REAL ESTATE MARKET RESEARCH DASHBOARD

Ronald Daley
DSCI 551 - Final Report

Table of Contents

Introduction

I have worked full-time in the real estate industry for four years, helping buyers and sellers start new chapters in their lives. I am also passionate about real estate investing and aspire to one day leverage data science skills to research and acquire properties, ultimately becoming a successful real estate investor.

The **Market Research Dashboard** is a data-driven tool designed to empower real estate investors by providing easy access to critical housing and demographic data. This dashboard tool allows real estate investors to interact with real estate-related databases to make data driven decisions based on recent market activity when investing in a new market. This tool will convert questions from investors into postgreSQL queries for efficient data retrieval, analysis, and modification.

This project integrates a natural language interface with real estate datasets, allowing users to interact with databases through human-like queries. The tool leverages Streamlit for its user interface and uses PostgreSQL for backend data management.

Team Members

This project was individually completed by Ronald Daley, covering all aspects of design, development, and deployment.

Planned Implementation

The project was designed to include three major data sources from Zillow: for sale inventory, new listings inventory, home value index, and median listing price. Key functionalities included schema exploration, natural language query execution, and data modification capabilities. All interactions were to be handled via a web app with ChatDB or similar NLP models converting natural language to MongoDB/PostgreSQL queries.

Core Functionalities:

Schema Exploration

- Retrieve a list of databases and tables (`SHOW DATABASES`, `SHOW TABLES`)

- Display sample data from tables (`SELECT * FROM table LIMIT n`)
- Show table structure (`DESCRIBE table` or query `information_schema`)

Query Handling

- Accept NL queries and convert them to SQL queries
  Support:
  - Select with projection: `SELECT column1, column2 FROM table WHERE condition`
  - Aggregation: `GROUP BY`, `HAVING`, `COUNT`, `SUM`, `AVG`
  - Sorting and pagination: `ORDER BY`, `LIMIT`, `OFFSET`
  - Joins: `INNER JOIN`, `LEFT JOIN`, etc. for multi-table queries

Data Modifications

- Support for `INSERT`, `UPDATE`, and `DELETE` operations via natural language

---

Architecture Design

My dashboard is deployed using the Streamlit platform, an open-source app framework that enables the creation of web applications entirely in Python. Streamlit allows me to efficiently and seamlessly develop a user-friendly web application and is able to read user questions from my back end database, as it seamlessly integrates both frontend and backend components within Python scripts. The detailed architecture of my web application is illustrated in Figure 1.

