**ANUDIP FOUNDATION**

A Project Report on

**VIRTUAL RENTAL AGREEMENT GENERATOR**

By

Batch: ANP-D0453

Student ID: AF0477187

Name: Laxmi Kolpyak

**Under the Guidance of**

Mrs. Rajshri Chandrabhan Thete

VIRTUAL RENTAL AGREEMENT GENERATOR

Introducing our Java-based **VIRTUAL RENTAL AGREEMENT GENERATOR**:

In today's digital era, automation and paperless transactions have become essential for improving efficiency, transparency, and convenience. The **Virtual Rental Agreement Generator** project is developed with the objective of simplifying the traditional process of creating rental agreements, which often involves manual paperwork, physical presence, and legal complexities. This system provides a smart and user-friendly solution that allows landlords and tenants to generate legally formatted rental agreements virtually, with minimal effort.

The project is implemented using **Java, Hibernate, and a relational database**, offering complete CRUD (Create, Read, Update, Delete) operations for managing users, properties, agreements, and payments. It also includes features such as **PDF generation for rental agreements** and a **placeholder for email integration**, enhancing usability and digital access. This system reduces human error, saves time, and promotes a structured way of maintaining rental records.

By automating the agreement generation process, this project bridges the gap between traditional documentation and modern technology, making rental transactions easier, faster, and more secure.

**Entities:**

* User (Landlord or Tenant)
* Property
* Agreement
* Payment

**ATTRIBUTES OF ENTITIES:**

1. **User (Landlord & Tenant)**

**Represents both landlords and tenants in the system.**

* **Attributes:**
  + user\_id – Unique ID for each user *(Primary Key)*
  + name – Full name of the user
  + email – Email ID
  + password – Encrypted password
  + phone\_number – Contact number
  + user\_type – Either 'Landlord' or 'Tenant'
  + created\_at – Timestamp of user creation
  + signature\_image – Digital signature image

1. **Property**

**Represents the properties available for rent.**

* **Attributes:**
  + property\_id – Unique ID for each property *(Primary Key)*
  + owner\_id – References user\_id of the landlord *(Foreign Key)*
  + address – Full address of the property
  + rent\_amount – Monthly rent in ₹
  + security\_deposit – Deposit amount in ₹
  + property\_type – Residential, Hostel, Commercial, etc.
  + availability\_status – Available / Unavailable
  + created\_at – Time of property listing

1. **Agreement**

**Represents the rental agreement between a landlord and tenant.**

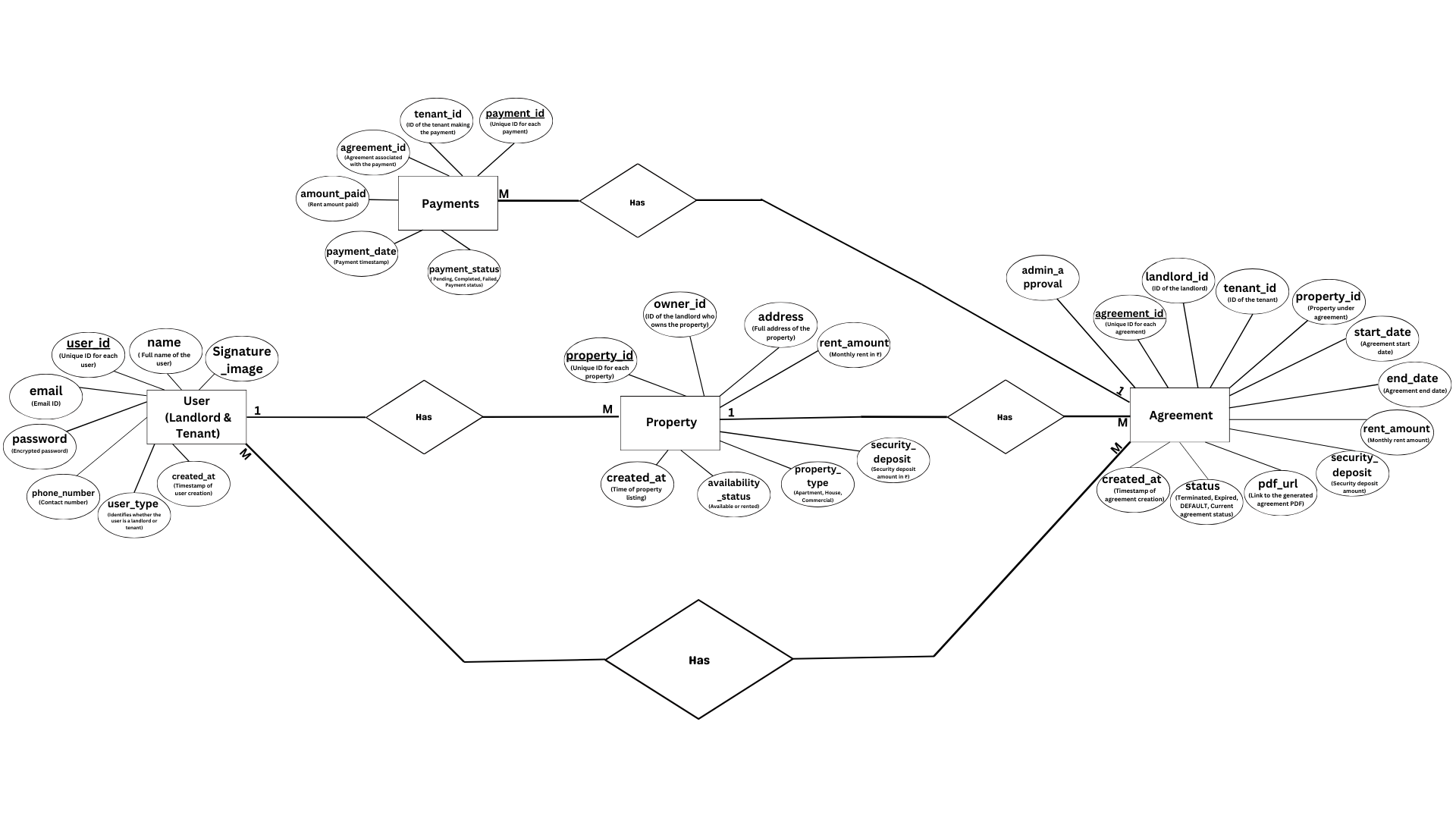
* **Attributes:**
  + agreement\_id – Unique ID *(Primary Key)*
  + landlord\_id – ID of the landlord *(Foreign Key to User)*
  + tenant\_id – ID of the tenant *(Foreign Key to User)*
  + property\_id – ID of the associated property *(Foreign Key)*
  + start\_date – Start date of agreement
  + end\_date – End date of agreement
  + rent\_amount – Monthly rent
  + security\_deposit – Deposit amount
  + status – Terminated, Expired, Current, DEFAULT
  + pdf\_url – Path to the generated PDF
  + admin\_approval – Admin approval status
  + created\_at – Timestamp of agreement creation

### ****Payments****

### Tracks rent payments made by tenants.

* **Attributes:**
  + payment\_id – Unique payment ID (Primary Key)
  + agreement\_id – Associated agreement (Foreign Key)
  + tenant\_id – ID of the paying tenant (Foreign Key to User)
  + amount\_paid – Amount paid
  + payment\_date – Timestamp of payment
  + payment\_status – Paid, Pending, Failed, Completed

**ENTITY RELATIONSHIP DIAGRAM - VIRTUAL RENTAL AGREEMENT GENERATOR**



**CONCLUSION:**

The *Virtual Rental Agreement Generator System* successfully addresses the need for a fast, secure, and digital platform to create and manage rental agreements. By integrating user management, property listings, automated agreement generation with PDF support, and payment tracking, the system simplifies the entire rental process for landlords and tenants alike. It eliminates traditional paperwork, reduces manual errors, and enhances accessibility by allowing agreements to be created and stored online. This project not only modernizes property rental management but also ensures transparency, legal clarity, and convenience for all parties involved. Future enhancements such as e-signatures, automatic rent reminders, and integration with government verification portals can further extend its practical utility and real-world adoption.

