CAR RENTAL SYSTEM

Name	ID
Mazen Elsayed	8247
Youssef Attia	8164
Youssef Ibrahim	8177
Youssef Ragab	8187

🚗 **CarRental**   Customer Login   Admin Login

Welcome to CarRental

Your premier destination for quality car rentals

Login   Register

Our Fleet
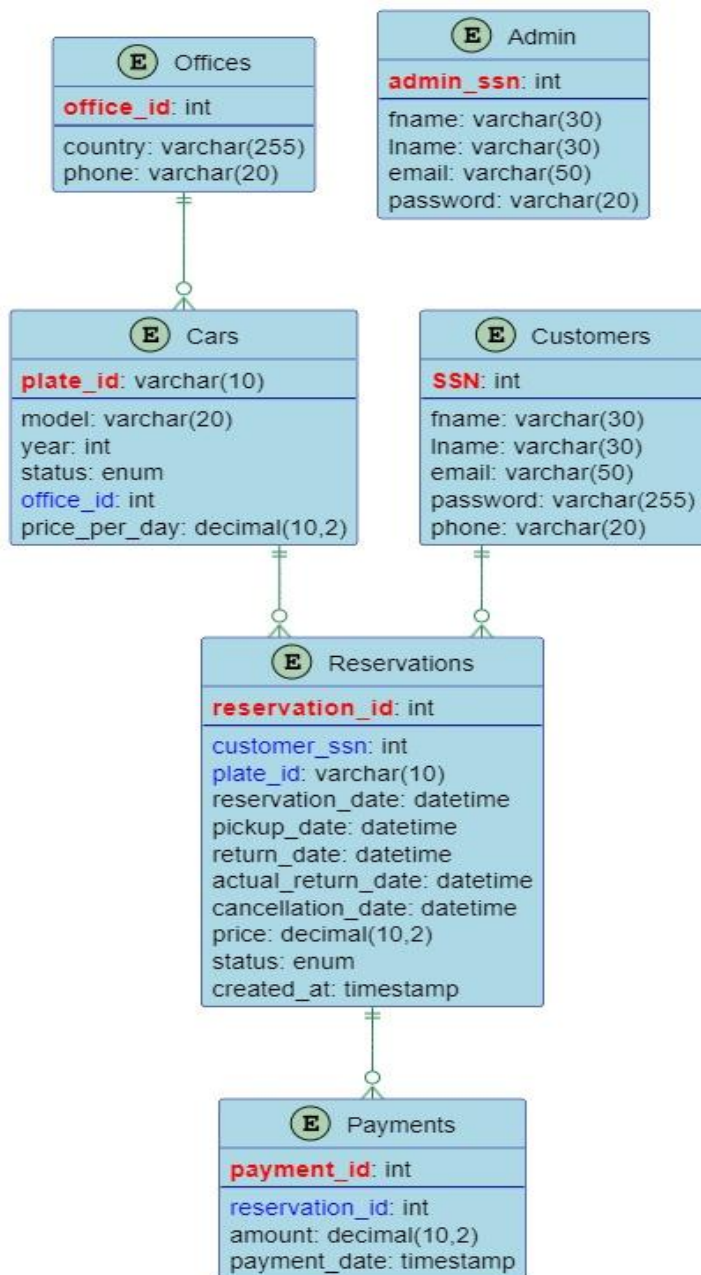
1. SYSTEM OVERVIEW

Our Car Rental System is a web-based application that manages car rentals across multiple offices.

It supports two user types:

- Customers: Can browse cars, make reservations, and manage their bookings

- Administrators: Can manage cars, view reports, and oversee the system

2. ENTITY RELATIONSHIP DIAGRAM

ERD Diagram

DATABASE SCHEMA

-Customer: Stores user account information and credentials

-Car: Maintains car inventory and details

-Reservation: Tracks booking information

-Payment: Records transaction details

-Office: Manages multiple rental locations

-Admin: Stores administrator accounts

3. IMPLEMENTED FEATURES

CUSTOMER FEATURES

- Account Registration and Login
- Car Search and Filtering
- Reservation Management
    - Create new reservations
    - View active reservations
    - Cancel reservations
- Secure Authentication

ADMIN FEATURES

- Secure Admin Login
- Dashboard with Statistics
- Car Management
    - Add new cars
    - Update car status
- Comprehensive Reporting
    - Reservation reports
    - Revenue analysis
    - Car status tracking
    - Payment history

4. SCREEN-BY-SCREEN DESCRIPTION

CUSTOMER SCREENS

REGISTRATION PAGE (**/CUSTOMER/REGISTER**)

