

Project Title:

AI-Driven Real Estate Document Verification & Intelligence System

Name and Email:

Sharan Muthu Krishna K – sharanmuthukrishna@gmail.com

GitHub:

<https://github.com/Sharan-Muthu-Krishna/AI-Powered-Real-Estate-Management>

| S. No | Index | Page No. |
|-------|---|----------|
| 1 | Introduction | 2 |
| 2 | Project Objective | 2 |
| 3 | System Overview | 2 |
| 4 | AI & Machine Learning Modules | 3 |
| 5 | Retrieval-Augmented Generation (RAG) System | 3 |
| 6 | Backend & Microservice Design | 4 |
| 7 | Technology Stack | 4 |
| 8 | Performance & Evaluation | 5 |
| 9 | How to Run the System | 5 |
| 10 | Screenshots | 6 |
| 11 | Conclusion | 15 |

1. Introduction

The real estate industry heavily depends on legal documents such as property agreements, identity proofs, contracts, and ownership records. These documents are often vulnerable to forgery, manipulation, and fraud, which can lead to major financial and legal losses.

This project presents an **AI-powered Real Estate Management System** that ensures:

- **Authenticity of uploaded documents**
- **Intelligent querying of legal records**
- **Secure handling of property data**

The system combines **Spring Boot enterprise backend** with **AI-driven Python microservices** for document verification and smart question answering.

2. Project Objective

The main goals of this project are:

- To detect forged or tampered images using CNN-based forgery detection
- To allow users to ask natural language questions about legal documents
- To retrieve accurate answers using Retrieval-Augmented Generation (RAG)
- To manage users and properties using a secure backend

3. System Overview

The system consists of three major components:

1. **Spring Boot Backend**
 - Handles users, properties, authentication, and API routing
2. **Forgery Detection Service (Flask)**
 - Uses CNN + Error Level Analysis (ELA) to detect image manipulation
3. **RAG Query Service (FastAPI)**
 - Allows intelligent search and question answering over uploaded documents

Together, they form a **secure, AI-driven real estate platform**.

4. AI & Machine Learning Modules

Image Forgery Detection

The system uses:

- **Error Level Analysis (ELA)** to expose manipulation patterns
- **A MobileNet-based Convolutional Neural Network** fine-tuned on real and forged document images

The CNN learns to distinguish:

- Authentic documents
- Digitally manipulated or altered documents

The model outputs:

- Probability of authenticity
- Probability of forgery

This allows automated verification of legal and property images.

5. Retrieval-Augmented Generation (RAG) System

The RAG pipeline enables users to **query documents using natural language**.

It works in four stages:

1. Document Chunking

- Uploaded documents are split into small meaningful parts

2. Embedding

- Each chunk is converted into a vector representation

3. Vector Database

- Embeddings are stored for semantic search

4. Answer Generation

- Relevant chunks are retrieved
- The Gemma 3N model generates a human-like answer

This allows users to ask:

“What is the owner name?”
“What is the policy duration?”
“What are the payment terms?”

and receive intelligent answers.

6. Backend & Microservice Design

Spring Boot (Core System)

It manages:

- Users and roles (Admin, Agent, Customer)
- Property listings
- API communication with AI services

Flask (Forgery Detection)

- Receives images
- Applies ELA
- Runs CNN model
- Returns authenticity and forged score

FastAPI (RAG Engine)

- Stores document embeddings
- Runs semantic search
- Calls Gemma 3N API
- Returns AI-generated answers

7. Technology Stack

| Layer | Technology |
|----------|--------------------------------------|
| Backend | Java 17, Spring Boot, JPA, Hibernate |
| Frontend | HTML, CSS, JavaScript, Thymeleaf |

| | |
|-------------------|---------------------------------|
| Forgery Detection | Python, Flask, TensorFlow/Keras |
| RAG System | Python, FastAPI, LangChain |
| LLM | Gemma 3N API |
| Database | MySQL |
| Vector Store | FAISS |

8. Performance & Evaluation

Forgery Detection

- CNN correctly identifies manipulated document images
- ELA improves visibility of tampered regions
- Processing time: **2–10 seconds per document (CPU)**

RAG Querying

- Query response time: **5–12 seconds**
- Depends on:
 - Document size
 - Number of chunks
 - External LLM API

Vector search is optimized using FAISS, keeping retrieval under **1 second**.

