Renters and Sellers

A PROJECT REPORT

Submitted by

Vasavada Het Jayeshbhai

210310116017

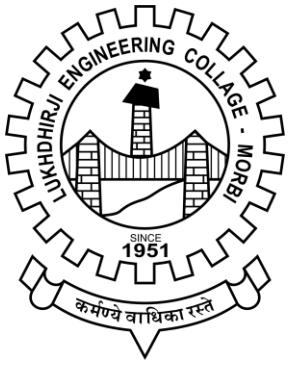
In partial fulfillment for the award of the degree of

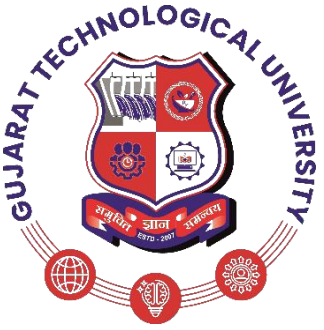
BACHELOR OF ENGINEERING

in

Information Technology

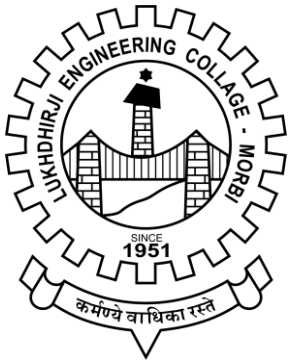
Lukhdhirji Engineering College, Morbi





Gujarat Technological University, Ahmedabad

April, 2025



Lukhdhirji Engineering College

Morbi

CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Renters and Sellers** has been carried out by **Vasavada Het** under my guidance in partial fulfillment for the degree of **Bachelor of Engineering in Information Technology**, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year 2024-25.

Prof. Rahul R. Keshwala
Internal Guide

Dr. Chirag A. Patel
Head of Department

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2024-2025

Date of certificate generation : 23 April 2025 (09:11:44)

This is to certify that, ***Het Jayeshbhai Vasavada*** (Enrolment Number - 210310116017) working on project entitled with ***Real Estate Analysis*** from ***Information Technology*** department of ***LUKHDHIRJI ENGINEERING COLLEGE, MORBI*** had submitted following details at online project portal.

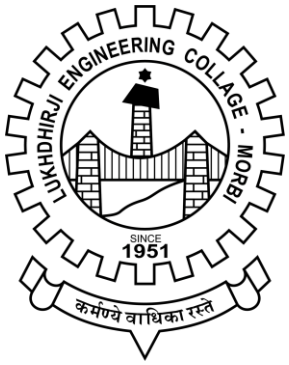
Internship Project Report	**Completed**

Name of Student : Het Jayeshbhai Vasavada

Name of Guide : Mr.keshwala rahul ranmalbhai

Signature of Student : _____

*Signature of Guide : _____



Lukhdhirji Engineering College

Morbi

DECLARATION

We hereby declare that the Project report submitted along with the Project entitled **Renters and Sellers** submitted in partial fulfillment for the degree of **Bachelor of Engineering in Information Technology** to Gujarat Technological University, Ahmedabad, is a Bonafide record of original project work carried out by me at **L. E. College** under the supervision of **Prof. Rahul R. Keshwala** and that no part of this report has been directly copied from any student's reports or taken from any other source, without providing due reference.

Name of Student **Sign**

Vasavada Het Jayeshbhai

ACKNOWLEDGEMENT

This project would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study. We are especially grateful to our Internal Guide **Prof. Rahul R. Keshwala** for her valuable support. He timely helped our team in making this project. We are also indebted to her for the reviews, suggestions and support extended whenever it was required.

This Project work has been the most practical and exciting part of our learning experience, which would be an asset for us and for our future career. We would like to express our special gratitude to **Dr. Chirag Patel (Head of Department)** and **Prof. Rakesh Parmar** for the guidance, inspiration, motivation and encouragement for the project.

We are also thankful to our batchmates and especially our seniors who directly or indirectly helped us in making our project.

Yours sincerely,

Het Vasavada

ABSTRACT

Renters and Sellers is an intelligent real estate platform designed to help users make smarter decisions when it comes to renting, selling, or buying properties. By integrating machine learning, the platform allows users to **compare property rates**, **find similar listings**, and gain **data-driven insights** — all in one place.

- Instantly compare current **rental** and **selling** prices of similar properties in the selected area.
- Leverage historical trends and ML predictions for better decision-making.
- Get **recommendations for similar properties** based on location, size, price, and features.
- Uses clustering and similarity algorithms to surface the most relevant options.
- Estimate the **realistic price** of any property using machine learning models trained on diverse real estate data.
- Helps users set the right price when listing or know what to expect when renting or buying.

Whether you're a first-time renter, seasoned investor, or someone looking to sell, **Renters and Sellers** simplifies the process with real-time insights and intelligent suggestions. With seamless comparison tools and a clean user interface, navigating the real estate market becomes effortless. Backed by machine learning, Renters and Sellers is your smart companion in every property journe

Table of Contents

CHAPTER – 1: INTRODUCTION TO PROJECT

1.1 Project Summary

Renters and Sellers is an intelligent real estate platform built to help users make smarter property decisions through real-time data and machine learning insights. It allows users to seamlessly **compare rental and sale prices**, **predict property values**, and **discover similar properties** based on key features such as location, size, and budget.

1.2 Purpose

Renters and Sellers is designed to provide a **structured**, **interactive**, and **efficient digital environment** for renters, property sellers, and administrators. The primary goal of the platform is to **streamline property listing, comparison, and price evaluation**, while ensuring **accessibility and ease of use** for all users.

1.3 Objective

- Develop a **robust and interactive real estate platform** tailored for structured property exploration and smart decision-making.
- Provide **secure and role-based features** for Sellers, and Renters
- Enable administrators to manage property listings, approve submissions, oversee agents, and monitor platform activity efficiently.
- Allow sellers to create and organize property listings with detailed information, images, and pricing insights.
- Offer renters an **engaging and personalized experience** through smart filters, real-time price comparisons, similar property suggestions, and location-based search tools.

1.4 Scope

- **User Management:** Role-based secure login and access for Sellers, and Renters.
- **Property Listing & Management:** Creation and organization of categorized property listings (for rent/sale), including detailed descriptions, images, and price details.
- **Price Prediction & Rate Comparison:** Machine learning models for predicting property prices and comparing rental vs. selling rates across similar properties.
- **Similar Listings:** Smart recommendations for properties based on user preferences, location, price range, and features.
- **Agent Contacts & Management:** Easy access to agent contact information and direct communication for each listed property.
- **Search & Filter System:** Advanced search functionality with filters for price, location, bedrooms, furnishing status, and more.
- **Scalability & Automation:** Enhance platform scalability by automating property submission, comparison, and recommendation processes to reduce manual efforts.

1.5 Technology

HTML, CSS, and JavaScript

- **HTML** is used for structuring the content on the platform, ensuring a clean and organized layout for both the property listings and user dashboards.

- **CSS** provides styling to make the platform visually appealing and responsive across devices, allowing for a smooth and intuitive user experience.

- **JavaScript** powers the interactivity of the platform, from form submissions and real-time updates to interactive maps and dynamic content loading.

React is utilized for building the frontend of the platform, creating a dynamic and responsive user interface. It ensures seamless rendering of property listings, real-time price comparisons, and filter functionalities, enhancing the overall user experience.

FastAPI serves as the backend framework for the platform, ensuring high performance and quick responses for API requests. It facilitates secure authentication, data processing, and communication between the frontend and database, especially for the property listing and price prediction features.

Python is used for developing machine learning models that predict property prices, compare rates, and recommend similar listings based on user behavior. It powers the core logic behind the price prediction and recommendation system, ensuring accuracy and scalability.

PostgreSQL (PSQL) is the relational database management system for storing and managing property data, user information, and transaction records. It ensures data integrity and supports efficient queries for searching, filtering, and comparing properties, enabling quick data retrieval for users.

VSCode is the primary code editor used for development, offering a versatile environment for writing, debugging, and testing both frontend and backend code. It enhances productivity with features like IntelliSense, code navigation, and integration with version control systems.

Git is used for version control, allowing the development team to collaborate efficiently. It tracks changes in the codebase, enables branching for feature development, and ensures smooth collaboration between frontend and backend developers, with a clear history of all modification.

pgAdmin is used as a graphical interface for managing the PostgreSQL database. It allows administrators and developers to interact with the database easily, execute queries, monitor performance, and manage tables and data structures, ensuring the efficient handling of property listings and user data.

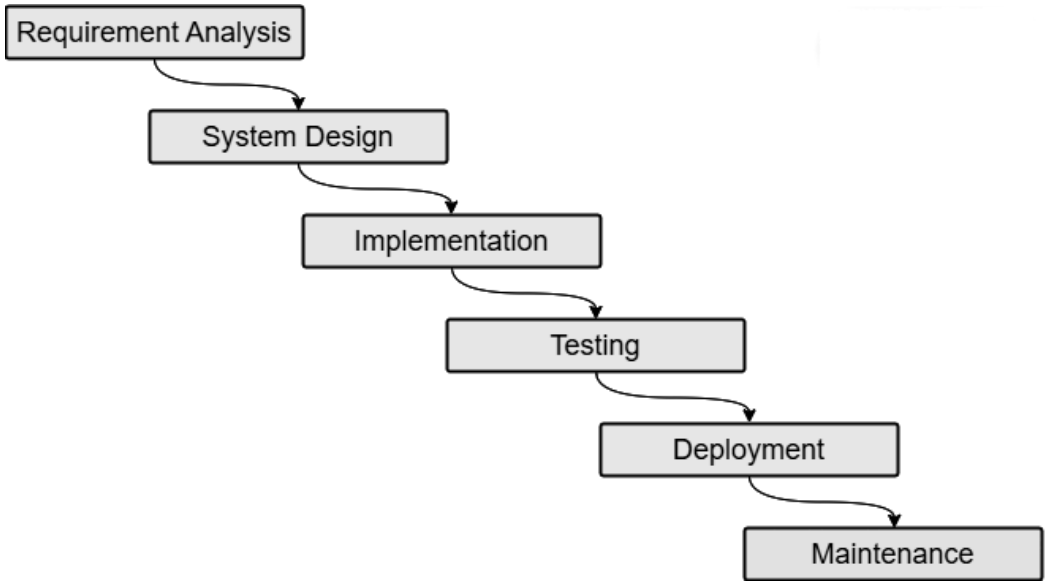
Jupyter Notebook is utilized for developing and testing the machine learning models behind the property price predictions and recommendations. It allows for interactive coding, data exploration, and visualization, helping to refine models and analyze the results in a collaborative and iterative manner.

.

1.6 Project Planning

The **Renters and Sellers** platform is developed to digitize and simplify the process of property exploration, buying, and renting. It provides structured property management, dynamic listing functionalities, and AI-powered features such as price prediction, rate comparison, and property recommendations. To ensure a smooth, well-organized development lifecycle, the **Waterfall Model** has been adopted. This linear and sequential approach ensures that each phase—**Requirement Analysis**, **System Design**, **Implementation**, **Testing**, **Deployment**, and **Maintenance**—is completed thoroughly before proceeding to the next, ensuring the platform is robust, scalable, and efficient.

1.6.1 Project Development Approach



Waterfall Model for Project Development: The Waterfall Model consists of five major phases: Requirement Analysis, System Design, Implementation, Testing, and Maintenance. Below is the short breakdown of each phase.

Phase 1: Requirement Analysis

This phase identifies the key system requirements such as **user authentication**, **property listing and management**, **admin control panels**, and **ML-powered features** like price prediction and similar property suggestions. A **Software Requirement Specification (SRS)** document is prepared to clearly define the project scope, functional components, and data security aspects.

