EX NO: 11	MINI PROJECT:Vehicle Rental System
DATE :	

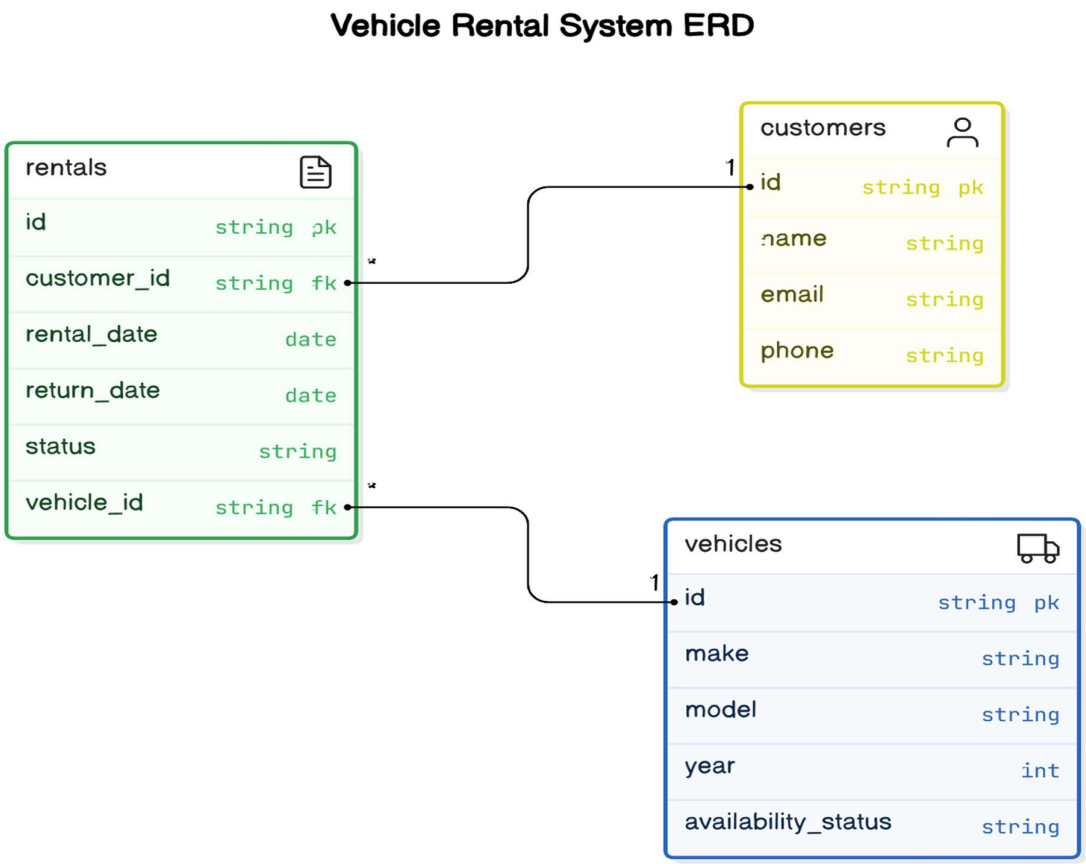
AIM:

The aim of the Vehicle Rental System is to streamline vehicle rentals and returns through a web-based application. It automates processes, ensures real-time updates, and maintains accurate records, enhancing efficiency and customer experience.

ABSTRACT:

The Vehicle Rental System is a web-based application developed using Python Flask and MySQL to simplify vehicle rental operations. It allows customers to view, rent, and return vehicles while administrators manage vehicle availability and rental records. The system ensures real-time updates, error handling, and a user-friendly interface. It aims to automate manual processes, improve efficiency, and enhance customer satisfaction with scalable features for future integration.