9. How to Run the System

1. Start Spring Boot

```
mvn spring-boot:run
```

2. Start RAG Service

```
cd rag
venv\Scripts\activate
uvicorn app.main:app --reload
```

3. Start Forgery Detection

```
cd forgery_detection_api  
venv\Scripts\activate  
python app.py
```

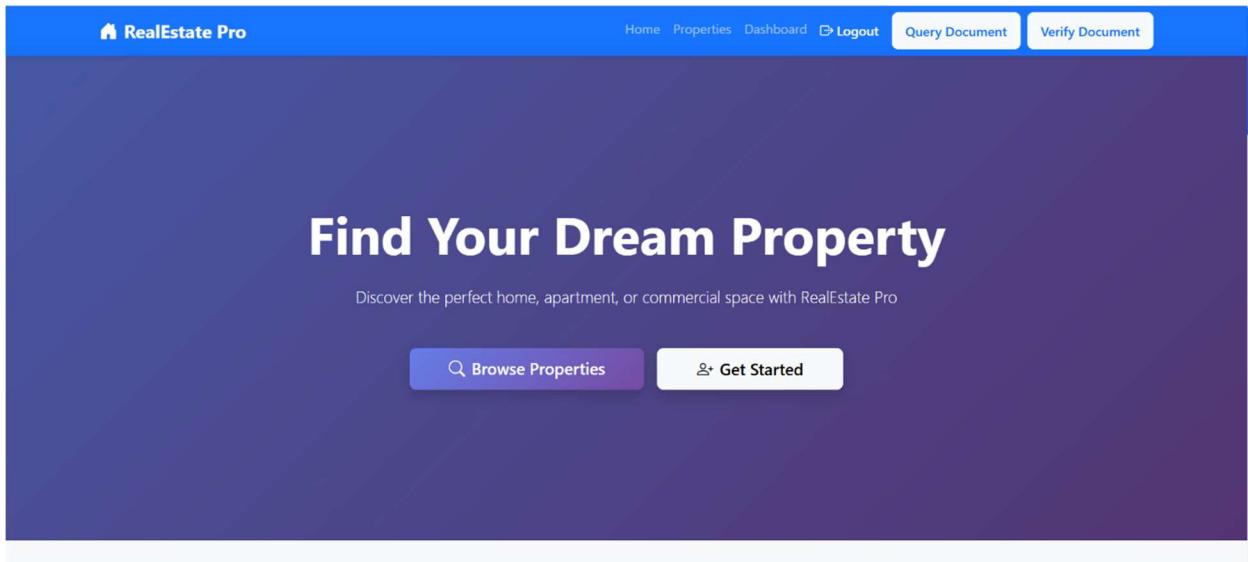
4. Start MySQL

```
mysql -u root -p
```

10. Screenshots

10.1 Public Home Page & Property Browsing

This screenshot shows the public home page of the Real Estate platform. Users can browse property listings, view featured properties, and navigate through different pages without logging in. This ensures open access for visitors to explore available properties before creating an account.



Why Choose Us

We provide the best real estate services



Quality Properties

Handpicked properties that meet the highest standards of quality and comfort



Expert Agents

Professional agents ready to guide you through every step of your journey

Secure Transactions

Safe and transparent transactions backed by legal expertise

Featured Properties

Explore our handpicked selection of premium properties

[← Back to Properties](#)

Interested in this property?

Send us an inquiry and our agents will contact you soon

Please [login](#) to send an inquiry

[Login to Inquire](#)

Need help?

+1 (555) 123-4567

All Properties

Browse through our extensive collection of properties

Property Type

All Types

Location

Enter location

Min Price

\$0

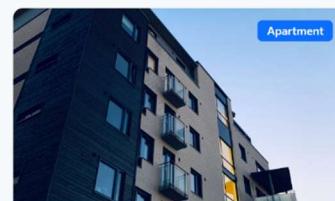
Max Price

\$999,999,999

[Search](#)

Pondicherry
vellore

Found 11 properties



10.2 User Registration & Login

This screenshot displays the user authentication system where users can create a new account and log in securely.

