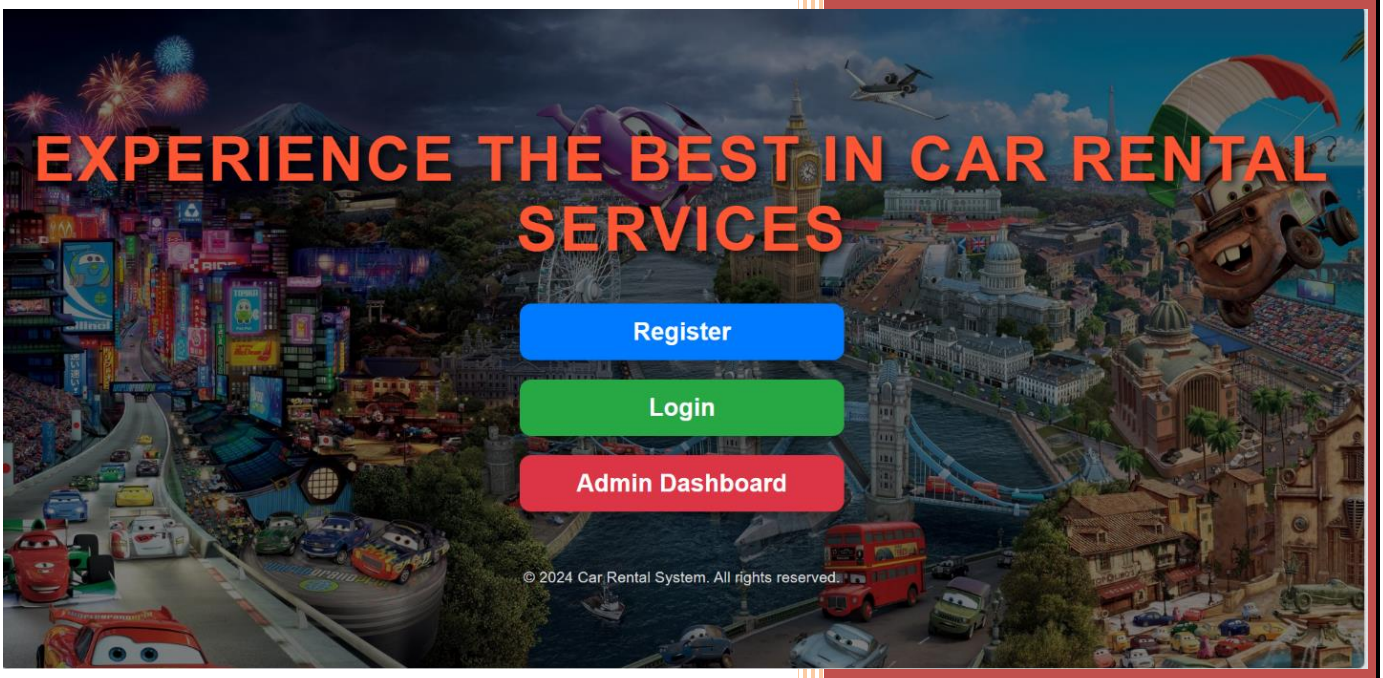
Car Rental System



Sama Ehab Ibrahim Adam 7975

Marie Magdi Sharl 8049

Kiria Elkess Daoud Botros 8301

Karen Sameh Sabry Nasr 8366

Car Rental System

Introduction

The **Car Rental System** project is a comprehensive web-based application designed to streamline the processes of renting cars, managing reservations, and processing payments. This system caters to both customers and administrators, offering user-friendly interfaces and robust backend functionalities to ensure a seamless and secure experience.

The project encompasses essential features, including user authentication, intuitive dashboards, advanced search capabilities, and secure reservation and payment modules. By leveraging modern web development technologies such as HTML, CSS, PHP, and MySQL, this system aims to provide a reliable solution for car rental operations while ensuring data integrity and usability.

This documentation outlines the implemented features, technical details, and methodologies employed to achieve a functional and efficient Car Rental System. The system is built with scalability and security in mind, adhering to best practices for database interactions and user data protection.