**DATABASE CREATION QUERY:**

**Query OK, 1 row affected (0.01 sec)**

**mysql> CREATE TABLE Admin (**

**-> admin\_id INT PRIMARY KEY,**

**-> admin\_name VARCHAR(255),**

**-> admin\_Phone VARCHAR(20),**

**-> admin\_Email VARCHAR(255)**

**-> );**

**mysql> CREATE TABLE Staff (**

**-> staff\_id INT PRIMARY KEY,**

**-> staff\_name VARCHAR(255),**

**-> staff\_rating INT,**

**-> staff\_Email VARCHAR(255)**

**-> );**

**Query OK, 0 rows affected (0.02 sec)**

**mysql> CREATE TABLE Items (**

**-> item\_id INT PRIMARY KEY,**

**-> category VARCHAR(255),**

**-> price DECIMAL(10, 2),**

**-> Name VARCHAR(255),**

**-> Order\_count INT**

**-> );**

**Query OK, 0 rows affected (0.01 sec)**

**mysql>**

**mysql> CREATE TABLE Customers (**

**-> customer\_id INT PRIMARY KEY,**

**-> customer\_name VARCHAR(255),**

**-> customer\_Phone VARCHAR(20),**

**-> customer\_email VARCHAR(255),**

**-> customer\_password VARCHAR(255)**

**-> );**

**Query OK, 0 rows affected (0.01 sec)**

**mysql> CREATE TABLE Parcel (**

**-> Parcel\_id INT PRIMARY KEY,**

**-> item\_id INT,**

**-> customer\_id INT,**

**-> amount DECIMAL(10, 2),**

**-> order\_id INT,**

**-> Time TIMESTAMP,**

**-> FOREIGN KEY (item\_id) REFERENCES Items(item\_id),**

**-> FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)**

**-> );**

**Query OK, 0 rows affected (0.02 sec)**

**mysql> CREATE TABLE Orders(**

**-> order\_id INT PRIMARY KEY,**

**-> time TIMESTAMP,**

**-> amount DECIMAL(10, 2),**

**-> customer\_id INT,**

**-> FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)**

**-> );**

**Query OK, 0 rows affected (0.04 sec)**

**mysql> CREATE TABLE Tables (**

**-> Table\_number INT PRIMARY KEY,**

**-> Total\_people INT,**

**-> Time TIMESTAMP,**

**-> customer\_id INT,**

**-> item\_id INT,**

**-> order\_id INT,**

**-> FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id),**

**-> FOREIGN KEY (item\_id) REFERENCES Items(item\_id),**

**-> FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)**

**-> );**

**Query OK, 0 rows affected (0.04 sec)**

**mysql> CREATE TABLE feedback(feedback\_id INT NOT NULL PRIMARY KEY,**

**-> Staff\_rating INT,**

**-> customer\_id INT,**

**-> staff\_id INT,**

**-> item\_rating VARCHAR(255),**

**-> FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id),**

**-> FOREIGN KEY (staff\_id) REFERENCES staff(staff\_id)**

**-> );**

**Query OK, 0 rows affected (0.05 sec)**

**mysql> CREATE TABLE payment (**

**-> payment\_id INT PRIMARY KEY AUTO\_INCREMENT,**

**-> payment\_type VARCHAR(50) NOT NULL,**

**-> amount DECIMAL(10, 2) NOT NULL,**

**-> customer\_id INT,**

**-> FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)**

**-> );**

**Query OK, 0 rows affected (0.05 sec)**

**mysql> show tables;**

**+----------------------------------------+**

**| Tables\_in\_restaurant\_management\_system |**

**+----------------------------------------+**

**| admin |**

**| customers |**

**| feedback |**

**| items |**

**| orders |**

**| parcel |**

**| payment |**

**| staff |**

**| tables |**

**+----------------------------------------+**

**9 rows in set (0.00 sec)**