

- Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).

In our stage 1 proposal, we aimed to provide a real estate analytics platform that helps users better understand US housing market trends. In our final deliverable, our direction remained aligned with this goal, but our implementation philosophy shifted. Originally, we planned to develop a more advanced analytical tool, but we ended up creating a lighter-weighted and more user-friendly version.

Specifically, instead of focusing on advanced market indicators such as the turnover ratios, liquidity, and market forecasting, we emphasized more on metrics directly tied to housing prices and transaction volumes. We also prioritized clearer visualizations to make data more interpretable for normal users.

- Discuss what you think your application achieved or failed to achieve regarding its usefulness.

We think that our application has mostly achieved its usefulness. Our application allows users to query different housing data with filters such as location type, time range, property type, and metric type. We also implemented a ranking system, metric visualizations (e.g. line and bar charts), and an interactive map to help users grasp market patterns more easily than simply by viewing spreadsheets or tables on other housing platforms. Additionally, we allow users to upload their findings with a user reporting table, and let them favorite/ save different queries made by others on the platform with a favorite query and a favorite report table.

However, there are also some aspects where we hope to improve in the future. While our current platform may be suitable for users with limited economic backgrounds who require clearer and more accessible insights, it may not be as helpful for users with more related knowledge. Thus, if we introduce a dedicated section with more advanced indicators (e.g. turnover ratio, liquidity), we may improve the application's overall usefulness and broaden our target audience.

- Discuss if you changed the schema or source of the data for your application

We did not change the schema nor the source of our data. Our final application uses the same schema and dataset as the ones mentioned in previous stages.

- Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?

We did not change our UML diagram or table implementations. The final design is the same as our original design. We believe this design is suitable for our application because it supports the core features we implemented, including metric querying with keyword inputs and filters, visualization, ranking, user login and reporting, and favoriting queries or reports. Additionally, even if we later add more advanced indicators such as liquidity or turnover, or extend the functionality to support comparisons between multiple regions, the table implementations would still remain sufficient and wouldn't require design changes.

- Discuss what functionalities you added or removed. Why?

Functionalities added:

1. House data exploration + Interactive map: Users can input state, property type, and time range to search for recent house data. We also show the locations of house data queried on an interactive map.
2. Economic metric visualization: Users can input time range, location, property type, metric type, and visualization type to query a selected metric, including median sale price, median list price, median sale price per square foot, median list price per square foot, number of homes sold, new listings, and inventory. The search results are shown in both a list and the graph users desired. Users are also allowed to favorite queries so that they can easily view the chart of the selected favorite queries.
3. City price ranking: Users can input city, state, property type, time range, and metric type to search for the metric value of that city and the state and nation ranking of the city compared to the amount of data entries during that period.
4. User report and favorite reports: Users can input report id(for update only), location, property type, prices, time range, and square feet of houses to add and update data to user reporting data. After every insert and update, the detailed information of reports will immediately show on the frontend as a table. To delete existing reports, simply select reports you want to remove and click the “Delete” button. Users can also search for updates made by other users by filling out the searching form. If they find data they want to keep, they can select the search results and add them to their favorite for easier access.

### Functionalities Removed:

1. Advanced Economic Indicators (e.g. liquidity, turnover ratio, growth): We removed these economic metrics because they require more calculations. Instead, we prioritized more intuitive indicators such as prices and volumes and data visualization to better support users without a strong economic background.
  2. Two cities comparison and visualization: We did not include this feature in our final application due to time constraints. The intended visualization (pie chart) for this section was removed as well.
  3. Forecast tool: This feature was also removed because of time constraints.
- Explain how you think your advanced database programs complement your application.

We think the stored procedures we implemented were very helpful. In our application, we had used two complex SQL queries that were very long (over 100 lines), and by using stored procedures, we were able to store them in our GCP database instead of adding them in our backend. This makes our backend code cleaner and since we only need a few lines of code to call our procedures, this makes our backend much easier to maintain. In addition, the stored procedures help reliability as they reduce the number of roundtrip messages that are sent between the backend and the database. They also help with performance since the database is able to optimize and execute the stored procedure far faster than a backend written in Python could.