ER-DIAGRAM



CODING:

App.py

```
from flask import Flask, render_template, request, redirect, url_for, flash
import mysql.connector
from datetime import datetime

app = Flask(__name__)
app.secret_key = 'your_secret_key'  # Required for flash messages

# Database connection
def connect_to_db():
    return mysql.connector.connect(
        host="localhost",
        user="root",  # Replace with your MySQL username
        password="guna19082006@S",  # Replace with your MySQL password
        database="VehicleRentalDB"
    )

# Home page
@app.route('/')
def index():
    return render_template('index.html')

# View available vehicles
@app.route('/view')
def view_vehicles():
    db = connect_to_db()
    cursor = db.cursor()
    cursor.execute("SELECT * FROM Vehicles WHERE is_available = TRUE")
    vehicles = cursor.fetchall()
    db.close()
    return render_template('view.html', vehicles=vehicles)

# Rent a vehicle
@app.route('/rent', methods=['GET', 'POST'])
def rent_vehicle():
    if request.method == 'POST':
        vehicle_id = request.form['vehicle_id']
        customer_name = request.form['customer_name']
        rental_date = datetime.now().strftime('%Y-%m-%d')

        db = connect_to_db()
        cursor = db.cursor()
        cursor.execute("SELECT is_available FROM Vehicles WHERE vehicle_id = %s",
(vehicle_id,))
        result = cursor.fetchone()
        if result and result[0]:
```

```
        cursor.execute("INSERT INTO Rentals (vehicle_id, customer_name,
rental_date) VALUES (%s, %s, %s)",
                       (vehicle_id, customer_name, rental_date))
        cursor.execute("UPDATE Vehicles SET is_available = FALSE WHERE
vehicle_id = %s", (vehicle_id,))
        db.commit()
        flash("Vehicle rented successfully!", "success")
    else:
        flash("This vehicle is not available or doesn't exist.", "error")
    db.close()
    return redirect(url_for('rent_vehicle'))
  return render_template('rent.html')

# Return a vehicle
@app.route('/return', methods=['GET', 'POST'])
def return_vehicle():
  if request.method == 'POST':
    rental_id = request.form['rental_id']
    return_date = datetime.now().strftime('%Y-%m-%d')

    db = connect_to_db()
    cursor = db.cursor()
    cursor.execute("SELECT vehicle_id FROM Rentals WHERE rental_id = %s",
(rental_id,))
    result = cursor.fetchone()
    if result:
        vehicle_id = result[0]
        cursor.execute("UPDATE Rentals SET return_date = %s WHERE rental_id =
%s", (return_date, rental_id))
        cursor.execute("UPDATE Vehicles SET is_available = TRUE WHERE
vehicle_id = %s", (vehicle_id,))
        db.commit()
        flash("Vehicle returned successfully!", "success")
    else:
        flash("Invalid Rental ID.", "error")
    db.close()
    return redirect(url_for('return_vehicle'))
  return render_template('return.html')

# Add a new vehicle
@app.route('/add', methods=['GET', 'POST'])
def add_vehicle():
  if request.method == 'POST':
    vehicle_name = request.form['vehicle_name']
    vehicle_type = request.form['vehicle_type']
    rental_price = request.form['rental_price']

    db = connect_to_db()
    cursor = db.cursor()
    cursor.execute("INSERT INTO Vehicles (vehicle_name, vehicle_type,
```

```
rental_price_per_day) VALUES (%s, %s, %s)",
                (vehicle_name, vehicle_type, rental_price))
        db.commit()
        flash("Vehicle added successfully!", "success")
        db.close()
        return redirect(url_for('add_vehicle'))
    return render_template('add.html')

if __name__ == '__main__':
    app.run(debug=True)
```

Index.html:

```
<!DOCTYPE html>
<html>
<head>
    <title>Vehicle Rental System</title>
    <link rel="stylesheet" href="global.css">
</head>
<body>
    <h1>Vehicle Rental System</h1>
    <a href="/view">View Available Vehicles</a><br>
    <a href="/rent">Rent a Vehicle</a><br>
    <a href="/return">Return a Vehicle</a><br>
    <a href="/add">Add a New Vehicle</a>
</body>
</html>
```