DDL

```
CREATE DATABASE CarRentalSystem;

USE CarRentalSystem;

-- Office Table

CREATE TABLE Office (

    OfficeID INT AUTO_INCREMENT PRIMARY KEY,

    Location VARCHAR(100) NOT NULL,

    Phone VARCHAR(20)

);

-- Car Table

CREATE TABLE Car (

    CarID INT AUTO_INCREMENT PRIMARY KEY,
```

```
    Model VARCHAR(50) NOT NULL,
    Year INT NOT NULL,
    PlateID VARCHAR(20) UNIQUE NOT NULL,
    Status ENUM('Active', 'Out of Service', 'Rented') DEFAULT 'Active',
    CostPerDay DECIMAL(10, 2) NOT NULL
);
-- Customer Table
CREATE TABLE Customer (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    LastName VARCHAR(50) NOT NULL,
    Email VARCHAR(100) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL,
    Phone VARCHAR(15),
    Address TEXT
);
-- Reservation Table
CREATE TABLE Reservation (
    ReservationID INT AUTO_INCREMENT PRIMARY KEY,
    CarID INT,
    CustomerID INT,
    ReservationDate DATE NOT NULL,
    StartDate DATE NOT NULL, -- Pickup date
    EndDate DATE NOT NULL, -- Return date
```

```
    OfficeID INT,
    Amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (CarID) REFERENCES Car(CarID),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);

-- Payment Table
CREATE TABLE Payment (
    PaymentID INT AUTO_INCREMENT PRIMARY KEY,
    ReservationID INT,
    TotalCost DECIMAL(10, 2),
    PaymentDate DATE NOT NULL,
    PaymentMethod ENUM('Cash', 'Credit Card', 'Online') NOT NULL,
    CardID VARCHAR(255) NOT NULL,
    FOREIGN KEY (ReservationID) REFERENCES Reservation(ReservationID)
);
-- Admin Table
CREATE TABLE Admin (
    AdminID INT AUTO_INCREMENT PRIMARY KEY,
    Username VARCHAR(50) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL, -- Encrypted passwords are recommended
    Email VARCHAR(100) UNIQUE NOT NULL,
    FullName VARCHAR(100) NOT NULL
);
```

```
-- Altering the 'Customer' table

ALTER TABLE customer MODIFY phone VARCHAR(15);

ALTER TABLE customer ADD UNIQUE (email);

ALTER TABLE customer ADD UNIQUE (phone);

-- Dropping the 'PaymentMethod' column from the 'Payment' table

ALTER TABLE payment DROP COLUMN PaymentMethod;

-- Dropping the 'Amount' column from the 'Reservation' table

ALTER TABLE reservation DROP COLUMN amount;
```

DML

```
USE CarRentalSystem;

-- Inserting data into the 'Office' table

INSERT INTO Office (Location, Phone) VALUES

('New York', '555-1234'),

('Los Angeles', '555-5678'),

('Chicago', '555-8765'),

('Miami', '555-4321'),

('San Francisco', '555-8766'),

('Dallas', '555-2468'),

('Boston', '555-1357'),

('Houston', '555-7531'),

('Seattle', '555-9512'),

('Denver', '555-4678');


-- Inserting data into the 'Car' table
```

```
INSERT INTO Car (Model, Year, PlateID, Status, CostPerDay) VALUES
('Toyota Camry', 2020, 'ABC123', 'Active', 50.00),
('Honda Accord', 2021, 'XYZ456', 'Rented', 55.00),
('Ford Mustang', 2022, 'LMN789', 'Out of Service', 80.00),
('Chevrolet Malibu', 2021, 'DEF987', 'Active', 60.00),
('BMW 3 Series', 2020, 'GHI654', 'Rented', 75.00),
('Audi A4', 2019, 'JKL321', 'Active', 85.00),
('Mercedes-Benz C-Class', 2022, 'MNO234', 'Out of Service', 95.00),
('Tesla Model 3', 2023, 'PQR567', 'Active', 120.00),
('Nissan Altima', 2022, 'STU890', 'Rented', 70.00),
('Hyundai Sonata', 2021, 'VWX345', 'Active', 65.00),
('Kia Optima', 2020, 'YZA678', 'Active', 55.00),
('Volkswagen Passat', 2019, 'BCD234', 'Out of Service', 60.00),
('Subaru Outback', 2023, 'EFG567', 'Rented', 80.00),
('Mazda CX-5', 2021, 'HIJ890', 'Active', 70.00);

-- Inserting data into the 'Customer' table
INSERT INTO Customer (FirstName, LastName, Email, Password, Phone, Address) VALUES
('John', 'Doe', 'johndoe@email.com', 'password123', '555-1111', '123 Main St, New York, NY'),
('Jane', 'Smith', 'janesmith@email.com', 'password456', '555-2222', '456 Oak St, Los Angeles, CA'),
('Michael', 'Johnson', 'mikejohnson@email.com', 'password789', '555-3333', '789 Pine St, Chicago, IL'),
('Sarah', 'Williams', 'sarahw@email.com', 'password321', '555-4444', '101 Maple St, Miami, FL'),
('David', 'Brown', 'davidb@email.com', 'password654', '555-5555', '202 Birch St, San Francisco, CA'),
('Emily', 'Davis', 'emilyd@email.com', 'password987', '555-6666', '303 Cedar St, Dallas, TX'),
```

