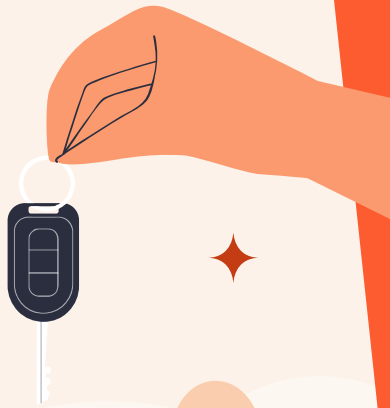
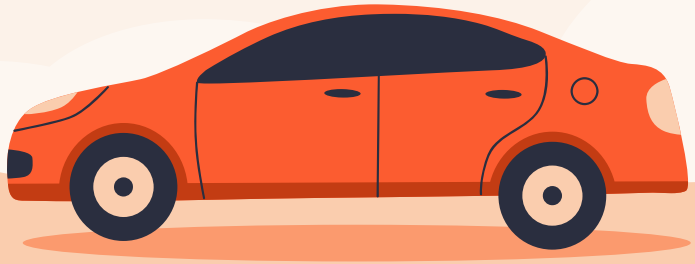
# World wheels

Drive your dreams-luxury cars at your fingertips

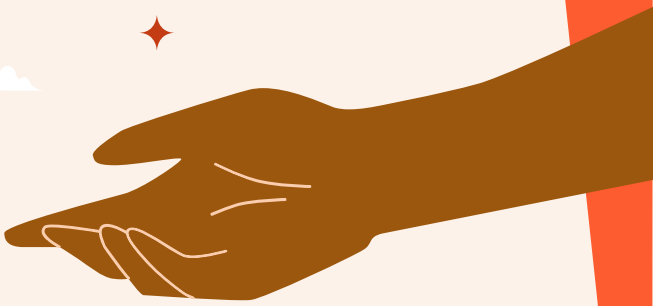




# Team members

| ID | Name |
| --- | --- |
| 2305049 | Youssef Hassan Youssef |
| 2305048 | Youssef Ahmed Abd Elmola |
| 2305073 | Mina Ayman Mossad |

01



# DATA DEFINITION LANGUAGE

# DDL

## users

```
CREATE TABLE users (
        user_id INT PRIMARY KEY,
    username VARCHAR(200) NOT NULL,
    email VARCHAR(200) UNIQUE,
    password VARCHAR(200) NOT NULL,
    phone_number VARCHAR(200) NOT NULL UNIQUE ,
    address text  NOT NULL,
    role ENUM("User","Admin") DEFAULT 'User' NOT NULL,
    sid VARCHAR(200) NOT NULL
);
```

## cars

```
CREATE TABLE cars(
    car_id INT PRIMARY KEY,
    model VARCHAR (250) NOT NULL,
    year INT NOT NULL,
    plate_id VARCHAR(100) UNIQUE,
    status ENUM("Available", "Not Available"),
    office_id INT,
    image_url TEXT NOT NULL,
    car_price INT NOT NULL,
    category VARCHAR(100) NOT NULL,
    seller_id INT,
    FOREIGN KEY (seller_id) references users(user_id),
    FOREIGN KEY (office_id) references office(office_id)
);
```

## office

```
CREATE TABLE office (
office_id INT auto_increment PRIMARY KEY,
    office_contry VARCHAR (100) NOT NULL,
    office_location text NOT NULL
);
```

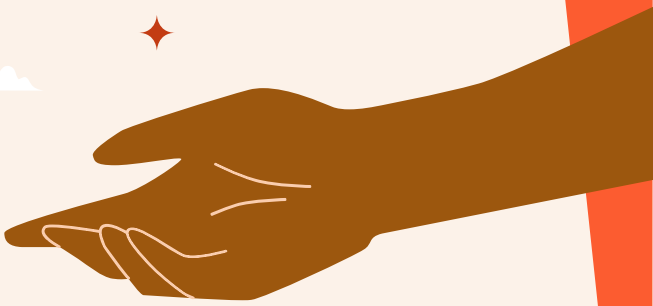
## Reservation

```
CREATE TABLE Reservation (
    reservation_id INT auto_increment,
    reservation_date DATE NOT NULL,
    pickup_date DATE NOT NULL,
    return_date DATE NOT NULL,
    payment_status ENUM("Paid", "Not paid","Pay on site") DEFAULT "Not paid",
    user_id INT,
    car_id INT,
    PRIMARY KEY (reservation_id,user_id,car_id),
    FOREIGN KEY (user_id) references users(user_id),
    FOREIGN KEY (car_id) references Car(car_id)
);
```

## Payment

```
CREATE TABLE Payment (
    payment_id INT PRIMARY KEY,
    payment_date DATE,
    payment_amount INT,
    method ENUM ("Cash", "VISA") DEFAULT "Cash" NOT NULL,
    reservation_id INT,
    FOREIGN KEY (reservation_id) references reservation(reservation_id)
);
```

02



# IMPLEMENTED FEATURES

## USER registration

Users can create accounts securely, as their passwords are encrypted to ensure data protection. Renting a car requires users to log in, adding an extra layer of security. Each user is assigned a single role, which is 'User,' as the platform designates only one administrator to maintain system safety and prevent unauthorized access.