Rent.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Rent a Vehicle - Vehicle Rental System</title>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap
" rel="stylesheet">
    <link rel="stylesheet" href="global.css">
</head>
<body>
    <div class="container">
        <h1>Rent a Vehicle</h1>
        <form method="post">
            <div>
                <label for="vehicle_id">Vehicle ID:</label>
                <input type="number" id="vehicle_id" name="vehicle_id" required>
            </div>
```

```
        <div>
            <label for="customer_name">Your Name:</label>
            <input type="text" id="customer_name" name="customer_name" required>
        </div>

        <button type="submit">Rent</button>
    </form>
    {% with messages = get_flashed_messages(with_categories=true) %}
        {% for category, message in messages %}
            <p class="{{ category }}">{{ message }}</p>
        {% endfor %}
    {% endwith %}
    <a href="/" class="back-link">Back to Home</a>
  </div>
</body>
  </html>
```

Add.html:

```
<!DOCTYPE html>
<html>
<head>
  <title>Add a New Vehicle</title>
  <link rel="stylesheet" href="global.css">
</head>
<body>
  <h1>Add a New Vehicle</h1>
  <form method="post">
    <label>Vehicle Name:</label>
    <input type="text" name="vehicle_name" required><br>
    <label>Vehicle Type:</label>
    <input type="text" name="vehicle_type" required><br>
    <label>Rental Price/Day:</label>
    <input type="number" step="0.01" name="rental_price" required><br>
    <button type="submit">Add Vehicle</button>
  </form>
  {% with messages = get_flashed_messages(with_categories=true) %}
    {% for category, message in messages %}
        <p class="{{ category }}">{{ message }}</p>
    {% endfor %}
  {% endwith %}
  <a href="/">Back</a>
</body>

</html>
```

Return.html

```
<!DOCTYPE html>
<html>
<head>
```

```
    <title>Return a Vehicle</title>
</head>
<body>
    <h1>Return a Vehicle</h1>
    <form method="post">
        <label>Rental ID:</label>
        <input type="number" name="rental_id" required><br>
        <button type="submit">Return</button>
    </form>
    {% with messages = get_flashed_messages(with_categories=true) %}
        {% for category, message in messages %}
            <p class="{{ category }}">{{ message }}</p>
        {% endfor %}
    {% endwith %}
    <a href="/">Back</a>
</body>
<style>

</style>
</html>
```

View.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Available Vehicles - Vehicle Rental System</title>
    <link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600&display=swap
" rel="stylesheet">
    <link rel="stylesheet" href="view-styles.css">
</head>
<body>
    <div class="container">
        <h1>Available Vehicles</h1>
        <div class="table-container">
            <table>
                <caption>Current Vehicle Inventory</caption>
                <thead>
                    <tr>
                        <th scope="col">ID</th>
                        <th scope="col">Name</th>
                        <th scope="col">Type</th>
                        <th scope="col">Price/Day</th>
                    </tr>
                </thead>
                <tbody>
                    {% for vehicle in vehicles %}
                        <tr>
                            <td>{{ vehicle[0] }}</td>
```

```
                <td>{{ vehicle[1] }}</td>
                <td>{{ vehicle[2] }}</td>
                <td>${{ vehicle[3] }}</td>
              </tr>
            {% endfor %}
          </tbody>
        </table>
      </div>
      <a href="/" class="back-link">Back to Home</a>
    </div>
</body>
<style>

</style>
</html>
```

Sql:

```
CREATE DATABASE VehicleRentalDB;

USE VehicleRentalDB;

CREATE TABLE Vehicles (
   vehicle_id INT AUTO_INCREMENT PRIMARY KEY,
   vehicle_name VARCHAR(50),
   vehicle_type VARCHAR(20),
   rental_price_per_day DECIMAL(10, 2),
   is_available BOOLEAN DEFAULT TRUE
);

CREATE TABLE Rentals (
   rental_id INT AUTO_INCREMENT PRIMARY KEY,
   vehicle_id INT,
   customer_name VARCHAR(50),
   rental_date DATE,
   return_date DATE,
   FOREIGN KEY (vehicle_id) REFERENCES Vehicles(vehicle_id)
);
```

