CS 411
Spring 2024

Final Project Report

PropertyHub

Divyam Arora, Mihir Shah, Abhiram Vodela, Sourya Sappa

May 6, 2024

1. Project Evolution and Modifications

Some of the changes we made from the final include not adding ratings and not adding the feature of houses next to certain landmarks. Some features we added were an even more advanced filtration system and a user-friendly system of interacting with properties.

2. Achievements and Shortcomings

We think our application achieved creating a user-friendly application. There were plenty of features that made it easy for users of our website to track properties on our website. One thing that we failed in doing is making our front end visually appealing but that's not an extremely fatal flaw.

3. Schema and Data Source Changes

During the course of development, we had to modify our data schema. The initial data received included unnecessary details, prompting us to streamline our database to better meet our project's needs.

4. Database Schema Adjustments

For our ER diagram, we changed multiple things. From our original original implementation, we removed the Sellers, Sales, and Search tables. We also added the audit_log table as well as the UserFavorites and UserProperties tables while removing the favorites attribute from the User table. The audit_log table is updated based on a trigger that adds an audit based on an update to a primary key of a propertyId. The UserProperties and UserFavorites tables are solely based on foreign keys of userId and propertyId to store user-specific information related to properties. This means all of these tables are weak entities. Additionally, the Address table is a weak entity as well as it only exists when the corresponding property exists. We also changed the primary key of the Address table to a specific ID because we changed our implementation to work based on latitude and longitude instead of a street address. We chose to make these changes for multiple reasons. We added tables for UserFavorites and UserProperties instead of keeping them as attributes because this allowed us to easily update and delete from these properties, making the application more efficient. If the database was made using MongoDB then an array would have made more sense but because it was made with SQL it makes more sense to keep them as separate tables. The reason we added the audit_log table is that we realized during development that keeping track of price changes is extremely important for a real estate website. We removed the Search table because it made more sense to keep that as a function within the javascript rather than involving SQL. In the future it would make sense to add a SearchHistory table which could store

searches made based on the userID foreign key. We believe that our updated design is the most suitable for the project. The only change would be to add the SearchHistory table.

5. Functionalities and Features

- (Removed) We originally planned to implement machine learning algorithms to provide personalized property recommendations to users based on their past search history, saved listings, and preferences. This functionality was proposed for extra credit opportunities but we were not able to implement this feature as we did not have enough data to train a model to provide useful results for the user.
- (Added) We did not plan to have user profiles but after the midpoint check-in with our TA, we received the recommendation to add the functionality of user profiles. For this, we implemented a new login/ signup page on our web app through which users could create their profiles and perform property searches on their individual login sessions. This allowed users to add property listings to their favorites, which they could view later in their profiles. In addition, the users had the ability to manage their specific listings by adding a new listing, viewing all their existing listings, deleting an existing listing, and modifying the price of an existing listing. The functionality was made such that each listing was associated with a specific user and they could manage their own listings only.

6. Advanced Database Programs Integration

- We made use of transactions as a part of our advanced database programs. The transactions played a very crucial role in managing the listings. Consider a case, where two users want to add a new listing with the exact same address. If both users add the listing at the exact same time, we would end up having duplicate listings in our database with different attributes. This is not good for the website in real-life scenarios. In order to avoid this situation, we made use of transactions. A transaction is initiated before attempting to insert the address into the Address table. If any error occurs during the transaction, the rollback function is called to revert any changes made within the transaction. If the address insertion is successful, the transaction is committed, ensuring that all changes are applied atomically.
- The property search feature on our website allows the user to filter properties in a specific mile radius. In order to select properties from our database within a specific mile radius, we made use of a stored procedure. This procedure takes user input as the radius of properties they want to search on. Since the Address table stores all the property addresses in the form of latitudes and longitudes, we

performed some mathematical calculations to define the range of latitudes and longitudes in that specific radius. The procedure then performs a select operation using those ranges and returns all the properties in the specific mile radius.