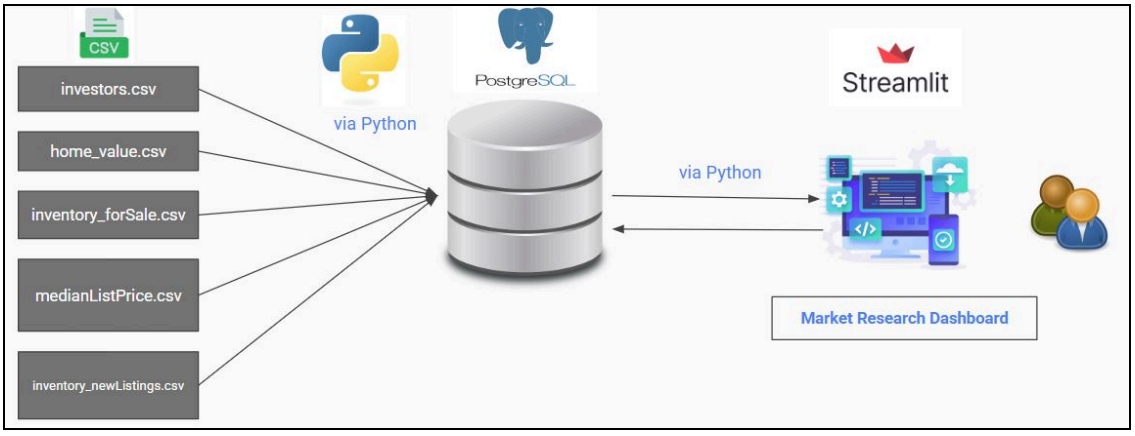


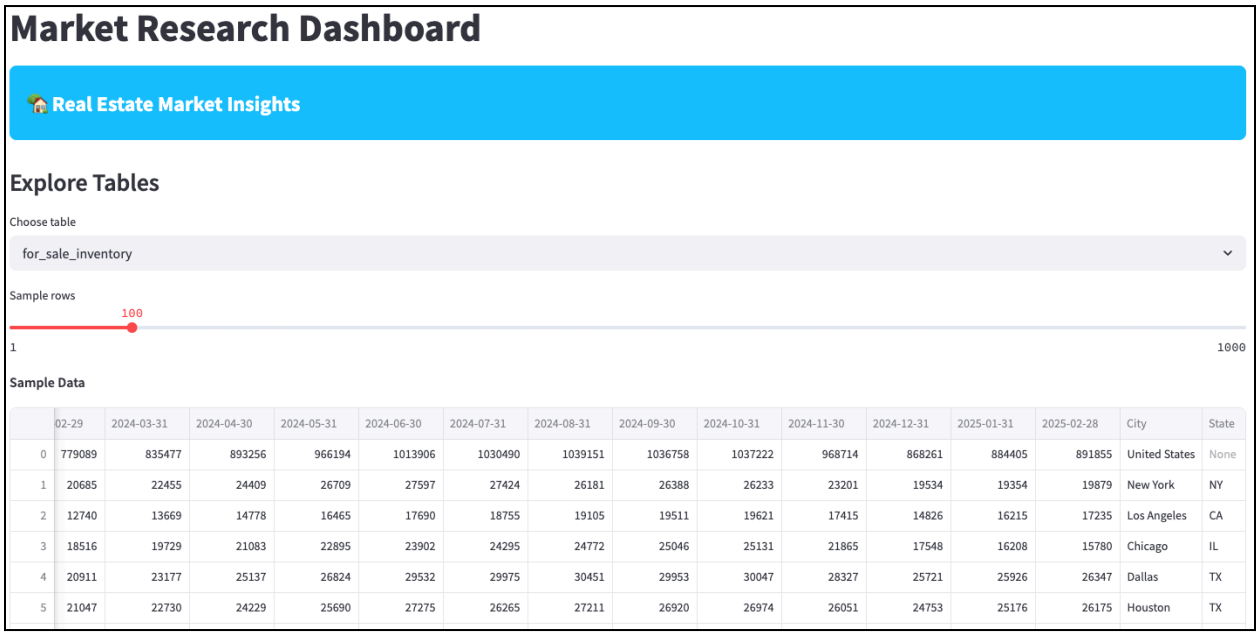
Fig. 1 Web App Architecture Diagram

The following architecture is the high-level workflow of the **Market Research Dashboard:**

1. Users interact with the system through a web-based dashboard built in Streamlit.
2. Natural language queries are passed through a query interpreter (ChatDB).
3. The interpreter translates queries into MongoDB or SQL syntax based on the data type.
4. Results are fetched from the database and displayed in the frontend.
5. Insert, update, and delete operations are also supported via NL input.

Components of Web Application

My web application includes one landing page and five sub-pages, each offering a distinct service (see Fig. 2). The landing page provides an introduction to the application and allows investors to explore the database schema. Users can navigate between the five pages using the dropdown menu on the left side of the screen.



