Pre-owned Cars Sales Management System

Project Description:

Pre-owned Car Portal is a web application where the pre-owned car owners can sell their cars and the car buyers can able to view and buy car in budget. This application can be access by buyer, seller and admin. Admin can login and manage the car details, buyer details, seller details and sales detail. The admin can manage the car categories. Each car posted in this portal includes commission. Admin can add and manage the commission. The sellers can sign up themselves and can. They can post their car details like brand, color, model, fuel type, engine, registration date, mileage, no of owners and description of the car that they want to sell.

They can view the applications from the buyers and can able to respond them back. The admin will review the details, add the commission and approve or reject it. The payment will be made through the application and the seller will transfer the commission to the admin. The sellers can able to view the cars sales history. The buyers can sign up and login with the application. Both buyers and sellers can update their profile details. Buyers can view all the available cars details based on the car categories and budget. If interested the buyer can send a message to the seller. The seller will respond to it back.

Also, the buyer can view the sellers contact details and they can contact them. They can review the car and can buy it by paying the quoted amount in the application.

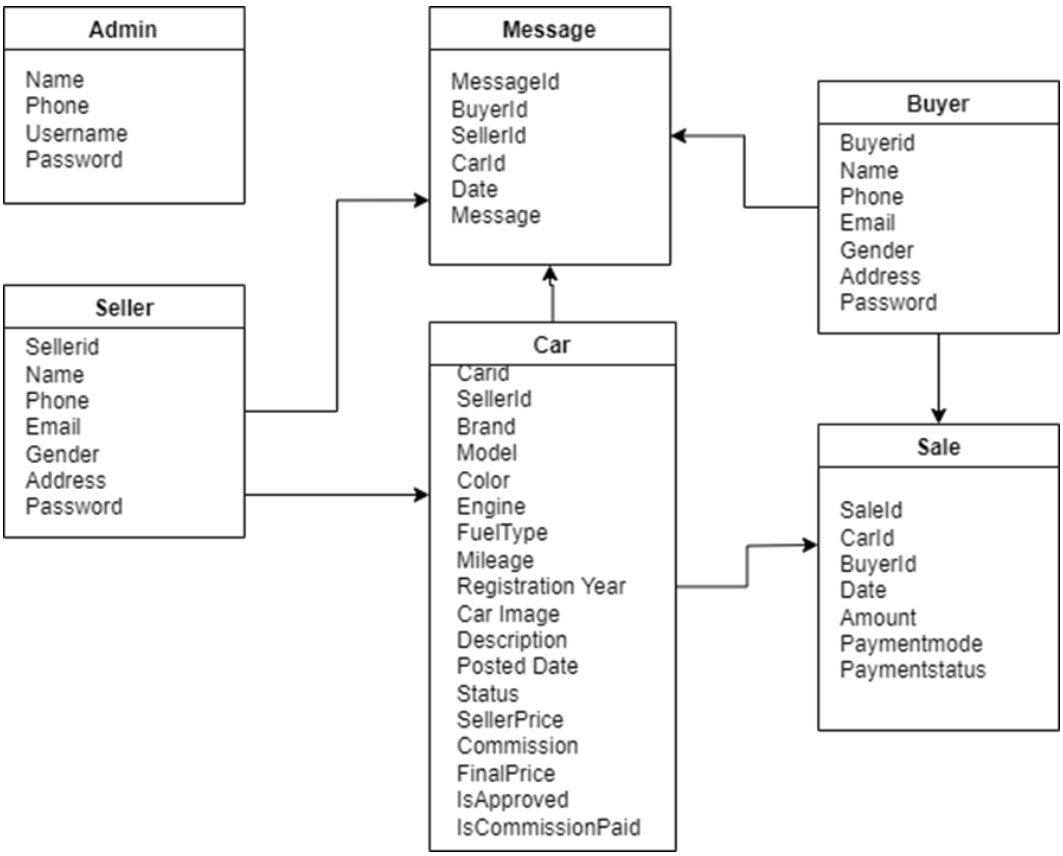
Database Description and Business Rules

Pre-owned Car Portal is a web application where the pre-owned car owners can sell their cars and the car buyers can be able to view and buy car in budget. This application can be access by buyer, seller and admin. Admin can login and manage the car details, buyer details, seller details and sales detail. The admin can manage the car categories. Each car posted in this portal includes commission. Admin can add and manage the commission.

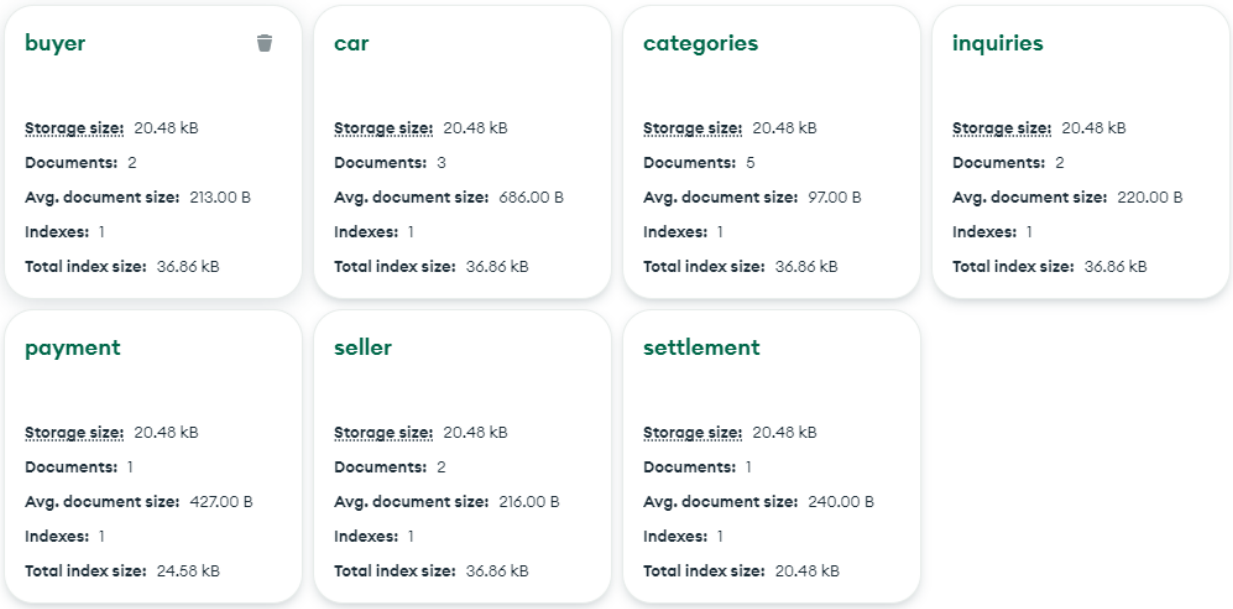
The sellers can sign up themselves and can. They can post their car details like brand, color, model, fuel type, engine, registration date, mileage, no of owners and description of the car that they want to sell. They can view the applications from the buyers and can be able to respond back to them. The admin will review the details, add the commission, and approve or reject it. The payment will be made through the application and the seller will transfer the commission to the admin. The sellers can be able to view the cars sales history.

The buyers can sign up and login with the application. Both buyers and sellers can update their profile details. Buyers can view all the available cars details based on the car categories and budget. If interested the buyer can send a message to the seller. The seller will respond to it back. Also, the buyer can view the sellers' contact details and they can contact them. They can review the car and can buy it by paying the quoted amount in the application.

NoSQL Database Model



NoSQL Database Model of this preowned car system contains the following collections:



1. Categories Collection:

```
_id: ObjectId('64c403b99d683b7132e7fdc3')
catId: "CAT2370001"
name: "HatchBack"
_class: "com.car.model.Category"
```

```
_id: ObjectId('64c403c09d683b7132e7fdc4')
catId: "CAT2370002"
name: "Sedan"
_class: "com.car.model.Category"
```

2. Seller Collection


```
_id: ObjectId('64c3ec4a4535da459c7e8193')
sellerId: "SELER64901"
name: "Seller - 1"
address: "2570 Heavner Court, Lynbrook, New York(NY), 11563"
mobile: "516-651-1336"
emailId: "seller1@gmail.com"
password: "1234"
_class: "com.car.model.Seller"
```


3. Cars Collection


```
_id: ObjectId('64c41184e998820e4a5f719a')
categoryId: "64c403b99d683b7132e7fdc3"
carId: "CAR0362353"
sellerId: "SELER64901"
brand: "Volkswagen"
model: "Polo"
regYear: "APR 2015"
regNumber: "GJ23-AN9894"
engineCapacity: "1198 cc"
transmission: "Manual"
kmDriven: "75151"
fuelType: "Petrol"
noOfOwner: 3
basePrice: 38100
description: "4 tyres that still have 100% of their tread life remaining, Rated 4 on…"
imgUrl: "64c41184e998820e4a5f719a.webp"
commissionPercentage: 10
postedDate: "2023-07-29T00:35:40.337993600"
isApproved: true
isSold: true
isBooked: true
```

4. Buyer Collection


```
_id: ObjectId('64c3ebf54535da459c7e8192')
buyerId: "BUYER64901"
name: "Buyer - 1"
address: "2662 Little Acres Lane, Kansas, Illinois(IL), 61933"
mobile: "708-266-2553"
emailId: "buyer1@gmail.com"
password: "1234"
_class: "com.car.model.Buyer"
```

```
_id: ObjectId('64c5171de31dfb774a6e3fc1')
buyerId: "BUYER64902"
name: "buyer-2"
address: "address"
mobile: "75698745632"
emailId: "buyer2@gmail.com"
password: "1234"
_class: "com.car.model.Buyer"
```