## Reservation table acts as junction table

Since the reservation table connects users to payments, it ensures that payments cannot be made without a corresponding reservation, preventing errors such as paying without saving the reservation.

## Full control over cars management

We have full control over a car, as we can add, edit, delete, and rent cars. The database structure of the car allows for seamless changes to the table, ensuring no errors occur like referential integrity errors when modifying car details.

## Email Confirmation After Rental for Verification

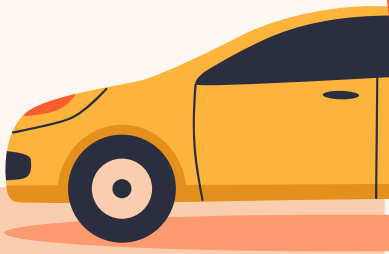
To confirm the user's car rental, an email is automatically sent upon completion of the purchase, leveraging the SMTP protocol for verification.

## Stripe API for secure payment

To enhance the safety and user experience, the Stripe API is integrated for secure payment processing. Stripe provides a reliable, PCI-compliant payment gateway, ensuring that user payment information is securely handled. By using Stripe's easy-to-integrate form, users can make payments through a smooth and intuitive interface, offering a seamless checkout experience. This integration also supports multiple payment methods, including credit/debit cards, ensuring flexibility for users while maintaining the highest standards of security.

## Rent car function efficiency

This function effectively uses the database to retrieve car, seller, and office details, ensuring the correct car is selected for rental. It fetches data from the Car, User, and Office tables to display relevant information, while validating dates to prevent errors. The rental cost is dynamically calculated using data from the Car and Office tables. Additionally, session management ensures that reservation details are stored temporarily , providing a seamless and consistent user experience throughout the rental and payment process. These database-driven features ensure data integrity and smooth functionality.



## Car Model Search Functionality

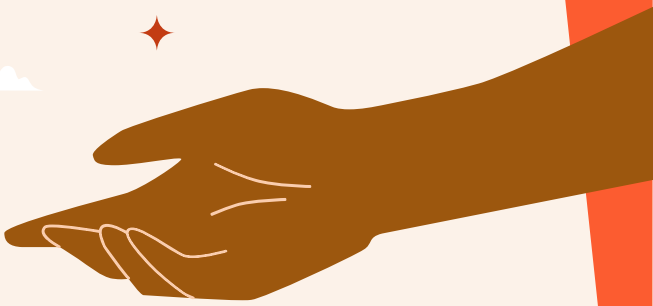
This function allows users to search for car models using the LIKE operator with the % wildcard for partial matches. The query is case-insensitive, enabling flexible searches in the Car table. The results are then displayed on the search.html page.