It verifies that the Spring Boot backend correctly handles user registration, credential validation, and role-based access control.

The screenshot shows the 'Create Account' page of the RealEstate Pro application. The header includes the logo, navigation links for Home, Properties, and Login, and buttons for Query Document and Verify Document. A 'User +' icon is also present. The main form is titled 'Create Account' with the sub-instruction 'Join our real estate community'. It contains fields for 'Full Name' (placeholder: Enter your full name), 'Email Address' (placeholder: alice@example.com), and 'Password' (placeholder:). Below the password field is a note: 'Password must be at least 6 characters long'. A tooltip indicates that the account will be registered as a 'Customer'. A large purple 'Create Account' button is at the bottom.

The screenshot shows the 'Welcome Back' page of the RealEstate Pro application. The header includes the logo, navigation links for Home, Properties, and Login, and buttons for Query Document and Verify Document. A 'User' icon is shown. The main message is 'Welcome Back' with the sub-instruction 'Sign in to your account'. A green success message box says 'You have been logged out successfully!' with a close button. The login form fields are 'Email Address' (placeholder: admin@realestate.com) and 'Password' (placeholder:). A purple 'Sign In' button is at the bottom. Below it, a link says 'Don't have an account? [Register here](#)'. A demo credentials box at the bottom lists 'Demo Credentials: Admin: admin@realestate.com / admin123'.

10.3 Admin Dashboard – Users, Properties & Inquiries

This screenshot shows the admin dashboard.

The administrator can:

- Manage users and roles
- Add, update, and delete property listings
- View and track customer inquiries

The screenshot displays the Admin Dashboard for RealEstate Pro. At the top, there's a blue header bar with the logo, navigation links (Home, Properties, Dashboard, Logout), and two buttons (Query Document, Verify Document). Below the header, the title "Admin Dashboard" is centered. A welcome message "Welcome back! Here's an overview of your system" follows. The dashboard features three main stats in cards: "Total Users" (3, Manage Users button), "Total Properties" (11, Manage Properties button), and "Active Inquiries" (1, Manage Inquiries button). Underneath these are sections for "Quick Actions" (Add Property, View Users, View Inquiries, View Public Site) and "Recent Properties" (a table with columns: Title, Location, Type, Price, Size).

This screenshot shows another view of the Admin Dashboard. It has a similar layout with the RealEstate Pro logo and navigation at the top. The "Quick Actions" section includes "Add Property" (purple button) and "View Users, Inquiries, Public Site" (light blue buttons). The "Recent Properties" section displays a table of five properties with columns: Title, Location, Type, Price, and Size. Each row includes a "View Details" button.

RealEstate Pro

Home Properties Dashboard [Logout](#) [Query Document](#) [Verify Document](#)

Manage Users

View and manage all registered users

| ID | Name | Email | Role | Actions |
|----|----------------|-----------------------|----------|------------------|
| 1 | Admin User | admin@realestate.com | ADMIN | |
| 2 | John Agent | agent1@realestate.com | AGENT | |
| 4 | Alice Customer | alice@example.com | CUSTOMER | |

[← Back to Dashboard](#)

RealEstate Pro

Your trusted partner in finding the perfect property. We offer the best real estate solutions for buying, selling, and renting.

Quick Links

- [Home](#)
- [Properties](#)

Contact Us

- 123 Real Estate Ave, City
- +1 (555) 123-4567

RealEstate Pro

Home Properties Dashboard [Logout](#) [Query Document](#) [Verify Document](#)

Manage Properties

Add, edit, or delete properties

[← Back to Dashboard](#)

Add New Property

| | | |
|----------------------|---|----------------------|
| Title | Price | Size (sq ft) |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Location | Type | |
| <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Description | <input type="text"/> | |
| Property Image | <input type="button" value="Choose File"/> No file chosen <small>Optional: Upload a property image</small> | |

RealEstate Pro

Home Properties Dashboard [Logout](#) [Query Document](#) [Verify Document](#)