('Joshua', 'Miller', 'joshuam@email.com', 'password147', '555-7777', '404 Elm St, Boston, MA'),

('Samantha', 'Garcia', 'samanthag@email.com', 'password258', '555-8888', '505 Pine St, Houston, TX'),

('Christopher', 'Martinez', 'chrism@email.com', 'password369', '555-9999', '606 Oak St, Seattle, WA'),

('Olivia', 'Hernandez', 'oliviah@email.com', 'password753', '555-0000', '707 Maple St, Denver, CO'),

('Daniel', 'Lopez', 'daniel@email.com', 'password741', '555-1234', '808 Birch St, Miami, FL'),

('Sophia', 'Gonzalez', 'sophiag@email.com', 'password852', '555-4321', '909 Cedar St, New York, NY'),

('Benjamin', 'Clark', 'benjamin@email.com', 'password963', '555-5678', '1010 Elm St, Los Angeles, CA'),

('Ava', 'Rodriguez', 'ava@email.com', 'password654', '555-6789', '1111 Oak St, Chicago, IL');

-- Inserting data into the 'Reservation' table (Note: Amount column removed)

INSERT INTO Reservation (CarID, CustomerID, ReservationDate, StartDate, EndDate, OfficeID)
VALUES

(1, 1, '2024-12-01', '2024-12-10', '2024-12-15', 1),

(2, 2, '2024-12-05', '2024-12-12', '2024-12-20', 2),

(3, 3, '2024-12-10', '2024-12-15', '2024-12-18', 3),

(4, 4, '2024-12-12', '2024-12-20', '2024-12-25', 4),

(5, 5, '2024-12-15', '2024-12-18', '2024-12-22', 5),

(6, 6, '2024-12-20', '2024-12-22', '2024-12-28', 6),

(7, 7, '2024-12-22', '2024-12-26', '2024-12-30', 7),

(8, 8, '2024-12-25', '2024-12-28', '2025-01-02', 8),

(9, 9, '2024-12-27', '2025-01-02', '2025-01-05', 9),

(10, 10, '2024-12-30', '2025-01-05', '2025-01-10', 10),

(11, 1, '2024-12-10', '2024-12-15', '2024-12-20', 2),

(12, 2, '2024-12-15', '2024-12-18', '2024-12-23', 3),

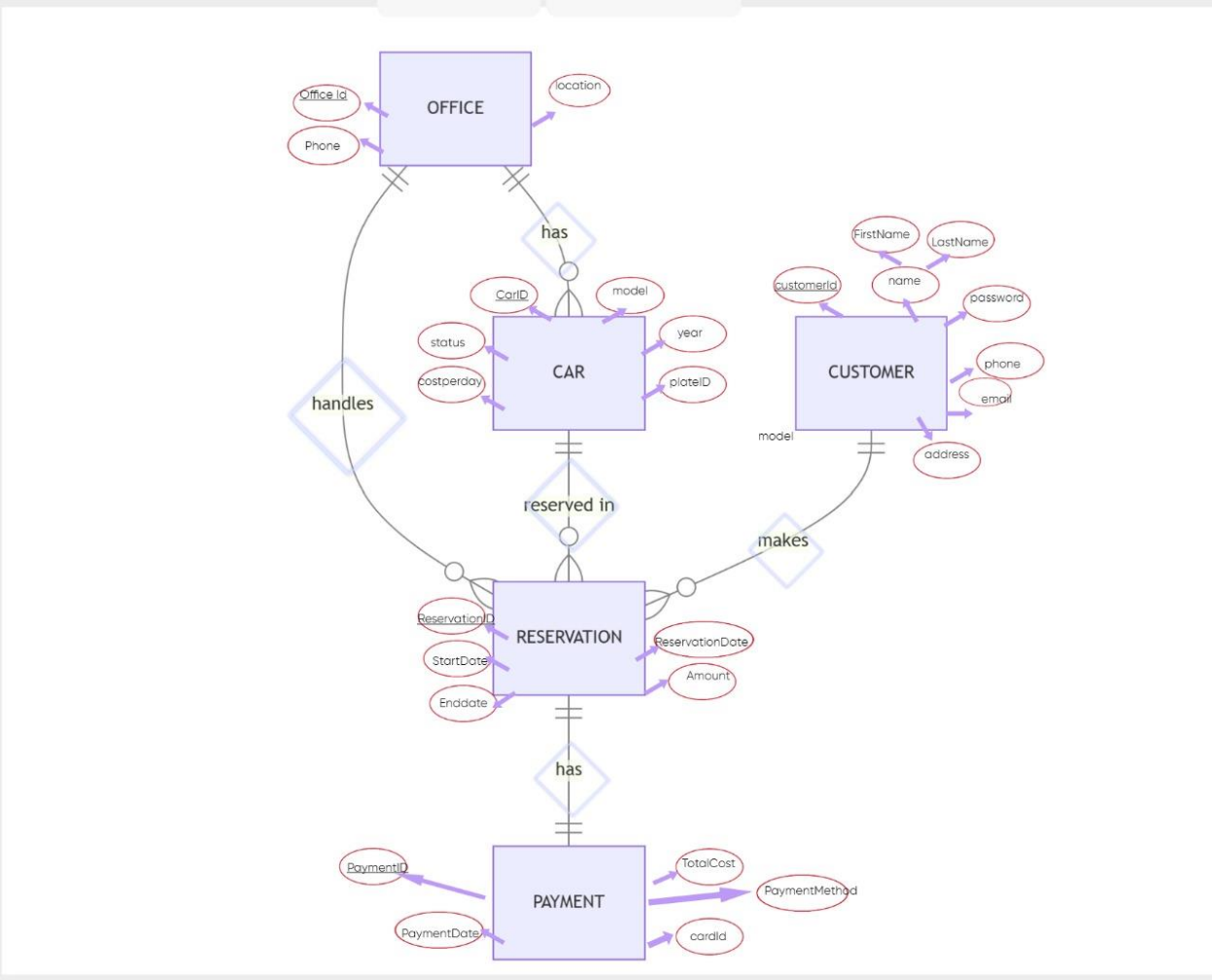
(13, 3, '2024-12-18', '2024-12-22', '2024-12-25', 4),

```
(14, 4, '2024-12-21', '2024-12-24', '2024-12-29', 5);


-- Inserting data into the 'Payment' table
INSERT INTO Payment (ReservationID, TotalCost, PaymentDate, CardID) VALUES
