September Milestone

Team Members

- **Sameer Gupta - 2021093**
- **Lakshay Chauhan - 2021060**
- **Chaitanya Arora - 2021033**
- **Ankit Kumar - 2021015**

Project Overview

The Real Estate Aggregator System is designed to facilitate the secure exchange and verification of property-related documents. The primary objective is to enable secure transactions during property buying, selling, and renting.

Schema

Attributes

User

- id (Portal created: Auto-incremented value)
- Username (User input)
- Roll Number (User input)
- Email (User input)
- Password (User input, but stored as hashed value)
- First Name (User input)
- Last Name (User input)
- Document (User input)
- Document Hash (Portal created: By hashing the Document)
- User Hash (Portal created: By hashing the combination of Username, Roll Number, and Email)
- Verification Code (Portal created: Randomly generated and sent to user's email)

Location

- Location Id (Portal created: Auto-incremented value)
- Latitude (Portal generated: from user input)
- Longitude (Portal generated: from user input)

Property

- Property id (Portal created: Auto-incremented value)
- Title (User input)
- Description (User input)

- price (User input)
- Bedrooms (User input)
- Bathrooms (User input)
- Area (User input)
- Status (User input)
- Availability Date (User input)
- Location (Foreign key: Links to Location entity)

Contract

- Property (Foreign key: Links to Property entity)
- Seller (Foreign key: Links to User entity)
- Buyer (Foreign key: Links to User entity)
- Verified By Buyer (Portal set: set to True on buyer verification)
- Verified By Seller (Portal set: set to True on seller verification)
- Verified By Portal (Portal set: set to True when both buyer and seller verification is complete)
- Contract Hash (Portal created: By hashing the contract details)
- Contract Address (Portal created: Based on the platform used, could be blockchain address)

Relationships

User to Property

- A user can have multiple properties.
- A property belongs to one user.

User to Contract

- A user can be a buyer or seller in multiple contracts.
- A contract involves one buyer and one seller.

Property to Location

- A property has one location.
- A location can be linked to multiple properties.

Property to Contract

- A property can be linked to one contract.
- A contract links to one property.

Fields Created by Portal

User

- Id: Auto-incremented

- Document Hash: Created by hashing the user's Document.
- User Hash: Created by hashing the combination of Username, Roll Number, and Email.
- Verification Code: Randomly generated and sent to user's email.

Location

- Location Id: Auto-incremented.
- Latitude: Generated from user input
- Longitude: Generated from user input

Property

- Property Id: Auto-incremented.

Contract

- Verified By Buyer: Set to True when the buyer verifies the contract.
- Verified By Seller: Set to True when the seller verifies the contract.
- Verified By Portal: Set to True when both the buyer and seller have verified.
- Contract Hash: Created by hashing the contract details.

The Portal (Application)

Property Listing

- Allows sellers to list properties.
- Portal validates the input and then saves it in the database.
- Generates a unique `propertyId` for each new property.
- Uses the `generatePropertyHashIdentifier` function to create a unique SHA-256 hash identifier for the property.
- Geocodes the provided location via the `geocode_location` function to obtain the latitude and longitude.

Contract Creation

- When a buyer expresses interest in a property and initiates a transaction, the portal facilitates the creation of a new contract.
- The portal captures the `propertyId`, `buyer`, and `seller` information.
- Sets the `verifiedByBuyer`, `verifiedBySeller`, and `verifiedByPortal` attributes to False by default.
- Generates a unique `contractHash` for security and data integrity purposes.
- If implementing blockchain, generates or captures a `contractAddress` for the smart contract.

Contract Verification

- Allows buyers and sellers to verify the terms and conditions of the contract.
- Once verified by both parties, sets the `verifiedByPortal` attribute to True.
- Might send notifications or reminders to both parties to complete their verifications.

User Registration and Authentication

- Registers users by capturing details and hashing the password before storing.
- Sends a verification code to the user's email.
- Handles user login by verifying the hashed password.

Bidding

- Facilitates placing of bids.
- Checks if the user is authenticated and the bid is valid.
- Updates the property's current bid and bidder information.
- Implements Django's CSRF protection.
- Uses @login_required decorator for specific functionalities.
- Hashes documents and other sensitive data.

The Buyer

Property Exploration

- Uses the `propertyList` function to explore available properties.
- Can filter properties based on criteria such as type, budget, and location.

Initiate Transaction

- Selects a property and initiates the contract creation process.
- Provides necessary details for the contract.

Contract Verification

- Interacts with the portal to review the contract's terms and conditions.
- Confirms acceptance of the contract's terms.

Bidding

- Places bids on properties via the `addBid` function.
- Checks if they have the necessary documents verified.

Completion of Transaction

- After verification by both parties, proceeds to make payments or any other necessary processes to finalize the property purchase/rental.

The Seller

Property Listing

- Lists the property on the portal using the `addProperty` function.
- Provides required details such as title, description, price, location, etc.

Review and Acceptance

- Reviews the initiated contracts by potential buyers.
- Accepts the terms and conditions of the contract.

Contract Verification

- Confirms the acceptance of the contract's terms through the portal.
- Engages in communication with the buyer if needed.

Finalizing the Transaction

- Once the contract is verified, coordinates property handover or other related activities.

Summary

For the system to function seamlessly

- The Portal provides a platform to facilitate property listings, user registrations, contract creations, and verifications. It also implements vital security and verification measures.
- The Buyer explores properties, initiates transactions, and verifies contracts.
- The Seller lists properties, reviews initiated contracts, and verifies them.
- Both Buyer and Seller need to collaborate through the portal to ensure smooth transactions and verification processes.

Security Protocol

- HTTPS (SSL/TLS): Ensures encrypted communication between the user's browser and the server. This prevents eavesdropping, man-in-the-middle attacks, and data tampering

- Data Encryption: By encrypting sensitive data, you protect it from being accessed or understood even if there's a data breach.
- Password Hashing: Passwords should never be stored in plain text. Hashing turns passwords into a fixed-size string of bytes, often in hexadecimal. If an attacker gains access to the database, they won't easily decipher actual passwords.
- Multi-Factor Authentication (MFA): Adds an additional layer of security by requiring users to provide two or more verification factors to gain access.
- CSRF (Cross-Site Request Forgery) Protection: Prevents attackers from tricking a victim into performing actions they didn't intend to. By using a token, the server ensures that the request made is legitimate and originating from a trusted source.
- XSS (Cross-Site Scripting) Protection: Ensures that attackers cannot inject malicious scripts into web pages. These scripts could be used to steal information or perform actions on behalf of the user without their knowledge.
- Secure File Uploads: Ensures that the file upload feature cannot be exploited. This can be achieved by checking file types, scanning for malware, and storing files outside the webroot.