Manage Inquiries

View and update customer inquiries

[← Back to Dashboard](#)

| ID | Customer | Property | Message | Status | Date | Actions |
|----|----------------|-----------------------------|---|-----------|--------------|--|
| 1 | Alice Customer | Investment Land Plot | Hello I would like to buy this plot. Let's fin... | CONTACTED | Nov 09, 2025 | |
| 2 | Alice Customer | Luxury Villa in Pondicherry | I would like to buy this. | CLOSED | Nov 09, 2025 | |
| 3 | Alice Customer | Luxury Villa in Pondicherry | 500000 | CLOSED | Nov 09, 2025 | |
| 4 | Alice Customer | flat | want to buy | CLOSED | Nov 10, 2025 | |

RealEstate Pro

Quick Links

Contact Us

10.4 Agent Dashboard – Customer Inquiries

This screenshot displays the agent panel.

Agents can view customer inquiries related to assigned properties and respond to potential buyers.

RealEstate Pro

Home Properties Dashboard Logout Query Document Verify Document

Agent Dashboard

Manage your properties and customer inquiries

My Properties 11 [View Properties](#)

Inquiries 4 [View Inquiries](#)

Recent Properties

RealEstate Pro

Home Properties Dashboard Logout Query Document Verify Document

Customer Inquiries

Manage customer inquiries

[← Back to Dashboard](#)

| Customer | Property | Message | Status | Date | Actions |
|----------------|-----------------------------|---|-----------|--------|-------------------------|
| Alice Customer | Investment Land Plot | Hello I would like to buy this plot ... | CONTACTED | Nov 09 | Contact |
| Alice Customer | Luxury Villa in Pondicherry | I would like to buy this. | CLOSED | Nov 09 | Contact |
| Alice Customer | Luxury Villa in Pondicherry | 500000 | CLOSED | Nov 09 | Contact |
| Alice Customer | flat | want to buy | CLOSED | Nov 10 | Contact |

10.5 Customer Dashboard – Property Inquiry

This screenshot shows the customer dashboard where a logged-in user can view property details and submit inquiries about properties they are interested in.

This connects customers directly with agents through the platform

RealEstate Pro

Home Properties Dashboard Logout Query Document Verify Document

My Dashboard

Welcome back, Alice Customer!

My Inquiries 4

[View All Inquiries](#)

Browse Properties

Find your dream home

[View Properties](#)

Recent Inquiries

| Property | Message | Status | Date |
|-----------------------------|---|-----------|--------------|
| Investment Land Plot | Hello I would like to buy this plot. Let's fin... | CONTACTED | Nov 09, 2025 |
| Luxury Villa in Pondicherry | I would like to buy this. | CLOSED | Nov 09, 2025 |
| Luxury Villa in Pondicherry | 500000 | CLOSED | Nov 09, 2025 |
| flat | want to buv | CLOSED | Nov 10, 2025 |

RealEstate Pro

Home Properties Dashboard Logout Query Document Verify Document

[← Back to Properties](#)



Interested in this property?

Send us an inquiry and our agents will contact you soon

Your Message

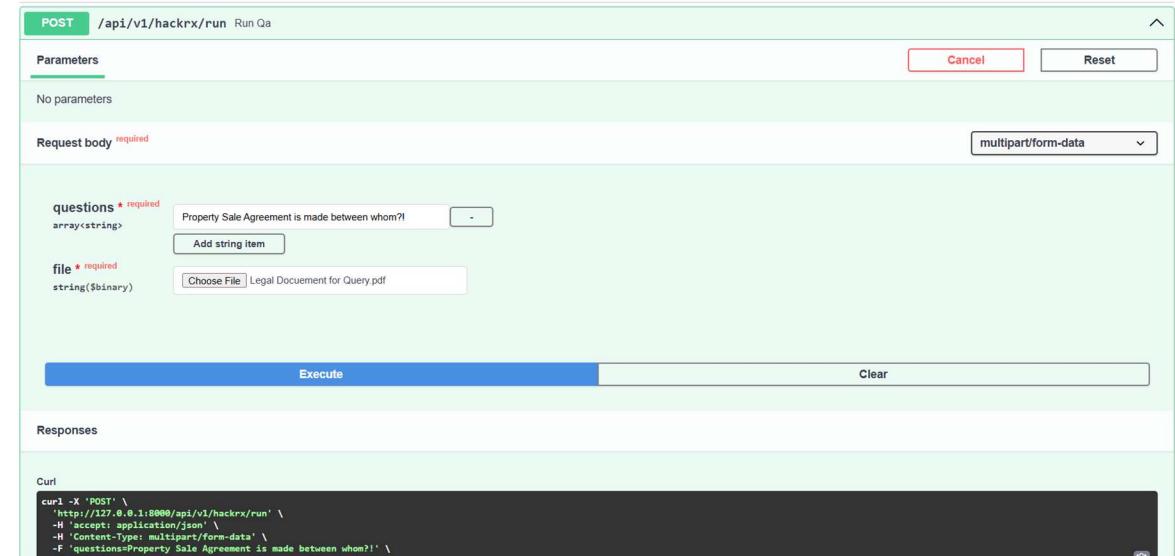
I want to buy this.

