CAR RENTAL WEBSITE

Project Synopsis

1. Introduction

The Car Rental Website is a web-based application developed using the Django framework. The system connects car owners and customers through a centralized digital platform. Customers can search, view, and book available cars online, while owners can list their vehicles for rental after admin approval. The platform ensures secure authentication, structured booking management, and role-based access control.

2. Objectives

- To develop a secure and scalable car rental platform.
- To enable car owners to list and manage their vehicles.
- To allow customers to search and book cars online.
- To implement role-based authentication and authorization.
- To provide administrative control for approvals and monitoring.
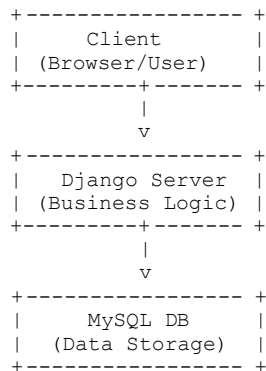
3. Technology Stack

Backend: Python, Django Framework Frontend: HTML, Tailwind CSS, JavaScript Database: MySQL Version Control: Git & GitHub

4. System Modules

- User Authentication Module (Registration, Login, OTP Verification)
- Become Owner Request Module
- Car Management Module (Add, Update, Delete Cars)
- Car Search & Details Module
- Booking & Payment Module
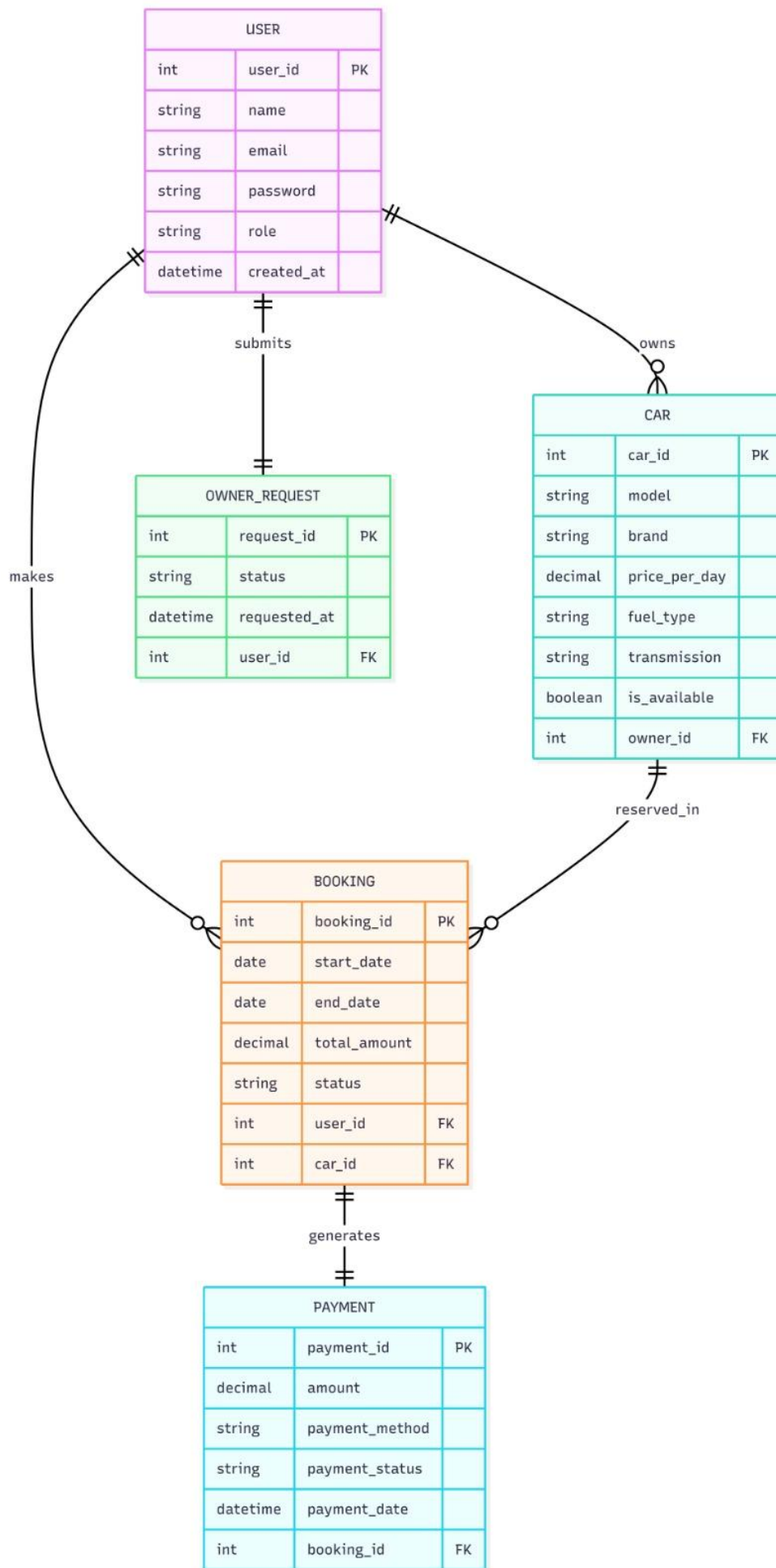- Admin Management Module

5. System Architecture Diagram

```
+---------------- +
|     Client      |
| (Browser/User)  |
+---------+------- +
          |
          v
+---------------- +
|  Django Server  |
| (Business Logic)|
+---------+------- +
          |
          v
+---------------- +
|    MySQL DB     |
|  (Data Storage) |
+---------------- +
```

Explanation: The client interacts with the system using a web browser. Requests are sent to the Django server, which processes business logic, handles authentication, booking validation, and admin approval workflows. The server communicates with the MySQL database for storing and retrieving data. The processed response is then returned to the client.

6. ER Diagram Explanation

Main Entities: 1. User: Stores user information including roles (Customer/Owner/Admin). 2. Car: Contains car details such as model, price per day, owner reference. 3. Booking: Stores booking details including rental dates and payment status. 4. OwnerRequest: Stores requests submitted by users to become car owners. 5. Payment: Stores payment transaction details. Relationships: - One User can make multiple Bookings. - One Owner (User) can list multiple Cars. - One Car can have multiple Bookings (on different dates). - Each Booking is associated with one Payment record.

USER

int	user_id	PK
string	name	
string	email	
string	password	
string	role	
datetime	created_at	

OWNER_REQUEST

int	request_id	PK
string	status	
datetime	requested_at	
int	user_id	FK

CAR

int	car_id	PK
string	model	
string	brand	
decimal	price_per_day	
string	fuel_type	
string	transmission	
boolean	is_available	
int	owner_id	FK

BOOKING

int	booking_id	PK
date	start_date	
date	end_date	
decimal	total_amount	
string	status	
int	user_id	FK
int	car_id	FK

PAYMENT

int	payment_id	PK
decimal	amount	
string	payment_method	
string	payment_status	
datetime	payment_date	
int	booking_id	FK

Relationships:
- USER submits OWNER_REQUEST
- USER owns CAR
- USER makes BOOKING
- CAR reserved_in BOOKING
- BOOKING generates PAYMENT

7. Advantages

- Reduces manual paperwork and improves efficiency.
- Provides real-time car availability.
- Secure and role-based system.
- Centralized monitoring and control by admin.

8. Conclusion

The Car Rental Website demonstrates a complete full-stack web application built using Django and MySQL. The system successfully integrates user authentication, owner management, car listing booking processes, and administrative control into a unified platform. The project showcases practical implementation of modern web development concepts and can be extended further for commercial deployment.