(1, 500.00, '2024-12-01', '1234-5678-9876-5432'),
(2, 600.00, '2024-12-05', '9876-5432-1234-5678'),
(3, 240.00, '2024-12-10', '1111-2222-3333-4444'),
(4, 300.00, '2024-12-12', '5555-6666-7777-8888'),
(5, 350.00, '2024-12-15', '8888-7777-6666-5555'),
(6, 420.00, '2024-12-20', '1111-4444-3333-2222'),
(7, 560.00, '2024-12-22', '1234-5678-9876-5432'),
(8, 600.00, '2024-12-25', '8765-4321-5678-1234'),
(9, 210.00, '2024-12-27', '2345-6789-1234-5678'),
(10, 300.00, '2024-12-30', '5555-4444-3333-2222'),
(11, 400.00, '2024-12-10', '9999-8888-7777-6666'),
(12, 500.00, '2024-12-15', '1111-2222-3333-4444'),
(13, 350.00, '2024-12-18', '5555-4444-3333-2222'),
(14, 600.00, '2024-12-21', '1234-5678-9876-5432');


-- Inserting data into the 'Admin' table
INSERT INTO Admin (Username, Password, Email, FullName) VALUES
('admin1', 'adminpassword1', 'admin1@email.com', 'Admin One'),
('admin2', 'adminpassword2', 'admin2@email.com', 'Admin Two');
```

ERD



Features Implemented

1. Admin Login System

- **Frontend (File: `adminLogin.html`)**:
  - Designed an interactive and responsive login page for administrators.
  - Features included:
    - Background image and gradient text effects for an engaging UI.
    - Styled input fields with focus effects to highlight user interaction.
    - A dynamic message display system to show login success or error messages based on query parameters.
    - A fully responsive layout adaptable to various screen sizes, ensuring usability across devices.
  - Implemented CSS animations to enhance user experience, such as hover effects on buttons and smooth transitions.

- **Backend (File: `adminlogin.php`)**:
    - Developed secure login functionality by integrating backend logic.
    - Ensured credentials were verified against the database.
    - Supported password hashing (using bcrypt) for secure storage and verification.