5. Payment Collection


```
_id: ObjectId('64c507c9f0eed4106c716553')
paymentId: "BOOK362353"
carId: "CAR0362353"
sellerId: "SELER64901"
buyerId: "BUYER64901"
bookingDate: "2023-07-29T18:06:25.569559400"
amount: 38100
bookingAmount: 19050
balanceAmount: 19050
paymentMode: "Credit Card"
paymentStatus: "Paid"
status: "Sold"
settlementStatus: "Settled"
cardNo: "8520741085209632"
cvv: "33"
expDate: "2025-01-22"
_class: "com.car.model.Payment"
```

6. Settlement Collection


```
_id: ObjectId('64c530638d531c55ccca2fc8')
paymentId: "BOOK362353"
sellerId: "SELER64901"
payDate: "29/07/2023"
saleAmount: 38100
commissionPercentage: 10
commissionAmount: 3810
settlementStatus: "Settled"
_class: "com.car.model.Settlement"
```

Sample Data:

Buyer


```
{
  "_id": {
    "$oid": "64c3ebf54535da459c7e8192"
  },
  "buyerId": "BUYER64901",
  "name": "Buyer - 1",
  "address": "2662 Little Acres Lane, Kansas, Illinois(IL), 61933",
  "mobile": "708-266-2553",
  "emailId": "buyer1@gmail.com",
  "password": "1234",
  "_class": "com.car.model.Buyer"
}
```


Car


```
{
  "_id": {
    "$oid": "64c515cbe31dfb774a6e3fbf"
  },
  "categoryId": "64c403c09d683b7132e7fdc4",
  "carId": "CAR0362354",
  "sellerId": "SELER64902",
  "brand": "Honda",
  "model": "Honda City I-VTEC V",
  "regYear": "AUG 2014",
  "regNumber": "GJ01-RF7831",
  "engineCapacity": "1497 CC",
```

```
  "transmission": "Manual",
  "kmDriven": "44,500 km",
  "fuelType": "Petrol",
  "noOfOwner": 1,
  "basePrice": 7600,
  "description": "Drive in style with sleek alloy wheels, 4 tyres that still have 100% of their tread life remaining, Equipped with 2 airbags and an Antilock Braking System (ABS)",
  "imgUrl": "64c515cbe31dfb774a6e3fbf.webp",
  "commissionPercentage": 10,
  "postedDate": "2023-07-29T19:06:11.560593200",
  "isApproved": true,
  "isSold": false,
  "isBooked": false,
  "_class": "com.car.model.Car"
}
```


Seller


```
{
  "_id": {
    "$oid": "64c3ec4a4535da459c7e8193"
  },
  "sellerId": "SELER64901",
  "name": "Seller - 1",
  "address": "2570 Heavner Court, Lynbrook, New York(NY), 11563",
  "mobile": "516-651-1336",
  "emaillId": "seller1@gmail.com",
  "password": "1234",
  "_class": "com.car.model.Seller"
}
```

Inquiries


```
{
  "_id": {
    "$oid": "64c52504b0d06b622e2fe4d4"
  },
  "carId": "CAR0362354",
  "buyerId": "BUYER64902",
  "sellerId": "SELER64902",
  "inquiry": "Inquiry from buyer2",
  "response": "Response from seller 2",
  "inquiryDate": "2023-07-29",
  "responseDate": "2023-07-29",
  "_class": "com.car.model.Inquiry"
}
```


Payment

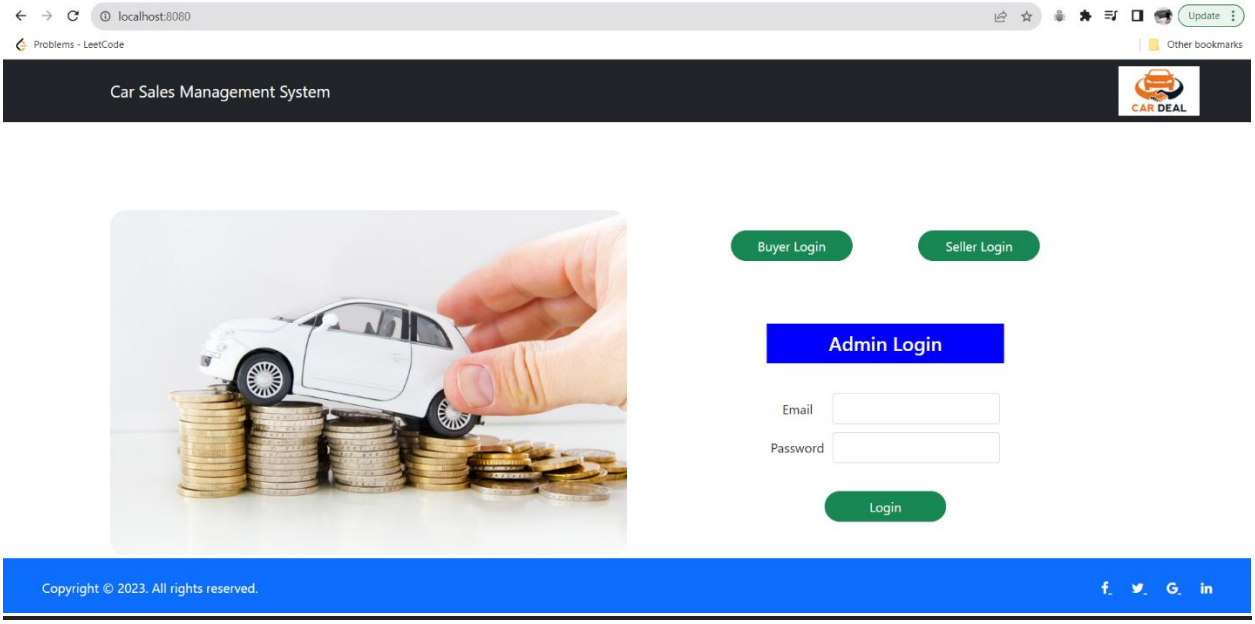

```
{
  "_id": {
    "$oid": "64c507c9f0eed4106c716553"
  },
  "paymentId": "BOOK362353",
  "carId": "CAR0362353",
  "sellerId": "SELER64901",
  "buyerId": "BUYER64901",
  "bookingDate": "2023-07-29T18:06:25.569559400",
  "amount": 38100,
  "bookingAmount": 19050,
  "balanceAmount": 19050,
```

```
  "paymentMode": "Credit Card",
  "paymentStatus": "Paid",
  "status": "Sold",
  "settlementStatus": "Settled",
  "cardNo": "8520741085209632",
  "cvv": "33",
  "expDate": "2025-01-22",
  "_class": "com.car.model.Payment"
}
```

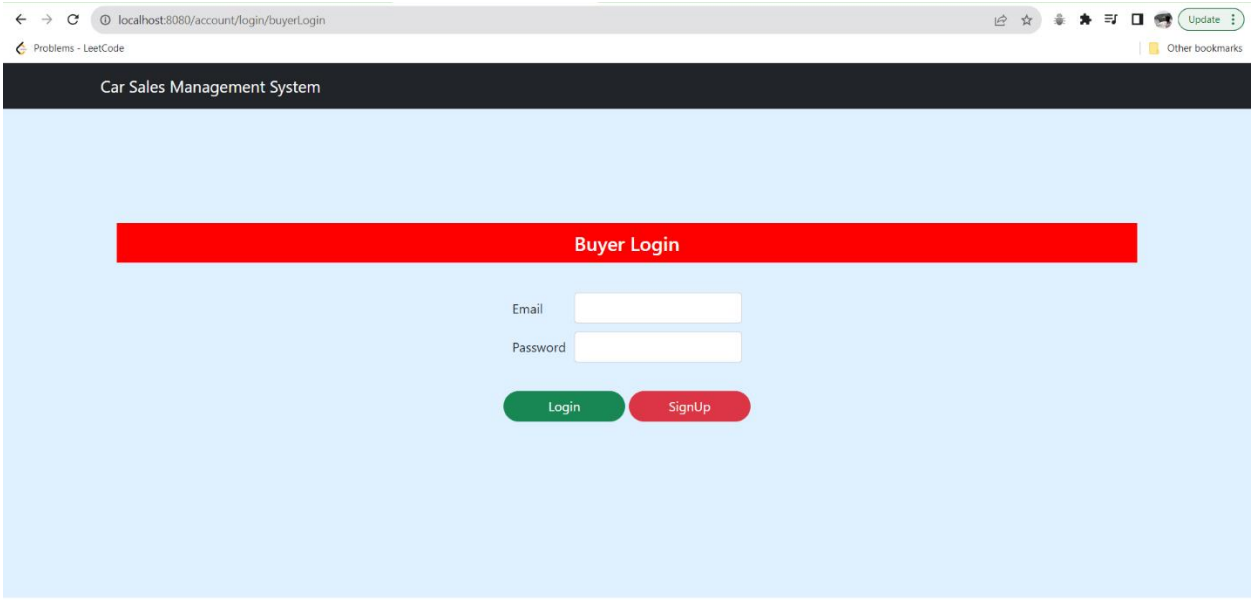

DATABASE COLLECTIONS	DESCRIPTION
Categories	Contains information about the car categories.
Seller	Contains information about Seller details.
Car	Contains about car details.
Buyer	Contains information about buyer details.
Settlement	Contains information about settlement details
payment	Contains information about payment details.
Inquiries	Contains inquiries made by Buyers