	02-29	2024-03-31	2024-04-30	2024-05-31	2024-06-30	2024-07-31	2024-08-31	2024-09-30	2024-10-31	2024-11-30	2024-12-31	2025-01-31	2025-02-28	City	State
0	779089	835477	893256	966194	1013906	1030490	1039151	1036758	1037222	968714	868261	884405	891855	United States	None
1	20685	22455	24409	26709	27597	27424	26181	26388	26233	23201	19534	19354	19879	New York	NY
2	12740	13669	14778	16465	17690	18755	19105	19511	19621	17415	14826	16215	17235	Los Angeles	CA
3	18516	19729	21083	22895	23902	24295	24772	25046	25131	21865	17548	16208	15780	Chicago	IL
4	20911	23177	25137	26824	29532	29975	30451	29953	30047	28327	25721	25926	26347	Dallas	TX
5	21047	22730	24229	25690	27275	26265	27211	26920	26974	26051	24753	25176	26175	Houston	TX

Fig. 2 Landing page screenshot

Web App & Database Implementation

1. Web App Functionalities

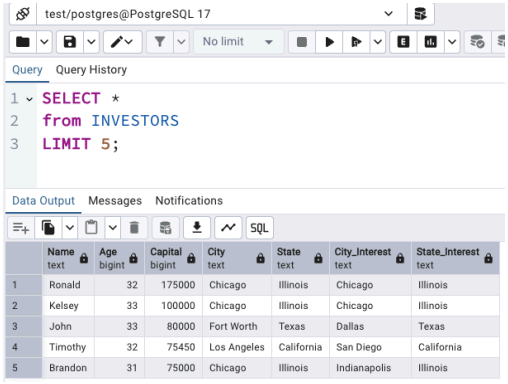
a. Explore schema and data (show collections/tables, attributes)
b. Execute NL queries (find, aggregate, JOIN, etc.)
c. Insert, update, and delete data through natural language
d. Interactive dashboard for querying and visualizing housing market trends

2. Tech Stack

- **Backend**: Python
  - **Libraries**: `pandas`, `SQLAlchemy`, `pymongo`, `requests`, `openai`,`psycopg2`, `streamlit`, `os`
- **Database**: PostgreSQL
- **Data:** Zillow housing data
- **Frontend**: Streamlit

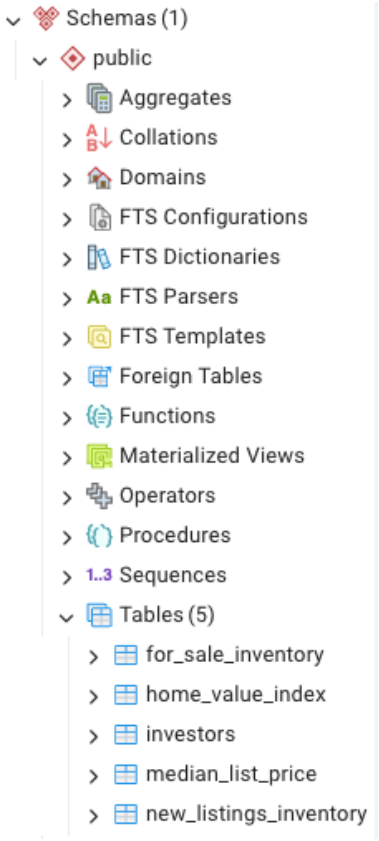
3. Database Implementation

a. PostgreSQL

i. 

	Name text	Age bigint	Capital bigint	City text	State text	City_Interest text	State_Interest text
1	Ronald	32	175000	Chicago	Illinois	Chicago	Illinois
2	Kelsey	33	100000	Chicago	Illinois	Chicago	Illinois
3	John	33	80000	Fort Worth	Texas	Dallas	Texas
4	Timothy	32	75450	Los Angeles	California	San Diego	California
5	Brandon	31	75000	Chicago	Illinois	Indianapolis	Illinois

