A PROJECT ON

Property Selling Site

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY
Hinjawadi

SUBMITTED BY:
Prasad Lanjile

UNDER THE GUIDENCE OF:
Snehal Jadhav
Faculty Member
Sunbeam Institute of Information Technology, Pune

<u>CERTIFICATE</u>

This is to certify that the project work under the title 'Web Portal for Student and teacher' is done by **Prasad Lanjile** in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

Project Guide : Snehal Jadhav   **Mr.Yogeh Kolhe**
                                                      Course Co-Ordinator

Date: 03/02/2026

ACKNOWLEDGEMENT

1. **INTRODUCTION TO PROJECT**

The web-based **Online Property Selling System** is a full-stack application developed to automate and simplify the process of buying and selling real estate properties through an online platform. The system is designed to provide an easy-to-use, secure, and efficient interface for different types of users involved in property transactions.

The application allows users to explore property listings, view detailed information, and interact with sellers through enquiries. A **guest or buyer** can browse available properties without logging in, enabling quick access to property information. However, to perform actions such as sending enquiries or managing personal interactions, the user must log in to the system.
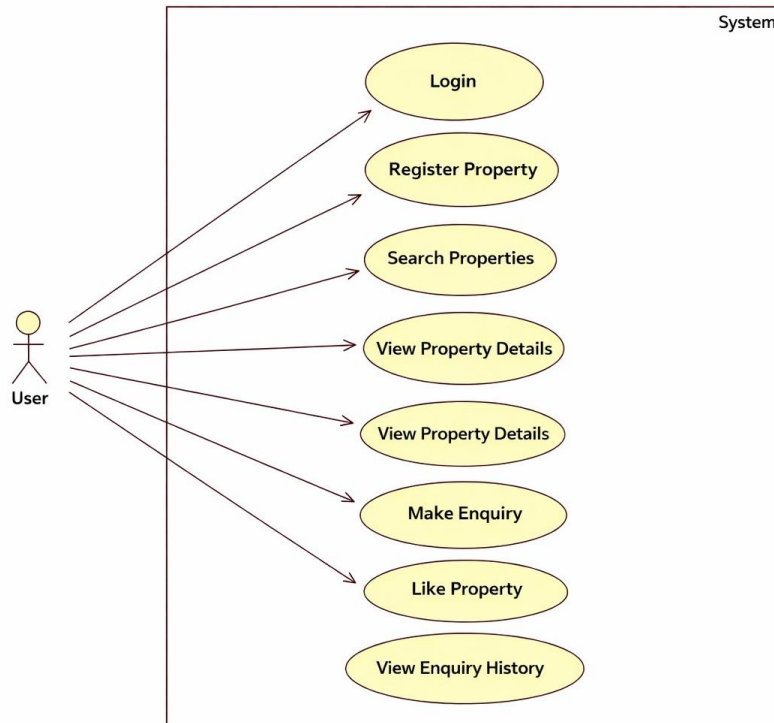
The system supports multiple user roles, including **Buyer**, **Seller**, and **Admin**, each with specific responsibilities. Sellers can list new properties, upload property images, and manage enquiries received from interested buyers. Buyers can search properties based on location, price, and type, and can send enquiries to sellers for further communication. The admin oversees the overall system by managing users and maintaining platform integrity.

The backend of the system is developed using **Spring Boot**, which handles business logic, REST API orchestration, user authentication, and property workflows. **MySQL** is used as the relational database to ensure reliable data storage and efficient retrieval of users, properties, images, and enquiries. The frontend is built using **React**, providing a responsive and interactive user experience.

To enhance system observability and debugging, the project integrates a **separate ASP.NET Core–based logging service** that uses structured logging to capture application events and errors. This microservice-based approach improves scalability, maintainability, and monitoring of the overall system.

Overall, the Online Property Selling System provides a secure, role-based, and scalable solution that demonstrates real-world full-stack application development with emphasis on clean architecture, authentication, and cross-technology integration.

2.REQUIREMENTS

2.1 FUNCTIONAL REQUIREMENTS

2.1   User Account

The Online Property Selling System supports multiple types of users, who are collectively referred to as **users**. The system presents different functionalities based on the user's role and authentication status. A user may interact with the system either as a **guest** or as a **registered user**.

A **guest user** is a visitor who has not logged into the system. A guest can browse property listings, view property details, and perform basic property searches without creating an account. However, a guest user is not permitted to perform actions such as sending enquiries or managing listings.

A **registered user** is a user who has created an account in the system by providing valid credentials. Registered users can log into the system using their email ID and password. Based on their role, registered users are categorized as **Buyer**, **Seller**, or **Admin**.