Phase 2: System Design

The database structure, system architecture, and UI wireframes are designed. Technologies like HTML, CSS,Python, React and PSQL are selected for development.

Phase 3: Implementation

The frontend is built using HTML, CSS, and React, while Python and PSQL handle backend operations. Features like user authentication and book management are implemented.

Phase 4: Testing

The system undergoes unit, integration, performance and security testing to fix bugs and improve performance.

Phase 5 & 6: Deployment & Maintenance

The system is deployed **locally using FastAPI** with a **PostgreSQL database** for managing and testing the application in a controlled environment. Post-deployment, regular **maintenance activities** are conducted to fix bugs, optimize performance, and implement feature enhancements based on user feedback, ensuring the platform remains stable, responsive, and up-to-date.

1.6.2 Project Effort, Time and Cost Estimation

1. Time & Effort Estimation

the total effort and time is estimated based on the tasks required to complete the project within 12 weeks. The estimated effort distribution is:

- **Requirement Analysis & Design:** 3 weeks
- **Development & Coding:** 5 weeks
- **Testing & Debugging:** 2 weeks
- **Deployment & Documentation:** 2 weeks
- **Total Effort & Time Required:** 12 weeks

2. Cost Estimation

The cost is calculated based on the developer's hourly rate, software tools, and additional expenses:

- **Developer Cost (Assuming ₹50 per hour):** ₹50 × 672 hour (84 days) = ₹33,600
- **Software & Tools (VS Code, PSQL - Free):** ₹0
- **Miscellaneous (Electricity, Internet):** ₹400
- **Total Estimated Cost:** ₹34,000

1.7 Roles and Responsibilities

1. Agent Roles and Responsibilities:

- Manage and publish property listings on behalf of sellers or landlords.
- Ensure all property details—such as pricing, images, location, and amenities—are accurate and up to date.
- Assist potential buyers and renters by responding to inquiries and scheduling visits.
- Monitor the performance of listings and make updates based on market trends or client feedback.
- Act as the bridge between the platform users (buyers/renters) and property owners.

2. Seller Roles and Responsibilities:

- List properties for sale or rent with clear descriptions, images, pricing, and contact information.
- Categorize listings appropriately (e.g., apartment, house, commercial) and mark them as furnished/unfurnished if needed.
- Manage and update listings regularly to reflect current availability.
- Respond to inquiries from interested renters or buyers.
- Ensure all property information complies with platform guidelines and policies.

3. Renter/Buyer Roles and Responsibilities:

- Register on the platform and securely log in to browse listings.
- Use filters and map views to search for properties based on preferences like price, location, and number of rooms.
- Compare similar listings using ML-based suggestions for smarter decisions.
- View detailed property pages and contact agents or sellers directly.
- Save favorite properties, manage profile settings, and track communication history.

CHAPTER 2: SYSTEM ANALYSIS

2.1 Study of Current System

Current property listing platforms and real estate portals typically provide basic functionalities such as posting ads, viewing listings, and contacting agents. However, many lack advanced features like **ML-powered property recommendations**, **dynamic rate comparisons**, and **interactive map-based browsing**..

Users often face challenges in **finding similar properties**, **comparing rent/sale rates**, or **predicting price trends** in real-time. Additionally, platforms rarely offer **category-specific filtering** (e.g., furnished vs. unfurnished) or seamless communication with verified agents, leading to an unstructured and generic property exploration experience.

2.2 Problem and Weaknesses of Current System

- **No Intelligent Recommendations:** Most platforms do not use ML to suggest similar or relevant properties, making it hard for users to explore alternative options effectively..

- **Lack of Price Comparison Tools:** Users cannot easily compare rental or sale prices across similar listings in a specific area, leading to poor decision-making.

- **Weak Categorization and Filters:** Many platforms lack structured categorization (e.g., furnished/unfurnished, number of rooms), making property search tedious and unorganized.

- **No Predictive Insights:** There's no support for features like price prediction based on market trends, which could help users plan better financially.

2.3 Requirement of New System

- **Category-Based Property Structuring:** Organize listings under clear categories such as Rent, Sell, Furnished, Unfurnished, Apartments, and Commercial spaces for seamless navigation.
- **ML-Powered Recommendations:** Integrate machine learning models to suggest similar properties based on user preferences and listing details..
- **Dynamic Price Comparison & Prediction:** Allow users to compare property prices across listings and predict future trends using regression models.
- **Responsive UI Design:** Clean, user-friendly interface developed with HTML, CSS, and React, optimized for both desktop and mobile experiences.
- **Real-Time Data Handling:** FastAPI and PostgreSQL backend support for live listing updates, interactive filtering, and fast user interactions

2.4 System Feasibility

1. Technical Feasibility

- Built using **FastAPI** and **PostgreSQL** for robust backend performance, fast routing, and secure data handling.
- Developed with modern frontend technologies like **HTML, CSS, JavaScript, and React** for a dynamic, responsive UI.
- Efficiently runs on local or remote development environments, using tools like **VSCode** and **PgAdmin** for seamless integration.

2. Economic Feasibility

- Utilizes open-source technologies, reducing infrastructure and development costs significantly.
- Automates property comparisons, recommendations, and agent interactions—saving time and improving decision-making for users.
- Designed to scale efficiently with increasing users and listings, without major architectural changes.

3. Operational Feasibility

- Easy for **agents** to add and manage listings.
- **Renters and buyers** can explore, compare, and make informed decisions with minimal guidance.

4. Schedule Feasibility

- Developed using the **Waterfall Model** over a structured timeline.
- Clear phase-wise delivery—Requirement Gathering, Design, Development, Testing, and Deployment—ensures timely progress and quality control.

5. Legal & Security Feasibility

- User data secured through FastAPI's security layers and role-based access control.
- Complies with standard data privacy practices to protect user contact info, property details, and platform interactions.

.

2.5 Activity

1. Agent Activities

- Add and manage property listings with complete details, images, and pricing.
- Respond to inquiries from interested renters or buyers.
- Update listing status (active, sold, rented) and manage visibility.
- Coordinate with sellers to keep property information up to date.

2. Seller Activities

- Submit property details for listing, including pricing, location, and amenities.
- Choose an agent for handling inquiries and negotiation.
- Track listing performance and receive updates on interest or offers.
- Edit or deactivate listings when needed.

3. Renter/Buyer Activities

- Browse and filter properties based on needs (location, price, size, type).
- View property details, compare listings, and find similar recommendations using ML.
- Contact agents for more information or to schedule a visit.
- Bookmark and manage favorite properties within their account.

2.6 Features of New System

- **Category-Based Property Organization:** Listings structured by categories like Rent, Sell, Furnished, Unfurnished, Apartments, and more for seamless search and navigation.

- **ML-Powered Similar Listings:** Intelligent recommendations to suggest similar properties based on user preferences and past interactions.

- **Dynamic Price Comparison and Prediction:** Real-time price comparisons and predictive models to estimate future property trends, helping users make informed decisions.

- **Responsive and Intuitive UI/UX:** Clean, user-friendly design using **React**, ensuring compatibility across desktop and mobile devices.

- **Comprehensive Analytics:** Track property views, user activity, and engagement metrics, providing insights to agents and sellers for better property management..

2.7 List of Main Modules

1. Agent Module
2. Seller Module
3. Buyer/Renter Module
4. Property Listing Management System
5. Price Comparison & Prediction Module
6. Search & Filtering Module

2.8 Selection of Technology

- **Frontend**: HTML, CSS, React
- **Backend**: Python, FASTAPI
- **Database**: PSQL
- **IDE & Tools**: VS Code, Jupyter Notebook, PGAdmin

2.9 Software Requirements Specification (SRS)

Renters and Sellers is an interactive web-based platform designed to streamline the process of renting, selling, and buying properties. It offers features such as categorized property listings, price comparison, machine learning-powered recommendations, and dynamic price prediction. The platform includes dedicated dashboards for **Agents**, **Sellers**, and **Renters/Buyers**, ensuring role-specific access and secure data handling. With real-time updates, intelligent search filters, and an intuitive UI, it empowers users to make informed decisions and interact efficiently within the property market.

1. Functional Requirements

- **Agents** and **Sellers** can create, edit, update, and delete property listings, with options for pricing, descriptions, and images.
- Categorize listings by type (Rent, Sell, Furnished, Unfurnished, etc.) and allow advanced search filters (location, price, size, etc.) for easy property discovery.
- Real-time price comparison across listings and a machine learning-based price prediction model to estimate future property prices.
- ML-powered system to recommend similar properties to **Renters/Buyers** based on their search history and preferences.
- Secure messaging system for **Renters/Buyers** to communicate directly with **Agents** or **Sellers** for property inquiries.

2. Non-Functional Requirements

- **Performance**: Fast loading with real-time updates.
- **Security**: Auth guards, encrypted sessions, and role-based access.
- **Scalability**: Easily extendable to support new properties, agents, buyers, sellers.
- **Usability**: Clean, mobile-first UI.
- **Availability**: Hosted online, accessible 24/7.

3. Hardware and Software Requirements

- Software Requirements
    a. HTML, CSS, JavaScript
    b. pandas, numpy, seaborn, scikit-learn
    c. Jupyter notebook
    d. PostgreSQL
    e. VS Code
    f. FastAPI

- Hardware Requirements
    a. Processor
    b. RAM
    c. Storage
    d. Network

4. System Models

- **Architecture**: MVC architecture via PostgreSQL
- **User Roles**: Agent, Seller, Rent/Buyer

5. Assumptions & Constraints

- Active internet connection required.
- Users must be assigned correct roles for proper access.
- Regular backups needed for data integrity.

6. Testing Requirements

- **Unit Testing**: Covers individual functions like price prediction, property similarity logic, and filtering modules
- **Integration Testing**: Ensures compatibility between FastAPI endpoints and the React frontend
- **Performance Testing**: Load testing for filters and Properties loading
- **Security Testing**: Ensure access restrictions and encrypted data

7. Dependencies

Software:

- **React**: Frontend UI
- **FastAPI**: Backend API
- **PostgreSQL**: Database
- **Python & Jupyter**: Backend logic and ML models
- **VS Code, Git, pgAdmin**: Development and database management tools

Hardware:

- Local machine with minimum 8GB RAM, 256GB storage
- Internet for API integration and package installation

External:

- Browser support: Chrome, Firefox, Edge

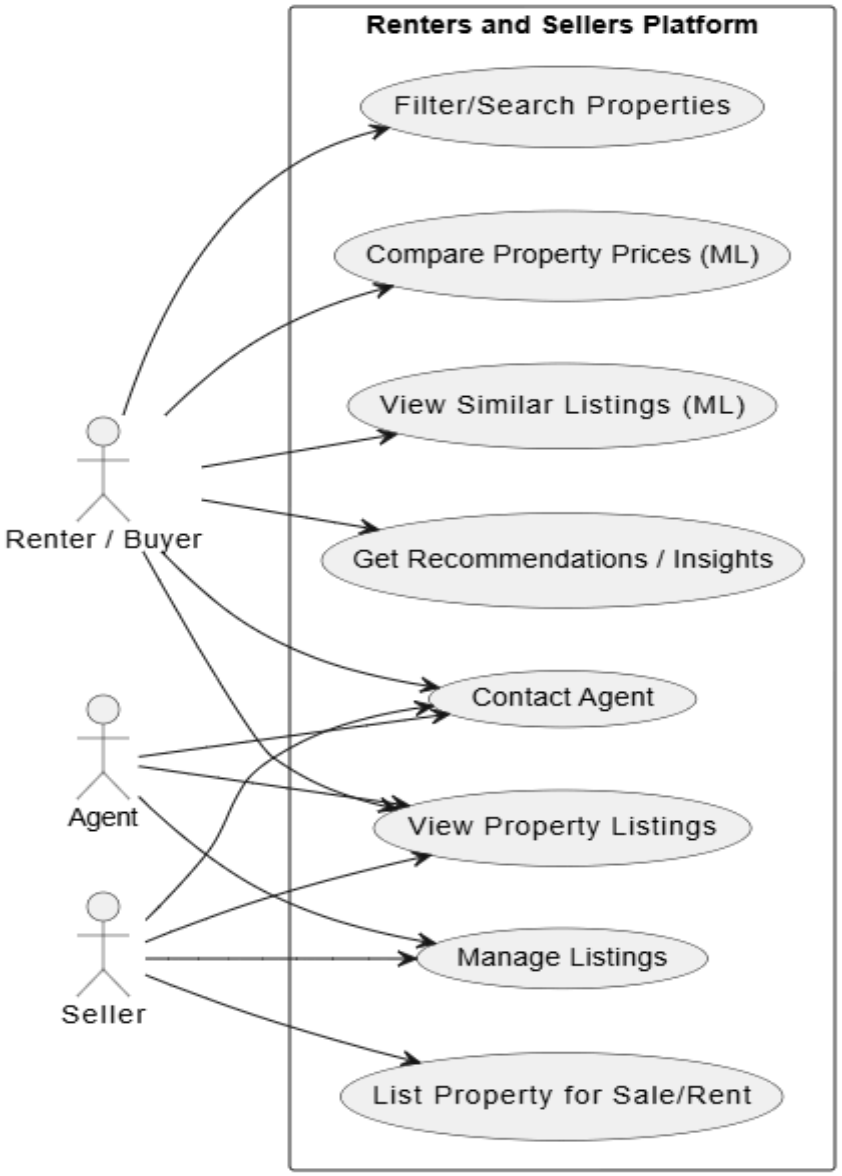
8. Change Management

- **Version Control:** Track system updates and modifications.
- **Feature Addition:** New features will be evaluated before implementation.
- **Bug Fixing:** Issues reported by users will be addressed in the next update cycle.
- **Backup & Recovery:** Regular backups will be taken before implementing changes.

CHAPTER 3: SYSTEM DESIGN

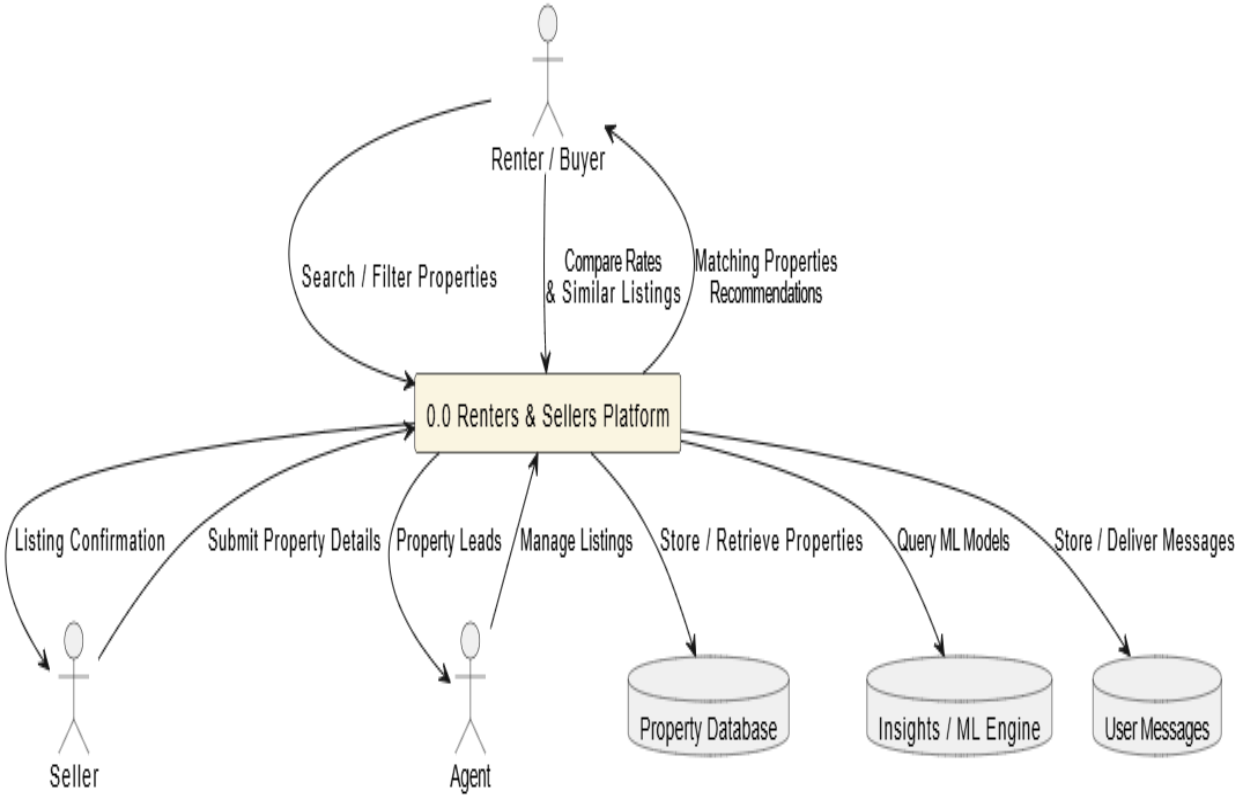
3.1 System Design & Methodology

3.1.1 Use Case Diagram



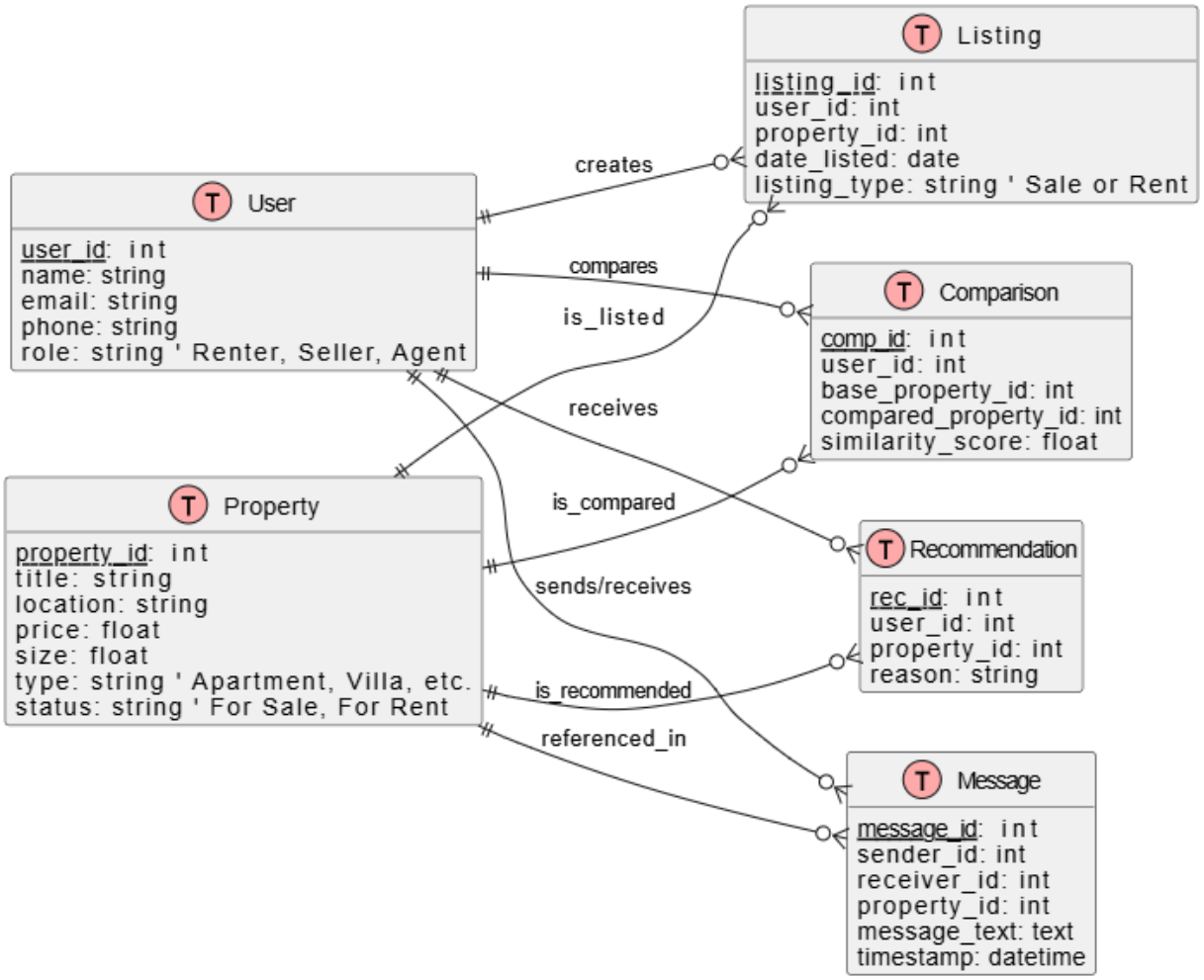
This use case diagram illustrates the interactions between the key users of the Renters and Sellers platform — Renters/Buyers, Sellers, and Agents — with the system's core functionalities.

3.1.2 DFD Level 0



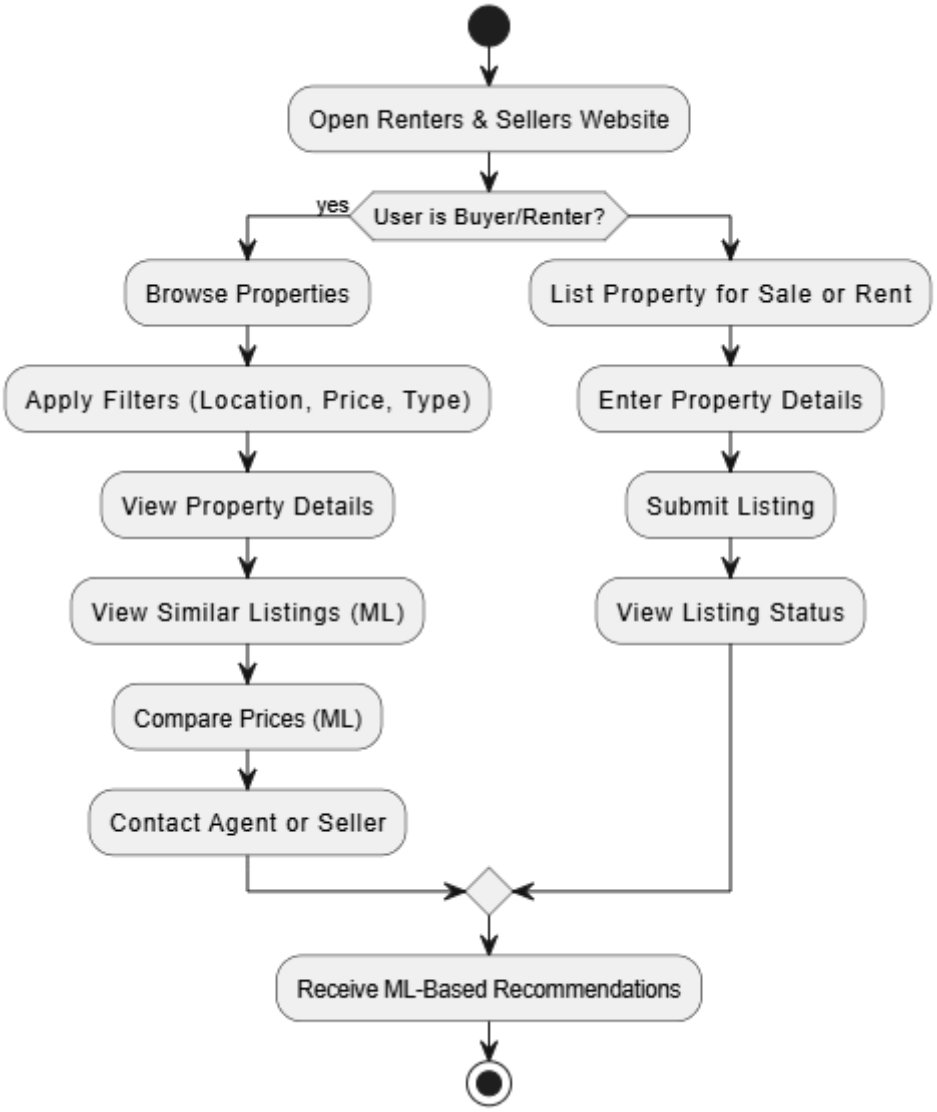
DFD Level 0 provides a bird's-eye view of the Renters and Sellers platform. It shows how Renters/Buyers, Sellers, and Agents interact with the system, with data flowing through core components like the Property Database, Machine Learning Engine, and Message Store.