## Category-Based Car Filtering

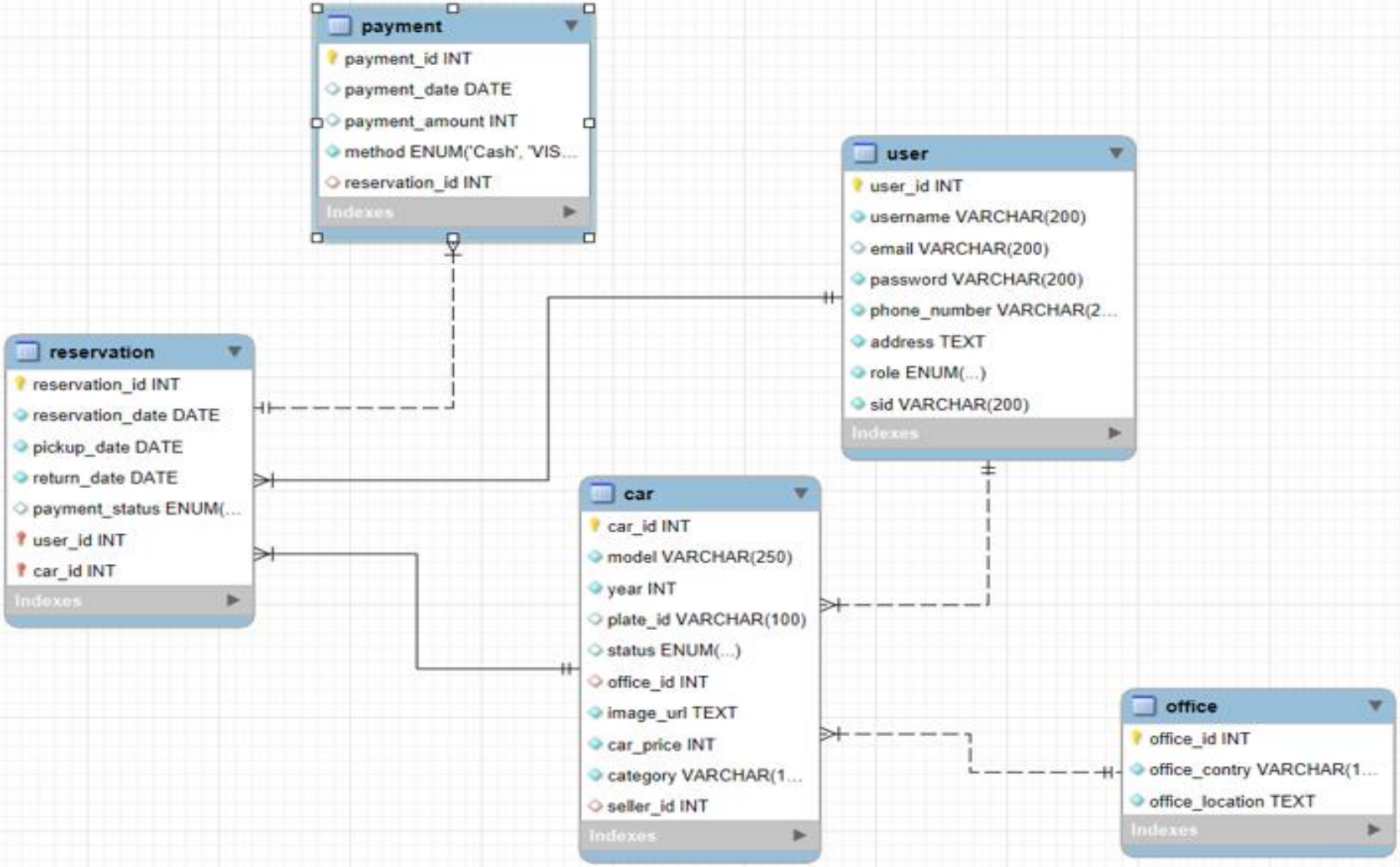
This function filters and displays cars based on a specific category. When a user navigates to a URL with a category (e.g., /category/suv), the function retrieves all cars from the Car table that match the given category using **(WHERE cars.category = category;)**. The filtered cars are then passed to the category_items.html template for display. This functionality allows users to view cars that belong to a particular category, providing a more targeted browsing experience.

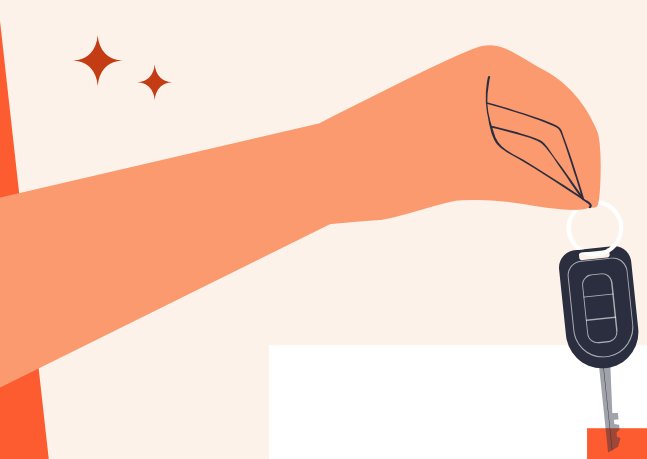
# 03

# DIAGRAMS




car_id
model
year
plate_id
status
image_url
car_price
category
seller_id
has
office
Office_id
office_contry
office_location
cars
M
1
M
1
has
Reservation
reservation_id
reservation_date
pickup_date
return_date
user_id
car_id
M
1
Rent
perform
has
user_id
email
username
phone_number
address
password
role
sid
user
1
1
1
1
Payment
payment_id
payment_date
payment_amount
method
reservation_id


payment
payment_id INT
payment_date DATE
payment_amount INT
method ENUM('Cash', 'VIS...
reservation_id INT
Indexes
user
user_id INT
username VARCHAR(200)
email VARCHAR(200)
password VARCHAR(200)
phone_number VARCHAR(2...
address TEXT
role ENUM(...)
sid VARCHAR(200)
Indexes
reservation
reservation_id INT
reservation_date DATE
pickup_date DATE
return_date DATE
payment_status ENUM(...
user_id INT
car_id INT
Indexes
car
car_id INT
model VARCHAR(250)
year INT
plate_id VARCHAR(100)
status ENUM(...)
office_id INT
image_url TEXT
car_price INT
category VARCHAR(1...
seller_id INT
Indexes
office
office_id INT
office_contry VARCHAR(1...
office_location TEXT
Indexes



# Thanks

WORLD WHEELS TEAM