

## **CS 411 Final Project Report**

# **AirDnB: All-in-One Platform**

**Your One-Stop Platform for Finding the Best AirBnB in New York City**

**Team 058 - dbQuest**

**Ayushe Nagpal, Reva Jethwani, Achintya Sanjay, Karan Shah**

## **1. Changes from the Original Proposal**

Our project stayed true to its original concept of being a comprehensive platform to view AirBnBs in NYC holistically integrated with crime data, transportation, and amenities. However, originally it was broader to explore the options through filters on an interactive map. So users would be able to put in filters of their categories which would limit results on the map. However, comparing the final application with the original proposal, the final implementation focused more on specific user inputs for Airbnb URLs, with functionality centered around finding nearby restaurants, subway stations, and crime data within proximity. This shift allowed for a more personalized user experience, although it narrowed the broader analytical scope initially proposed.

## **2. Achievements + Failures**

The application succeeded in providing a highly user-specific experience, allowing individuals to assess the safety and convenience of their chosen Airbnb locations by analyzing nearby amenities and transportation. This site-specific information allowed users to make informed decisions about their accommodation choices based on their personal safety and convenience criteria. However, the application did encounter some limitations in its current functionality. One significant shortcoming was its inability to perform a broader, multi-neighborhood search using various filters simultaneously. This feature would have allowed users who are in the initial stages of planning their stay to efficiently explore and compare different areas within New York City. This absence restricted the application's utility, particularly for users who were not yet certain about the specific area they wished to stay in and were interested in conducting a comparative analysis of various neighborhoods based on multiple criteria. Enhancing this aspect of the application could potentially transform it into a great tool for anyone looking to optimize their stay in New York City.

## **3. Schema or Data Source Changes**

Our data sources changed multiple times over the course of this project. This was because we realized after running different advanced queries on our data, that some of the data related to Crime Log was outdated as the query would not display anything about new crimes committed. Thus, we had to resource our data table to a more recent Crime database from the internet. Also, our Subway Station dataset did not have Subway Stations in one of the boroughs of New York (Staten Island), so we resourced that dataset as well. The schema never changed for any of the datasets as all of them had the same important data we were looking for - an ID to serve as a primary key, a more common identifier like a name, and latitude and longitude points. The

general nature of the schema we looked for made it easy for us to move forward with our intended functionality regardless of the problems we have with the dataset itself.

#### **4. ER Diagram Changes**

The changes we made to our ER diagram were because our first diagram was not correctly created following ER guidelines. After discussing with our TA how we misrepresented the relationships between certain entities on the diagram, we had to go back and fix how it showed up on the table. After that, there were no more changes to the diagram because our schema never changed and neither did the data we were looking to use for our project. Even though our data sources changed, the relationships and overall general structure of the tables stayed the same for the ER diagram.

#### **5. Functionality Changes**

We added the functionality to input an Airbnb URL directly, which tailored the application more closely to individual needs. Conversely, we removed the broader filtering capabilities that would have allowed users to explore multiple listings simultaneously based on various safety and amenity criteria on a Google Maps interface. This shift was due to time constraints and technical challenges in implementing dynamic map updates with extensive filters through the Google Maps API.

#### **6. Advanced Database Programs**

Our advanced database programs were a transaction, stored procedure, trigger, and constraints.

##### **Transactions:**

Our transaction was placed on the main functionality of the project. When a user submits a Airbnb link, a transaction was placed on all the queries ran related to that Airbnb. This meant that if one of the queries failed to display information (ex: the restaurant query failed to display the x closest restaurants for whatever reason) then all the other queries would be locked as they were part of the transaction too.

##### **Trigger:**

Our trigger was placed on the user related portion of the website. When a user tried to modify their username in the modify account information section, a trigger would be activated. We placed this trigger because we didn't want the user to be able to modify their username. Any time a username is tried to update in the query, the trigger is activated.

**Constraints:** We defined the appropriate primary keys, and foreign keys in our application which played an essential role in the entire structure of the schema.