- Purpose: New customer account creation
- Fields:
    - First Name, Last Name
    - Email (with validation)
    - Phone (11 digits)
    - Password (with complexity requirements)
- Validation: Client-side and server-side input validation

CUSTOMER LOGIN (**/CUSTOMER/LOGIN**)
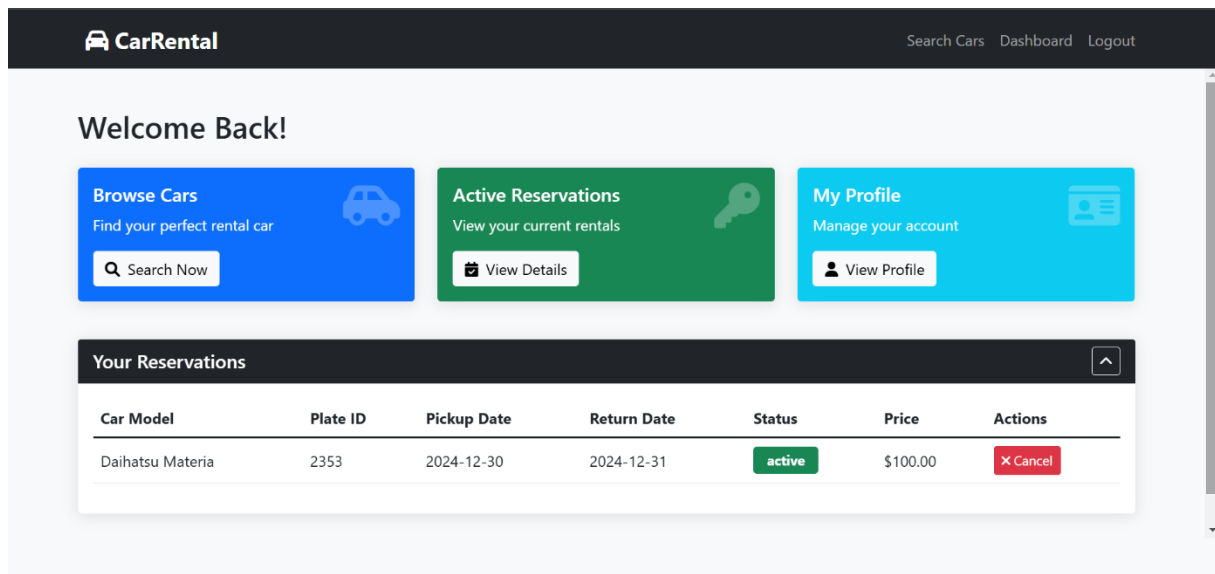
- Purpose: Authenticate existing customers
- Features:
    - Email/password authentication
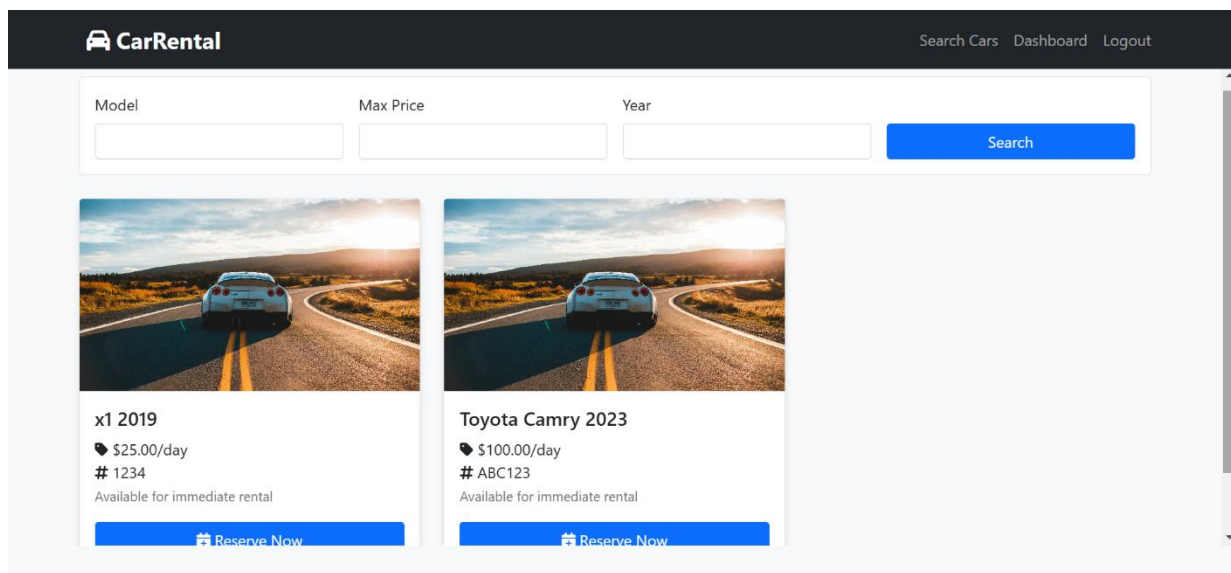   - Error handling