For triggers, we used them to validate inserts and updates on the User\_Report table. One trigger checks whether the sale price, list price, sold time, list time, and square feet are reasonable after an insert. If any value is invalid, the trigger will delete the inserted row. Another trigger checks before an update. If there is unreasonable data, it prevents the update and outputs an error message to the terminal.

For transactions, I add the transaction button on the User Reporting page. And this transaction will help the system filter out some invalid data such as some data that might already be stored inside the database (with user\_name start with "A") or the sale\_price might be too high that can be deemed as a malicious input.

- Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future

team could use this as helpful advice if they were to start a similar project or where to maintain your project.

Andrew: One technical challenge I've met is about using Git. Since I'm not really familiar with Git but still want to try to do some fancy operation to protect the main branch. So, I mistakenly set up the rule that forbids others except the admin to push or send merge requests to the main branch. Then, I downgrade everyone from admin to maintain. This causes nobody to access the main branch. Fortunately, when I ask for TA's help, TA upgrades us to admin in only a few hours. I think this gives me a lesson to think carefully before changing every setting in any project or collaboration.

Sam: One technical challenge we encountered was that we couldn't import our csv files into GCP properly. We later found out this was an issue +with Windows vs. Unix mismatches, so we wrote a script to convert the files to Unix. After that, we were able to import the csv files successfully. Another challenge was that one of the csv files we needed to upload was very large (around 6 million rows), and it often failed to be uploaded. We later successfully uploaded it after repeatedly uploading it. We also found that we couldn't upload a SQL file through the GCP GUI if it contained code like "DELIMITER //". We solved this issue by copying and pasting the code into the GCP console.

Hsin-Yi Lu: One technical challenge I faced was the design of favorite queries. At first, I was thinking of whether users should favorite queries or the searching results returned by those queries. I initially decided to let users favorite the results, since the search form is quick to fill out so saving the queries seemed less meaningful. This approach allowed users to directly access the exact data they wanted and generate charts. I designed the charting function of the favorite results by letting users select multiple saved results with the same location, property type. The chart uses the minimum start date and maximum end date among the selections. However, after completing the function, I realized that favoriting queries would still be more effective since running a saved query can provide a broader dataset with a larger time range.

Chinmay: One technical challenge I faced was in implementing the map visualization. The majority of the work related map visualization came from integrating with a React library with poor documentation. Due to the poor documentation, it took a lot of exploring the codebase of the library, checking out issues on GitHub and samples on StackOverflow before I was able to figure out

how to render all the markers correctly. In addition, I am not an expert in React or Javascript so figuring out how to use state and hooks without breaking the Rule of Hooks in addition to leveraging the library was also difficult. To fix my issues, I mainly turned to GitHub issues and examples of React states and hooks in the documentation to understand the mechanics and how to use it properly.

- Are there other things that changed comparing the final application with the original proposal?

Since our features are quite different from the original proposal, we also changed up the layout of our application a bit. We believe this improves the user experience.

- Describe future work that you think, other than the interface, that the application can improve on.

In the future, we can extend the functionality of the application to support more advanced economic indicators such as liquidity, turnover ratios, and growth. We can also implement the market forecast as mentioned in the original proposal. We can also add other features such as comparisons between multiple regions. Since our current version doesn't allow users to delete their account, we can also add that feature in our future version as well.

- Describe the final division of labor and how well you managed teamwork.

Andrew Liu (al92): Log In + Sign Up + Transaction + import data + write sql command

Syuan-Fu Hwang (syuanfu2): metric query + visualization + price ranking (w/ stored procedure) + import data + gcp setup

Hsin-Yi Lu (hsinyil3): User Report (w/ trigger and CRUD) + Favorite Query + Favorite Report + UI design

Chinmay (cud2): write sql command + add map visualization