

Project Report Vehicle Registration System

Stephen Dias sdias3@gmu.edu (G01387625)
Aditya Milind Limbekar alimbeka@gmu.edu (G01384408)

Link to source code: https://github.com/STEPHENDIAS10/INFS740_Project_vehicle-registration-system

• Introduction

The **Vehicle Registration System** is a comprehensive solution designed to manage information related to vehicles, user details, and reviews. The system provides users with the ability to perform Create, Read, Update, and Delete (CRUD) operations on vehicles, reviews, and user details. This report outlines the key features, components, and functionalities of the system.

• System Overview

The Vehicle Registration System consists of three main modules:

- a. **Vehicle Management:** - Allows users to register new vehicles (cars/bikes) by providing necessary details such as make, model, year, and registration information. - Provides the ability to view a list of registered vehicles. - Allows users to update the information of existing vehicles or delete them from the system.
- b. **Review Management:** - Enables users to submit reviews for registered vehicles, including ratings and comments. - Allows users to view and manage their own reviews. - Provides administrators with the ability to moderate and manage all reviews in the system.
- c. **User Management:** - Allows users to create accounts by providing personal information. - Provides the ability to view and update user profiles. - Administrators can manage user accounts, including creating, updating, and deleting user profiles.

• CRUD Operations

a. Vehicle CRUD Operations:

Create: Users can add new vehicles to the system by providing relevant information.

Read: Users can view a list of all registered vehicles along with their details.

Update: Users can modify the information of existing vehicles.

Delete: Users can remove vehicles from the system.

INFS 740

Project Report Vehicle Registration System

→ Add a vehicle

The screenshot shows a web browser at localhost:3000/add-car. A dark blue header contains the 'Vehicle Booking' logo and navigation links: 'My Bookings', 'All Bookings', 'Settings', and 'Log Out'. A toast notification from 'localhost:3000' says 'Vehicle added' with an 'OK' button. The main form, titled 'Enter Car/Bike details...', includes input fields for 'Brand' (Harley Davidson), 'Model' (Street Bob 114), 'Year' (2017), 'Price' (800), and 'Image URL' (https://cdn1.cycletrader.com/vt/media/654dda0fd1408b705d4ef44a.jpg?width=400&height=31). An 'Add Vehicle' button is at the bottom.

→ View/Read a vehicle

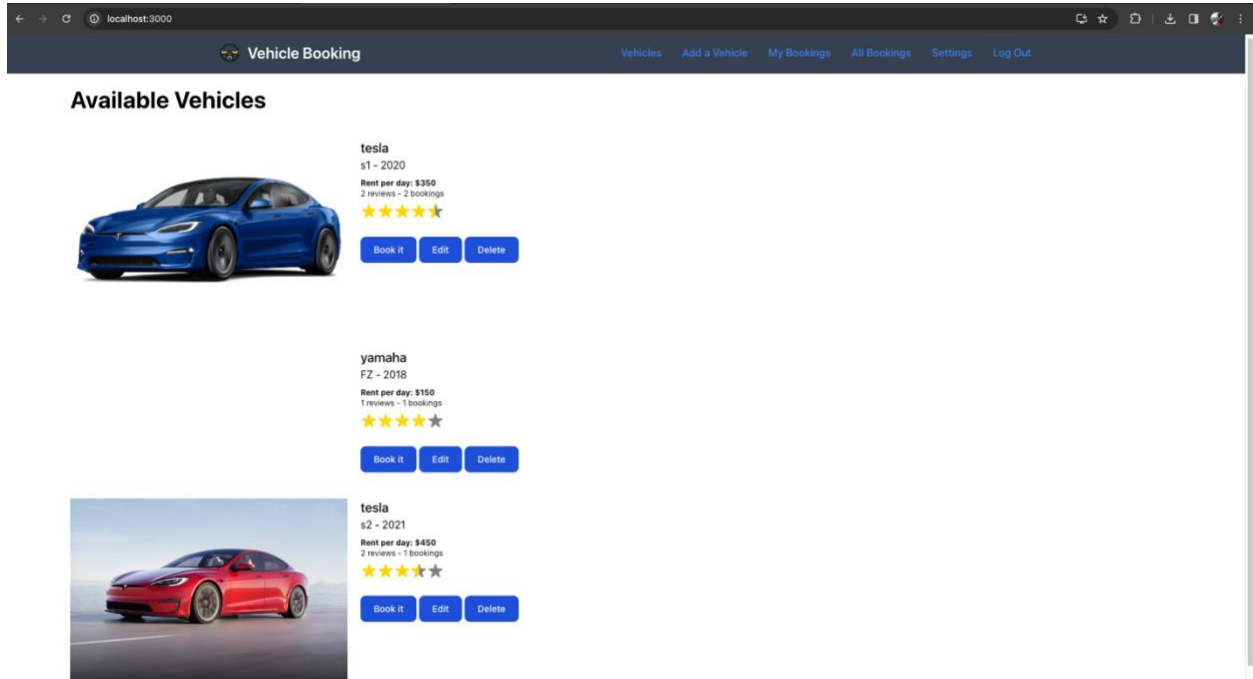
The screenshot shows the 'Available Vehicles' page. The header is identical to the previous page. The page lists three vehicles:

- Harley Davidson Street Bob 114 - 2017**: Rent per day: \$800, 0 reviews - 0 bookings, 5 empty star icons. Buttons: Book it, Edit, Delete.
- tesla s1 - 2020**: Rent per day: \$380, 2 reviews - 2 bookings, 5 half-filled star icons. Buttons: Book it, Edit, Delete.
- yamaha FZ - 2018**: Rent per day: \$150, 1 review - 1 bookings, 5 half-filled star icons. Buttons: Book it, Edit, Delete.

INFS 740

Project Report Vehicle Registration System

→ Update a vehicle



The screenshot displays the 'Enter Car/Bike details...' form in the 'Vehicle Booking' web application. The form fields are as follows:

- Brand**: yamaha
- Model**: FZ
- Year**: 2018
- Price**: 150
- Image URL**: https://www.yamaha-motor-india.com/theme/v3/image/webp/fz25bs6/color/color_01_fz.webp

The 'Update Vehicle' button is located at the bottom of the form.

INFS 740

Project Report Vehicle Registration System

localhost:3000

Vehicle Booking

localhost:3000 says
Vehicle updated

My Bookings All Bookings Settings Log Out

Enter Car/Bike details...

Brand
yamaha

Model
FZ

Year
2018

Price
150

Image URL
https://www.yamaha-motor-india.com/theme/v3/image/webp/fz25bs6/color/color_01_fz.webp