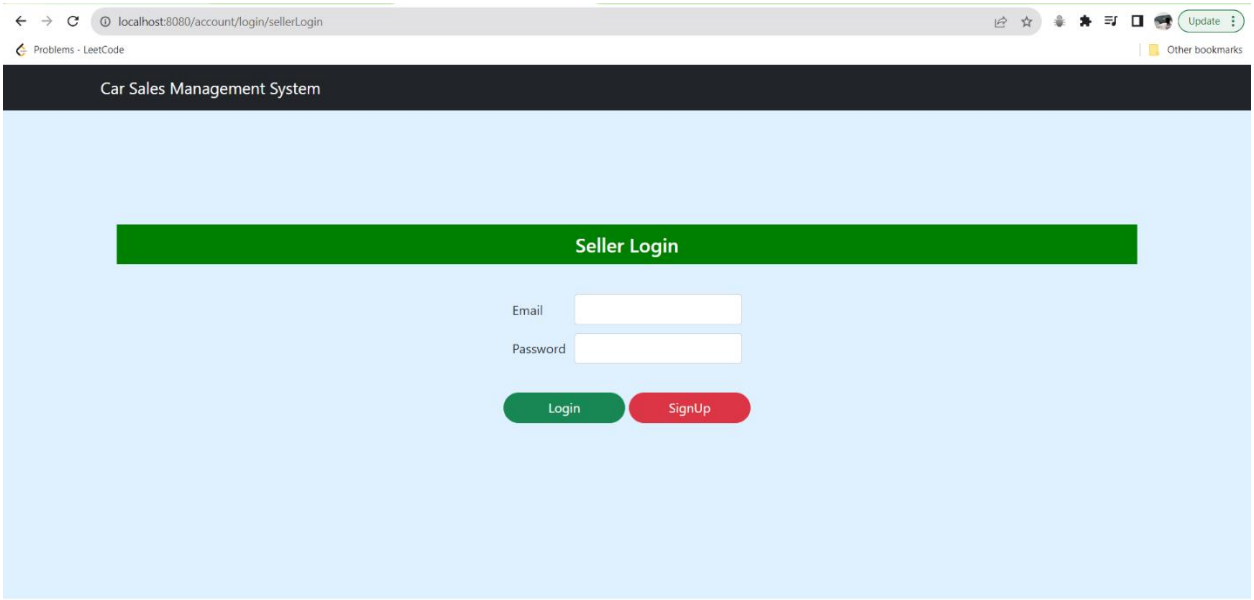
Screenshots

Homepage

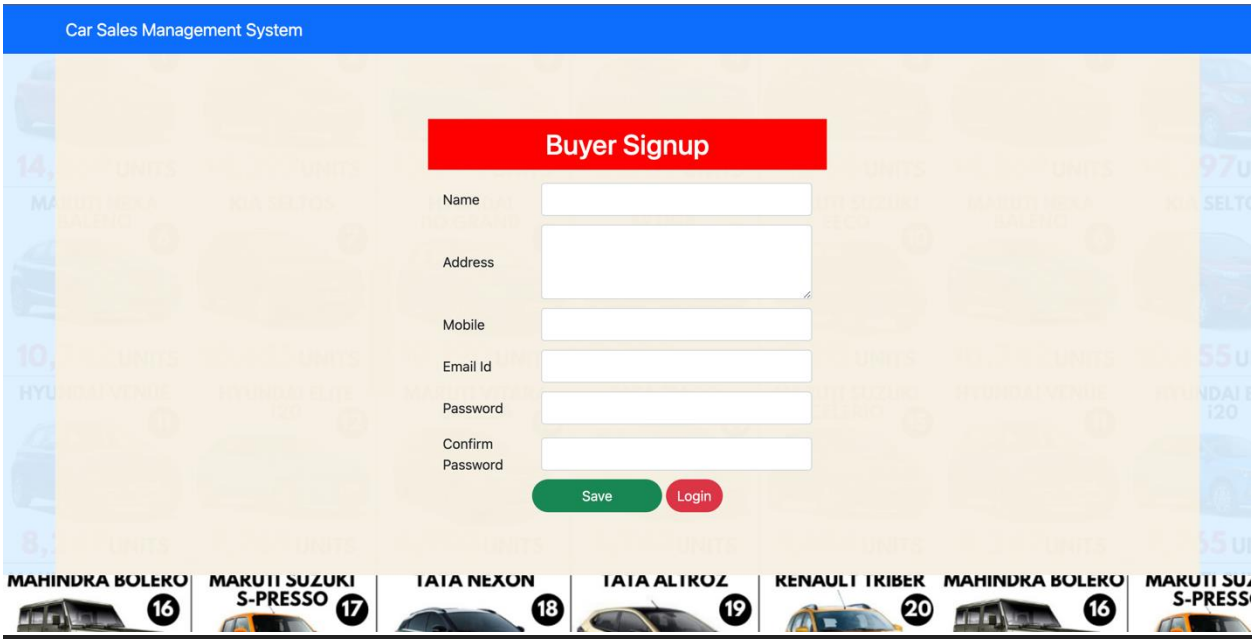


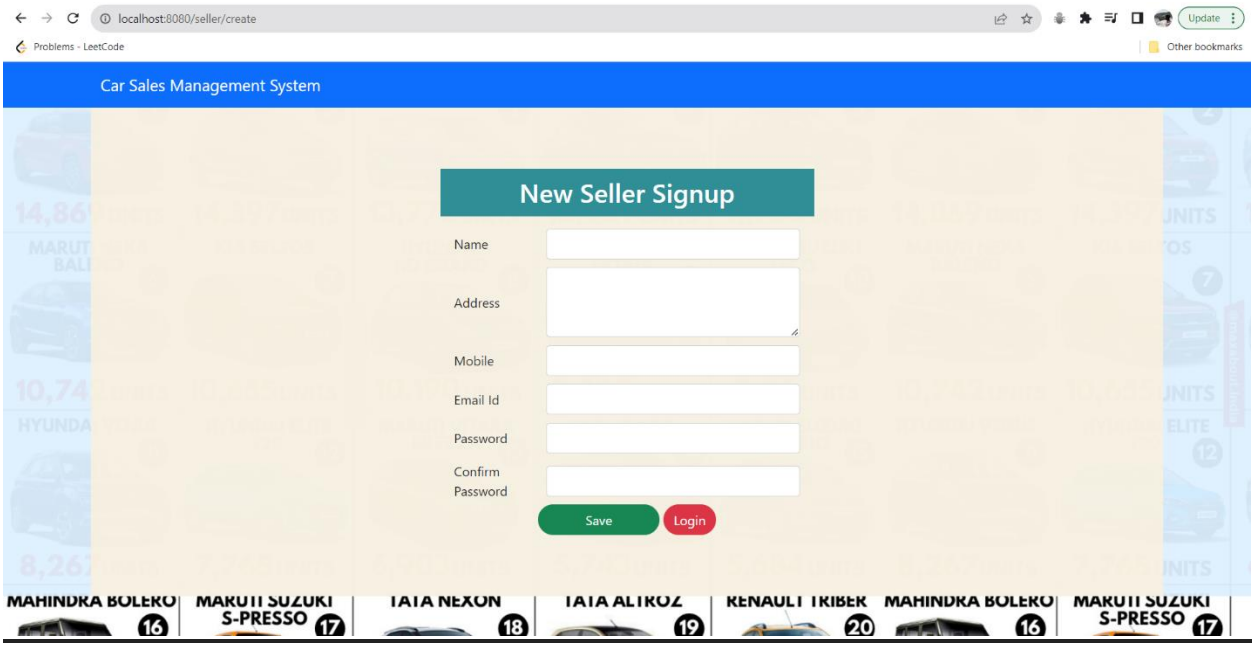
Login


localhost:8080/account/login/buyerLogin
Update
Problems - LeetCode
Other bookmarks
Car Sales Management System
Buyer Login
Email
Password
Login
SignUp