3.1.3 ER Diagram



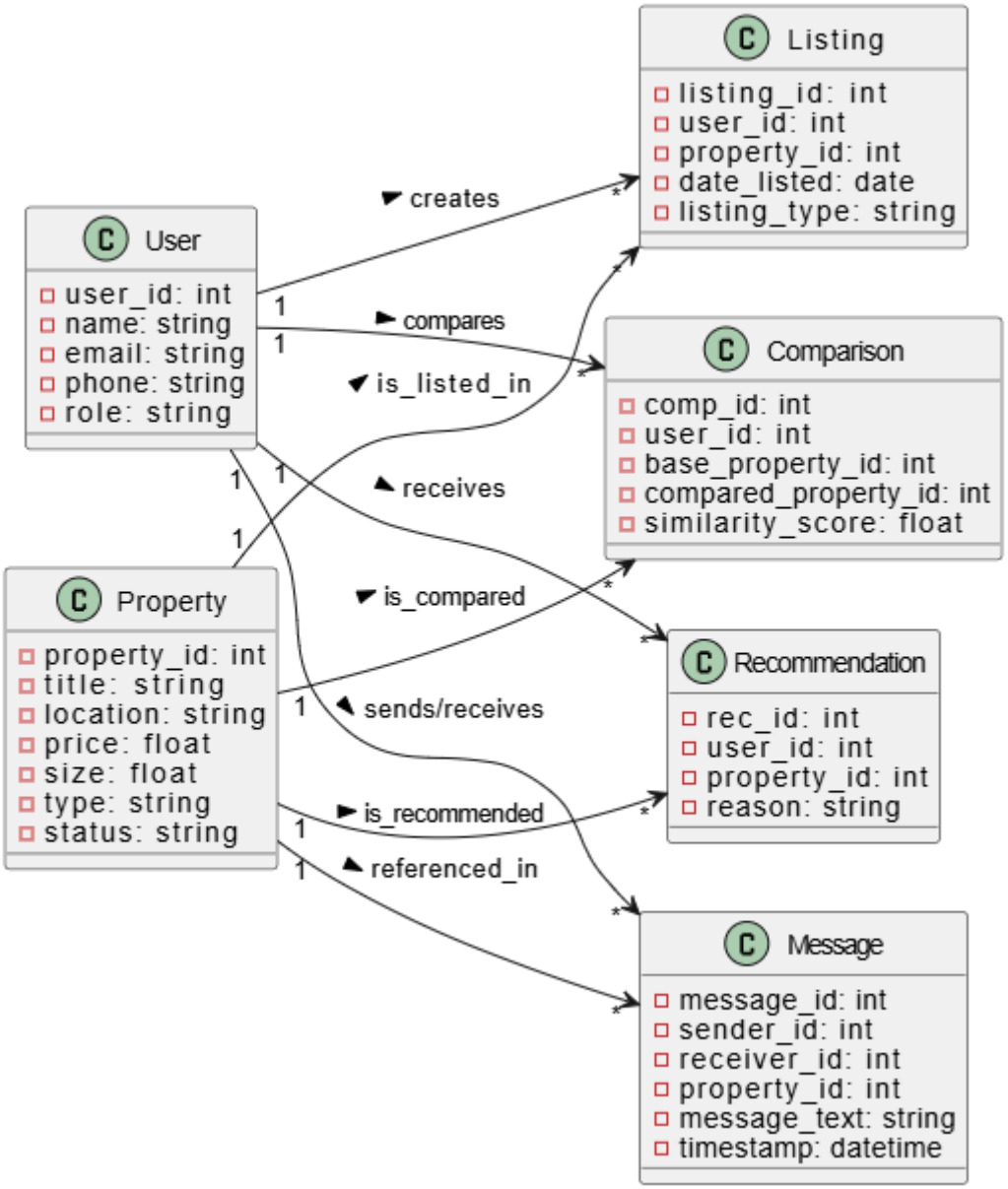
This **ER diagram** models the key entities and relationships in the Renters and Sellers platform. Users (acting as Renters, Sellers, or Agents) can list properties, send messages, and receive machine learning-based recommendations and comparisons. Each property can be associated with listings, messages, and AI-powered insights.

3.1.4 Activity Diagram



This **activity diagram** represents a user's flow through the Renters and Sellers platform. It supports both renters/buyers browsing and comparing listings and sellers listing their properties. Intelligent recommendations are integrated to assist decision-making.

3.1.5 Class Diagram



This **class diagram** models the core classes in the Renters and Sellers platform. Users can create listings, send messages, and receive ML-based comparisons and recommendations. The design promotes extensibility and supports all three user roles.

3.2 Database Design

3.2.1 Cleaned CSV file

Column Name	Data Type	Constraint	Description
id	bigint(20)	NOT NULL	Primary Key (unique user ID)
number_of_bedrooms	integer	>0	number of bedrooms
number_of_bathrooms	real	>0	number of bathrooms
living_area	numeric	>0	living area (in square foot)
condition_of_the_house	integer	1 to 5	furnished, semi furnished
grade_of_the_house	integer	1 to 13	condition vs grade
built_year	integer	after 1900	House built year
lattitude	numeric	-90 to 90	location
longitude	numeric	-180 to 180	location
price	numeric	>0	Price of property

3.2.2 Category classification

Column Name	Data Type	Constraint	Description
id	bigint(20)	NOT NULL	Primary Key (unique user ID)
number_of_bedrooms	integer	>0	number of bedrooms
number_of_bathrooms	real	>0	number of bathrooms
grade_of_the_house	integer	1 to 13	condition vs grade
built_year	integer	after 1900	House built year
price	numeric	>0	Price of property

3.2.3 Price Prediction

Column Name	Data Type	Constraint	Description
id	bigint(20)	NOT NULL	Primary Key (unique user ID)
number_of_bedrooms	integer	>0	number of bedrooms
number_of_bathrooms	real	>0	number of bathrooms
grade_of_the_house	integer	1 to 13	condition vs grade
built_year	integer	after 1900	House built year
price	numeric	>0	Price of property

3.2.4 Similar Properties

Column Name	Data Type	Constraint	Description
id	bigint(20)	NOT NULL	Primary Key (unique user ID)
number_of_bedrooms	integer	>0	number of bedrooms
number_of_bathrooms	real	>0	number of bathrooms
grade_of_the_house	integer	1 to 13	condition vs grade
built_year	integer	after 1900	House built year
price	numeric	>0	Price of property
lattitude	numeric	-90 to 90	location
longitude	numeric	-180 to 180	location

CHAPTER 4: IMPLEMENTATION

4.1 Implementation Platform

The system is developed, tested, and deployed in a local server environment.

Number	Description	Type
1	Operating System	Windows
2	Tools and Technologies	HTML, CSS, React, FastAPI, Python
3	Database	PSQL
4	IDE	Visual Studio Code, Jupyter Notebook

4.2 Outcomes

The outcomes of the Renters and Sellers project include the successful design, development, and local deployment of an intelligent real estate platform. The system provides a seamless experience for agents, renters, sellers, and buyers by offering features such as property comparison, rental and sale listings, price prediction, and similar property recommendations powered by machine learning.

