# AK Car Rental Services

# Project Report

## Group Members:

1. Kantesh Kumar (21K-3426)
2. Ahsan Ashraf (21K-3186)

Project Introduction:

The Car Rental Website project aims to create a robust online platform facilitating the rental of cars in Karachi, Pakistan. This comprehensive solution caters to both individual car owners and rental companies, providing a seamless interface for listing and renting vehicles. Developed using React for the front-end, Node.js for the backend, and MySQL for the database, the system promises a user-friendly experience for customers and efficient management tools for car owners.

Project Features (main):

1. User Profiles:
   * Description: This feature allows both individual users and companies to create profiles on the car rental platform. Users can register by providing essential details such as their name, contact information, address, and any additional information deemed necessary. Companies can also register by furnishing relevant business details.
   * Functionality:
     + Users and companies can sign up on the platform with a unique username and password.
     + Users provide personal details, including name, contact information, and address during registration.
     + Companies furnish business-related information, such as company name, address, and contact details.
   * Benefits:
     + Personalized user experience with individualized profiles.
     + Enhanced trust and transparency between users and car owners.
     + Enables targeted communication and notifications based on user preferences.
2. Car Listings:
   * Description: Car owners can showcase their vehicles on the platform, offering potential renters’ comprehensive information about each car. This includes details like the car's make, model, year, color, transmission type, and any additional features or specifications.
   * Functionality:
     + Car owners log in to their accounts to list their cars for rent.
     + Owners provide detailed information about each listed car, including its specifications, features, and rental terms.
     + High-quality images of the cars are uploaded to give renters a visual representation.
     + Owners can set rental rates and specify availability dates for each listed vehicle.
   * Benefits:
     + Enables a wide variety of cars for users to choose from.
     + Provides transparency about the condition, features, and pricing of each listed car.
     + Facilitates effective communication between car owners and potential renters.
3. Booking System:
   * Description: This feature allows users to browse through the available car listings and make reservations for specific dates. It involves a secure and user-friendly booking process that ensures a seamless experience for both renters and car owners.
   * Functionality:
     + Users search for available cars based on their preferences, such as location, dates, and car type.
     + Detailed information about each available car is displayed to help users make informed decisions.
     + Users select the desired car, specify rental dates, and initiate the booking process.
     + The platform confirms the reservation, and notifications are sent to both the renter and the car owner.
   * Benefits:
     + Streamlines the process of finding and reserving a suitable car.
     + Provides a secure payment gateway for online transactions.
     + Reduces the likelihood of booking conflicts through real-time availability updates.

Technology Stack:

* Frontend: React, HTML, CSS
* Backend: Node.js, Express
* Database: MySQL

Entity Relationship Diagram:

A black and white diagram

Description automatically generated

Normalized Schema:

Car:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Reg\_no | C\_name | Model | Color | Owner\_Id | Available | Description | Price\_Per\_Day | Transmission |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Mileage | Int\_image | Ext\_image | Passengers | Luggage | Doors | Year | AC |

Owner:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| O\_id | name | ph\_Number | Address | Gender | Password |

Customer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| C\_id | Name | Address | ph\_number | Age | Zip\_code | Gender |

rental\_reg

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Reg\_id | Customer\_cus\_id | Car\_Reg\_no | Total\_Price | Commission | Pick\_up\_TD | Drop\_off\_TD | Days |

transactions

|  |  |
| --- | --- |
| T\_id | Rental\_Reg\_Reg\_id |

car\_registration

|  |  |
| --- | --- |
| CR\_id | Car\_Reg\_no |

Functional Dependencies:

1. **Car:**

Reg\_no → {C\_name, Model, Color, Owner\_Id, Available, Description, Price\_Per\_Day, Transmission, Mileage, Int\_image, Ext\_image, Passengers, Luggage, Doors, Year, AC}

1. **Owner:**

O\_id → {name, ph\_Number , Address, Gender, Password}

1. **Customer:**

C\_id → {Name, Address, ph\_number, Age, Zip\_code, Gender}

1. **Rental\_Reg:**

Reg\_id → {Customer\_cus\_id, Car\_Reg\_no, Total\_Price, Commission, Pick\_up\_TD, Drop\_off\_TD, Days}

1. **Transactions:**

T\_id → {Rental\_Reg\_Reg\_id}

1. **Car\_Registration:**

CR\_ID → {Car\_Reg\_no}

Normalization:

**1NF (First Normal Form):**

All tables are already in 1NF, as each column contains atomic values, and there are no repeating groups.

**2NF (Second Normal Form):**

No partial dependencies exist in the tables. All non-prime attributes are fully functionally dependent on the primary keys.

**3NF (Third Normal Form):**

All transitive dependencies are removed. No non-prime attribute depends on another non-prime attribute.

Class Diagram:

