

# Housing Market Trends



Presented By:  
Chisom Ozo  
Jorge Nardy  
Muhammad Kashif  
Omar Salloum

December 11, 2023

Welcome everyone, our group tonight is comprised of (mention members names, including yourself). And our project aimed to provide housing Market trends in Austin, Texas.

# AGENDA

Introduction	Front-End Development
Project Plan	Visualizations & Demo
Data Sources	Conclusions
Challenges & Limitations	Q&A
Flask API	



In our agenda today we will take you through a brief introduction of the project, followed by the project plan, our data sources, challenges & limitations, and how we created our Flask API, Front-End Development and Visualizations. Finally we will have time for conclusions and a Q&A.

# Introduction



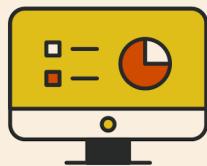
Our project focuses on exploring housing market trends in Austin, TX - one of the fastest growing cities in the USA according to CNBC - examining factors such as housing prices, property types, and nearby amenities.

Looking at the booming Austin Metropolitan Area and the ongoing market for properties. This website would provide a structure where individuals moving to Austin would be able to see schools, grocery stores, public transit, restaurants, gyms and parks around properties in the market.

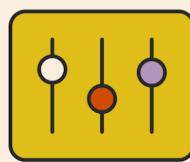
Source: [10 U.S. cities that are growing the fastest](#)

Our project focuses on exploring housing market trends in Austin Metropolitan Area, TX, examining factors such as housing prices, property types, and nearby amenities. We chose to implement web scraping and Leaflet to gather data and present it interactively through maps. This combination allows us to provide comprehensive insights into the housing market in an engaging manner.

# Project Plan



ATTOM API  
Yelp API



PostgreSQL Database  
Flask API



User interaction  
Data aggregation  
Visualization

For our data collection, we utilized ATTOM API and Yelp API to gather comprehensive information about housing prices, property types, and surrounding amenities. To manage and store the collected data, we established a robust database system using PostgreSQL, providing a foundation for our data retrieval. Alongside a Flask API. Finally, HTML/Java were used to create a user-friendly interface through interactions and visualizations.

# **Project Plan**

---

- What are the trends in housing prices in the Austin Metropolitan Area?
  - How do the prices vary based on the proximity to schools, grocery stores, and other amenities?
  - Can we identify patterns in housing features and prices within the same area?
  - Are there clusters of houses with similar amenity patterns?
- 

And here we have some of the main questions our project aimed to answer to help homebuyers moving to the Austin Metropolitan Area.

# Data Sources

For up-to-date information about market prices and sales of ongoing listing in Austin Metropolitan Area, an API was required.

<b>Data Sources</b>	<ul style="list-style-type: none"><li>• ATTOM API</li><li>• Yelp API</li></ul>
<b>Sample Size</b>	<ul style="list-style-type: none"><li>• Over 40,000 properties currently on the market</li><li>• Over 2,000 businesses</li></ul>
<b>Geographical Location</b>	<ul style="list-style-type: none"><li>• Austin Metropolitan Area, Texas</li></ul>
<b>Data Cleaning</b>	<ul style="list-style-type: none"><li>• Cleaning and removing outliers</li></ul>
<b>Database</b>	<ul style="list-style-type: none"><li>• PostgreSQL</li></ul>
<b>API</b>	<ul style="list-style-type: none"><li>• Flask</li></ul>

Our data sources involved a combination of ATTOM and Yelp APIs with thousands of entries relating properties on the market to businesses within a 1km radius. Users may also check ratings for businesses and neighborhood profiles by Zipcode based on this data.

# Data Sources

The details and steps used during the ETL process to ensure a robust and relevant dataset.

- What?
- Why?
- How?

Criteria Selection

Data Completeness

Outlier removals

Our data sources involved a combination of ATTOM and Yelp APIs with thousands of entries relating properties on the market to businesses within a 1km radius. Users may also check ratings for businesses and neighborhood profiles by Zipcode based on this data.

# Data Sources

Captured visualizations highlighting the outlier removals from the property data frames.

## Property size to room ratio

square_foot	bedrooms	bathrooms
974	840	4
984	480	3
1064	480	3
1144	648	3
1145	576	3

Our data sources involved a combination of ATTOM and Yelp APIs with thousands of entries relating properties on the market to businesses within a 1km radius. Users may also check ratings for businesses and neighborhood profiles by Zipcode based on this data.

# Data Sources

Captured visualizations highlighting the outlier removals from the property data frames.