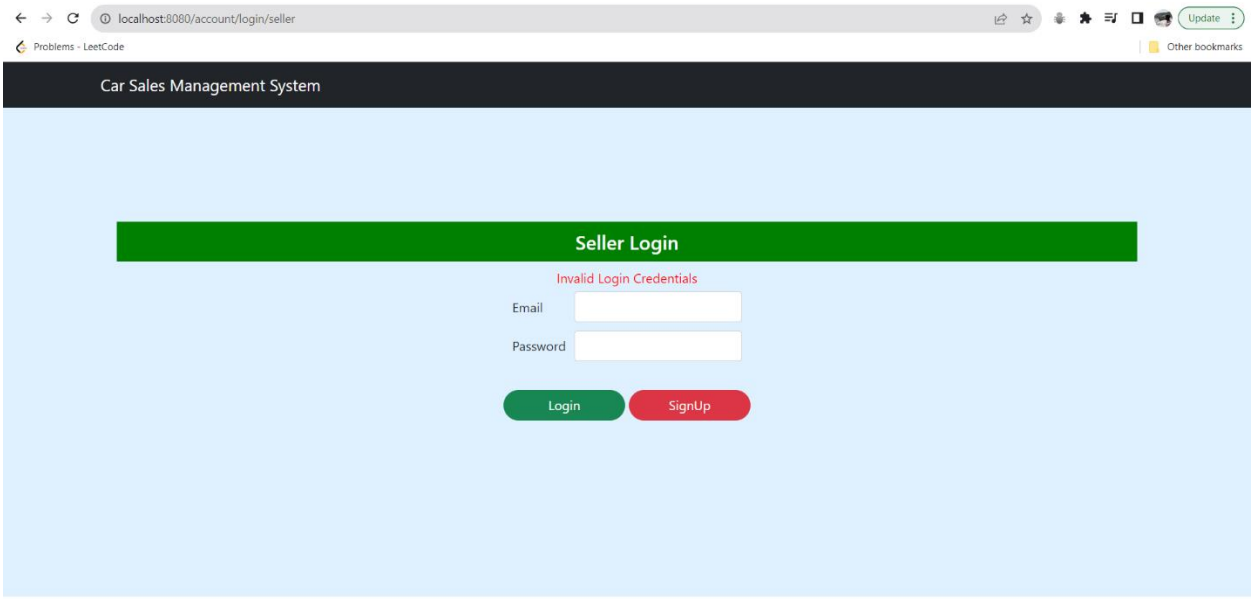

localhost:8080/account/login/sellerLogin
Update
Problems - LeetCode
Other bookmarks
Car Sales Management System
Seller Login
Email
Password
Login
SignUp

Signup


Car Sales Management System
Buyer Signup
Name
Address
Mobile
Email Id
Password
Confirm Password
Save
Login


localhost:8080/seller/create
Update
Problems - LeetCode
Other bookmarks
Car Sales Management System
New Seller Signup
Name
Address
Mobile
Email Id
Password
Confirm Password
Save
Login

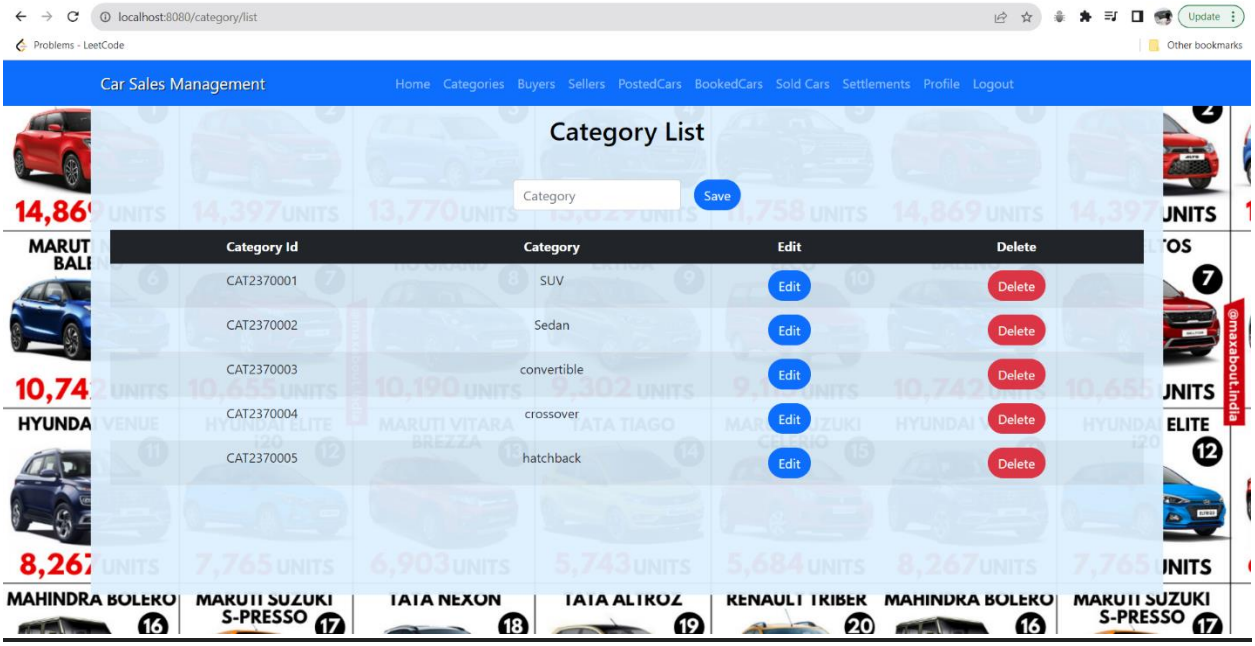
Validation: Checks the authorization credentials of the users.



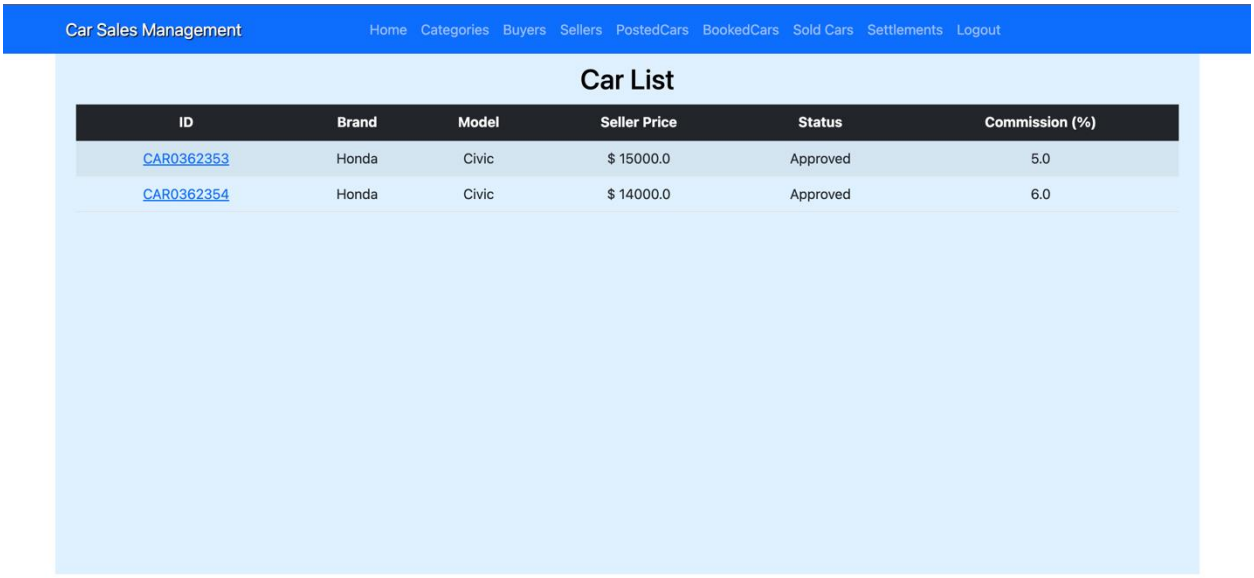
Admin Page

Admin page consists of the following:

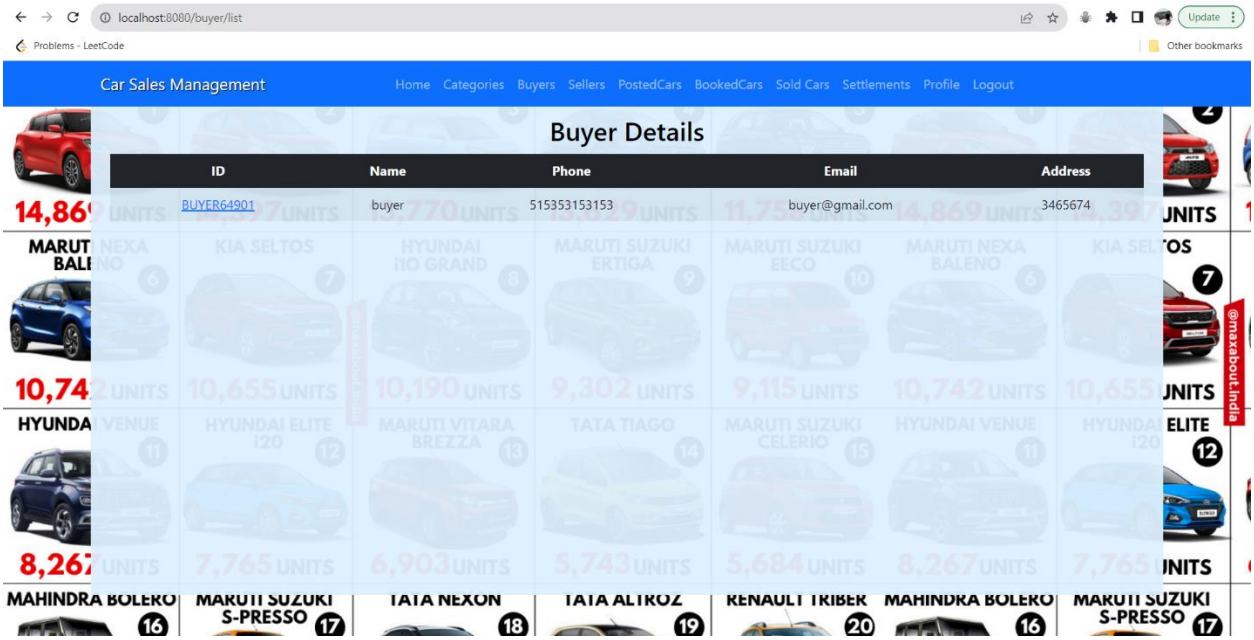
Categories: The Admin can add car categories such as SUV, sedan, sportscar etc.