- A **Buyer** can search properties, view detailed listings, and send enquiries to sellers.
- A **Seller** can add new property listings, upload property images, and view enquiries received from buyers.
- An **Admin** can manage users, monitor properties, and maintain the overall integrity of the system.

The system allows registered users to log out or exit the application at any time during their interaction.

2.2 Registration and creation of user profile

The system requires users to register in order to access protected features such as sending enquiries or managing properties. During registration, the system collects essential information including user name, email address, password, contact number, and role selection.

Once the registration process is completed successfully, the system automatically creates a **user profile** in the database. The profile stores user credentials and role information, which is later used for authentication and authorization. Passwords are stored in encrypted form to ensure data security.

2.1.3 Property Search and Quick Browse

The system provides a **Quick Search** facility that allows users to browse available properties without logging into the system. This feature enables users to search and compare properties based on parameters such as city, state, property type, and price range.

After entering search criteria, the system retrieves matching property listings from the database and displays them in a structured format. Each listing includes details such as property title, price, location, property type, and a representative image.

Registered users can access additional features after logging in, such as saving preferences or initiating enquiries. The system ensures that invalid or incompatible search inputs are handled gracefully and displays appropriate messages when no matching properties are found.

2.1.4 Property Enquiry and Interaction

The system allows a **Buyer** to send enquiries to sellers regarding properties of interest. To create an enquiry, the buyer must be logged into the system. The enquiry includes a message and is linked to both the buyer and the selected property.

Once an enquiry is created, the system stores it in the enquiry database and updates relevant property statistics. Sellers can view and respond to enquiries received for their listed properties. Each enquiry maintains a status such as *Open*, *Responded*, or *Closed* to track its lifecycle.

2.5 Property Listing Management

A **Seller** can add new property listings by entering property details such as title, description, location, price, and property type. The system allows sellers to upload multiple images for each property.

Sellers can view, update, or remove their own property listings. The system ensures that sellers can modify only the properties that they own, enforcing ownership-based access control.

2.1.6 Administrative Functions

The Admin role has full access to system management features. An admin can view user information based on user ID, monitor property listings, and take administrative actions such as blocking users or removing inappropriate content. The admin ensures smooth operation of the platform by overseeing users and maintaining data integrity across the system.
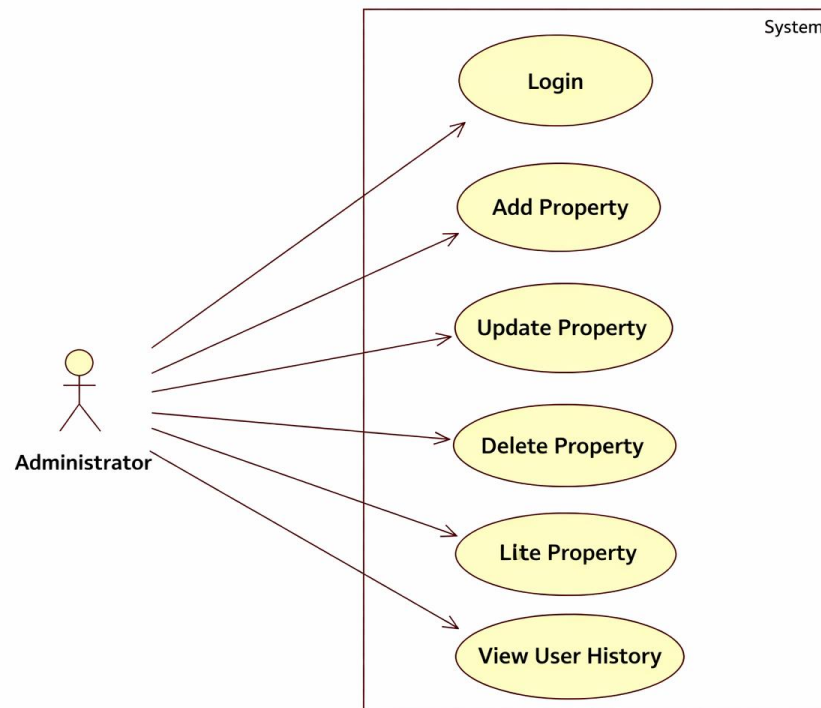
Admin should be able to login ,add airline information , add flight information, Delete flight and see user Information according to user Id.

2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Interface Requirements

The system provides a responsive and user-friendly web interface developed using React. The interface is designed to be intuitive and accessible across different devices and screen sizes. Detailed user interface designs are provided in the appendix section.