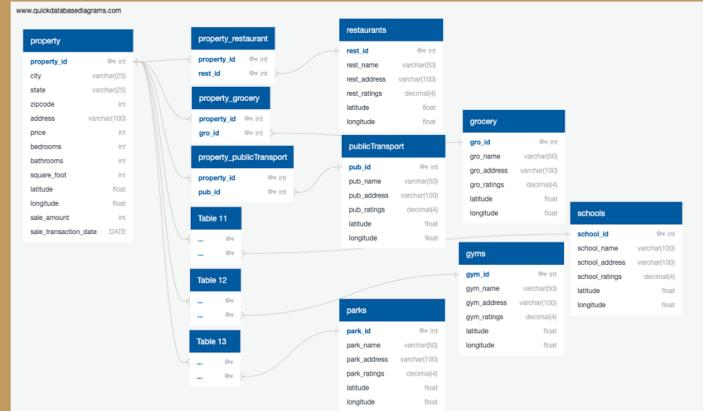
## Yearly price change

	sale_transaction_date	sale_amount	price
4572	2023-06-09	3063	4833563
28	2022-02-08	1250	741046
4159	2023-10-02	15953	450080
3477	2021-07-08	6250	1897071
6659	2023-09-15	367318	6754950

Our data sources involved a combination of ATTOM and Yelp APIs with thousands of entries relating properties on the market to businesses within a 1km radius. Users may also check ratings for businesses and neighborhood profiles by Zipcode based on this data.

# Challenges & Limitations

Throughout our project journey, we encountered a few challenges that tested our problem-solving skills and teamwork.



Some of our challenges involved the relationships between properties and amenities and how we captured which properties related to which amenities, we went through many schemas, as the example in this slide shows how businesses and properties are related by their IDs. Other limitations include for some of the properties, the market sales were not provided. Regardless, of the challenges and limitations we were able to create a flask that would provide a bridge between the clean data CSVs and the creation of the HTML.

# Flask-API

## Overview

Flask serves as the backbone of this application, handling both HTML rendering for web pages and JSON data for API endpoints. The flask was able to provided a visual view of what the website was to produce. It showed the properties(prices, bedrooms etc), the amenities close by and the rating of those amenities.

```
[{"latitude": 30.2773015358008, "longitude": -97.7440630452746, "publictransport_address": "209 W 9th St, Ste 100, Austin, TX 78701", "publictransport_id": 1, "publictransport_name": "Capital Metro", "publictransport_ratings": 2.0}, {"latitude": 30.34838, "longitude": -97.71251, "publictransport_address": "N Lamar Blvd And Research Blvd, Austin, TX 78751", "publictransport_id": 2, "publictransport_name": "North Lamar Transit Center", "publictransport_ratings": 2.0}, {"latitude": 30.58769, "longitude": -97.8559, "publictransport_address": "800 N US Hwy 183, Leander, TX 78641", "publictransport_id": 3, "publictransport_name": "Metrorail - Leander Station", "publictransport_ratings": 5.0}, {"latitude": 30.27754518, "longitude": -97.7481989, "publictransport_address": "1000 Brazos St, Ste 100, Austin, TX 78701", "publictransport_id": 4, "publictransport_name": "Metrobike Austin", "publictransport_ratings": 2.0}, {"latitude": 30.328698622504, "longitude": -97.716183335162, "publictransport_address": "6400 Airport Blvd, Austin, TX 78752", "publictransport_id": 5, "publictransport_name": "Highland Metrorail Station", "publictransport_ratings": 3.0}, {"latitude": 30.26136, "longitude": -97.72837, "publictransport_address": "2010 E 6th St, Austin, TX 78702", "publictransport_id": 6, "publictransport_name": "Carta", "publictransport_ratings": 5.0}]
```

Flask serves as the backbone and bridge of this application, handling both HTML rendering for web pages and JSON data for API endpoints. With the clean data from the CSVs and HTML/Json files, we were able to create a flask documentation that would provide a list of amenities such as school, grocery, restaurants, public transports, gyms and parks that show up near the property selected. It would also show the rating of each amenities with the help of the Yelp api. With the properties it will show the amount of beds, bath and how much the property is selling at. When running the flask API, it shows a list of available routes which the list of properties in the surrounds Austin area, and when putting the link in the website it shows all near by amenities within a 1km radius in the surrounding area. There was more in depth information that is need when creating the HTML/ front-end part will be taken over by Muhammad.