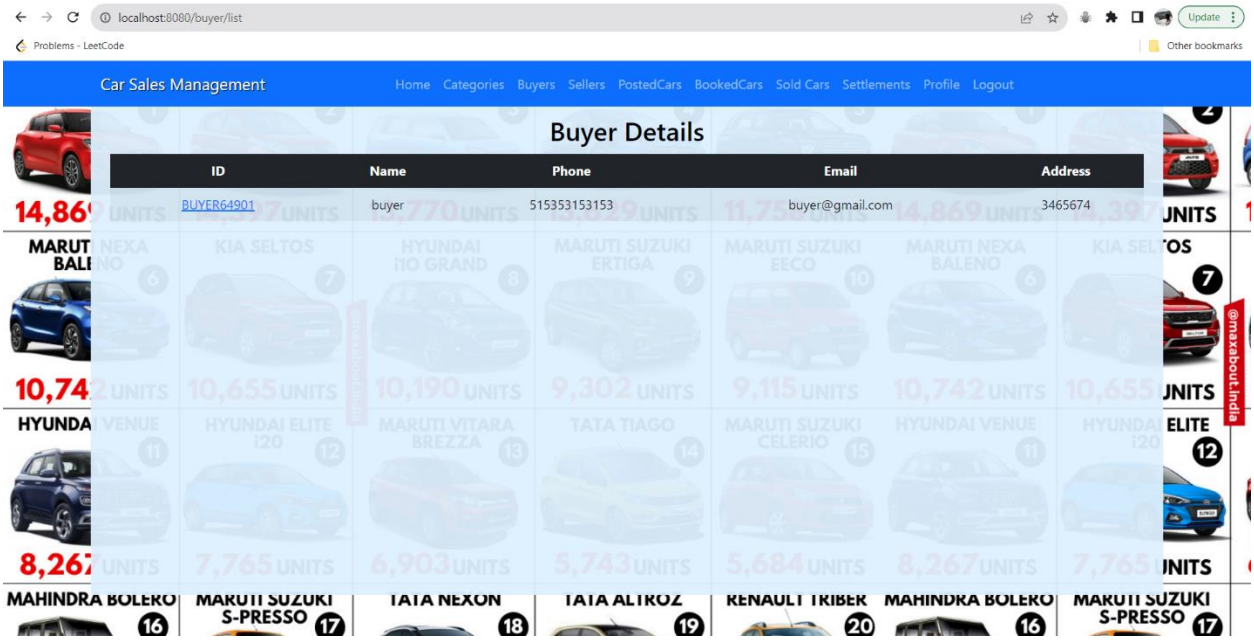
Posted Car List: List all the cars' details such as ID, brand, model, seller price, status, commission.



Buyers: Lists all the buyers' details such as buyerId, Name, Phone Number, Email, address.

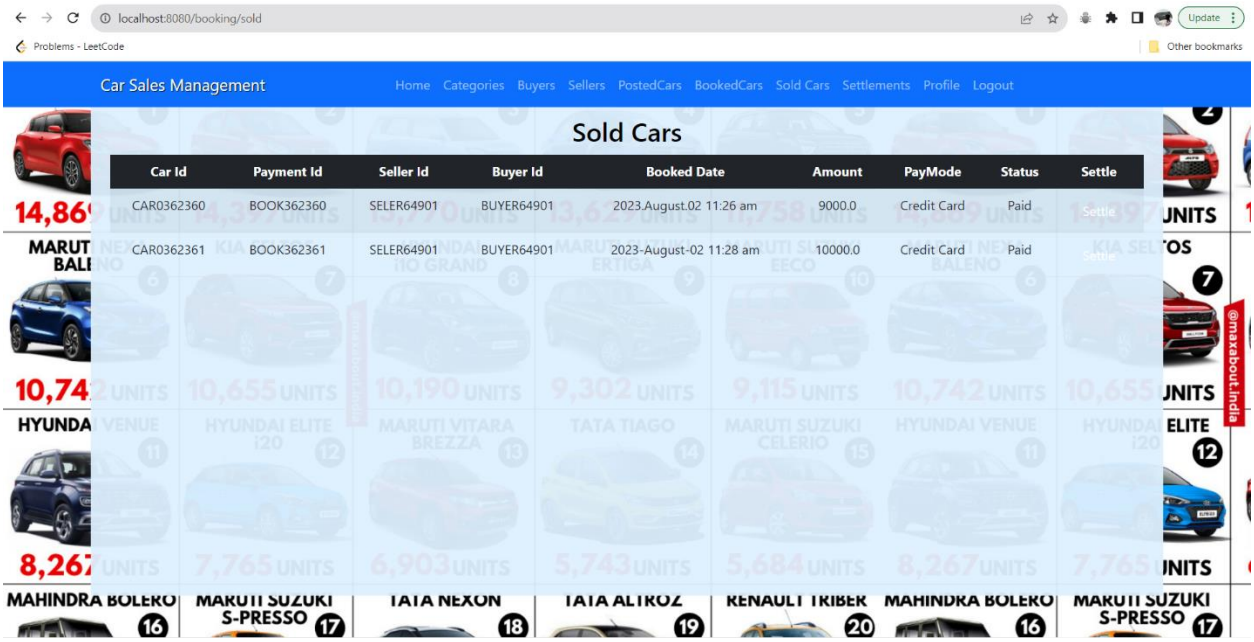


Sellers: Lists all the seller's details such as sellerId, Name, Phone Number, Email, address.

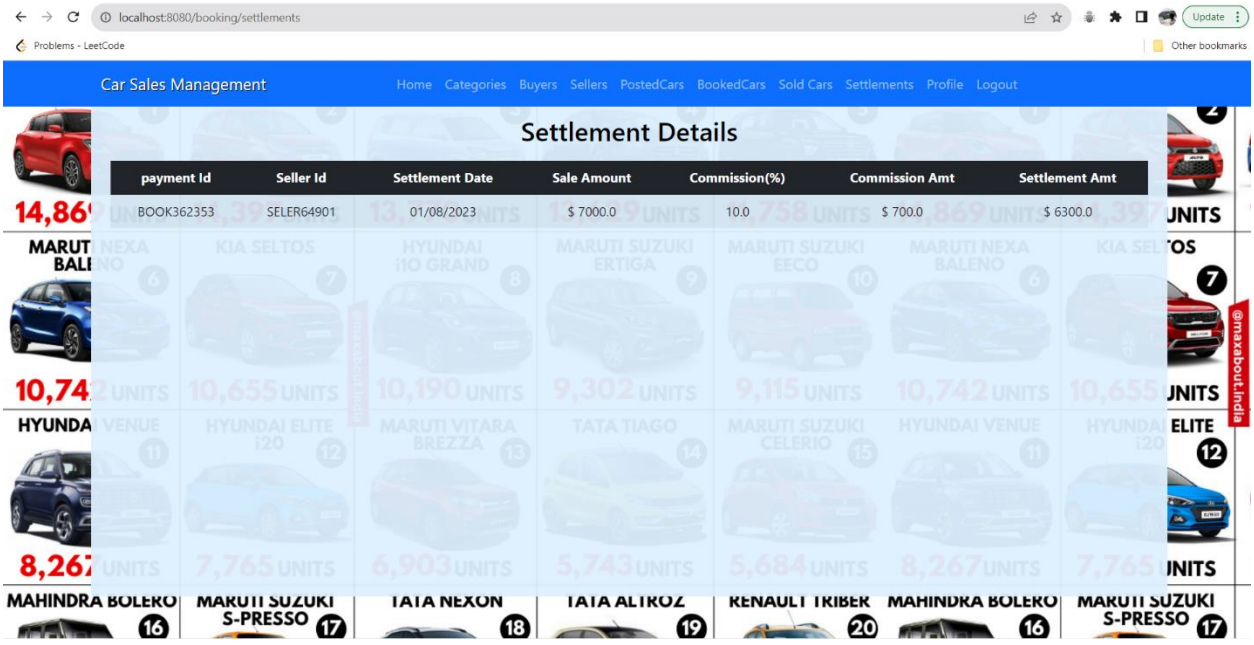


Booked cars: List all the cars' details such as carID, booking id, seller id, buyer id, booked date, total amount, booking amount, payment mode.