Output:

VEHICLE RENTAL SYSTEM

View Available Vehicles

Rent a Vehicle

Return a Vehicle

Add a New Vehicle

AVAILABLE VEHICLES

Current Vehicle Inventory

ID	NAME	TYPE	PRICE/DAY
1	tvs	car	$50.00
2	csd	car	$45.00
5	yamaha	car	$45.00
6	suzuki	car	$500.00

Back to Home

Limit to 1000 rows

```
1  SELECT * FROM Vehicles;
2
3
4
5
```

Result Grid | Filter Rows: | Edit: | Export/Import:

vehicle_id	vehicle_name	vehicle_type	rental_price_per_day	is_available
1	tvs	car	50.00	1
2	csd	car	45.00	1
3	tvs	car	54.00	0
4	yamaha	bike	50.00	0
5	yamaha	car	45.00	1
6	suzuki	car	500.00	1
NULL	NULL	NULL	NULL	NULL

Add a New Vehicle

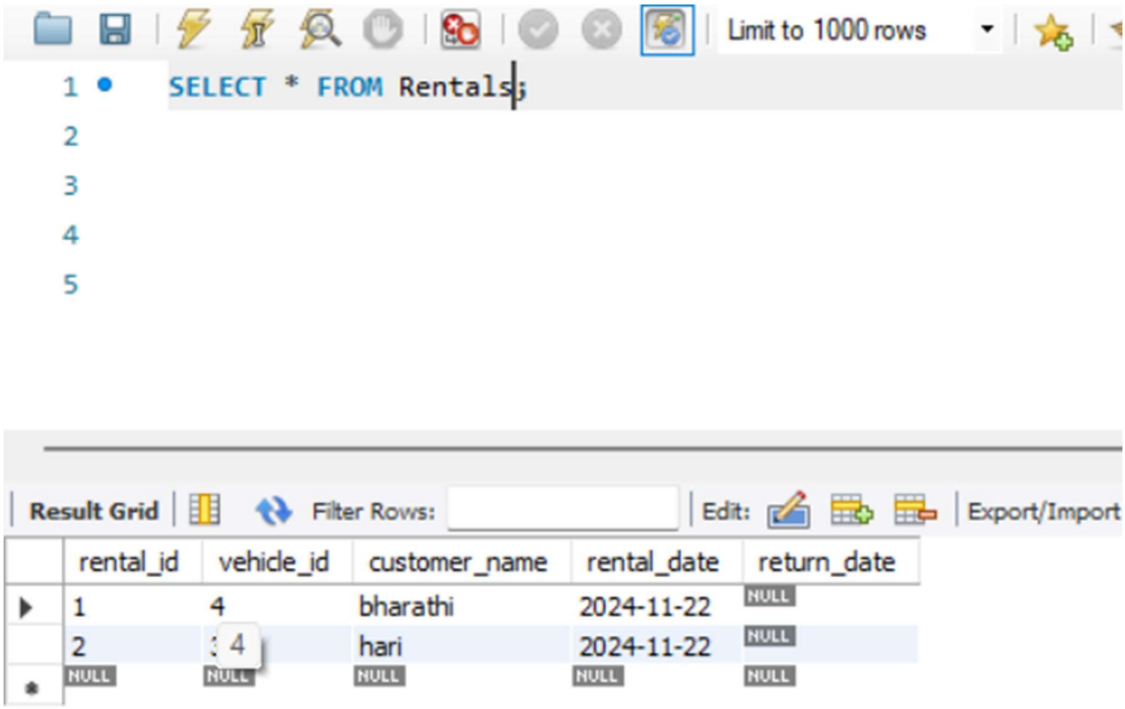
Vehicle Name:

Vehicle Type:

Rental Price/Day:

Add Vehicle

Back



Conclusion

The Vehicle Rental System provides an efficient and automated solution for managing vehicle rentals. It simplifies the process of renting and returning vehicles, ensuring real-time updates and accurate records. The system enhances both customer experience and administrative efficiency, with potential for future scalability and feature integration.