CUSTOMER DASHBOARD (**/CUSTOMER/DASHBOARD**)

- Purpose: Central hub for customer activities
- Features:
  - Quick actions menu
  - Active reservations display
  - Car search shortcut
  - Profile management
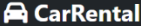

CAR SEARCH (**/CUSTOMER/SEARCH**)

- Purpose: Browse available cars
- Features:
  - Filter by price, model, year
  - Real-time availability check
  - Booking initiation

ADMIN SCREENS

ADMIN LOGIN (/ADMIN/LOGIN)

- Purpose: Secure admin authentication
- Features:
  – Enhanced security measures
  – Session management
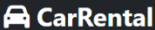 – Access control

ADMIN DASHBOARD (/ADMIN/DASHBOARD)

- Purpose: System overview and management
- Features:
  – Car statistics
  – Quick actions
  – System status
  – Navigation to reports

REPORTS PAGE (**/ADMIN/REPORTS**)

- Purpose: Comprehensive system reporting
- Features:
  - Date range filtering
  - Multiple report types:
    - Reservation history
    - Car status
    - Revenue analysis
    - Payment tracking
  - Export capabilities

Office Statistics

Office	Total Cars	Total Reservations	Total Revenue
Test Country	3	1	$100.00
Egypt	0	0	$0.00

Daily Payments (1)

Date	Number of Payments	Total Amount
2024-12-29	1	$100.00

ADD CAR (**/ADMIN/CARS/ADD**)

- Purpose: Vehicle inventory management
- Features:
  - Car details input
  - Office assignment
  - Pricing setup

CarRental   Dashboard   Logout

Add New Car

Plate ID

Model

Year

Office ID

Price per Day

[Add Car]

TECHNICAL IMPLEMENTATION

SECURITY MEASURES
- Password hashing using bcrypt
- Input validation and sanitization

USER INTERFACE
- Responsive design using Bootstrap
- Interactive elements with JavaScript
- Real-time validation

FRAMEWORKS AND TECHNOLOGIES USED

BACKEND
- **Python**
- **Flask**: Web framework
    - Route handling
    - Session management
- **MySQL Connector**: Database connectivity
- **bcrypt**: Password hashing and security

FRONTEND
- **Bootstrap 5**: UI framework
    - Responsive design
    - Component styling
    - Grid system
- **Font Awesome**: For icons
- **JavaScript**: Client-side validation

DATABASE
- **MySQL**: Relational database
    - Foreign key constraints
    - Transaction management
    - Index optimization

IMPLEMENTATION APPROACH

1. **Database Design**
   - Created ERD diagram
   - Implemented database schema
   - Set up relationships and constraints