2.2.2 Performance Requirements

- Concurrent Users:
  The system should be capable of handling multiple concurrent users without performance degradation.
- Response Time:
  Property search, listing retrieval, and enquiry creation operations should be completed within acceptable response times under normal load conditions.
- Scalability:
  The backend architecture supports scalability through REST-based services and stateless JWT authentication.

2.2.3 Security Constraints

- The system uses **JWT-based authentication** to ensure secure access to APIs.
- Role-based access control ensures that users can access only permitted functionalities.
- Sensitive data such as passwords are encrypted before storage.
- Unauthorized access attempts are restricted by the security framework.

2.2.4 Other Requirements:

Hardware Interfaces

The system is designed to run on standard computing hardware with moderate processing and memory capabilities. It can be accessed from devices with modern web browsers.

Software Interfaces

- Backend Framework: Spring Boot
- Frontend Framework: React
- Database: MySQL
- Security: Spring Security with JWT
- Logging Service: ASP.NET Core with Serilog

The system is platform-independent and can be accessed using modern browsers such as Chrome, Firefox, or Edge.

3. DESIGN

3.1 Database Design

The Online Property Selling System uses a **relational database design** implemented using **MySQL**. The database is designed to store user information, property listings, enquiries, property interactions such as likes and views, and related metadata. The database follows normalization principles to avoid redundancy and ensure data integrity.

The following table structures depict the database design of the system.

Table1: User
This table stores information about all registered users of the system, including buyers, sellers, and administrators.

Key Type / Constraint	Column Name	Data Type	Length	Allow Null (1=Yes; 0=No)
Primary Key	user_id	BIGINT	—	0
—	created_at	DATETIME	—	1
Unique	email	VARCHAR	100	0
—	name	VARCHAR	100	0
—	password	VARCHAR	255	0
—	phone	VARCHAR	20	1
—	role	ENUM (ADMIN, BUYER, SELLER)	—	0

Table2: properties
This table stores details of properties list

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
Primary Key	property_id	BIGINT	—	0
—	title	VARCHAR	150	0
—	description	TEXT	—	1
—	city	VARCHAR	100	0
—	state	VARCHAR	100	0
—	pincode	VARCHAR	10	1
—	price	DECIMAL	12,2	0
—	property_type	ENUM (APARTMENT, HOUSE, LAND, VILLA)	—	0

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
—	status	ENUM (AVAILABLE, PENDING, SOLD)	—	0
—	created_at	DATETIME	—	1
Foreign Key	seller_id	BIGINT	—	1

Table3: property_images

This table stores images associated with each property. A property can have multiple images.

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
Primary Key	image_id	BIGINT	—	0
—	image_url	VARCHAR	255	0
Foreign Key	property_id	BIGINT	—	1

Table4: Enquiries

This table records enquiries sent by buyers for specific properties.

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
Primary Key	enquiry_id	BIGINT	—	0
—	message	TEXT	—	1
—	status	ENUM (OPEN, RESPONDED, CLOSED)	—	0
—	created_at	DATETIME	—	1
Foreign Key	buyer_id	BIGINT	—	0
Foreign Key	property_id	BIGINT	—	0

Table5: property_likes

This table tracks properties liked by buyers, allowing users to mark properties as favorites.

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
Primary Key	like_id	BIGINT	—	0
—	liked_at	DATETIME	—	1
Foreign Key	buyer_id	BIGINT	—	0
Foreign Key	property_id	BIGINT	—	0

Table6: property_views

This table stores information about property views to support analytics and tracking.

Key Type / Constraint	Column Name	Data Type	Length	Allow Null
Primary Key	view_id	BIGINT	—	0
—	viewed_at	DATETIME	—	1
Foreign Key	user_id	BIGINT	—	0
Foreign Key	property_id	BIGINT	—	0

3.2 Entity Relationship Overview

The database design establishes the following relationships:

- A User (Seller) can list multiple Properties
- A Property can have multiple Images
- A User (Buyer) can create multiple Enquiries
- A Property can receive multiple Enquiries
- A Buyer can like multiple properties
- A Property can be viewed multiple times by different users

These relationships ensure logical data organization and support scalability and analytics requirements.

E-R Diagram,Dataflow diagram and Class Diagram:

4. CODING STANDARDS IMPLEMENTED

Naming and Capitalization

Below summarizes the naming recommendations for identifiers in Pascal casing is used mainly (i.e. capitalize first letter of each word) with camel casing (capitalize each word except for the first one) being used in certain circumstances.