b. **Database Schema & Tables**

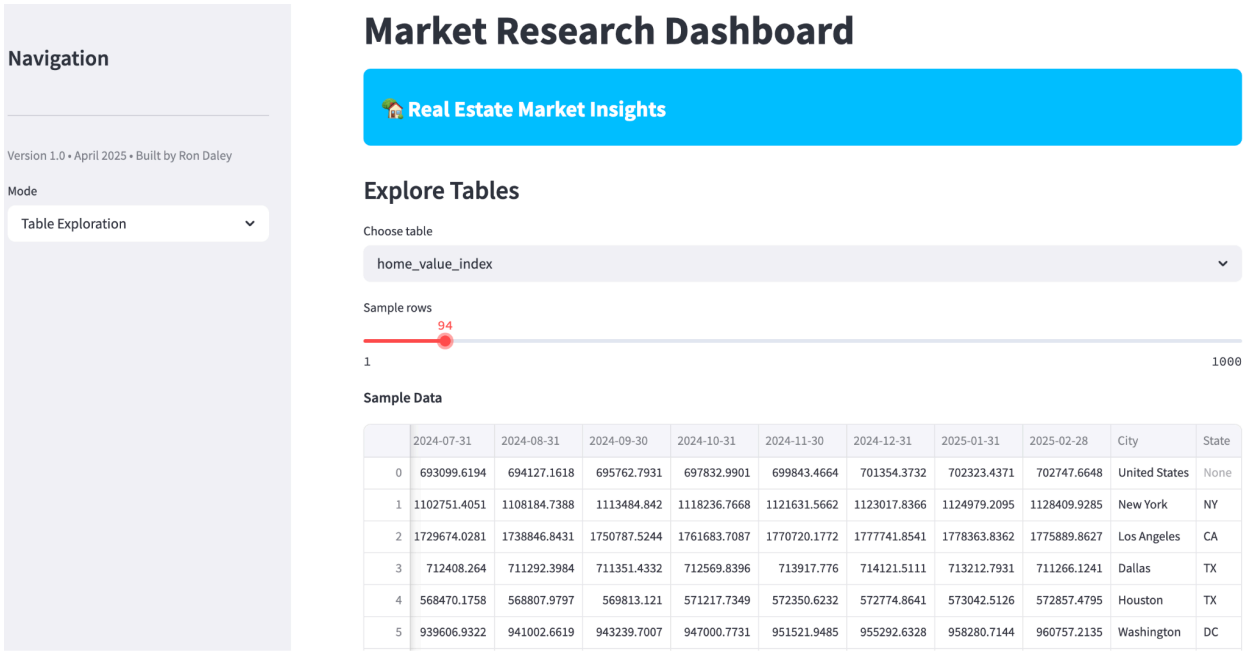
i. 

ii. **<u>Table Descriptions</u>**

1. **For-Sale Inventory:** The count of unique listings that were active at any time in a given month.
2. **Home Value Index:** A measure of the typical home value and market changes across a given region and housing type. It reflects the typical value for homes in the 35th to 65th percentile range.
3. **Investor:** List of investors
4. **New Listings:** Indicates how many new listings have come on the market in a given month.
5. **Median List Price:** The median price at which homes across various geographies were listed.