4.2.1 Data Science

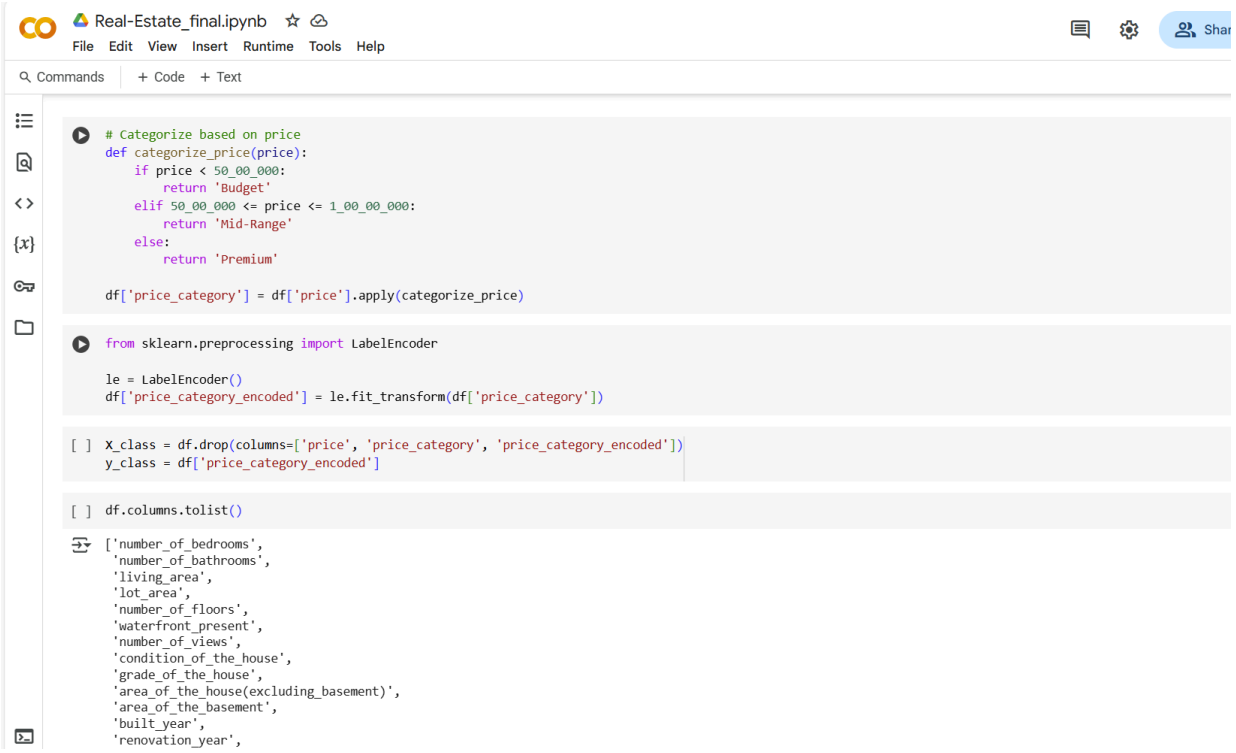
- **Original Dataset(csv)**

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	id	Date	number of	number of	living area	lot area	number of	waterfront	number of	condition c	grade of th	Area of the	Area of the	Built Year	Renovation	Postal Cod	Lattitude	Longitude
2	6.76E+09	42491	5	2.5	3650	9050	2	0	4	5	10	3370	280	1921	0	122003	52.8645	-114.557
3	6.76E+09	42491	4	2.5	2920	4000	1.5	0	0	5	8	1910	1010	1909	0	122004	52.8878	-114.47
4	6.76E+09	42491	5	2.75	2910	9480	1.5	0	0	3	8	2910	0	1939	0	122004	52.8852	-114.468
5	6.76E+09	42491	4	2.5	3310	42998	2	0	0	3	9	3310	0	2001	0	122005	52.9532	-114.321
6	6.76E+09	42491	3	2	2710	4500	1.5	0	0	4	8	1880	830	1929	0	122006	52.9047	-114.485
7	6.76E+09	42491	3	2.5	2600	4750	1	0	0	4	9	1700	900	1951	0	122007	52.9133	-114.59
8	6.76E+09	42491	5	3.25	3660	11995	2	0	2	3	10	3660	0	2006	0	122008	52.7637	-114.05
9	6.76E+09	42491	3	1.75	2240	10578	2	0	0	5	8	1550	690	1923	0	122006	52.9254	-114.482
10	6.76E+09	42491	3	2.5	2390	6550	1	0	2	4	8	1440	950	1955	0	122009	52.8014	-114.598
11	6.76E+09	42491	4	2.25	2200	11250	1.5	0	0	5	7	1300	900	1920	0	122010	52.9145	-114.391
12	6.76E+09	42491	5	2.5	2820	67518	2	0	0	3	8	2820	0	1979	0	122011	52.8094	-114.215
13	6.76E+09	42491	4	2	1820	5000	1.5	0	1	3	7	1640	180	1945	0	122006	52.9115	-114.459
14	6.76E+09	42491	4	2	1520	6200	1.5	0	0	3	7	1520	0	1945	0	122006	52.908	-114.459
15	6.76E+09	42491	4	2.75	2710	37277	2	0	0	3	9	2710	0	2000	0	122012	52.6934	-114.177
16	6.76E+09	42491	3	2.25	1750	1572	2.5	0	0	3	9	1470	280	2005	0	122013	52.8798	-114.511
17	6.76E+09	42491	4	2.5	2820	8408	2	0	0	3	9	2820	0	2014	0	122014	52.9838	-114.515
18	6.76E+09	42491	4	3.25	2730	54014	1	0	0	3	9	1560	1170	2007	0	122015	52.7433	-114.3
19	6.76E+09	42491	3	1.75	2360	7291	1	0	0	4	8	1360	1000	1948	0	122016	52.7574	-114.574
20	6.76E+09	42491	4	2.5	2730	12261	2	0	0	3	9	2730	0	1991	0	122017	52.9719	-114.395
21	6.76E+09	42491	3	2.5	3240	33151	2	0	2	3	10	3240	0	1995	0	122018	52.5556	-114.568
22	6.76E+09	42491	3	1.75	2330	14892	1	0	0	3	8	1970	360	1980	0	122019	52.8567	-114.236
23	6.76E+09	42491	4	2.5	3310	6500	2	0	0	3	8	3310	0	2012	0	122008	52.745	-114.06
24	6.76E+09	42491	4	2.5	1940	10500	1	0	0	4	7	1140	800	1976	0	122005	52.913	-114.304
25	6.76E+09	42491	3	1.75	2910	35200	1.5	0	0	3	8	2910	0	1979	0	122020	52.8047	-114.225
26	6.76E+09	42491	4	2.5	2860	3345	2	0	0	3	8	2190	670	2004	0	122021	53.0035	-114.348
27	6.76E+09	42491	4	1.75	1600	6380	1	0	0	3	8	1130	470	1959	0	122022	52.931	-114.496
28	6.76E+09	42491	4	1.75	2190	125452	1	0	2	3	9	2190	0	1968	0	122023	52.5003	-114.259

- Image showcasing the 'House Price India' dataset, which forms the core of the Renters and Sellers platform. It contains critical property information such as location, price, size, and amenities, which are used for price prediction and property comparison.

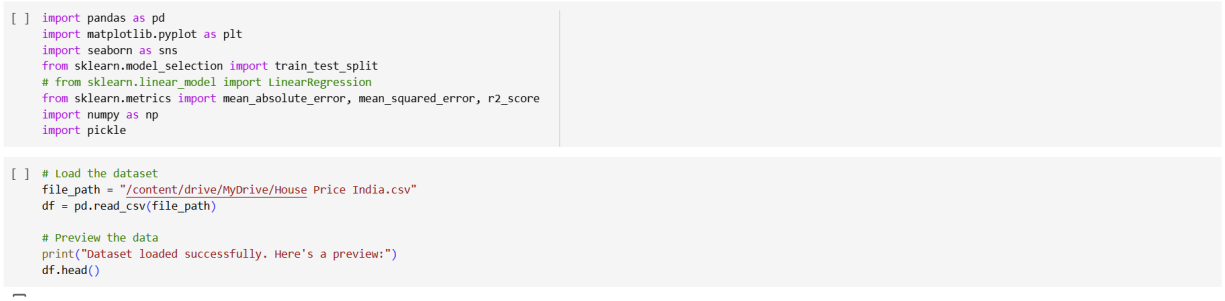
- A snapshot of the 'House Price India' dataset, providing essential attributes for real estate modeling, including BHK configuration, price, and building status to power the Renters and Sellers platform's property comparison features

Google collab



- Image of Google Colab showcasing the machine learning models trained on the 'House Price India' dataset. This environment is used for building and fine-tuning regression and classification models to predict property prices and categorize listings..

- "Image of Google Colab displaying the training of machine learning models, including K-Nearest Neighbors (KNN), Random Forest Regressor, and classification models. These models are utilized to predict property prices and categorize listings based on the 'House Price India' dataset.



```
[ ] df = df.drop(columns=['area_of_the_house(excluding_basement)', 'area_of_the_basement', 'lot_area', 'number_of_floors','waterfront_present', 'number_of_views', 'renovation_year', 'li

    df.columns.tolist()

    ['number_of_bedrooms',
     'number_of_bathrooms',
     'living_area',
     'condition_of_the_house',
     'grade_of_the_house',
     'built_year',
     'lattitude',
     'longitude',
     'lot_area_renov',
     'price']

[ ] df = df.drop(columns=['grade_of_the_house'])

[ ] df.columns.tolist()

    ['number_of_bedrooms',
     'number_of_bathrooms',
     'living_area',
     'built_year',
     'lattitude',
     'longitude',
     'price']
```

1. In this Google Colab image, essential libraries like **Pandas** and **NumPy** were used for data cleaning and preprocessing, handling missing values, and transforming categorical variables into numerical formats. These steps ensured the 'House Price India' dataset was ready for modeling with algorithms like KNN, Random Forest Regressor, and classification models.

2. The image shows the use of **Scikit-learn** for implementing machine learning models such as KNN and Random Forest Regressor. Data cleaning techniques, including feature scaling and handling missing data with **Pandas**, were applied to prepare the dataset for accurate predictions and classification.

3. Libraries like **Pandas**, **NumPy**, and **Matplotlib** were leveraged for data manipulation, cleaning, and visualization in this Google Colab setup. Columns like 'Price', 'Size', and 'Location' were cleaned and encoded to ensure the smooth training of KNN and Random Forest models for price prediction and categorization.

- **Visualization**

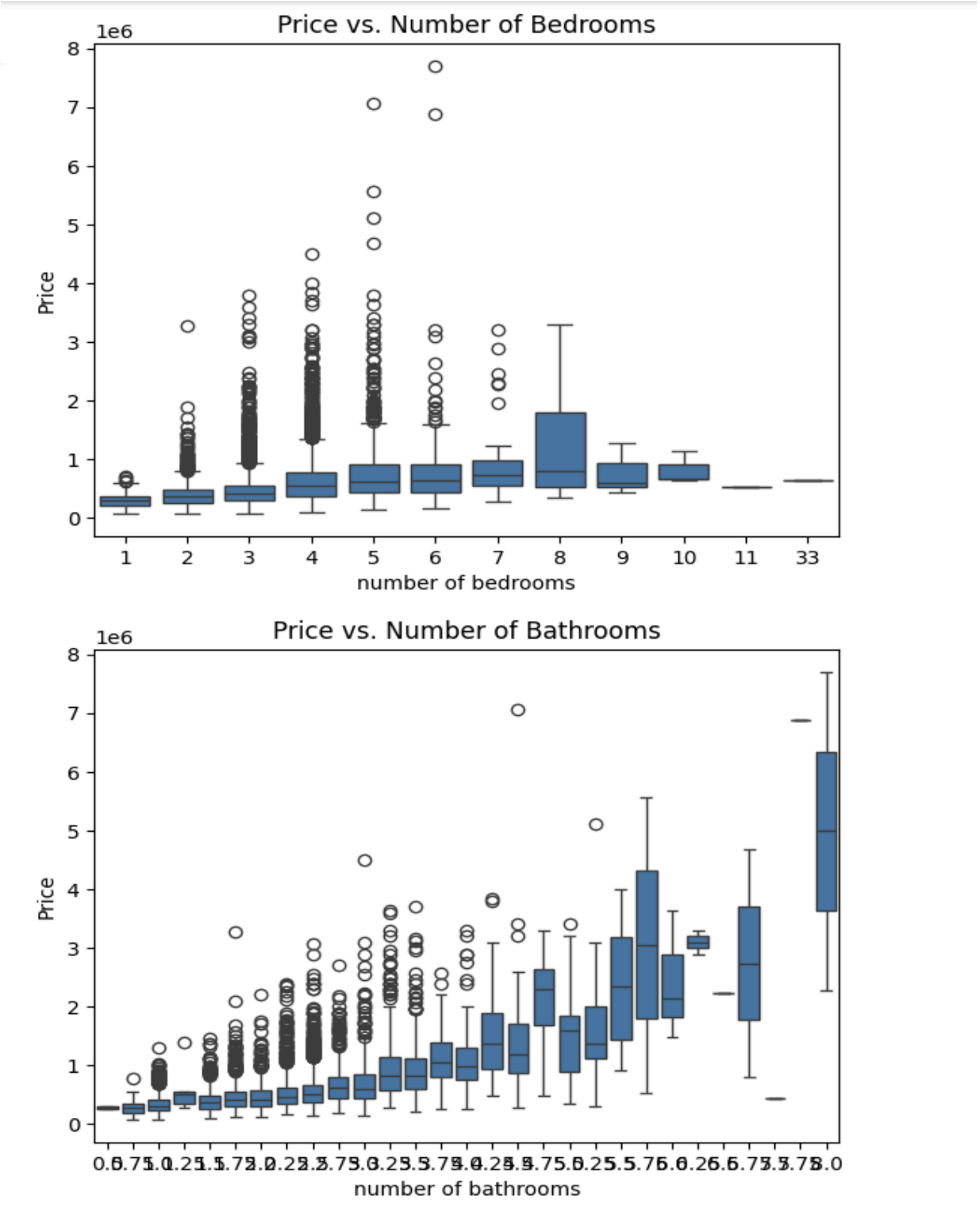


Image of a 'Price vs. Number of Bedrooms' graph, visualizing the relationship between property price and bedroom count. This plot helps identify trends in how the number of bedrooms affects property pricing, providing valuable insights for price prediction models