Identifier	Case	Examples	Additional Notes
Class	Pascal	Person, BankVault, SMSMessage, Dept	Class names should be based on "objects" or "real things" and should generally be **nouns**. No '_' signs allowed. Do not use type prefixes like 'C' for class.
Method	Camel	getDetails, updateStore	Methods should use **verbs** or verb phrases.
Parameter	Camel	personName, bankCode	Use descriptive parameter names. Parameter names should be descriptive enough that the name of the parameter and its type can be used to determine its meaning in most scenarios.
Interface	Pascal with "I" prefix	Disposable	Do not use the '_' sign
Property	Pascal	ForeColor, BackColor	Use a noun or noun phrase to name properties.
Associated private member variable	_camelCase	_foreColor, _backColor	Use underscore camel casing for the private member variables

Exception Class	Pascal with "Exception" suffix	WebException,	

Comments

- Comment each type, each non-public type member, and each region declaration.

- Use end-line comments only on variable declaration lines. End-line comments are comments that follow code on a single line.

- Separate comments from comment delimiters (apostrophe) or // with one space.

- Begin the comment text with an uppercase letter.

- End the comment with a period.

- Explain the code; do not repeat it.

5. TEST REPORT

Another group called Linux did the testing and the report of the testing is given hereunder.
GENERAL TESTING:

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
1	User Registration	User account should be created successfully	OK	Nothing
2	Login Page	User should be able to login with valid credentials	OK	Nothing
3	Invalid Login	System should reject invalid credentials	OK	Invalid email or password
4	Logout	User should be logged out successfully	OK	Nothing
5	Browse Properties (Guest)	Properties should be displayed without login	OK	Nothing
6	Property Search	Properties should be filtered based on search criteria	OK	Nothing
7	View Property Details	Complete property details should be shown	OK	Nothing
8	Add Property (Seller)	Seller should be able to add a new property	OK	Nothing
9	Upload Property Images	Images should be uploaded and linked to property	OK	Nothing
10	Send Enquiry (Buyer)	Enquiry should be saved and sent to seller	OK	Nothing
11	View Enquiries (Seller)	Seller should see enquiries for own properties	OK	Nothing
12	Property Like	Buyer should be able to like a property	OK	Nothing
13	Property View Tracking	Property view should be recorded in database	OK	Nothing
14	Role-Based Access	Unauthorized users should be restricted	OK	Access Denied
15	Admin Login	Admin should login successfully	OK	Nothing

SR-NO	TEST CASE	EXPECTED RESULT	ACTUAL RESULT	ERROR MESSAGE
16	Admin View Users	Admin should view user details	OK	Nothing
17	Security (JWT Validation)	Invalid or expired token should be rejected	OK	Unauthorized
18	Logging Service Integration	Logs should be sent to logging service	OK	Nothing

6. PROJECT MANAGEMENT RELATED STATISTICS

The development of the **Online Property Selling System** was carried out in a phased manner following standard software engineering practices. The project progressed through requirement analysis, design, implementation, testing, and deployment phases. The following table summarizes the project management activities and milestones achieved during the development lifecycle.

DATE	WORK PERFORMED	SDLC PHASE	ADDITIONAL NOTES
Jan 05, 2026	Project Allotment and Problem Definition	Feasibility Study	Project scope and objectives were discussed and finalized
Jan 07, 2026	Requirement Gathering and Domain Study	Requirement Analysis	Functional and non-functional requirements were identified
Jan 09, 2026	Preparation and Validation of SRS Document	Requirement Analysis (Elicitation)	Initial SRS reviewed and refined
Jan 11, 2026	Design of Use Case Diagram and ER Diagram	Requirement Analysis & Design	Database entities and relationships finalized
Jan 13, 2026	Database Schema Design	Design Phase	MySQL database tables created
Jan 15, 2026	Backend Architecture Design	Design Phase	Layered architecture using Spring Boot finalized
Jan 17, 2026	Coding Phase Started (Backend APIs)	Coding Phase	Core APIs for users and properties implemented
Jan 19, 2026	Implementation of Authentication & Authorization	Coding Phase	JWT-based Spring Security integrated
Jan 21, 2026	Frontend Development Started	Coding Phase	React UI components implemented
Jan 23, 2026	Integration of Frontend and Backend	Coding Phase	REST API integration completed
Jan 24, 2026	Implementation of Enquiry and Property Interaction Modules	Coding Phase	Likes, views, and enquiries implemented
Jan 26, 2026	Integration of ASP.NET Core Logging Service	Coding Phase	Structured logging using Serilog implemented
Jan 27, 2026	Unit Testing of Individual Modules	Coding Phase & Unit Testing	Core modules tested independently
Jan 28, 2026	Integration Testing	Testing Phase	End-to-end workflows validated