4. Web App Implementation

Streamlit dashboard



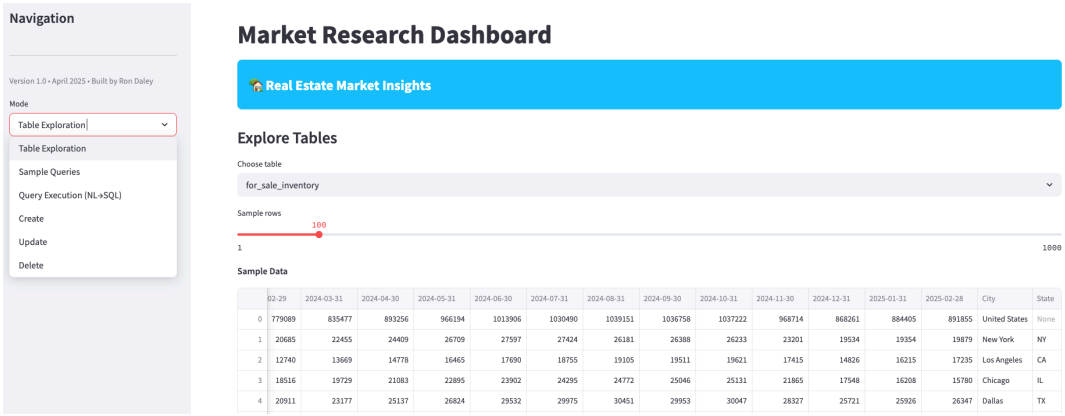
	2024-07-31	2024-08-31	2024-09-30	2024-10-31	2024-11-30	2024-12-31	2025-01-31	2025-02-28	City	State
0	693099.6194	694127.1618	695762.7931	697832.9901	699843.4664	701354.3732	702323.4371	702747.6648	United States	None
1	1102751.4051	1108184.7388	1113484.842	1118236.7668	1121631.5662	1123017.8366	1124979.2095	1128409.9285	New York	NY
2	1729674.0281	1738846.8431	1750787.5244	1761683.7087	1770720.1772	1777741.8541	1778363.8362	1775889.8627	Los Angeles	CA
3	712408.264	711292.3984	711351.4332	712569.8396	713917.776	714121.5111	713212.7931	711266.1241	Dallas	TX
4	568470.1758	568807.9797	569813.121	571217.7349	572350.6232	572774.8641	573042.5126	572857.4795	Houston	TX
5	939606.9322	941002.6619	943239.7007	947000.7731	951521.9485	955292.6328	958280.7144	960757.2135	Washington	DC

Fig. 3 Streamlit Web Application Screenshot

5. Dashboard User Experience (Web App UX)

a. Landing page

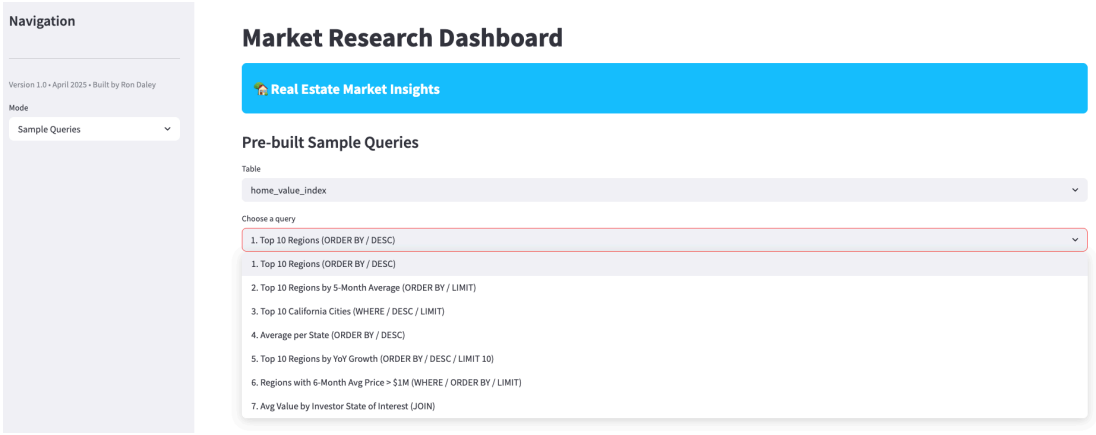
i.



