATETIME
Comment: VARCHAR(400)
OverallReviewerScore: REAL

Description of the functionality that your website offers

Rate My Apartment allows users to view information, ratings, and comments about different apartments. When opening up the website, there is a list of some limited number of apartments. Users can then filter the apartments based on certain characteristics, like number of bedrooms, management group, ease of access to bus stops, location information (state, city, zip code), minimum price, and maximum price. These would be some of the more simple features Rate My Apartment would have.

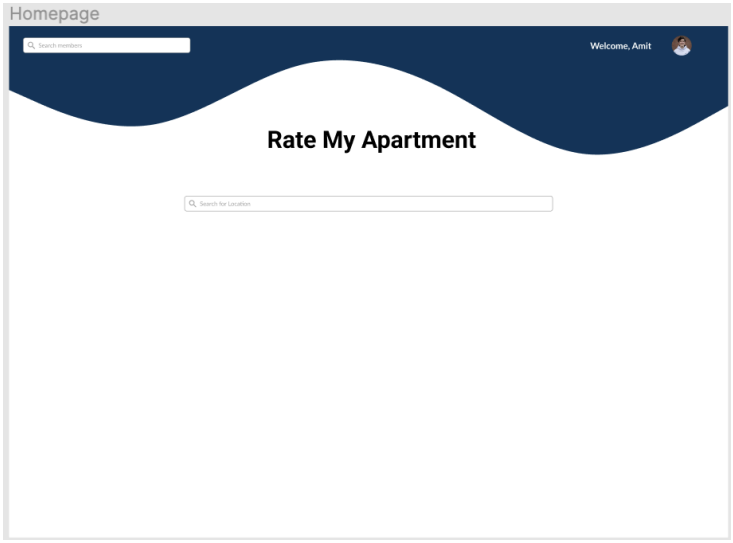
Users can also view apartment “pages” which display all relevant information about the apartment, which is the same as what the user can filter by. On this page, they can leave comments about their experiences with the apartment. These comments will originally be anonymous without the ability to edit or delete them. Additionally, on an apartment “page,” users can see an average rating for the apartment, as well as ratings in key categories (management, amenities, location, WIFI, apartment quietness). Filtering might be a more complicated query, especially since it would require “unioning” multiple resulting tables, but displaying the ratings on the pages would likely be an easier task since it is simply an average of all ratings that have been left in each respective category.

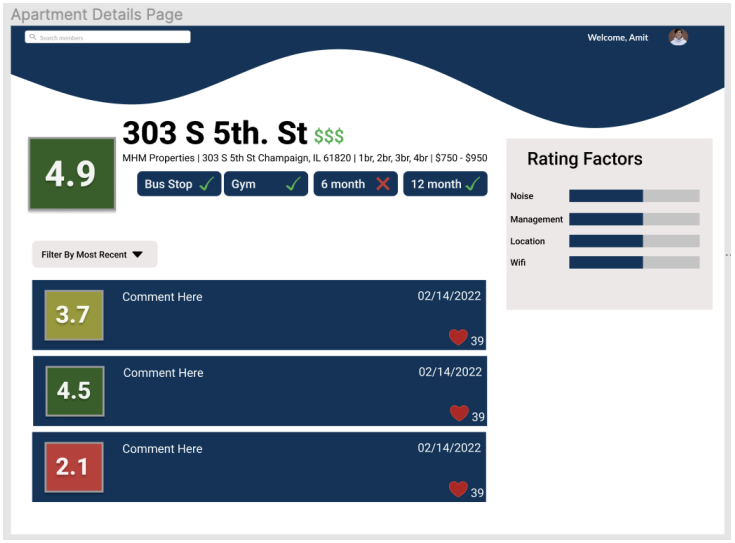
If we have no blockers during development, we also hope to have user roles. This would be supported by a log-in page authenticating users. Users, as well as admins, would be able to access all of the functionality above. Users and admins would also be able to edit or delete their comments. The only difference between the two is that admins would also be able to delete comments which do not meet guidelines. Implementing authentication and roles would be a more complicated feature since having special permissions will require some more thought and time. In addition to adding roles and account authentication, we would also allow users to “like” and “unlike” apartments to save and view for later.