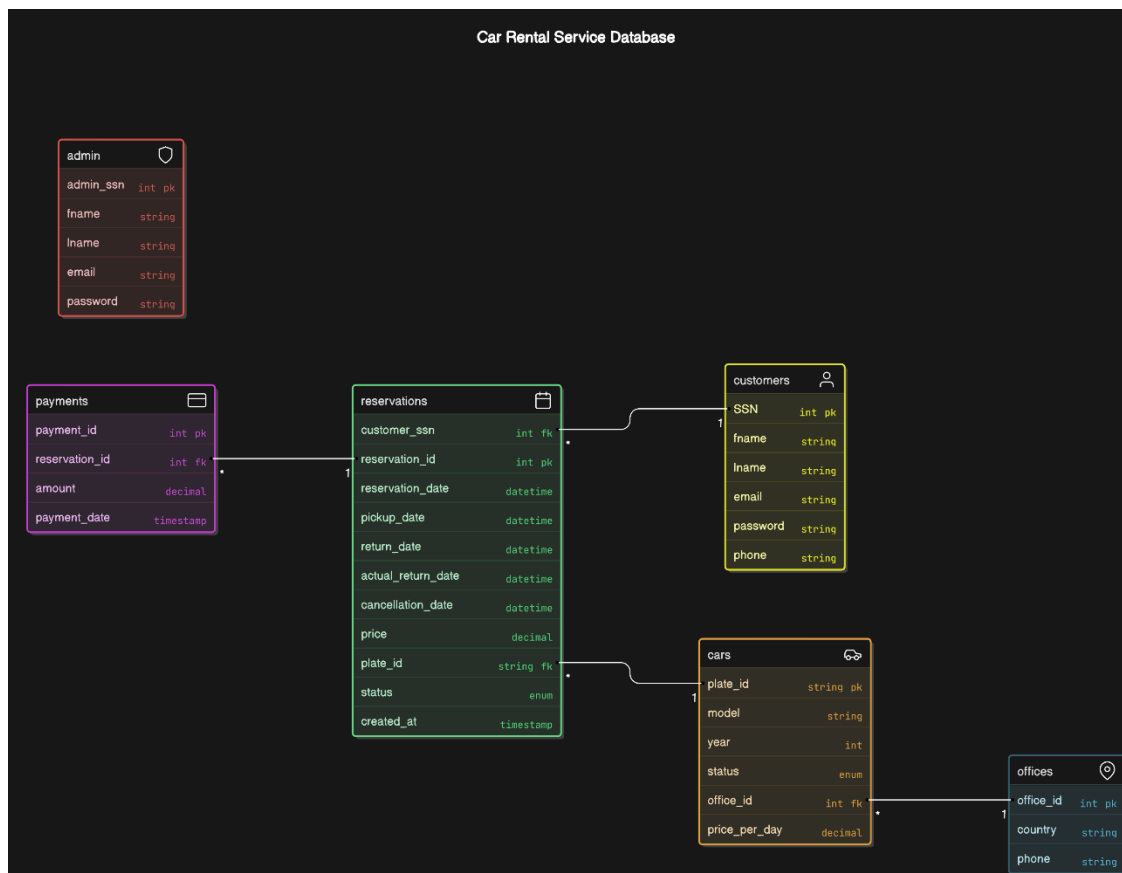2. **Backend Architecture**
   - Used OOP with separate classes for:
     - CarRentalSystem
     - Admin
     - Customer
   - Implemented secure authentication
   - Created reusable database methods

3. **Frontend Development**
   - Template inheritance using base.html
   - Responsive design for all devices
   - Client-side form validation
   - Interactive UI components
   - Real-time feedback

ERD

DDL

```sql
CREATE DATABASE car_rental;

USE car_rental;

CREATE TABLE offices (
    office_id INT PRIMARY KEY AUTO_INCREMENT,
    country VARCHAR(255) NOT NULL,
    phone VARCHAR(20)
);

CREATE TABLE cars (
    plate_id VARCHAR(10) PRIMARY KEY,
    model VARCHAR(20) NOT NULL,
    year INT NOT NULL,
    status ENUM('active', 'rented','out_of_service') DEFAULT 'active',
    office_id INT NOT NULL,
    price_per_day DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (office_id) REFERENCES offices(office_id) ON DELETE RESTRICT
);

CREATE TABLE customers (
    SSN INT PRIMARY KEY AUTO_INCREMENT,
    fname VARCHAR(30) NOT NULL,
    lname VARCHAR(30) NOT NULL,
    email VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(20) NOT NULL,
    phone VARCHAR(20) NOT NULL
);
```

```sql
CREATE TABLE reservations (
    reservation_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_ssn INT NOT NULL,
    plate_id VARCHAR(10) NOT NULL,
    reservation_date DATETIME NOT NULL,
    pickup_date DATETIME NOT NULL,
    return_date DATETIME NOT NULL,
    actual_return_date DATETIME,
    cancellation_date DATETIME,
    price DECIMAL(10, 2) NOT NULL,
    status ENUM('active','picked_up','completed', 'cancelled') DEFAULT 'active',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (customer_ssn) REFERENCES customers(SSN) ON DELETE RESTRICT,
    FOREIGN KEY (plate_id) REFERENCES cars(plate_id) ON DELETE RESTRICT
);

CREATE TABLE admin (
    admin_ssn INT PRIMARY KEY AUTO_INCREMENT,
    fname VARCHAR(30) NOT NULL,
    lname VARCHAR(30) NOT NULL,
    email VARCHAR(50) NOT NULL,
    password VARCHAR(20) NOT NULL
);

CREATE TABLE payments (
    payment_id INT PRIMARY KEY AUTO_INCREMENT,
    reservation_id INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (reservation_id) REFERENCES reservations(reservation_id) ON DELETE RESTRICT
);

ALTER TABLE customers MODIFY COLUMN password VARCHAR(255);
```