	02-29	2024-03-31	2024-04-30	2024-05-31	2024-06-30	2024-07-31	2024-08-31	2024-09-30	2024-10-31	2024-11-30	2024-12-31	2025-01-31	2025-02-28	City	State
0	779089	835477	893256	966194	1013906	1030490	1039151	1036758	1037222	968714	868261	884405	891855	United States	None
1	20685	22455	24409	26709	27597	27424	26181	26388	26233	23201	19534	19354	19879	New York	NY
2	12740	13669	14778	16465	17690	18755	19105	19511	19621	17415	14826	16215	17235	Los Angeles	CA
3	18516	19729	21083	22895	23902	24295	24772	25046	25131	21865	17548	16208	15780	Chicago	IL
4	20911	23177	25137	26824	29532	29975	30451	29953	30047	28327	25721	25926	26347	Dallas	TX

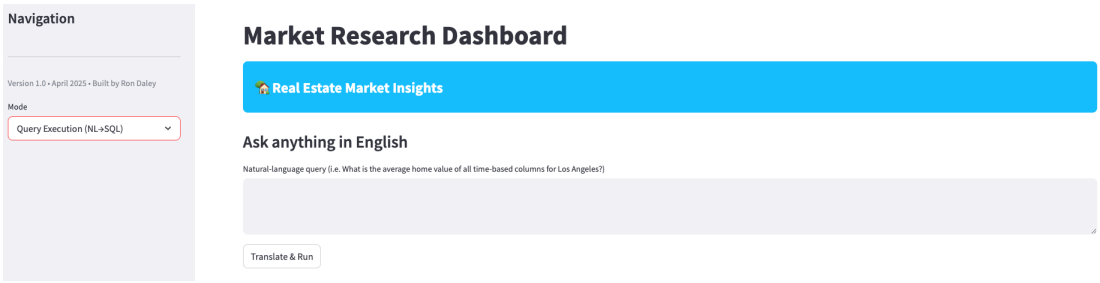
b. Sample Queries page

i.



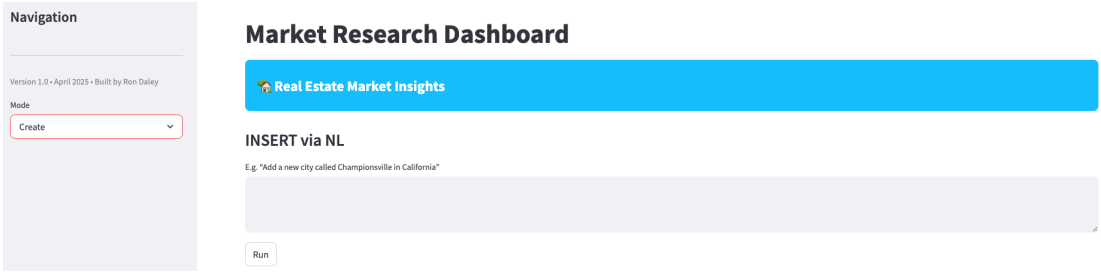
c. Query Execution (NL to SQL) page

i.