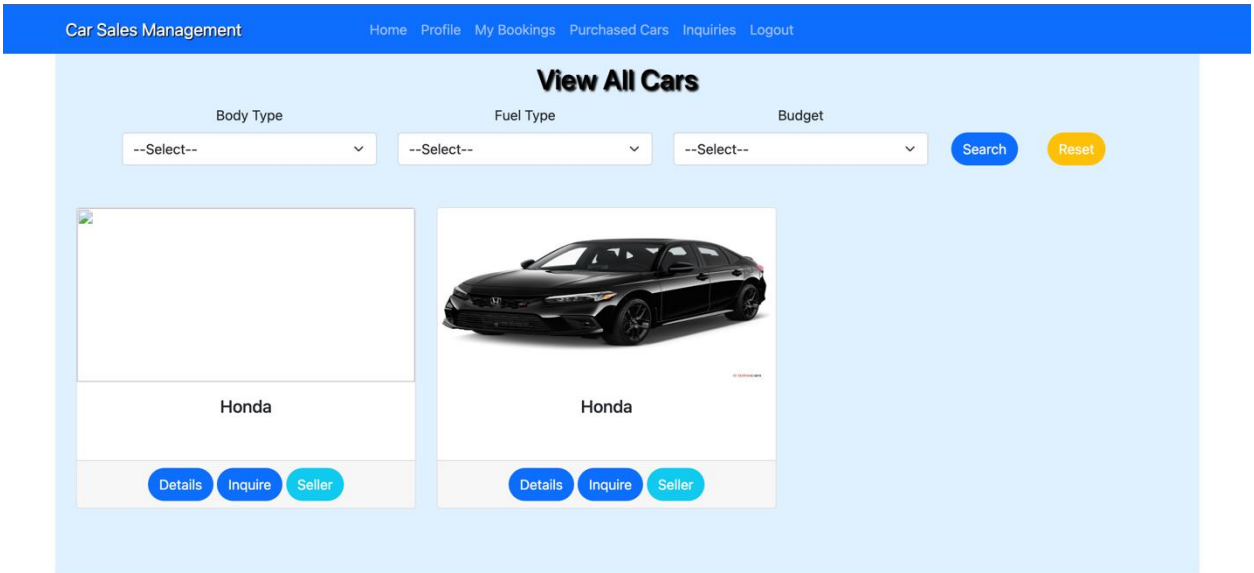
Sold cars: Lists all the sold cars details.



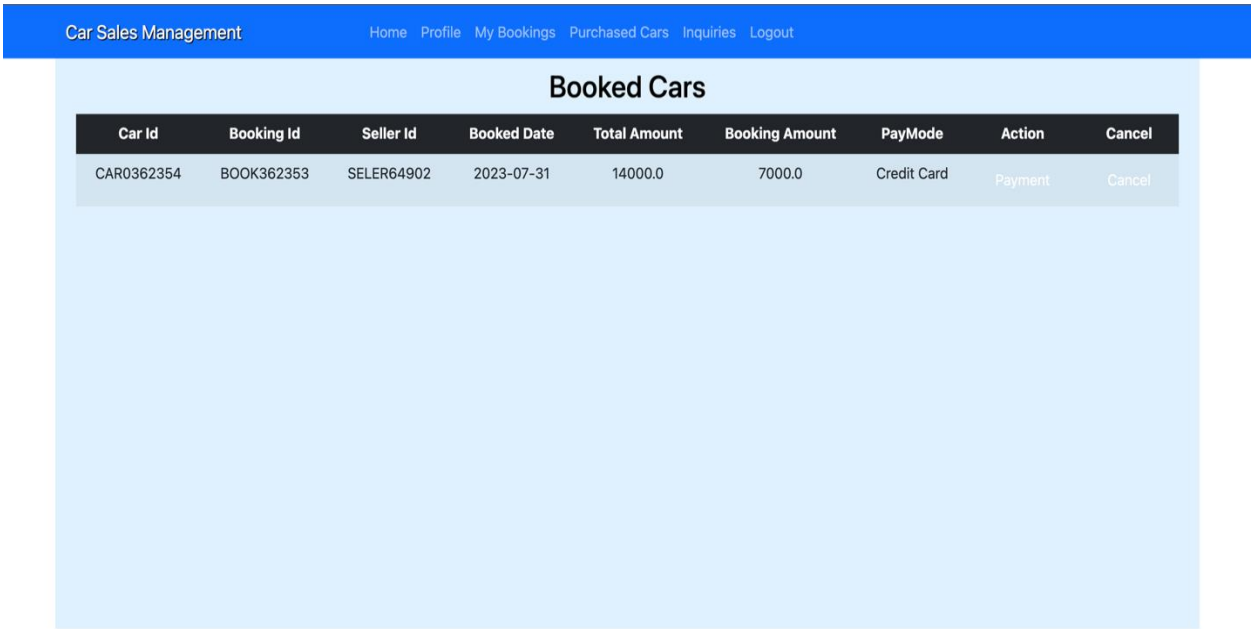
Settlements: Lists all the settlement details such as payment id, seller id, settlement id, settlement date, sale amount, commission percentage, commission percentage.



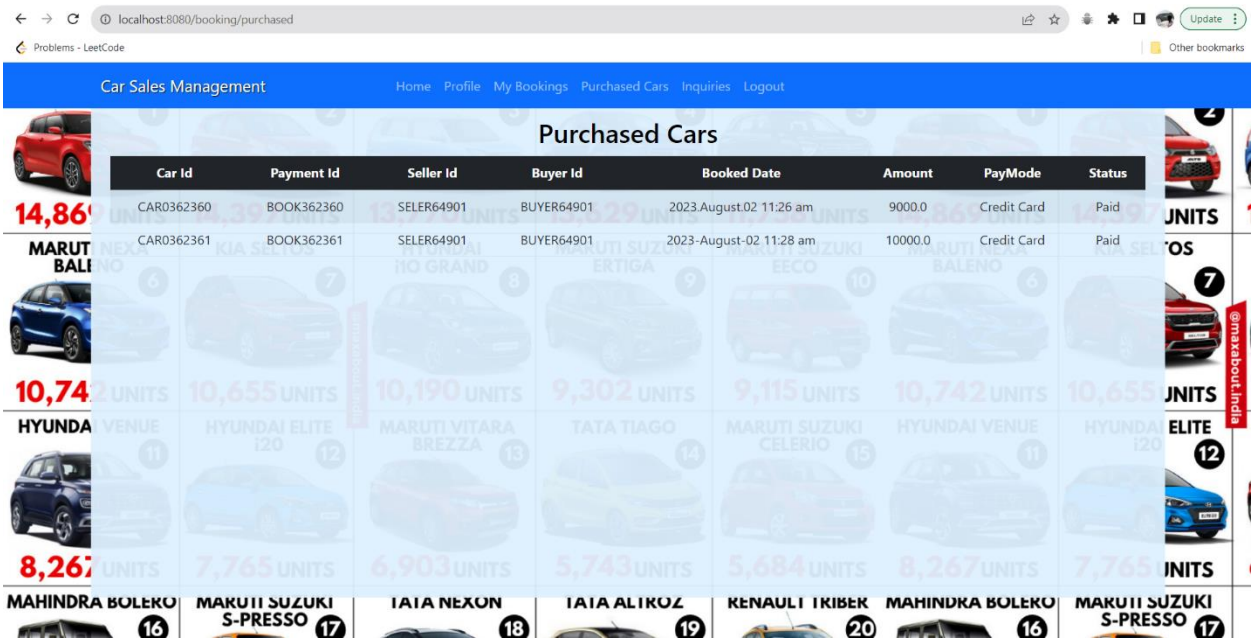
The Buyer page consists of the following:



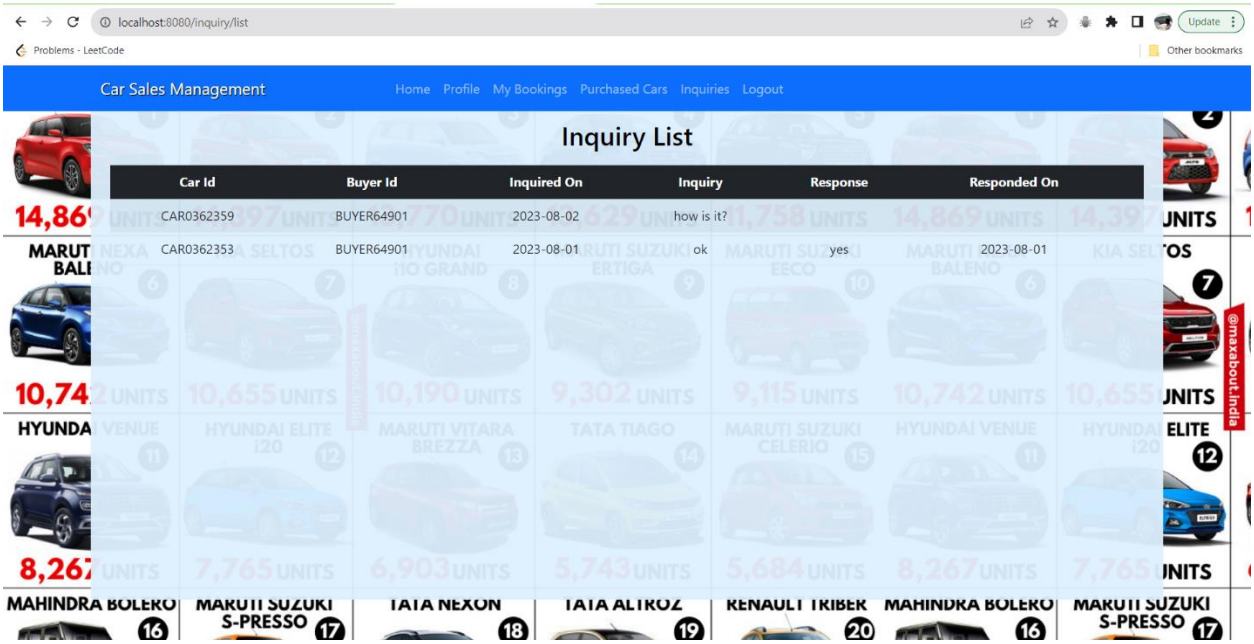
My Bookings: Lists all the bookings made by the buyer (Not paid total amount).