---

2. Customer Login and Registration

- **Customer Login (Files: `customerLogin.html`, `login.php`)**:
    - Designed a login interface similar to the admin system with a vibrant yet professional style.
    - Backend logic validated user credentials and managed user sessions upon successful login.
    - Error messages were displayed dynamically based on login results, such as incorrect email or password.
- **Customer Registration (Files: `customerRegister.html`, `register.php`)**:
    - Developed a multi-field registration form, requiring:
        - First Name, Last Name, Email, Password, Confirm Password, Phone, and Address.
    - Implemented comprehensive frontend and backend validation:
        - Passwords must match and conform to security standards.
        - Phone numbers were validated for numeric input and length constraints.
        - Checked for duplicate email or phone number entries in the database.
    - Backend integration included:
        - Data insertion into the `Customer` table.
        - Secure password storage using password hashing.

---

3. User Dashboards

- **Admin Dashboard (File: `adminMenu.html`)**:
    - Designed a centralized dashboard for administrators to perform critical actions such as:
        - Searching for car information.
        - Generating reports.
        - Adding or updating car statuses.
    - Styled the interface with gradient backgrounds and hover effects for a modern look.
    - Organized options in a column-based layout to facilitate intuitive navigation.
- **Customer Dashboard (File: `customerMenu.html`)**:
    - Created a user-friendly dashboard for customers, featuring:
        - A simple car search system.
        - Quick access to the reservation module.

---

4. Search Systems

- **Basic Search (File: `basicSearch.html`)**:
  - Enabled users to search cars by model and status.
  - Used a form submission system to pass user input to the backend for processing.
  - Displayed results in a styled table or showed a message if no results were found.
- **Advanced Search (File: `advancedSearch.html`, `search.php`)**:
  - Designed for administrators, this feature allowed for advanced queries using:
    - Customer email.
    - Reservation dates.
    - Car plate numbers.
  - Backend logic used flexible query construction to return results dynamically based on available parameters.
  - Results included reservation details, customer names, and car information, displayed in a tabular format with alternating row colors for clarity.

5. Reservation and Payment System

Reservation Module:

- Implemented a customer-facing reservation system (Files: `confirm_reservation.php`, `reserve.php`).
- Key features:
  - Customers could select car models, start and end dates, and view total costs dynamically.
  - Backend validated customer input and calculated total costs based on the number of rental days and cost per day fetched from the database.
  - Inserted reservation data into the `Reservation` table, including fields such as `CarID`, `CustomerID`, `ReservationDate`, `StartDate`, `EndDate`, and `OfficeID`.
  - Retrieved and displayed a confirmation summary with customer name, car details, and total cost.

Payment Module:

- Integrated secure payment handling (Files: `payment.php`, `confirmReservation.php`).
- Key features:
  - Dynamically calculated the total payment amount based on reservation details.
  - Secured payment information by hashing sensitive fields like `CardID` using SHA-256.
  - Recorded payment details into the `Payment` table, including `ReservationID`, `PaymentDate`, `CardID`, and `TotalCost`.
  - Displayed success or error messages based on the result of the database operations.

Technical Implementation

Frontend Development

- Used **HTML5** for semantic markup and **CSS3** for styling, ensuring compatibility across browsers.
- Incorporated JavaScript to manage form interactions, dynamic result rendering, and error handling.

Backend Development

- Scripts written in **PHP** handled data validation, database interactions, and user session management.
- Used **MySQL** database with prepared statements for enhanced security against SQL injection attacks.

Database Interactions

Authentication

- **Files**: `adminlogin.php`, `login.php`, `register.php`
  - Validated user credentials (admins and customers) by querying the database for matching records.
  - Utilized prepared statements for secure execution of SQL queries.
  - Inserted new customer records with hashed passwords to prevent plaintext storage.

Search System

- **Files**: `search.php`
  - Supported two search modes:
    - **Basic Search**: Query cars based on model and status from the `Car` table.
    - **Advanced Search**: Fetch reservations and associated car and customer details by joining the `Reservation`, `Customer`, and `Car` tables.
  - Constructed queries dynamically based on user-provided parameters.
  - Delivered search results as JSON for frontend rendering.

Reservation and Payment:

- **Files**: `reserve.html`, `reserve.php`, `confirm_reservation.php`, `payment.php`
- Reserved cars for customers by inserting reservation details into the **`Reservation`**
- table.Recorded payments securely in the `Payment` table with hashed card information and accurate transaction details.

Customer and Admin Dashboards

- **Files**: `adminMenu.html`, `customerMenu.html`
    - Displayed dynamically retrieved data for available actions (e.g., adding cars, updating statuses, viewing reservations).

Registration Validation

- **Files**: `register.php`
    - Queried the database to check for duplicate emails or phone numbers.
    - Used conditional checks to ensure no existing records matched the new entry, preventing conflicts.