DATE	WORK PERFORMED	SDLC PHASE	ADDITIONAL NOTES
Jan 29, 2026	Bug Fixing and Performance Improvements	Debugging Phase	Identified issues resolved
Jan 30, 2026	Final Testing and Validation	Testing Phase (Acceptance Testing)	Application validated for submission
Jan 31, 2026	Deployment on Cloud Platform	Deployment Phase	Application deployed and made publicly accessible
Feb 01, 2026	Final Documentation and Project Submission	Closure Phase	Project documentation completed

Appendix A
Entity Relationship Diagram
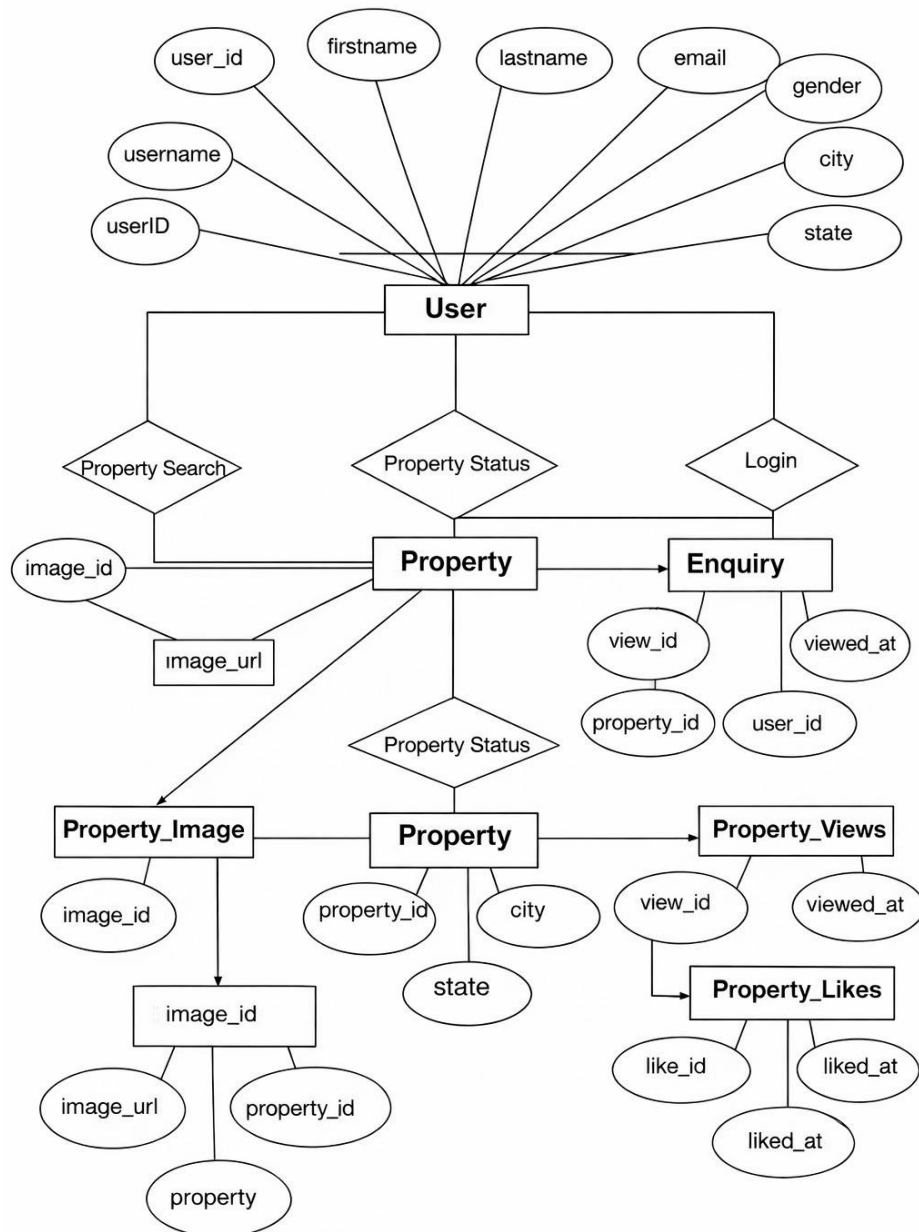
Data Flow Diagram:
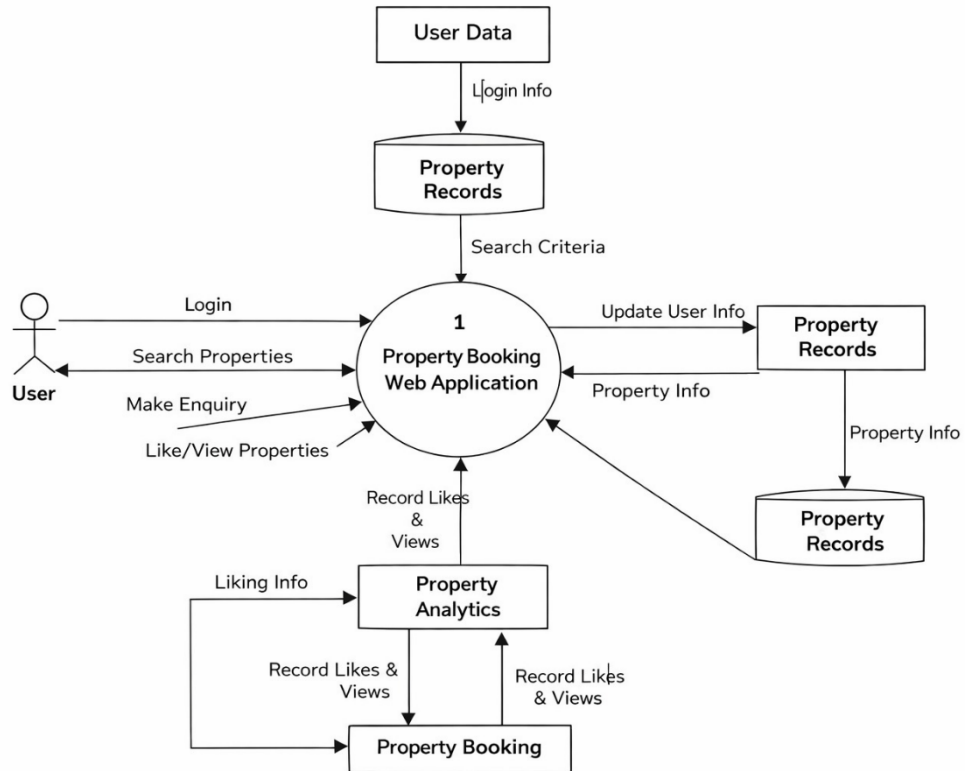
Class Diagram
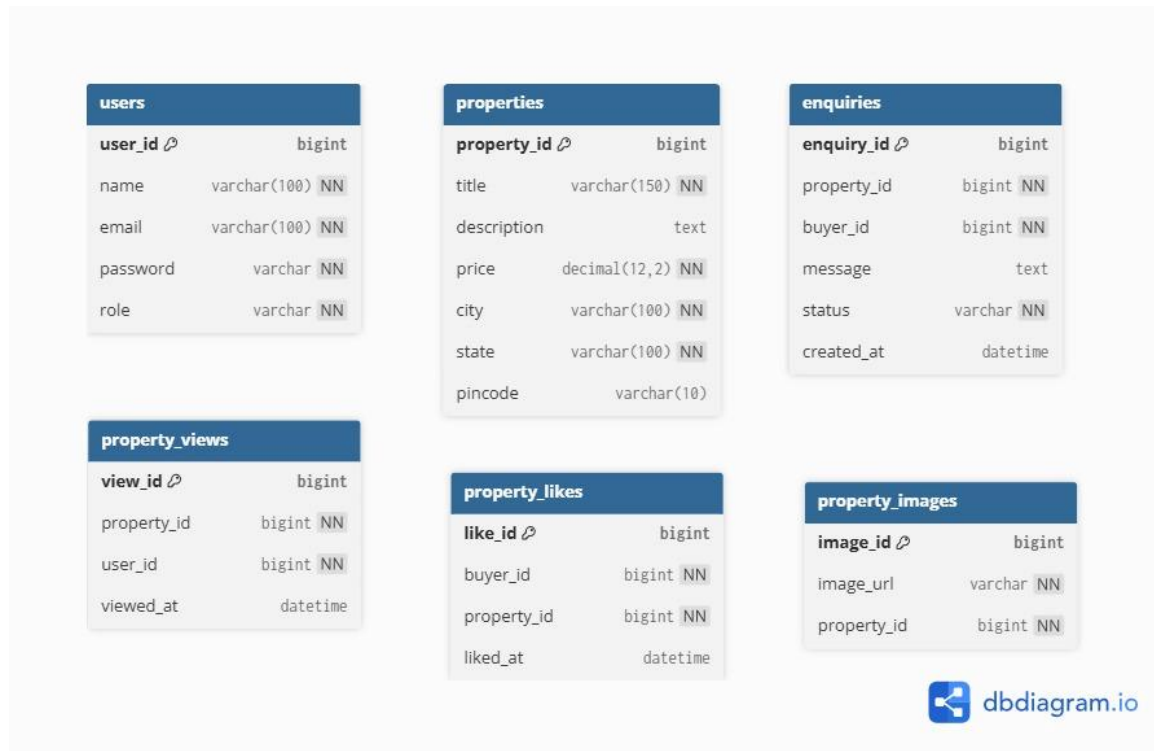
Appendix B
Homepage:
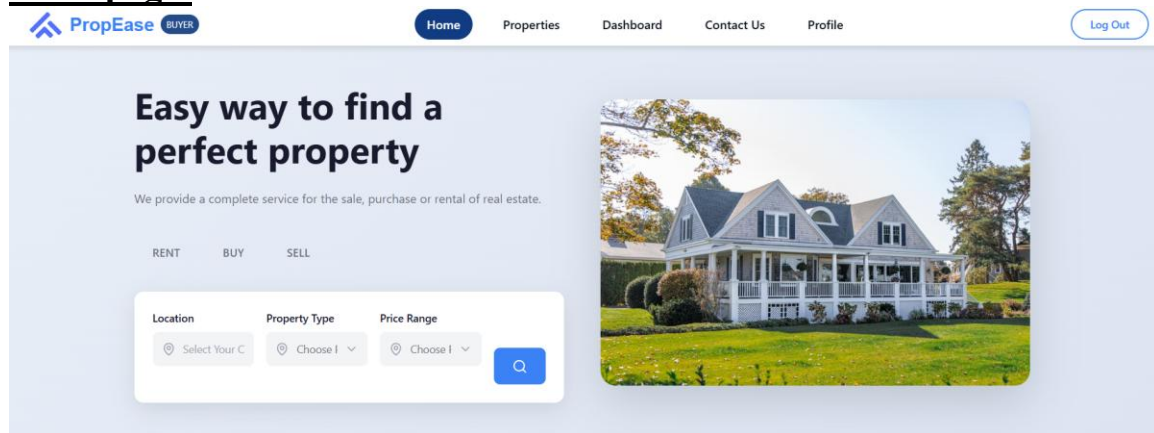
LoginPopup:

Administrators Login:

User Registration:

Create account

Join us today and find your perfect property

Full Name

👤 Enter your full name

Email

✉ pr@gmail.com

Phone Number

📞 Enter your phone number

Choose Role

👤 Buyer

Password

🔒 ••••   👁

Confirm Password

Add Property

PropEase SELLER   Seller Dashboard   My Properties   **Add Property**   Profile

Add New Property

Title

Description

Price (INR)	Property Type
	HOUSE
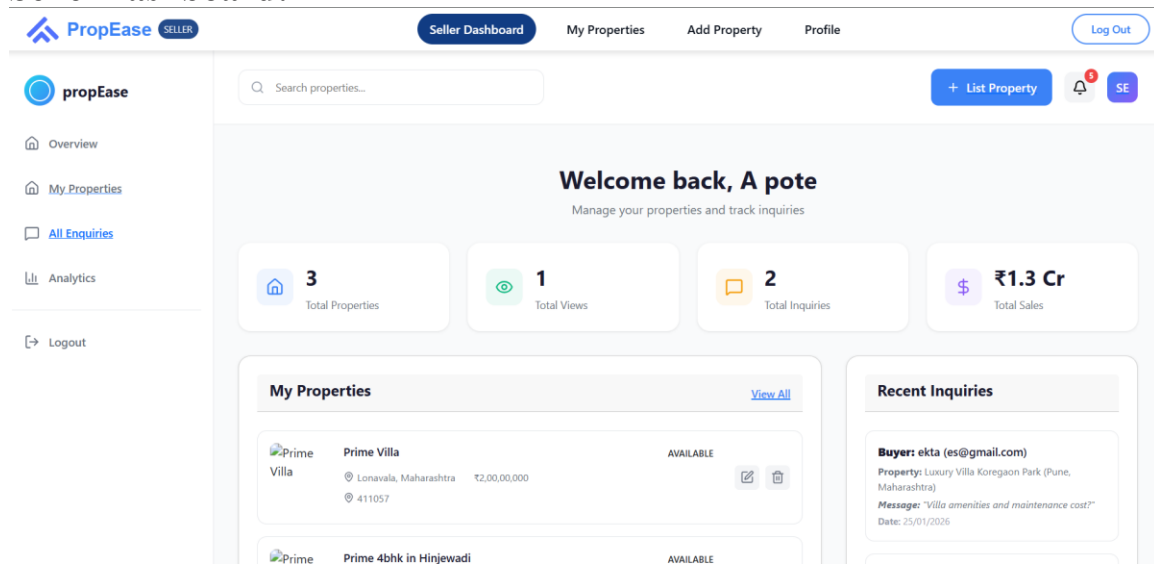
City	State
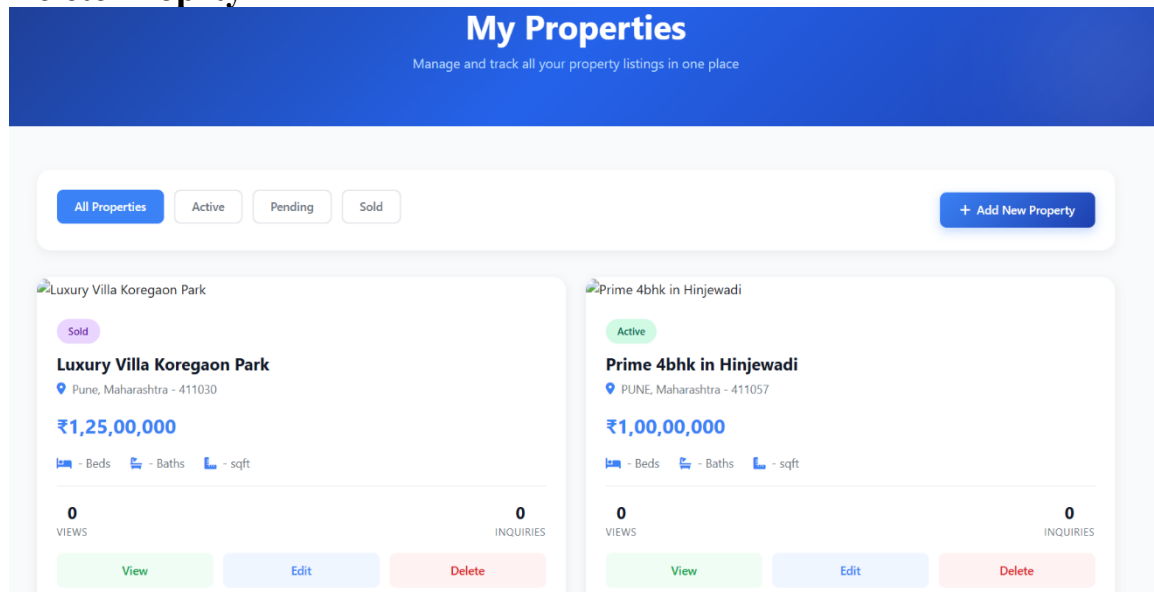
Pincode	Status
	AVAILABLE

SellerDashboard:

Delete Proprty

PropertyDetails:

BuyerDashboard:

AdminDashboard:

UserProfileDetails:

UserDetails:

Users Management

Manage buyers and sellers on the platform

11 Total Users	**4** Total Sellers	**6** Total Buyers	**1** Total Admins

Search by name, email, or phone... | All Roles

Showing **10** users

NAME	EMAIL	CONTACT	ROLE	JOIN DATE	ACTIONS
PR Prasadd R	pr@gmail.com	7894561230	BUYER	21/01/2026	👁 ✏ 🗑
AG A Gojegave	ag@gmail.com	1122334455	SELLER	21/01/2026	👁 ✏ 🗑