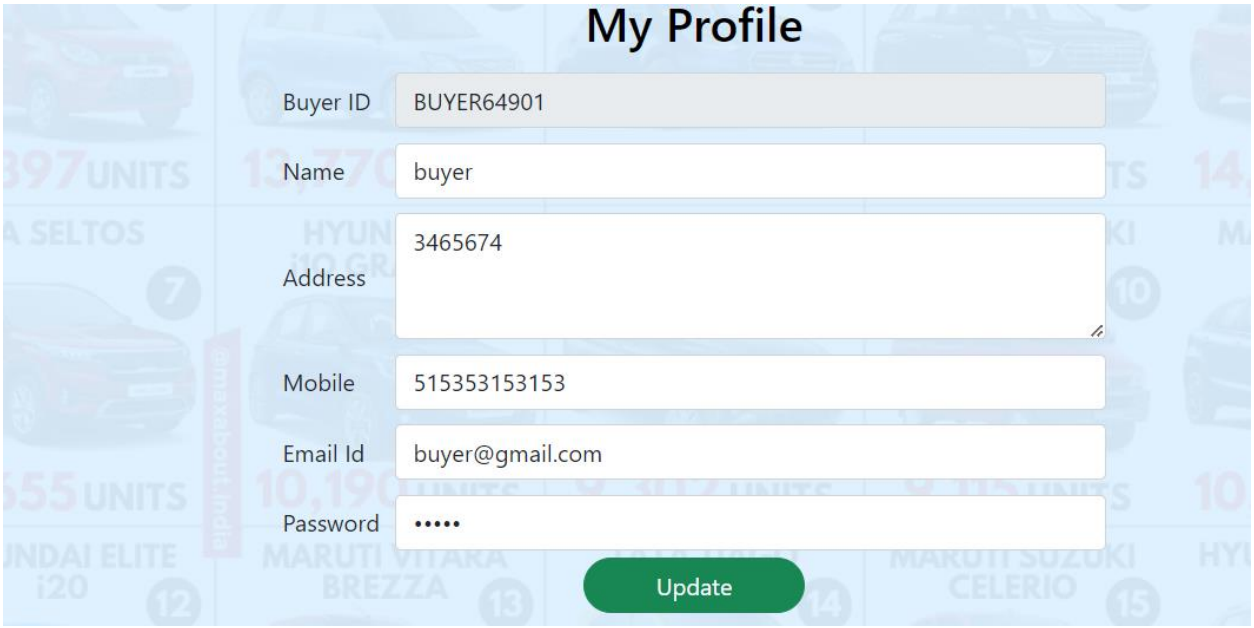
Purchased cars: List all the payment completed bookings.



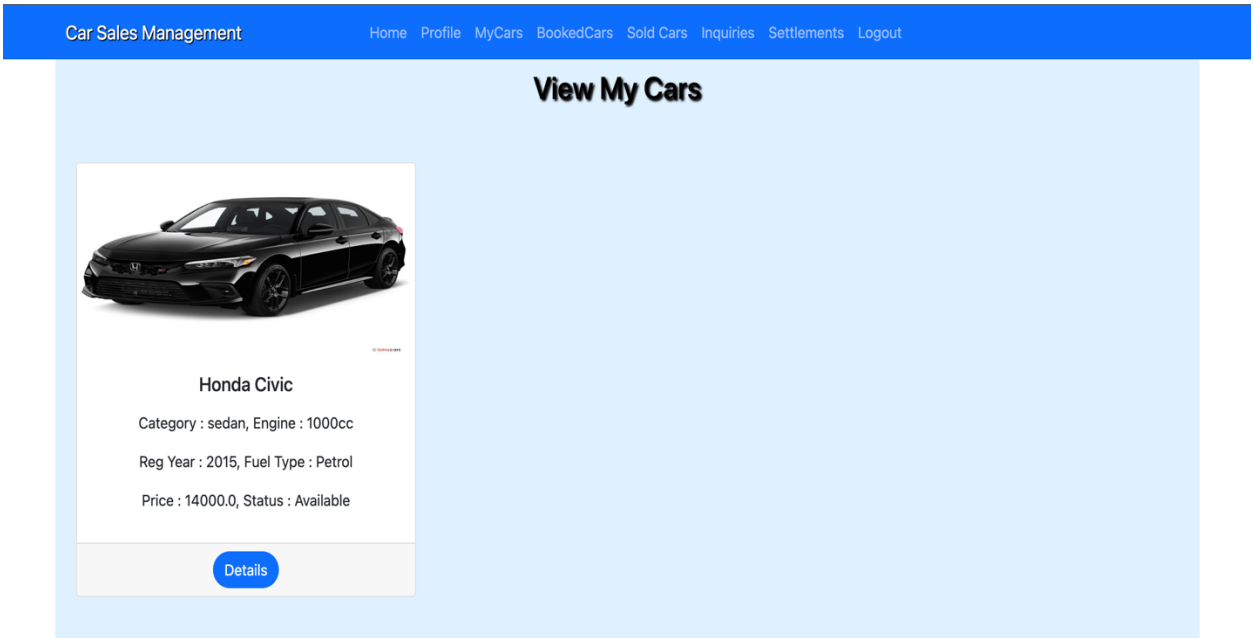
Inquiries: Can ask the seller for more details about the posted car.



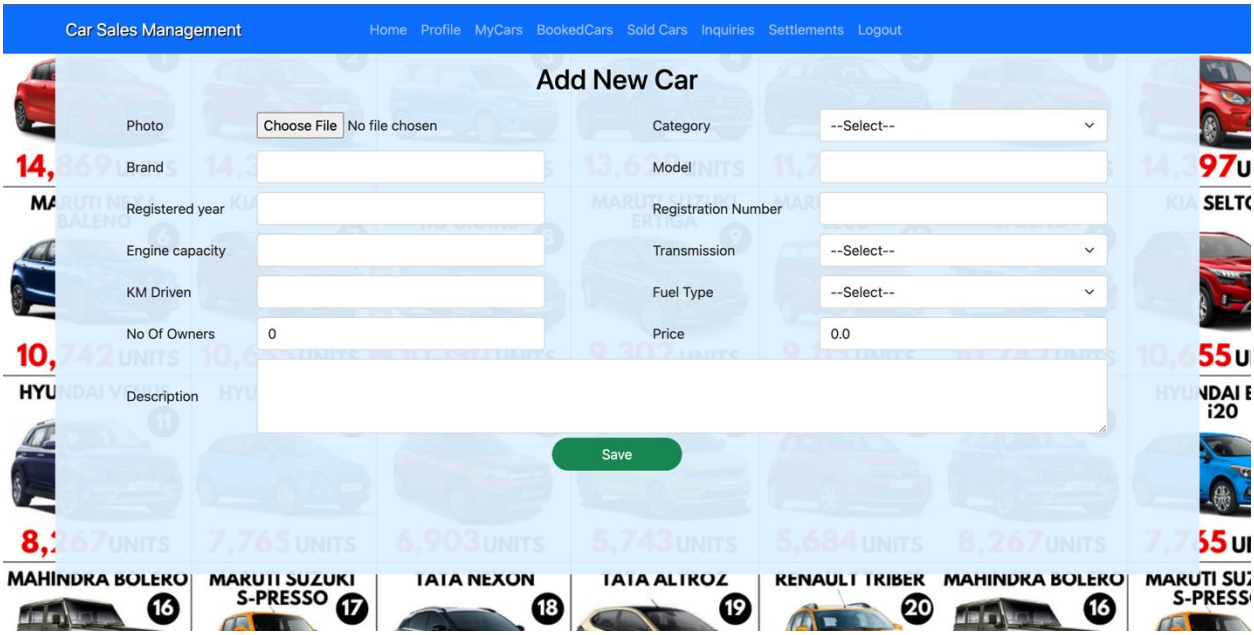
Profile: Can update his / her details.



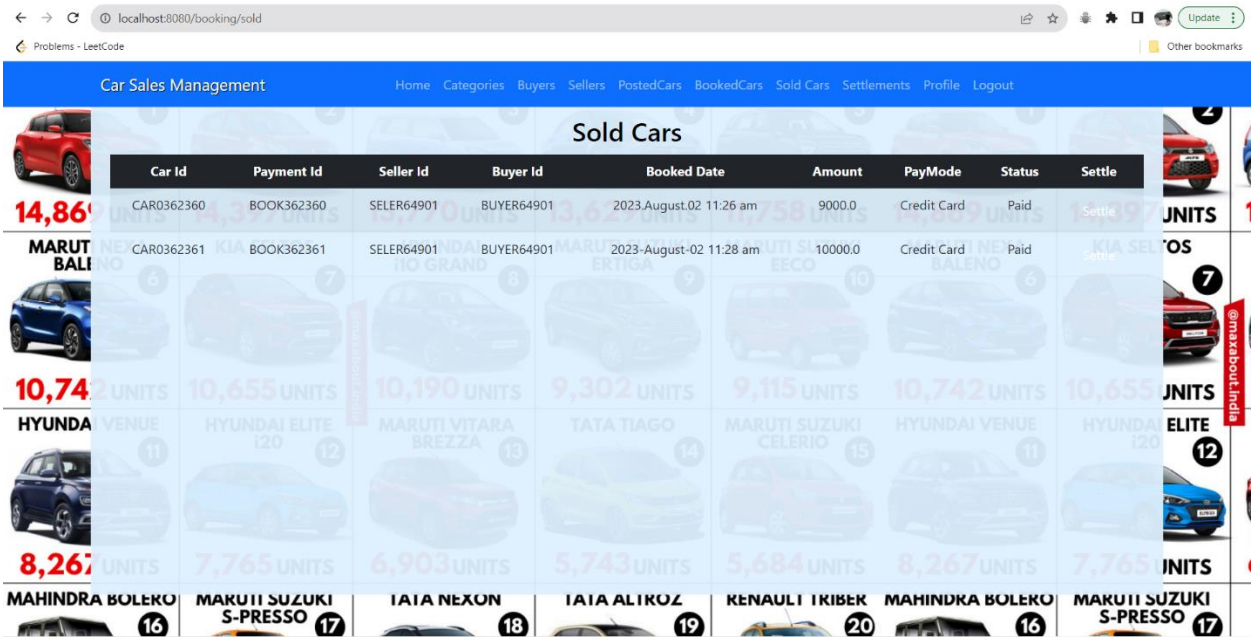
The Seller page consists of the following:



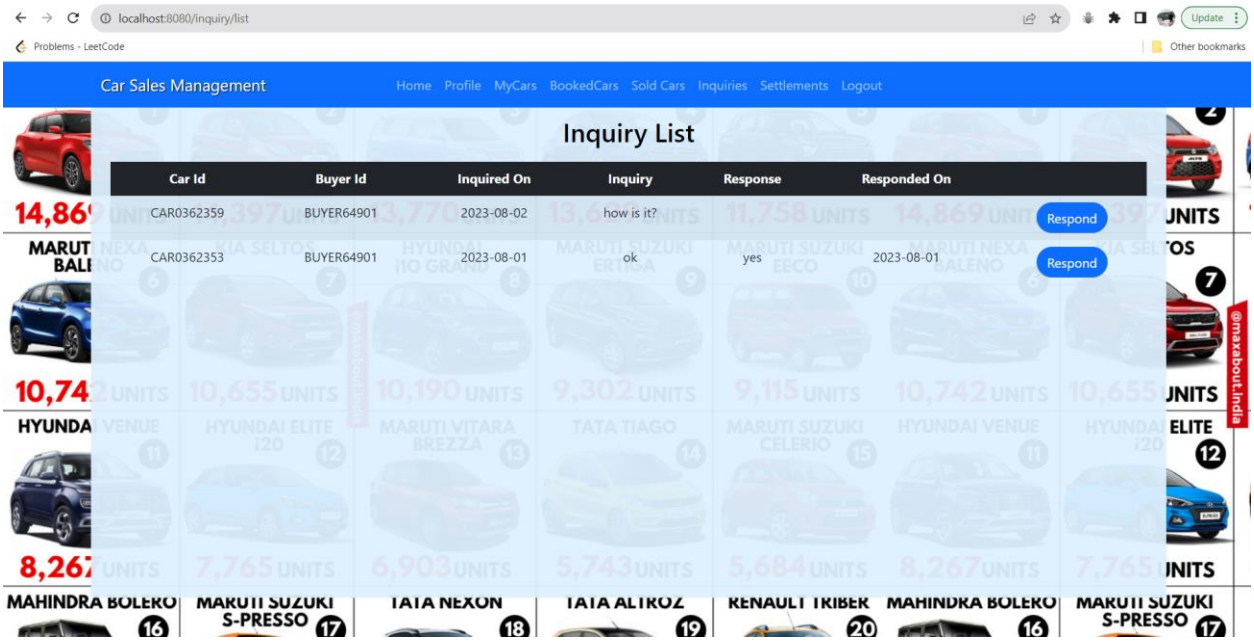
Mycars: Seller can add new sale post.



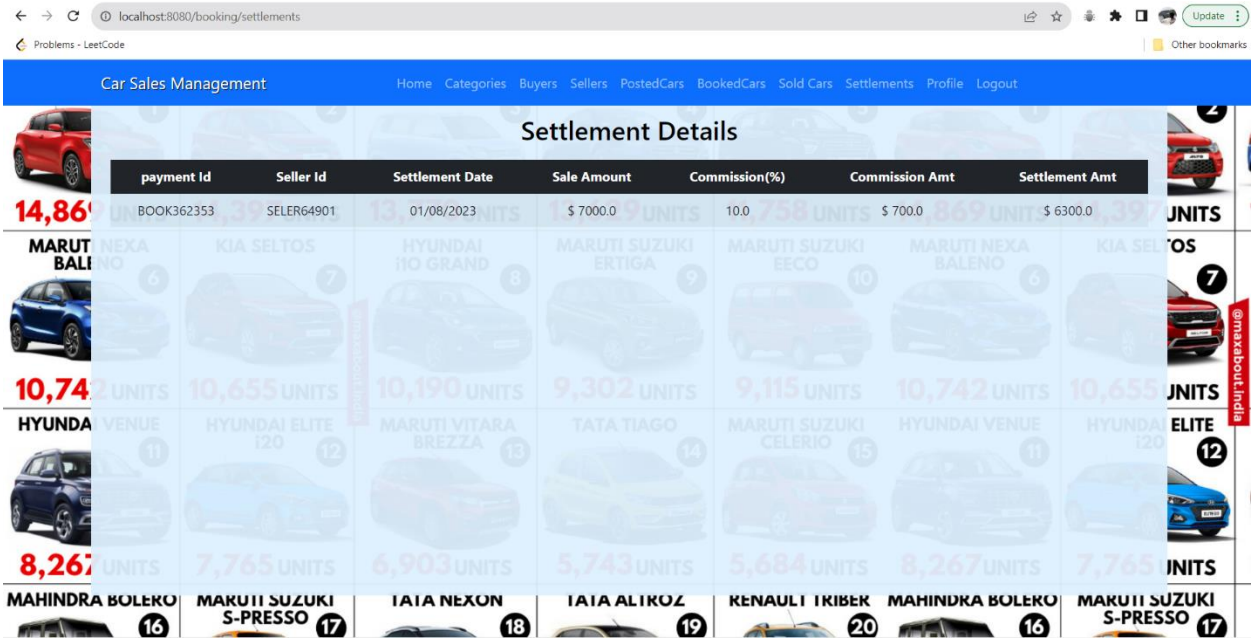
Sold cars: Lists all the sold cars.



Inquiry : The seller can respond to the buyer's inquiries.



Settlements: Same as Admin, the Seller can find all the settlement details here.



Source Code

```
@PostMapping("/login/admin")
public String adminLogin(Login login, Model model, HttpServletRequest
request) {
    if (login.getEmail().equals("admin") &&
login.getPassword().equals("admin")) {
        model.addAttribute("name", "Admin");
        request.getSession().setAttribute("id", "Admin");
        request.getSession().setAttribute("userid", "Admin");
        request.getSession().setAttribute("usertype", "admin");
        request.getSession().setAttribute("name", "Admin");
        return "redirect:/account/home";
    } else {
        model.addAttribute("adminmsg", "Invalid Login Credentials");
        return "login";
    }
}
```

This code checks the credentials of the user logged in. If the credentials match the page it redirects to the account home page, if not, then displays invalid Login credentials message. The same code will be similar to all the users(Buyer, Seller and Admin).

```
@RequestMapping("/list")
public String bookingList(Model model, HttpServletRequest req) {
    String utype = req.getSession().getAttribute("usertype").toString();
    String uid = req.getSession().getAttribute("userid").toString();
    if (utype.equals("buyer"))
        model.addAttribute("datalist",
payRepo.findAllByBuyerIdAndStatusEquals(uid, "Booked").get());
    else if (utype.equals("seller"))
        model.addAttribute("datalist",
payRepo.findAllBySellerIdAndStatusEquals(uid, "Booked").get());
    else
        model.addAttribute("datalist",
payRepo.findAllByStatus("Booked").get());

    return "booking_list";
}
```

This code displays a booking list based on the type of the user logged in.

The booking creates and update request calls the carRepo function and update the existing data.

```
public String saveBooking(Payment obj, HttpServletRequest req) throws
IOException {
    Optional<Payment> idobj = payRepo.findTopByOrderByIdDesc();
    String id = null;
    if (idobj.isPresent()) {
        int idnum =
Integer.parseInt(idobj.get().getPaymentId().substring(5));
```

```
        idnum++;
        id = "BOOK3" + idnum;
    } else {
        id = "BOOK362353";
    }
```

The above code saves the booking posted. The booking Id starts with the number 362353 initially and increases by one for every booking.

- All the model classes have variables, constructors, setters, and getters.
- All the repository classes extend the Spring function called CRUD that extends the respective model class.

```
• public static void saveFile(String uploadDir, String fileName,
            MultipartFile multipartFile) throws IOException {
        Path uploadPath = Paths.get(uploadDir);

        if (!Files.exists(uploadPath)) {
            Files.createDirectories(uploadPath);
        }

        try (InputStream inputStream = multipartFile.getInputStream()) {
            Path filePath = uploadPath.resolve(fileName);
            Files.copy(inputStream, filePath,
    StandardCopyOption.REPLACE_EXISTING);
        } catch (IOException ioe) {
            throw new IOException("Could not save image file: " + fileName,
    ioe);
        }
    }
```

The above code is used to upload car images. It uses the MultipartFile and Paths that are provided by the Spring.

```
spring.data.mongodb.uri=mongodb://localhost:27017/carsales

spring.autoconfigure.exclude=org.springframework.boot.autoconfigure.jdbc.Data
SourceAutoConfiguration
```

Database URL and using the spring autoconfigure feature.

All the HTML files are places under the static folder.