# Front-End Development

```
<nav>
  <div class="search-container">
    <label for="zipcode">Zip Code:</label>
    <select id="zipcode">
      <!-- Options will be populated dynamically -->
    </select>
    <label for="bedrooms">Number of Bedrooms:</label>
    <select id="bedrooms">
      <!-- Options will be populated dynamically -->
    </select>
    <button onclick="searchProperties()">Search</button>
  </div>
  Property Recommendation in Texas USA. Group#0 Jorge,Omar,Chisom,Muhammad</nav>

<div class="details-container" id="details">
  <!-- Property details will be displayed here on click -->
</div>
<div id="barChart">Sales History</div>
<div id="amenities">Amenities</div>
<div id="map">Area Map</div>
<div id="recommendations">Recommendations</div>
<!-- Side bar customization -->
<div id="sideBar">

  <div id="result-box">
    <div class="results-container" id="results">
      <!-- Results will be displayed here -->
    </div>
  </div>
</div>
```

```
// Function to display search results
async function searchProperties() {
  const zipCode = document.getElementById('zipcode').value;
  const bedrooms = document.getElementById('bedrooms').value;
  const resultsContainer = document.getElementById('results');
  resultsContainer.innerHTML = ''; // Clear previous results

  const data = await fetchData();
  const filteredData = data.filter(property => property.zipcode === zipCode && property.bedrooms === bedrooms);

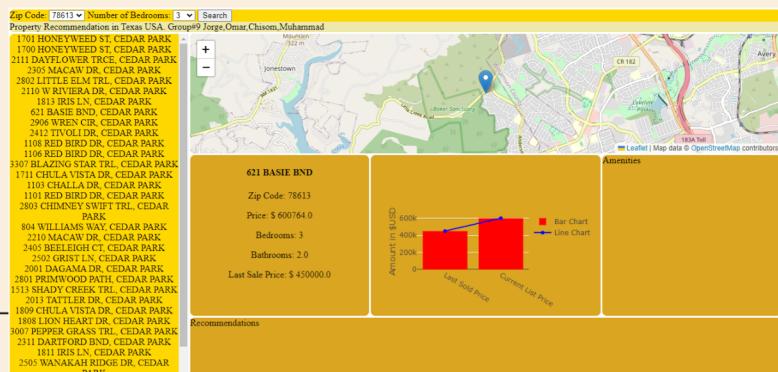
  filteredData.forEach(property => {
    const result = document.createElement('div');
    result.classList.add('result');
    result.textContent = `${property.address}, ${property.city}`;
    result.addEventListener('click', () => displayDetails(property));
    result.addEventListener('mouseover', () => highlightResult(result));
    result.addEventListener('mouseout', () => removeHighlight(result));
    resultsContainer.appendChild(result);
  });
}
```

Our front-end is crafted using HTML, and JavaScript. The snippets above and to the left showcase the design and functionality that enhance user interaction and exploration of housing market data.

Emphasizing user experience, we've incorporated intuitive elements such as dropdowns and filters. These features empower users to tailor their exploration, providing a customized view of the housing market.

# Visualizations & Demo

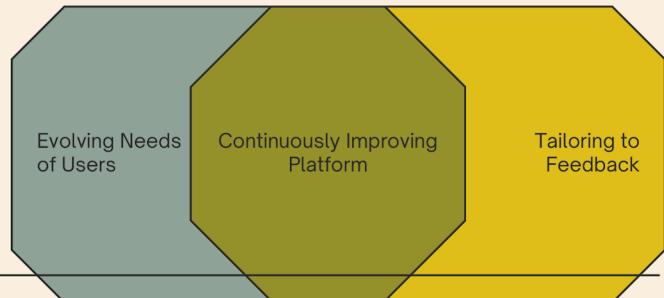
Explore our project through various lenses.



Now let's demonstrate how the tool works and the visualizations it presents.

# Areas of Improvement

Our roadmap includes exploring additional data sources, implementing more advanced visualizations, and introducing features that offer deeper insights into housing market trends.



As we reflect on our project, we envision several avenues for future enhancements and refinements to elevate the overall user experience.

Considering the evolving needs of our users, we plan to incorporate feedback mechanisms to tailor our project to their preferences, ensuring a continuously improving platform.

# Q&A

---



# THANK YOU!

---

Thank you very much for your attention! Now we will open the floor to a Q&A.



# References

[Flask Documentation](#)

[Leaflet Documentation](#)

[Plotly Documentation](#)

[Jupyter Notebooks Documentation](#)

[QuickDBD for Schema Design](#)

[Yelp API](#)

[ATTOM Real Estate API](#)

[10 U.S. cities that are growing the fastest](#)