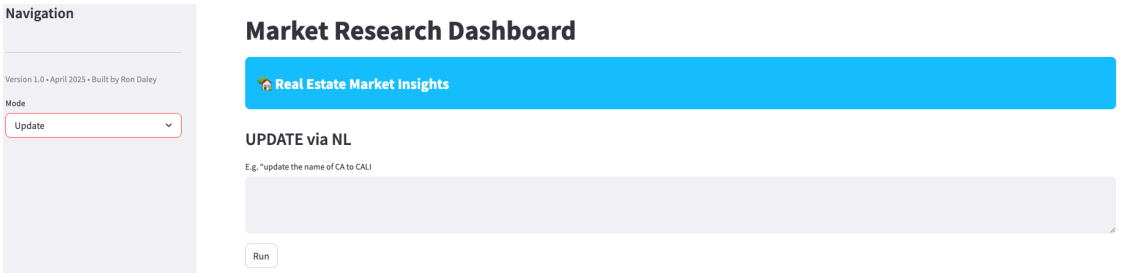


d. Create page

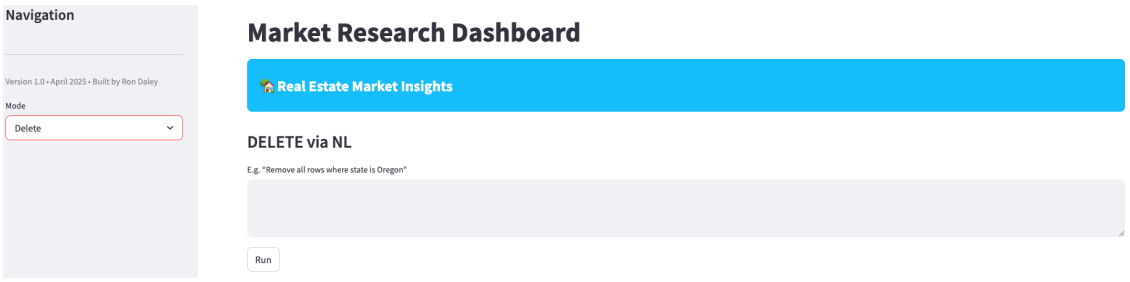
i.



e. Update page

i. 

f. Delete page

i. 

Learning Outcomes

This project significantly deepened my understanding of SQL database structures, real estate market datasets, and the integration of natural language processing (NLP) tools within data-driven interfaces. I gained hands-on experience obtaining real estate market data, cleaning and loading time-series data, and designing efficient database schemas to support complex queries. Additionally, I developed the skills needed to build a full-stack web application, bridging the gap between backend logic and a responsive, user-friendly frontend.

Beyond the technical skills, this project also enhanced my ability to think critically about data accessibility and user experience. Working with real-world datasets taught me the importance of data quality, consistency, and scalability. I also learned how to translate user requirements into practical features, balancing performance, usability, and functionality. These experiences have better prepared me for future roles in data science and analytics, particularly in environments that require seamless integration of data engineering and user-centered design.

Challenges Faced

One of the most significant challenges I encountered during the project was transitioning the database from a NoSQL solution (MongoDB) to a relational database (MySQL). Toward the end of the development phase, I learned that my upcoming internship would involve working with a data lake built on relational database technologies. I strategically decided to shift from MongoDB to MySQL to better align my project with the tools and systems I'll be using this summer.

This transition was far from trivial—it required extensive rework, including redesigning parts of the database schema, rewriting backend logic to accommodate SQL-based operations, and ensuring data consistency throughout the application. To further solidify my understanding of relational databases, I also took the initiative to learn PostgreSQL, gaining hands-on experience with its advanced features and syntax differences.

Beyond the backend, I faced the challenge of developing a user-friendly UI in Streamlit that seamlessly integrated these changes. I also had to ensure that all functional requirements were met, particularly those involving natural language processing, which required additional effort to fine-tune performance and usability within the new database structure.

Conclusion

The Market Research Dashboard is a robust and scalable real estate market analysis solution. It simplifies interaction with complex datasets by enabling natural language querying and delivers an intuitive, customizable interface tailored to user needs. This tool enhances data accessibility and empowers users to derive actionable insights without requiring technical expertise in database querying.