AP A pote	ap@gmail.com	9988774400	SELLER	26/01/2026	👁 ✏ 🗑
K karan	kb@gmail.com	5556330011	SELLER	26/01/2026	👁 ✏ 🗑

PropertyList:

PropEase **ADMIN**   Admin Dashboard   **Manage Properties**   Users   Profile   Log Out

All Properties

Manage and monitor all listed properties on the platform

Search by title, seller, or location...   All Status   All Types

Showing **7** properties

PROPERTY	SELLER	LOCATION	TYPE	PRICE	STATUS	DATE
Prime 2BHK in Pimpri	A Gojegave	Pimpri, Maharashtra, 411018	APARTMENT	₹4,500,000	Sold	Jan 20, 2026
Luxury Villa Koregaon Park	A pote	Pune, Maharashtra, 411030	VILLA	₹12,500,000	Sold	Jan 22, 2026
Bandra 3BHK House	karan	Mumbai, Maharashtra, 400001	HOUSE	₹8,500,000	Available	Jan 23, 2026
Nashik Residential Plot	A Gojegave	Nashik, Maharashtra, 422001	LAND	₹2,500,000	Available	Jan 24, 2026

7.REFERENCES:

http://www.google.com

http://www.airIndia.com

http://www.webdevelopersjournal.com/

http://www.lufthansa.com

http://www.w3.org

http://www.wikipedia.org

http://www.delta.com

http://www.priceline.com