**Stored Procedures:**

- Our stored procedures calculated the number of subway stations, restaurants, and Airbnb's issues in each borough around New York City
- Since the calculations were the same for every borough, we stored that procedure and called the procedure with the borough we needed it for whether we were looking for Airbnb's, restaurants, or subway stations.

**7. Technical Challenges****a. Achintya**

**GCP Set-Up and Management.** Setting up the project on GCP was challenging and required us to navigate through extensive documentation and tutorials. We faced a few hurdles, such as understanding the best practices for configuring our environment and managing resources, and getting everyone on the same page. We made mistakes along the way like not realizing we had to shut down instances when not in use, which led to the rapid depletion of our allocated credits. This experience taught us valuable lessons about cloud resource management and the importance of meticulous attention to operational details in cloud settings.

**b. Ayushe**

**Data Preprocessing and Design Decisions.** One of the technical challenges we faced was understanding the significance of data preprocessing and the necessity of thorough planning before interacting with the data. We realized that the quality of output heavily depends on the quality and structure of input. Early in the project, we mapped out our data processing steps, which included data cleaning, selecting relevant attributes, and structuring our data to fit our application's needs. This planning phase was crucial and brought attention to the numerous design decisions that need consideration in a data-intensive application. For instance, deciding how to handle missing values or determining the granularity of crime data significantly impacted our application's responsiveness and accuracy. This experience underscored the importance of a solid foundation in data handling to ensure the efficiency and effectiveness of the application. All the small design decisions had to be meticulously chosen since it would have rippling effects in the future.

**c. Karan**

**Bridging front end to back end.** While engineering advanced queries and running them independently on GCP was simple, actually integrating the queries with the database on a react page was much harder. It was hard to figure out how to extract the inputs from the webpage and use them in a static query and then make a request to the database that was being stored on the cloud. Another challenge we faced was connecting the front end to the back end. A lot of values

and fields we were trying to get to show up on the front end that was established on the back end was not showing up. A lot of values that were submitted through the front end were also not being recorded on the backend. Through a lot of backend and research about transmitting data through multiple react pages we were able to fix this.

#### **d. Reva**

**Optimizing Queries for Efficiency.** One significant challenge was optimizing database queries for speed and efficiency, particularly when handling large datasets for crime and transportation. We learned the importance of indexing and query optimization techniques, which would be beneficial for any future team working on similar large-scale data applications.

### **8. Future of the Project**

Apart from enhancing the user interface, future work could reintroduce some of the original filtering capabilities to allow for broader searches across neighborhoods. Additionally, incorporating real-time data updates and expanding the database to include more dynamic elements like event listings could significantly enhance user experience. We also wanted to incorporate a Google Maps API connection on the home page and the results page to directly direct the user to the locations of their interests, whether it be restaurants or subway stations. These features would make it a one-stop website for all information related to Airbnbs and the amenities near them. The final thing that may be useful to incorporate in the future would be to allow users to add their own Airbnbs to the database. We initially wanted only verified Airbnbs to be shown as was the reason for using a dataset. However, it might be useful to allow people to add their own Airbnbs.

### **9. Teamwork**

Throughout our project, our team managed collaboration effectively, with each of us taking the lead on specific aspects—front end, back end, data integration, and dynamic components—while supporting each other across different areas.

Team members and their roles:

Achintya Sanjay - Google Cloud Platform Set-Up/FrontEnd (Color Scheme & Design) /Backend (User Routes - Login, Logout, User Page, Register, Modify, Delete)

Karan Shah - Advanced Queries/Dynamic Components/Backend(Home Page - AirDnb Link Parsing, Stored Procedures)

Reva Jethwani - Advanced Queries/Indexing/Dynamic Components/Backend(Results Routes, Transactions, Stored Procedures)

Ayushe Nagpal - Dynamic Components/FrontEnd(Color Scheme and Design)/Backend(Results Routes - Results Page, Transitions)

This balanced approach facilitated comprehensive learning and a well-rounded project development experience. We would all help each other when we were stuck on our parts, which would allow us to move quickly past issues since one of our team members could usually quickly figure out the issue and get past roadblocks.