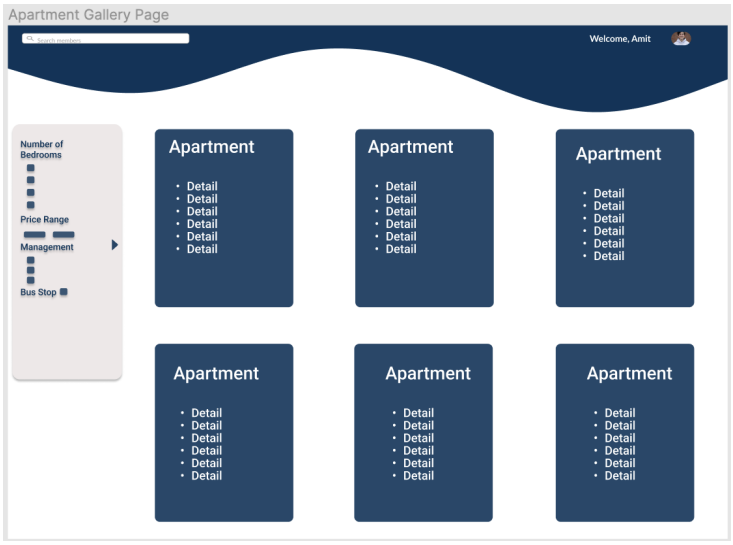
Finally, if we are able to exceed our development schedule, users would be able to upload photos of apartments and request to add apartment names. Additionally, the list view of liked apartments and filtered apartments would be paginated.

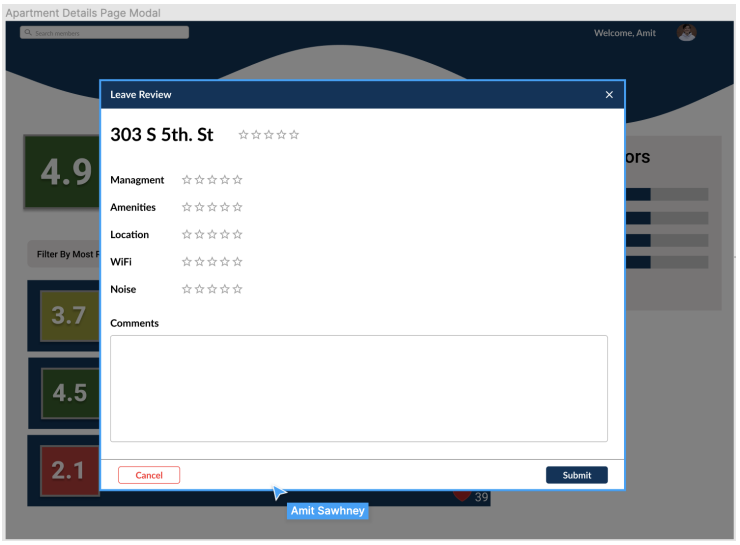
A good creative component (function) that can improve the functionality of our application would be to add an automatic location detector that will request the user's location and then use that location to find apartments nearby. We can implement this feature by using the Google Maps API.

Low fidelity UI mockup









Project Work Distribution

The following breakdown is tentative because we plan on working on these features together as much as possible, but in the case that our schedules do not work out, we plan on splitting the features up as such.

We have split this project into 3 sets of priorities (P0s, P1s, and P2s, respectively). The first set of priorities are key features we plan to accomplish by the end of the semester. The second set of priorities are features we could potentially get done within the semester, depending on how development goes for the first set of features and we have no blockers. Finally, the third set of priorities are stretch goals that can only be accomplished if development goes *extremely well* and we have a lot of additional time.

For the backend distribution, we plan on initially setting up the schema for the database together since it's something we all want to learn. We will then split up setting up different API endpoints. These are our P0s. The breakdown of who is responsible for which endpoint is as follows:

- **Insert** Comments - Archna
- **Query** Apartments Based on Key Characteristics (Filtering) - Ishaan
    - Minimum Price, Maximum Price, Number of Bedrooms, Bus Stop Access, Location, etc.
- **Query** Apartments Based on Name - Vasu
- **Insert/Update/Delete** Numerical Apartment Ratings - Amit
- **Query** Average Rating - Archna

Similarly, the breakdown for the corresponding frontend features is outline below:

- Text Input for Comments - Archna
- Filtering Panel - Ishaan
- Search Bar - Vasu
- Apartment Rating Display - Amit
- List of Apartments (Results from Searches/Filtering) - Archna & Ishaan
- Apartment Page Layout - Amit & Vasu

Given that development is going well, we hope to accomplish the following P1s for backend endpoints:

- **Insert/Query** OAuth ID's for Authentication Using Passport - Amit
    - Authentication (using Passport.js)
- **Update/Delete** Comments Corresponding to the User - Archna
- **Insert/Update/Delete** Liked Apartments - Ishaan
- **Query** User Information, like Liked Apartments - Vasu
- **Insert** Photos of Apartments - Archna

These features on the frontend will be divided as such:

- (Simple) Log-In Portal - Amit
- Like Button for Apartments - Ishaan

- Profile Button and Page for Users - Vasu
- Upload Image Button and Image Display for Apartments - Archna

For our stretch features (P2s), we have planned the backend tasks to be done by the following members of our team:

- **Insert/Delete** Admins - Amit
- **Update** Apartment Listings - Vasu

The frontend work for these P2s will be split as follows:

- Admin Portal to Add/Delete Admins - Ishaan & Archna