- Our other stored procedure provides the user with a price rating based on property prices within a 5-mile radius. This was done by doing something similar to the search procedure where we found all properties joined with addresses within the radius, which was determined through an equation. The procedure then finds the average price of those properties and then uses a switch case to compare the average price to the price of the desired property.
- Our triggers complement our application in several ways. One of our triggers automatically deletes a property from UserFavorites, Properties, and Address if the user chooses to delete it. This makes it so that the database is managed properly and only stores properties that actually exist. This prevents users from being able to see a property in both the search function and their favorites if it no longer exists. Our other trigger automatically adds to the audit_log table if a price is updated. Although this is not shown in the front end, it complements the application because it is extremely important to track price changes when it comes to real estate. Additionally, if a realtor feature is ever added to the website, they should be able to see important attributes such as price changes.

7. Technical Challenges and Resolutions

- Divyam: We hosted our web application on GCP and used a Virtual Machine (VM) to run the application. Whenever the server is running on a specific port and the VM is closed without closing the server, it keeps the node running. Therefore, when you try to run it on the same port again, it gives you an error that the port is already in use. To overcome this, we started running the application on a different port, and we kept getting the same error for different ports as well. After a point, there were multiple ports running on the same server and the web app became unresponsive. To fix this issue, we had to close the running port instances using the following commands - ps aux | grep node (command to view the currently running ports), and then kill node x (x is the node number). An advice would be to close the server of your web app before you stop working on the project.
- Mihir: One challenge we ran into was learning how to use Node.js and Express.js as none of us had ever used either framework prior to the project. This led to us facing several issues involving the integration of our backend with our frontend. For example, when dealing with the POST and GET requests that come with these frameworks, we were very focused on how to implement them as well as how to

pass variables to the backend. If a group ever runs into a problem like this, I recommend using a framework they are more comfortable with or learning how to use these frameworks properly before starting to code.

- Sourya: One technical challenge that we ran into was when implementing the filtering system. We ended up realizing that one of our stored procedures was messing up the functionality of the filters. In fact, the filters would not even get stored or recognized. This was particularly true with the way we were handling all of the property attributes, especially the null values. After debugging our code for a long time, we narrowed the issue down to the stored procedure. After modifying the stored procedure, the filtering system finally worked.
- Abhiram: One technical challenge that I ran into was when I was working on the login system, when a user inputted their passwords, it was not being hashed properly. After a grueling period of many trials and errors, I realized that I needed specific libraries that would do the hashing for me and after I understood how those libraries worked, I was able to make the login system.

8. Future Enhancements

- Features such as forums or community pages where users can discuss neighborhoods, share experiences, and provide tips could foster a community around the PropertyHub platform. This could lead to higher engagement as users can return to not only look at properties but also interact and engage with others. This would also give more transparency to the website and allow users to trust the website more. Community features provide a direct channel for feedback on the platform's services and features, allowing for continuous improvement based on user input.
- Integrating a rating system could significantly enhance the functionality and appeal of PropertyHub, making it a more comprehensive and user-friendly platform. These ratings can help future buyers gauge the quality of a listing beyond just the description. We could allow ratings based on various factors like safety, cleanliness, local services, and public transportation. On top of this, we could add functionality that shows how many reviews a user has made and an average rating score on their profile. This would increase community interaction with the website. Users could then possibly filter properties based on ratings.

9. Final Division of Labor

The implementation Sourya was in charge of was adding the filter system to the website. This allowed the user on the website to filter properties by price, square footage,

bedrooms, bathrooms, property type, and year built. Abhiram implemented the front end and back end for the login page. This included HTML and JavaScript logic. He also implemented the front end for the page where the user is greeted upon launching the website. Mihir implemented the logout functionality that would allow the user to log out of their profile. Divyam ended up implementing the part of the website where the user could add listings, favorite listings, and edit listings. All of these components were crucial to our website. The effectiveness of how we were able to implement these features means that we managed our teamwork in a clean and efficient way.