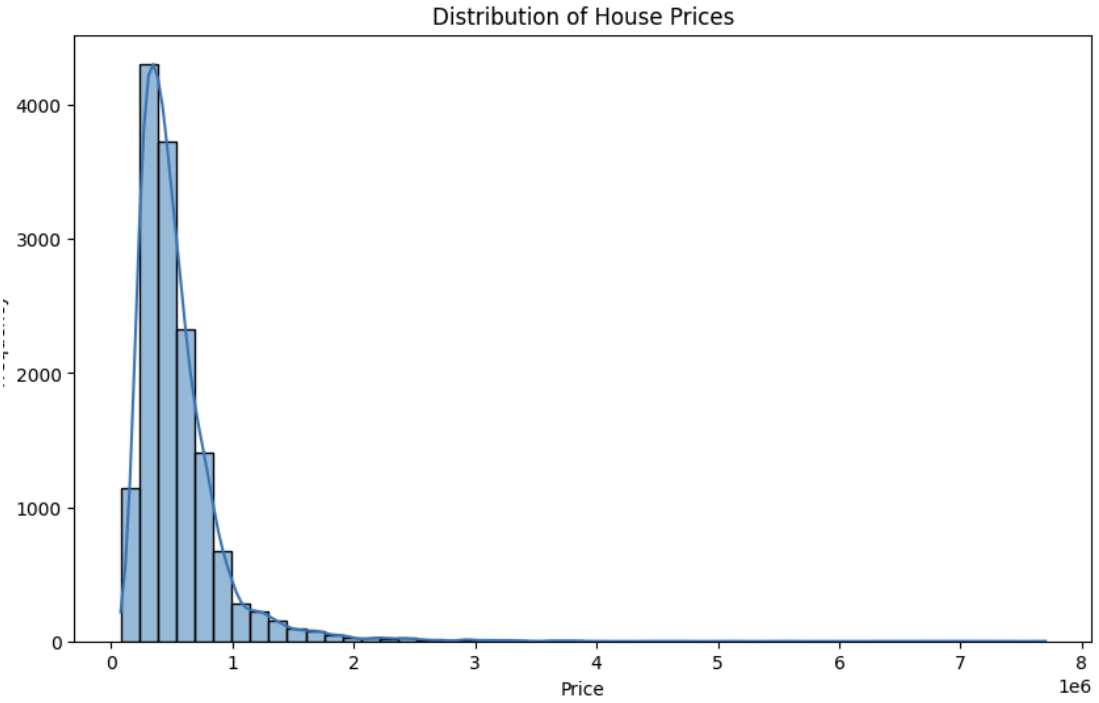


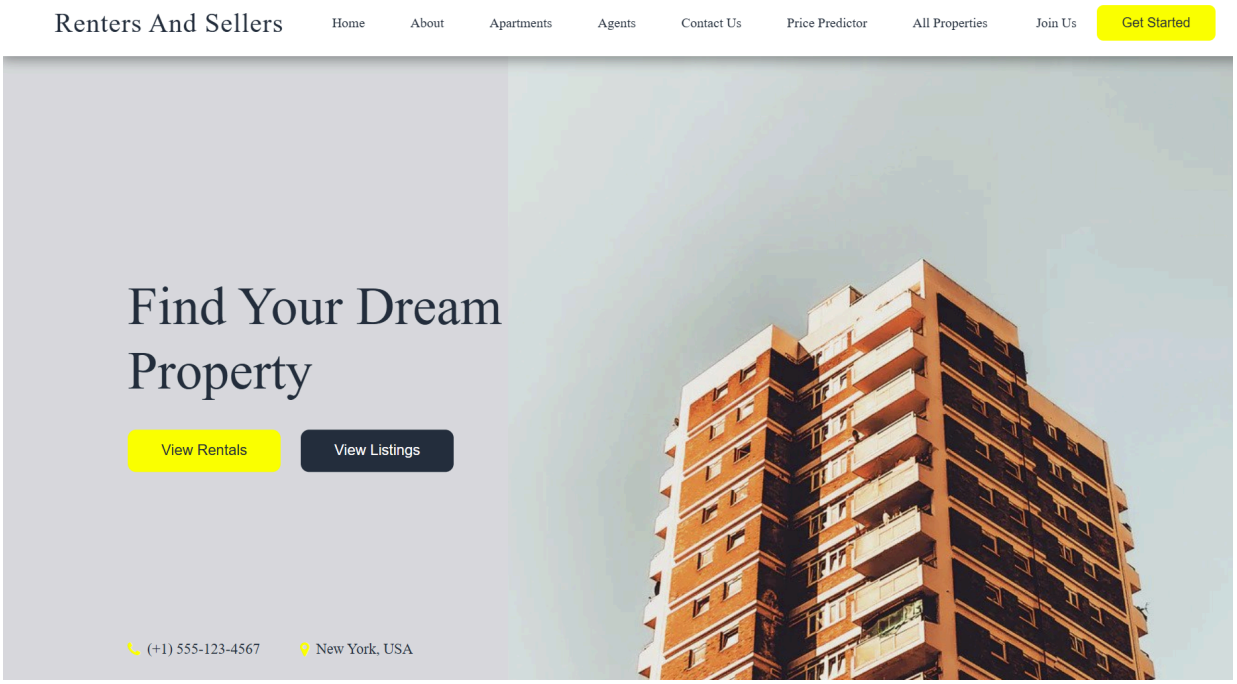
Image of a 'Price Distribution vs. Square Foot' graph, illustrating how property prices vary with the size of the property in square feet. This visualization highlights key patterns and outliers, offering insights into the correlation between property size and its price, crucial for predictive modeling.

- Process of downloading the trained machine learning models as **pickle** files, allowing easy storage and deployment. These files contain the trained KNN, Random Forest Regressor, and classification models, enabling fast model loading and predictions without retraining.

- Here, the trained machine learning models are saved as **pickle** files for efficient deployment. By downloading these files, the models can be loaded directly into the backend of the Renters and Sellers platform, ensuring quick price predictions and classification without the need for retraining

4.2.2 Frontend

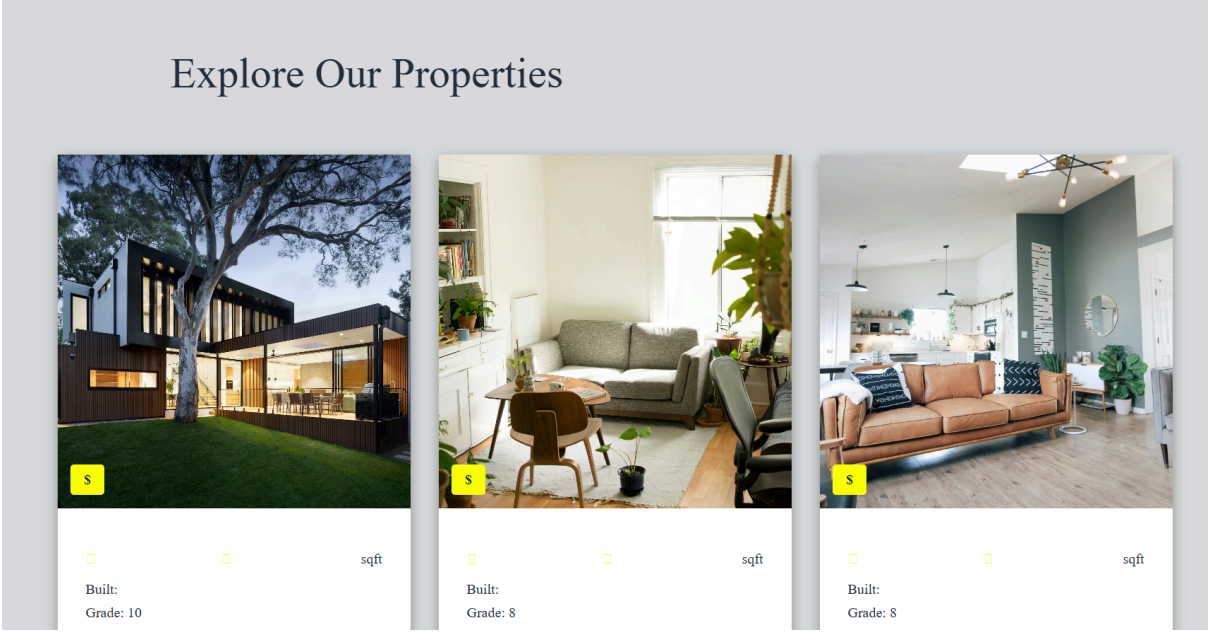
- **Website Header**



- The image showcases the website header with a well-structured navbar, offering easy access to important pages like 'About', 'Price Prediction', 'All Properties', and 'Apartment Listings'. It also includes clear call-to-action buttons for users to quickly navigate to 'View Rentals' and 'View Sellings'

- It is designed and routed in such a way that it is easier for user to navigate among different features of the website.

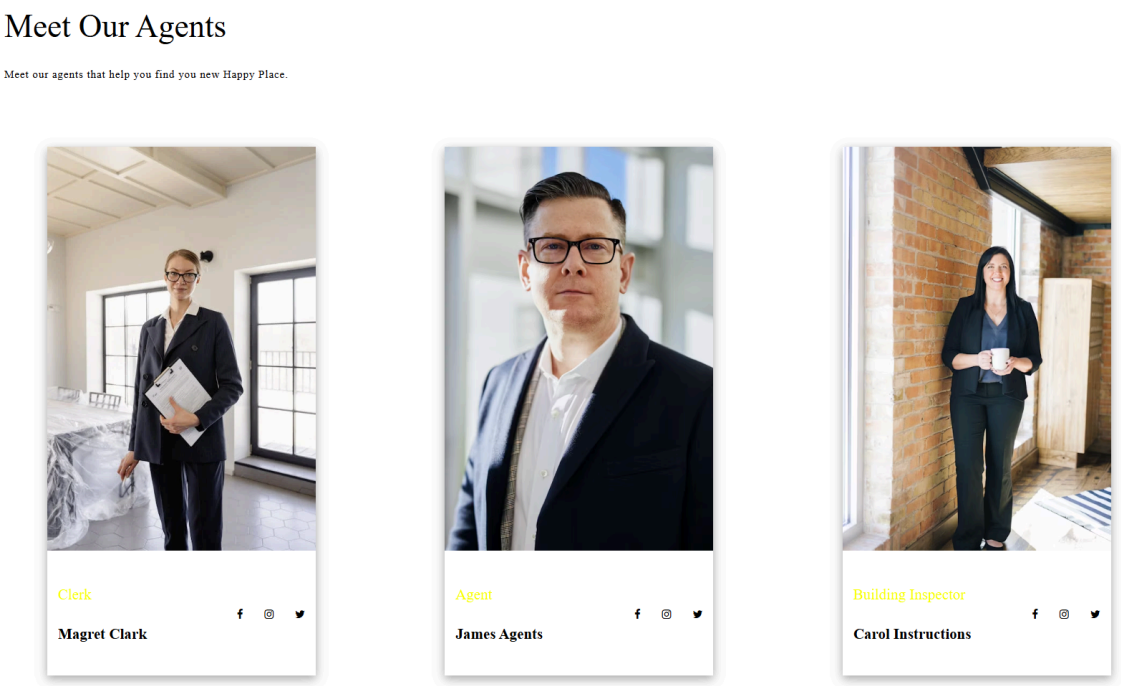
- **All Properties Listing**



- "The image displays the **listing of all properties** on the website, organized in a user-friendly grid or card layout. Each property listing includes key details such as price, size, number of bedrooms, location, and a thumbnail image. This setup allows users to quickly scan through available properties, making it easier to compare options and find the perfect fit."