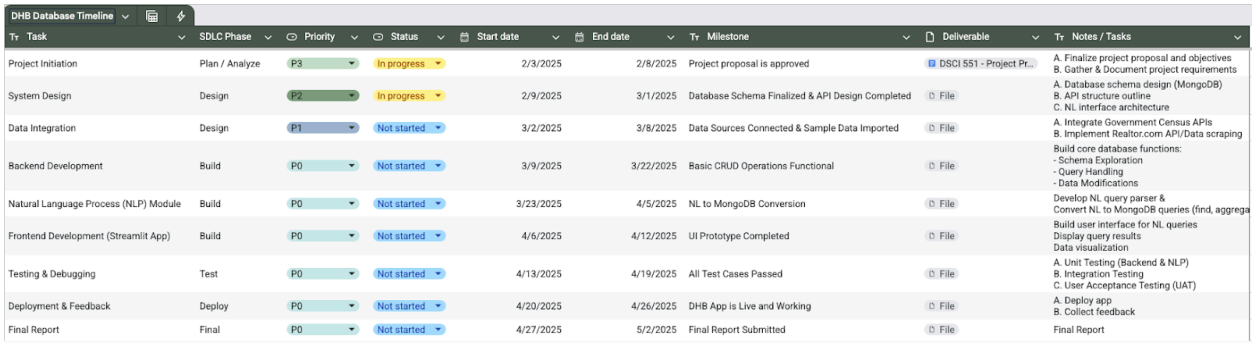
Future Scope

In the future, there are several opportunities to enhance the functionality and impact of the Market Research Dashboard. One key area is the integration of additional demographic and economic statistics, such as crime rates, employment data, and population trends, which would provide users with a more comprehensive view of the real estate landscape. Enhancing the user experience and dashboard performance is also a priority, with plans to optimize loading times, streamline navigation, and improve overall responsiveness.

Further developments include adding interactive maps to visualize geospatial data and potential integration with visualization tools like Tableau for advanced analytical capabilities. Expanding the natural language processing (NLP) model to support more complex queries and broader linguistic variations will significantly improve usability and accessibility. Incorporating more diverse datasets will also help extend the platform's reach and analytical power across different regions and property types.

Appendix

- **GitHub Repository link:**
  - https://github.com/radaley1906/dsci551_final
- **Google Drive Link (Project Code):**
  - https://drive.google.com/drive/folders/1y2ApCcwplcZuJkWTy-6gslN7Krclixko?usp=sharing
- **Full List of Requirements:**
  - https://docs.google.com/document/d/1qd-S46bjl3lkbZs98lAYasZzZzwDA09Dt-F1vPKbObI/edit?tab=t.0
- **Final Project Demo:**
  - https://docs.google.com/presentation/d/1rrwcptagOpgs6M8lIP3Ed8Gucl4mi1EW-a_NH2MCo6U/edit?usp=sharing
- **Project Timeline:**
  - https://docs.google.com/spreadsheets/d/1-ge6lWMzC7-TCVMF5dGbl1z1nnS_xWrP353anCWY_HY/edit?gid=1771400549#gid=1771400549



DHB Database Timeline

Task	SDLC Phase	Priority	Status	Start date	End date	Milestone	Deliverable	Notes / Tasks
Project Initiation	Plan / Analyze	P3	In progress	2/3/2025	2/8/2025	Project proposal is approved	DSCI 551 - Project Pr...	A. Finalize project proposal and objectives B. Gather & Document project requirements
System Design	Design	P2	In progress	2/9/2025	3/1/2025	Database Schema Finalized & API Design Completed	File	A. Database schema design (MongoDB) B. API structure outline C. NL interface architecture
Data Integration	Design	P1	Not started	3/2/2025	3/8/2025	Data Sources Connected & Sample Data Imported	File	A. Integrate Government Census APIs B. Implement Realtor.com API/Data scraping
Backend Development	Build	P0	Not started	3/9/2025	3/22/2025	Basic CRUD Operations Functional	File	Build core database functions: - Schema Exploration - Query Handling - Data Modifications
Natural Language Process (NLP) Module	Build	P0	Not started	3/23/2025	4/5/2025	NL to MongoDB Conversion	File	Develop NL query parser & Convert NL to MongoDB queries (find, aggrega
Frontend Development (Streamlit App)	Build	P0	Not started	4/6/2025	4/12/2025	UI Prototype Completed	File	Build user interface for NL queries Display query results Data visualization
Testing & Debugging	Test	P0	Not started	4/13/2025	4/19/2025	All Test Cases Passed	File	A. Unit Testing (Backend & NLP) B. Integration Testing C. User Acceptance Testing (UAT)
Deployment & Feedback	Deploy	P0	Not started	4/20/2025	4/26/2025	DHB App is Live and Working	File	A. Deploy app B. Collect feedback
Final Report	Final	P0	Not started	4/27/2025	5/2/2025	Final Report Submitted	File	Final Report