**Parking Lot Management**

**Home Controller**

Controller for server-side rendering

TotalSpots (int): Total number of parking spots in the parking lot.

HourlyFee (decimal): Hourly fee for parking in the lot.

**Actions**:

Index(): IActionResult: Renders the main view for the parking lot management web app.

CheckIn(string tagNumber): IActionResult: Handles the check-in process and updates 'Area B' via AJAX.

CheckOut(string tagNumber): IActionResult: Handles the check-out process and updates 'Area B' via AJAX.

Stats(): IActionResult: Renders the modal with statistics.

**Views:**

Index.cshtml

Contains "Area A" with input controls for check-in or out.

Uses JavaScript to handle AJAX requests to the backend for updating "Area B" dynamically.

Uses JavaScript Fetch API for AJAX requests.

Stats.cshtml

**Model:**

**Cars**

**Properties:**

SpotId (int): Unique identifier for the car.

TagNumber (string): Unique identifier for the car's tag.

EntryTime (DateTime): Timestamp for when the car entered the parking lot.

ExitTime (DateTime?): Timestamp for when the car exited the parking lot (null if still parked).

**ParkingSpots**

Contains the modal with statistics displayed when the "Stats" button is clicked.

**Properties**:

TotalSpots (int): Total number of parking spots in the parking lot.

HourlyFee (decimal): Hourly fee for parking in the lot.

AvailableSpots (int): Number of available parking spots in the parking lot.

Revenue (decimal): Total revenue earned by the parking lot as of now.

AverageCarsPerDay (double): Average number of cars parked per day for the past 30 days.

AverageRevenuePerDay (double): Average revenue earned per day for the past 30 days.

**Data**

**ParkingLotContext**

Contains database CRUD operation functions

LoadDataFromDatabase():Loads DB on page load

SaveCarToDatabase():save car with checkIn

DeleteCarFromDatabase(): Delete car on checkout

**Utility**:

LoadDataFromDatabase(): To Load database for general purpose