- "This image illustrates the **property listing page**, where users can explore all available properties in a clean, structured layout. Each property is presented with its essential information, such as price, square footage, number of rooms, and location, helping users make informed decisions. The page also features filters for sorting by price, size, or property type to enhance the user experience.

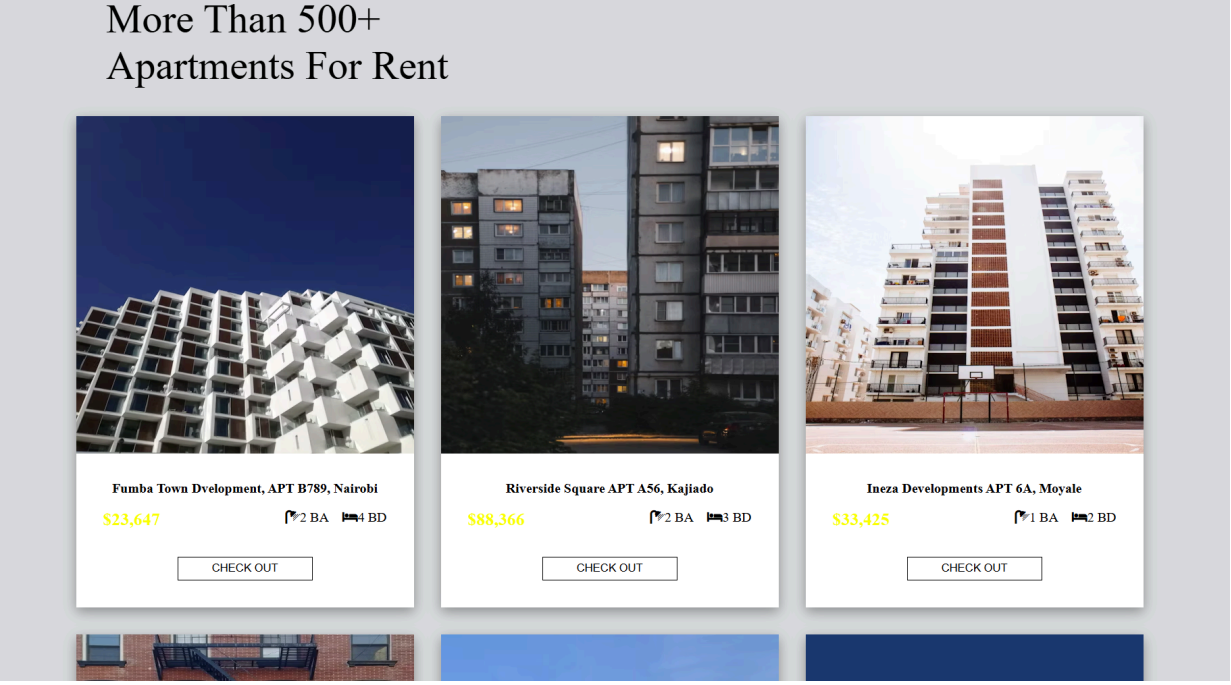
Agents



- The image features a section dedicated to **agents to contact**, providing users with direct access to the agents handling specific properties. Each agent's contact information, including phone number, email, and a contact form, is displayed, allowing users to easily get in touch for more details or inquiries.

- This image shows the **Get in Touch** section, where users can find contact details for property agents. With a list of available agents and their contact information, users can reach out directly to get more information about properties, schedule viewings, or ask any questions related to the listings.

- **Home Page Listings**

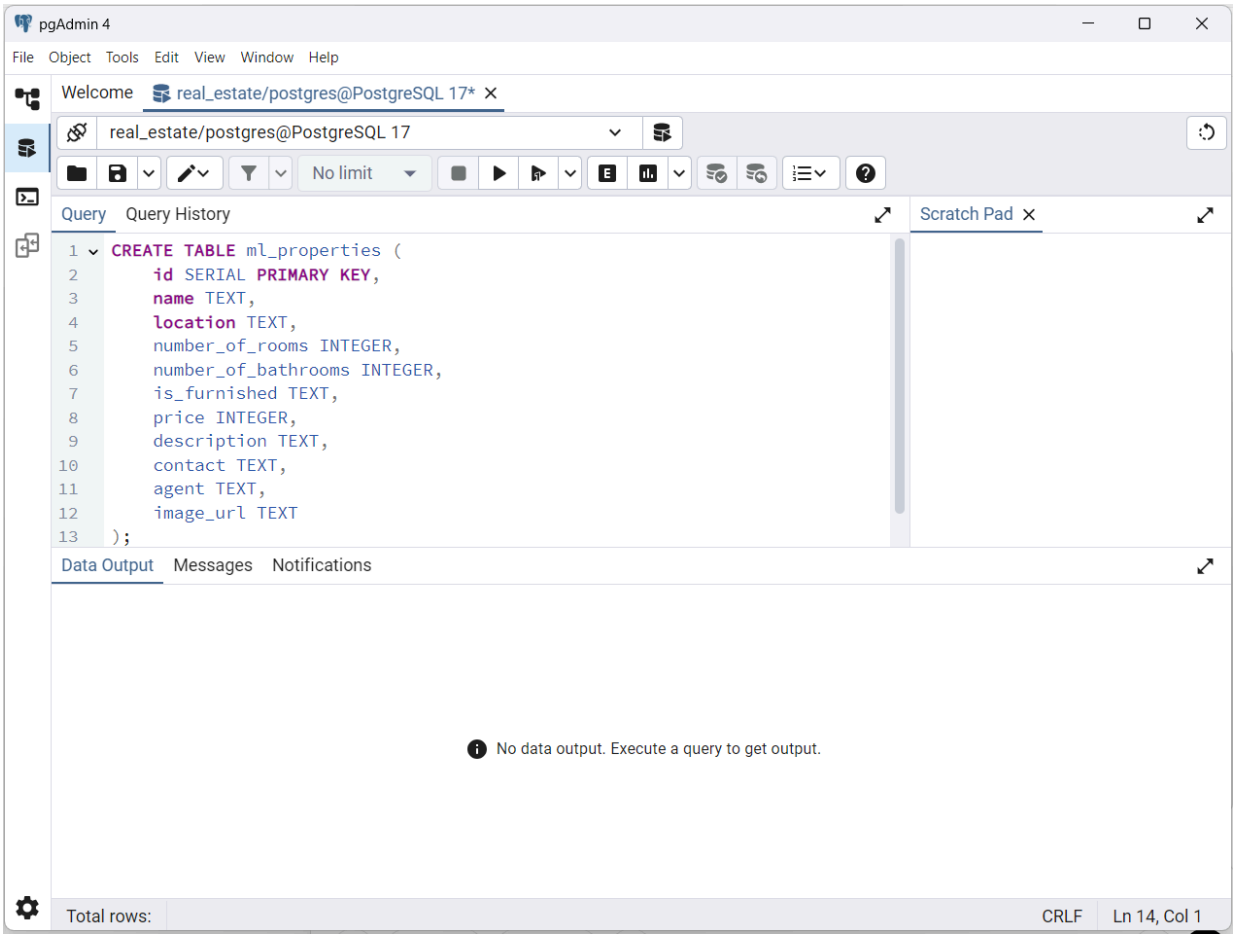


- The image showcases the **home page listings**, where featured properties are displayed in an appealing and organized grid layout. Each listing provides key details such as price, location, size, and property type, giving users an easy way to browse through the most relevant or highlighted properties on the platform.

- Here is the **home page listings** section, highlighting featured properties for quick viewing. The page displays properties with essential information like price, size, and location, providing a user-friendly experience that helps visitors quickly navigate and discover properties that match their preferences.

4.2.3 Database

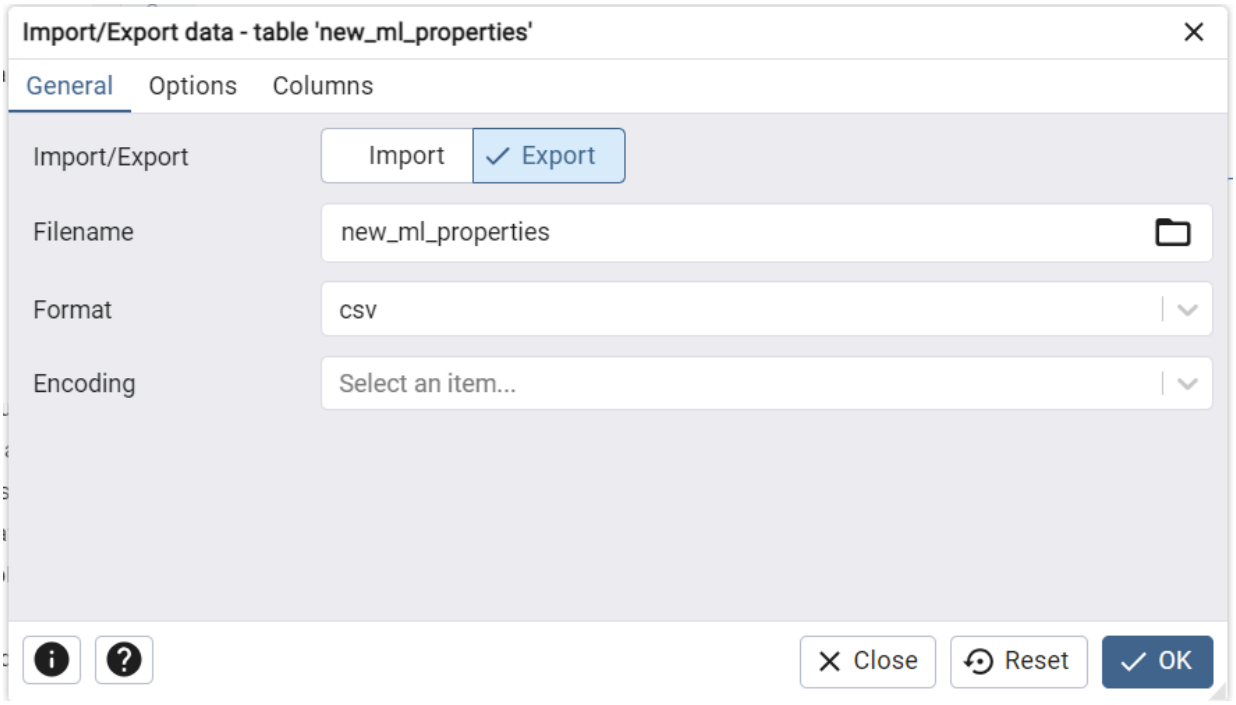
- **PostgreSQL Schema**



- The image shows the **PSQL schema** for the platform's database, outlining the structure of tables used to store property information, user data, and other related entities. This schema ensures efficient organization and management of real estate data, including properties, users (agents, sellers, renters), and transaction records.

- Here is the **PSQL schema** representing the relational database for the Renters and Sellers platform. It includes tables for properties, users, agents, and transactions, designed to support seamless data operations like listings, price prediction, and user interactions with a scalable and normalized structure.

- **Connection to DB**

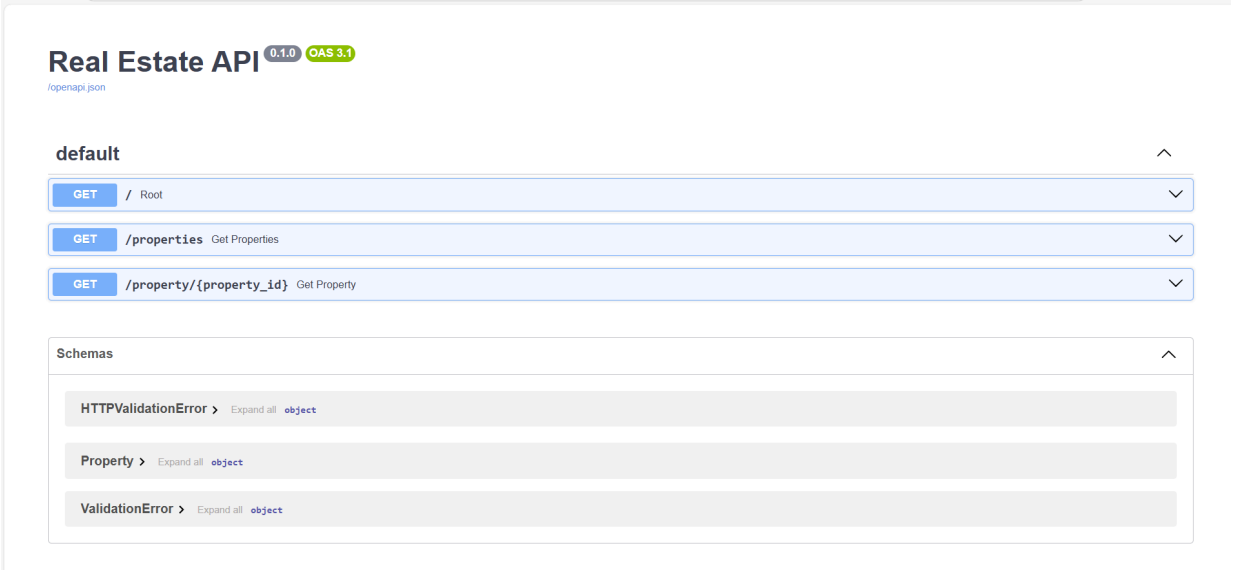


- "The image demonstrates the **connection to the database**, where the platform securely links to PostgreSQL using connection strings to retrieve and manipulate data. This ensures that all user interactions, such as property searches, rentals, and sales, are dynamically handled by the backend with real-time updates."

- "Here is the **database connection setup**, showcasing how the Renters and Sellers platform connects to the PostgreSQL database. This connection allows the application to fetch property details, process user inputs, and update records for rental and selling properties efficiently."

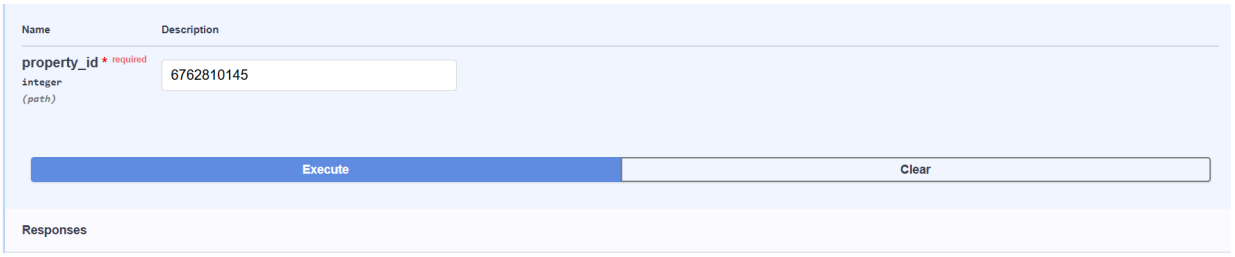
4.2.4 Backend

- **FastAPI**



FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints. It allows you to create web applications and APIs efficiently, with automatic validation, interactive documentation, and high performance. FastAPI is built on top of Starlette and Pydantic, ensuring easy handling of data validation, authentication, and routing.

➢ **Testing by ID**



➢ **JSON output**

The image shows **testing an API endpoint** by providing an ID and receiving a JSON output. This allows developers to verify the functionality of the endpoint, ensuring that the database query works correctly and returns the expected data in a structured JSON format. It demonstrates how FastAPI handles requests and responses efficiently for smooth user interactions.

CHAPTER 5: TESTING

5.1 Testing Plan / Strategy

In this project, all testing was performed manually to ensure that every functionality worked correctly. The testing process involved creating test cases, executing them in Xdebug, verifying expected vs. actual results, and fixing errors before final deployment. Through this approach, we successfully conducted the following types of testing:

1. Unit Testing

- **Unit Testing** was performed on each module of the Renters and Sellers platform, including property listings, search functionality, agent contact, and price prediction. Test cases were created to validate the accuracy and efficiency of each feature, ensuring that the expected output was delivered. Any issues found during testing were addressed using debugging tools, with modules being retested until they performed as intended.

2. Integration Testing

- **Integration Testing** was carried out to evaluate the interaction between various modules of the Renters and Sellers platform. For example, after a user filtered properties by price, it was tested whether the correct results appeared in real-time on the property listings page. The integration of FastAPI for backend processing, React for the frontend, and PostgreSQL for data management was tested to ensure seamless data synchronization and smooth user experience across the platform.

3. Performance Testing

- **Performance Testing** was conducted to assess the responsiveness and speed of the Renters and Sellers platform. Actions such as filtering properties, viewing agent details, and price predictions were manually tested to ensure quick load times. Dynamic features like real-time property search and price predictions were monitored to guarantee smooth interactions. SQL queries were optimized, and the performance of the PostgreSQL database was evaluated for efficient handling of large datasets, ensuring seamless user experience even with higher traffic and data volumes

4. Security Testing

- **Security Testing** was performed on the Renters and Sellers platform to identify and fix potential vulnerabilities, such as SQL injection, Cross-Site Scripting (XSS), and session hijacking. Various input fields, including property listings and agent contact forms, were tested with malicious code to ensure that the database and system remain secure. Additionally, user authentication methods, such as secure password hashing and session management, were rigorously tested to prevent unauthorized access and ensure data privacy

CHAPTER 6: CONCLUSION AND DISCUSSION

6.1 Conclusion

The Renters and Sellers platform successfully achieves its goal of simplifying property rentals and sales through its intuitive interface and advanced features. The system enables users to compare property prices, find similar listings, and predict prices using machine learning models. With secure login, interactive property listings, and agent contact details, the platform ensures a seamless experience for renters, sellers, and agents alike. The inclusion of real-time search filters and personalized recommendations enhances user engagement and satisfaction, making the platform an efficient and reliable tool for all users.

6.2 Overall Analysis of This Project

The Renters and Sellers platform is developed to modernize and streamline the property rental and sales process, providing a seamless, automated experience for users. By integrating machine learning for price prediction and similar listing recommendations, the system enhances the decision-making process for both renters and sellers.

Key strengths of the project include

- **Secure User Authentication**: The platform ensures safe access for agents, renters, and sellers through encrypted login features.
- **Property and Listing Management**: Sellers and agents can easily list, manage, and update properties, while renters can easily browse, filter, and compare listings.
- **Interactive Features**: Real-time search filters, dynamic recommendations, and agent contact options create an engaging, user-friendly experience for all.
- **Data Integrity and Security**: Built with robust technologies like FastAPI for backend processes, PostgreSQL for data storage, and secure authentication mechanisms, the platform ensures reliable data management and protection.

6.3 Problem Encountered and Possible Solution

During the development of the Renters and Sellers platform, several challenges were faced, and each was addressed with effective solutions. Some key issues and their solutions include:

Problem 1:Data Quality and Cleaning for Machine Learning Models:

- **Problem**: The dataset for property prices had missing values, inconsistencies, and noise that could affect the accuracy of the machine learning models.
- **Solution**: Data preprocessing techniques like filling missing values, handling outliers, and encoding categorical variables were applied to clean the data before training the models, resulting in better predictions.

Problem 2: Integration of Machine Learning Models with the Web Interface:

- **Problem**: Connecting machine learning models (like KNN and RandomForestRegressor) to the web interface for real-time predictions posed technical difficulties.
- **Solution**: The models were serialized using pickle files and integrated into the FastAPI backend. This allowed seamless communication between the frontend and the backend, enabling real-time predictions for users.

Problem 3: Performance Bottlenecks with Large Data:

- **Problem:** The platform faced slow query performance when retrieving large property listings, affecting user experience.
- **Solution**: Optimization techniques such as indexing database columns (like property price and location) and pagination for property listings were implemented, improving the speed and responsiveness of the platform.

6.4 Limitation and Future Enhancement

Limitations:

- **Limited Filtering Options:** The platform currently offers basic filters, but more advanced search options like filtering by property features (e.g., garden, parking spaces) or proximity to schools could improve the user experience.

- **No Mobile App:** While the website is responsive, the absence of a dedicated mobile app reduces convenience for on-the-go users who might want to list properties or check details from their mobile devices.

- **Manual Property Listings:** Property owners and agents have to manually list properties, and this process can lead to potential errors or delays in getting listings live.

- **Price Prediction Accuracy:** While the price prediction feature exists, the accuracy could be enhanced by incorporating more detailed data, including external factors like market trends or nearby amenities.

- **Limited Notifications:** The platform lacks an automated notification system for important events, such as price drops, new similar listings, or property status updates, which would improve user engagement and retention.

Future Enhancements:

1. **Mobile App Development:** Creating a dedicated mobile app for both iOS and Android would enhance accessibility, allowing users to manage listings, track rental/selling processes, and receive real-time notifications from anywhere.

2. **Advanced Search Filters:** Adding more detailed filters for properties, such as amenities (e.g., garden, parking, pool), proximity to key locations (e.g., schools, shopping malls), and property age, would allow users to refine their searches for a more tailored experience.

3. **Price Prediction Enhancement:** Integrating additional data sources, like market trends, economic conditions, and neighborhood data, into the price prediction models will improve the accuracy and reliability of property price estimations.

4. **Automated Notifications System:** Implementing a notification system for significant events like price changes, new property listings, upcoming payments, or property status updates will keep users engaged and informed in real-time.

5. **Property Verification and Certification:** Introducing a verification system for property listings to ensure that all details are accurate and that the property meets required standards would build trust among renters, buyers, and sellers.

6. **AI-powered Chatbot Support:** Implementing an AI-based chatbot for instant customer support can provide immediate assistance to users for common queries related to property listings, pricing, and more.

7. **Virtual Tours and Augmented Reality (AR) Integration:** Offering virtual tours of properties or incorporating AR for users to view properties in 3D will enhance the user experience and help potential renters and buyers visualize properties more effectively.

8. **Real-time Collaboration Tools:** Allowing renters, sellers, and agents to collaborate in real-time via messaging, document sharing, or video calls could streamline communication and make the rental or buying process more efficient.

6.5 Summary of Project Work

Project Title	Renters and Sellers	
Aim	The aim of your project, **Renters and Sellers**, is to create an intelligent and user-friendly real estate platform that connects renters, buyers, sellers, and agents. It aims to simplify the property search, buying, selling, and renting processes by offering key features like property listings, price prediction, similar property recommendations, and agent contact functionality.	
Developed by	Vasavada Het	
Project Category	Web-Based Application	
Tools	**IDE**	Visual Studio Code
	Language/ Framework	HTML, CSS,React, Python, FastAPI, Git
	Server & Database	PostgreSQL
	Testing & Debugging Tools	Console, FastAPI Docs, Develop Tools
	External Tools	Visual Paradigm, ChatGPT
Duration	3 Months	

REFERENCES

- Google Collab
- Chatgpt
- Visual Studio Code
- FastAPI
- PostgreSQL
- Stack OverFlow
- React
- Jupyter Notebook
- Diagrams