[Send Inquiry](#)

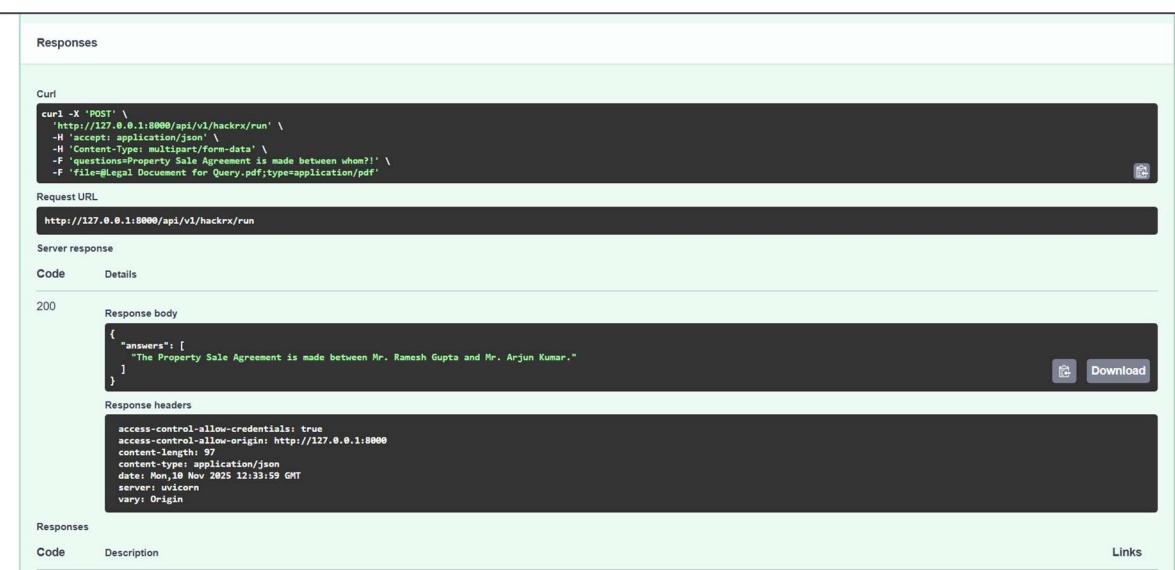
Need help?
+1 (555) 123-4567

10.6 Document Query (RAG System)

This screenshot shows the document querying interface powered by the RAG pipeline. Users upload legal or property documents and ask questions in natural language. The system retrieves relevant information from the documents and generates accurate answers using the Gemma 3N model.



The top section shows the API endpoint `/api/v1/hackrx/run` with a `POST` method. It has a green "Parameters" section indicating "No parameters" and a red "Request body required" section. The request body contains fields for `questions` (array of strings) and `file` (string binary). The `questions` field has a dropdown with the value "Property Sale Agreement is made between whom?". The `file` field has a button labeled "Choose File" with the value "Legal Document for Query.pdf". Below the request body are "Execute" and "Clear" buttons. The "Responses" section is collapsed.



The bottom section shows the "Responses" tab expanded. It includes a "Curl" command:

```
curl -X 'POST' \
'http://127.0.0.1:8000/api/v1/hackrx/run' \
-H 'accept: application/json' \
-H 'Content-Type: multipart/form-data' \
-F 'questions=Property Sale Agreement is made between whom?' \
-F 'file=@Legal Document for Query.pdf;type=application/pdf'
```

Below the curl command is the "Request URL": `http://127.0.0.1:8000/api/v1/hackrx/run`. The "Server response" tab is expanded, showing a code table for status 200. The "Details" column shows the response body as JSON:

```
{"answers": [ "The Property Sale Agreement is made between Mr. Ramesh Gupta and Mr. Arjun Kumar." ]}
```

The "Response headers" section shows the following headers:

```
access-control-allow-credentials: true
access-control-allow-origin: http://127.0.0.1:8000
content-length: 97
content-type: application/json
date: Mon, 18 Nov 2025 12:33:59 GMT
server: unicorn
vary: Origin
```

The "Responses" tab is collapsed at the bottom left, and the "Links" tab is collapsed at the bottom right.

10.7 Forgery Detection

This screenshot shows the forgery detection module. An uploaded document image is processed using Error Level Analysis (ELA) and a deep learning model to determine whether the document is authentic or manipulated. The system returns the forgery probability and authenticity score.

The screenshot shows a web-based application for forgery detection. At the top, there is a file input field labeled "Choose File" with "verify1.jpg" selected. Below the file input is a blue "Execute" button and a "Clear" button. To the right of the "Execute" button is a dropdown menu for "Response content type" set to "application/json". Under the "Responses" section, there is a "Curl" command:

```
curl -X POST "http://127.0.0.1:5000/detect-forgery" -H "accept: application/json" -H "Content-Type: multipart/form-data" -F "image=@verify1.jpg;type=image/jpeg"
```

Below the "Curl" command is the "Request URL": <http://127.0.0.1:5000/detect-forgery>. The "Server response" section shows a status code of 200. The "Details" tab is selected, and the "Response body" contains the following JSON output:

```
{
  "authentic_probability": 0.8842,
  "forgery_detected": false,
  "forgery_probability": 0.1158
}
```

At the bottom right of the "Response body" panel are "Copy" and "Download" buttons.

10.8 MySQL Database View

This screenshot displays the MySQL database tables storing: User information, Property data, Inquiry records. It confirms that all system data is securely stored and managed through a relational database.

The screenshot shows a terminal window within a development environment. The terminal is connected to a MySQL database named "realstate_db". The user has run several commands to view the database structure and data:

```
mysql> use realestate_db;
Database changed
mysql> show tables;
+-----+
| Tables_in_realestate_db |
+-----+
| inquiries               |
| properties              |
| users                   |
+-----+
3 rows in set (0.01 sec)

mysql> select * from properties;
+----+-----+-----+-----+-----+-----+-----+
| id | location | price | size | title           | type   | image_url
+----+-----+-----+-----+-----+-----+-----+
| 1  | Stunning 5-bedroom villa with panoramic city views, infinity pool, and modern amenities. | https://images.unsplash.com/photo-1613490493576-7fde63acd817w=800 | Beverly Hills, CA | 2500000 | 4500 | Villa
| 2  | Contemporary 2-bedroom apartment in the heart of downtown with gym and rooftop access. | https://images.unsplash.com/photo-1545324418-cc1a3fa10c007w=800 | New York, NY | 450000 | 1200 | Modern Downtown Apartment | Apartment
| 3  | Beautiful 4-bedroom house with large backyard, perfect for families. | https://images.unsplash.com/photo-1568605114967-8130f3a3699a2w=800 | Austin, TX | 675000 | 3200 | Spacious Family House | House
| 4  | Luxurious 3-bedroom condo with direct beach access and ocean views. | https://images.unsplash.com/photo-151291774080-9991fc4c750w=800 | Miami, FL | 890000 | 1800 | Beachfront Condo | Apartment
| 5  | Prime commercial property in business district, 10,000 sq ft with parking. | https://images.unsplash.com/photo-1486406146926-c627a92ad
```

11. Conclusion

This project demonstrates how **AI, Deep Learning, and RAG** can be integrated into a **real enterprise system**.

By combining:

- **CNN-based forgery detection**
- **LLM-powered document querying**
- **Spring Boot backend**

the system delivers a **secure, intelligent, and scalable** real estate platform.

This project reflects strong skills in:

- AI system design
- Backend engineering
- Machine learning integration