Update Vehicle

localhost:3000


Vehicle Booking

Vehicles Add a Vehicle My Bookings All Bookings Settings Log Out

Available Vehicles



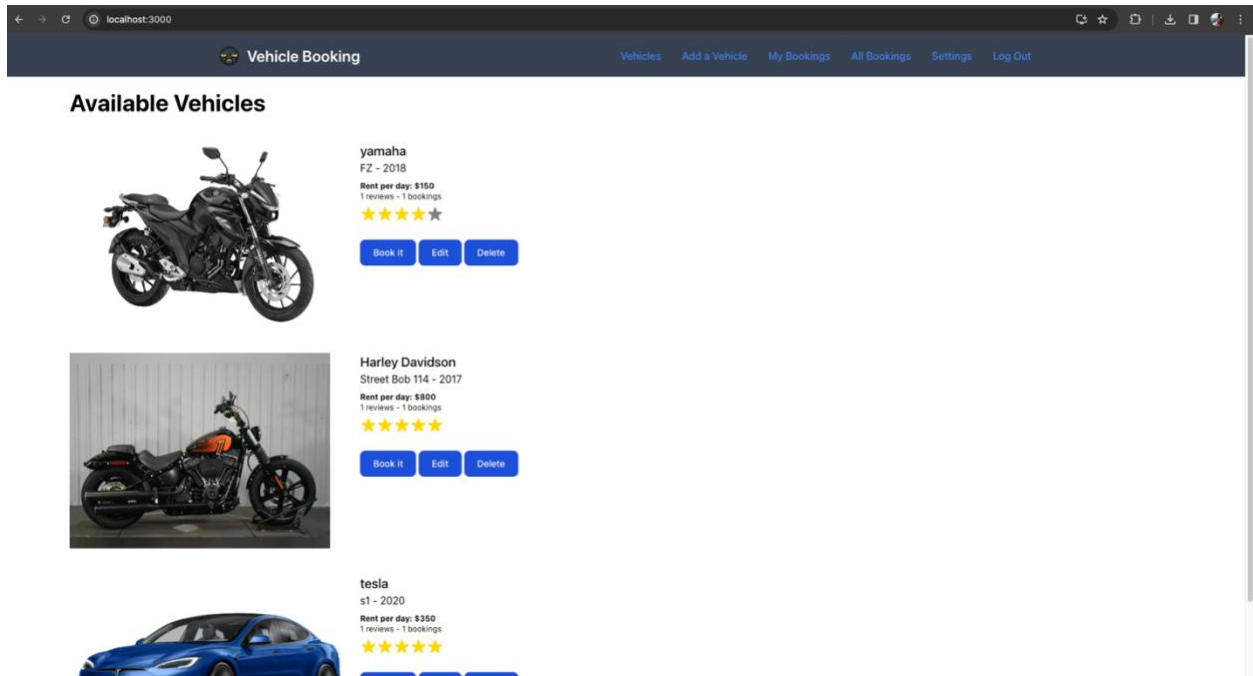
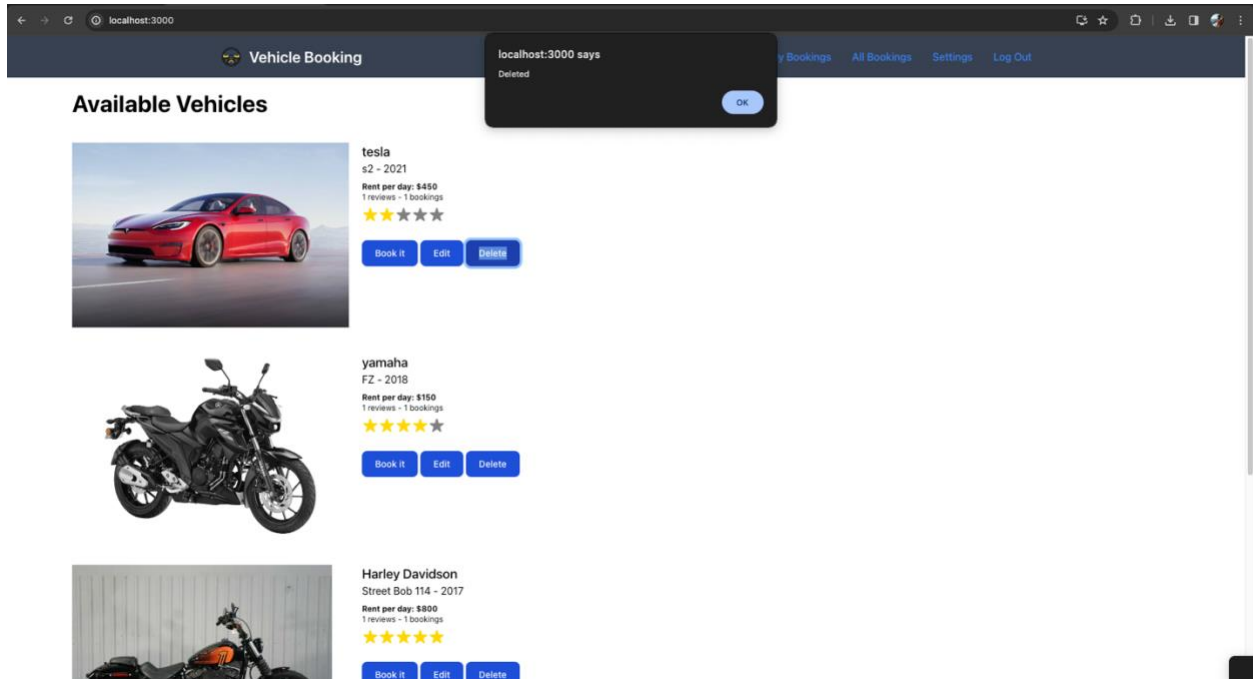
tesla
s2 - 2021
Rent per day: \$450
1 reviews - 1 bookings
★★★★★
Book It Edit Delete



yamaha
FZ - 2018
Rent per day: \$150
1 reviews - 1 bookings
★★★★★
Book It Edit Delete

Project Report Vehicle Registration System

→ Delete a vehicle



Project Report Vehicle Registration System

b. Review CRUD Operations:

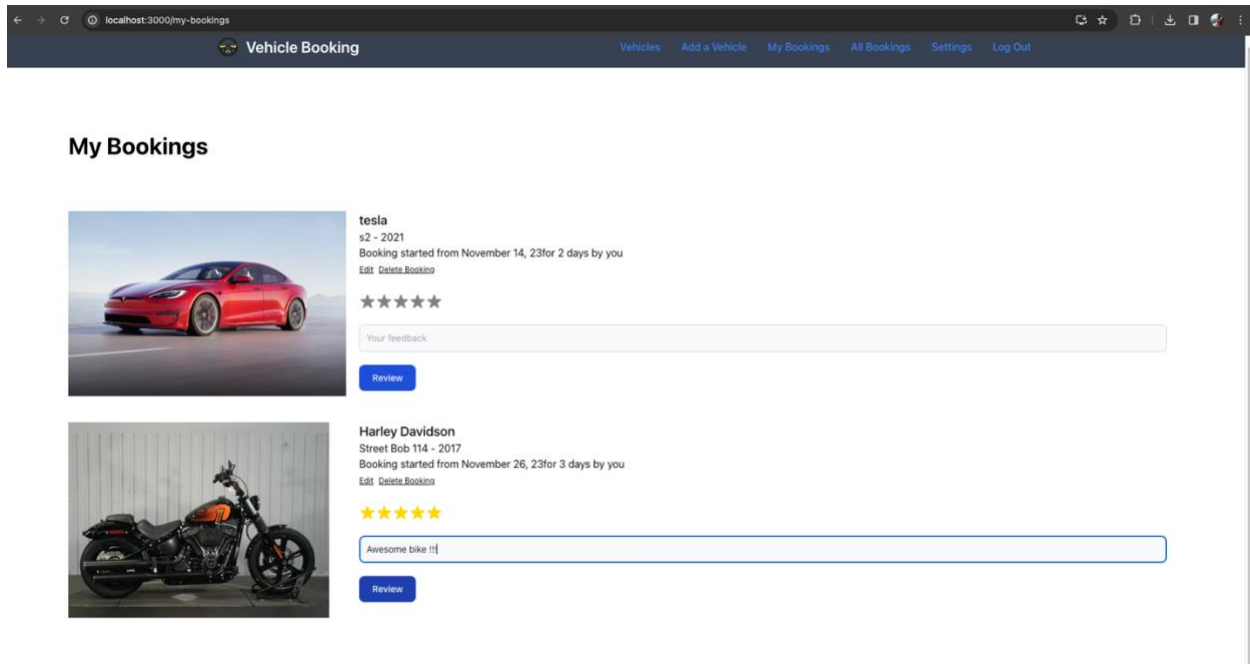
Create: Users can submit reviews for specific vehicles.

Read: Users can view their own reviews and administrators can view all reviews.

Update: Users can edit their own reviews.

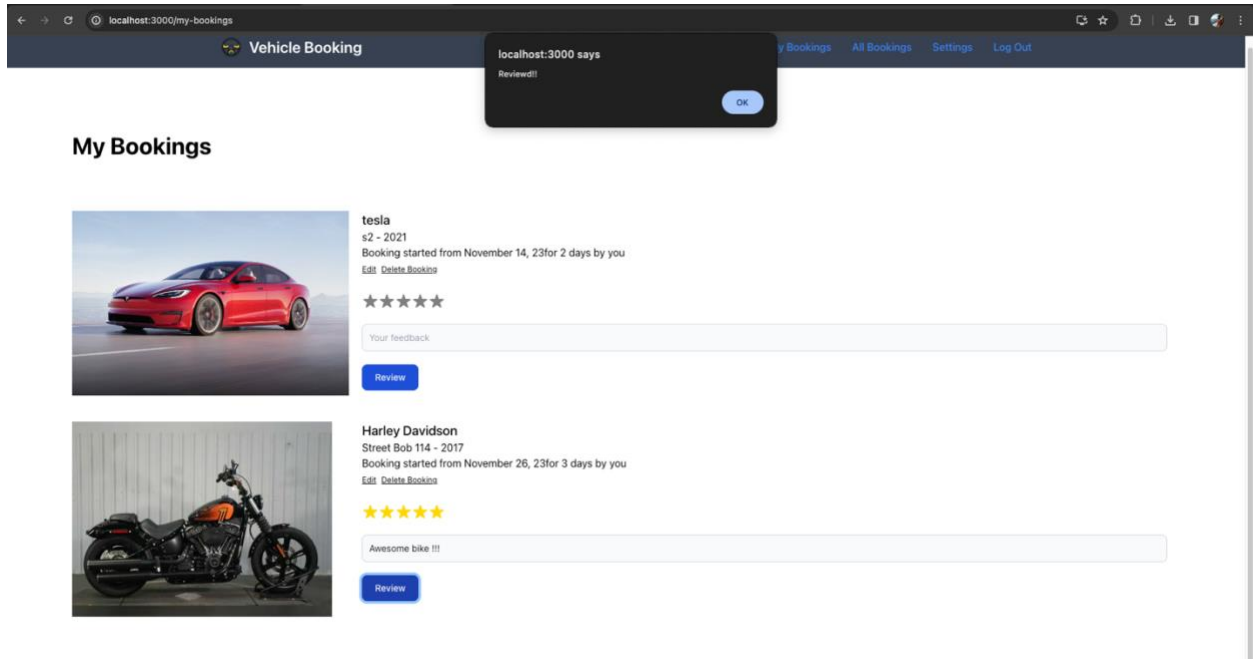
Delete: Users can delete their own reviews, and administrators can delete any review for moderation purposes.

→ **Add a review**

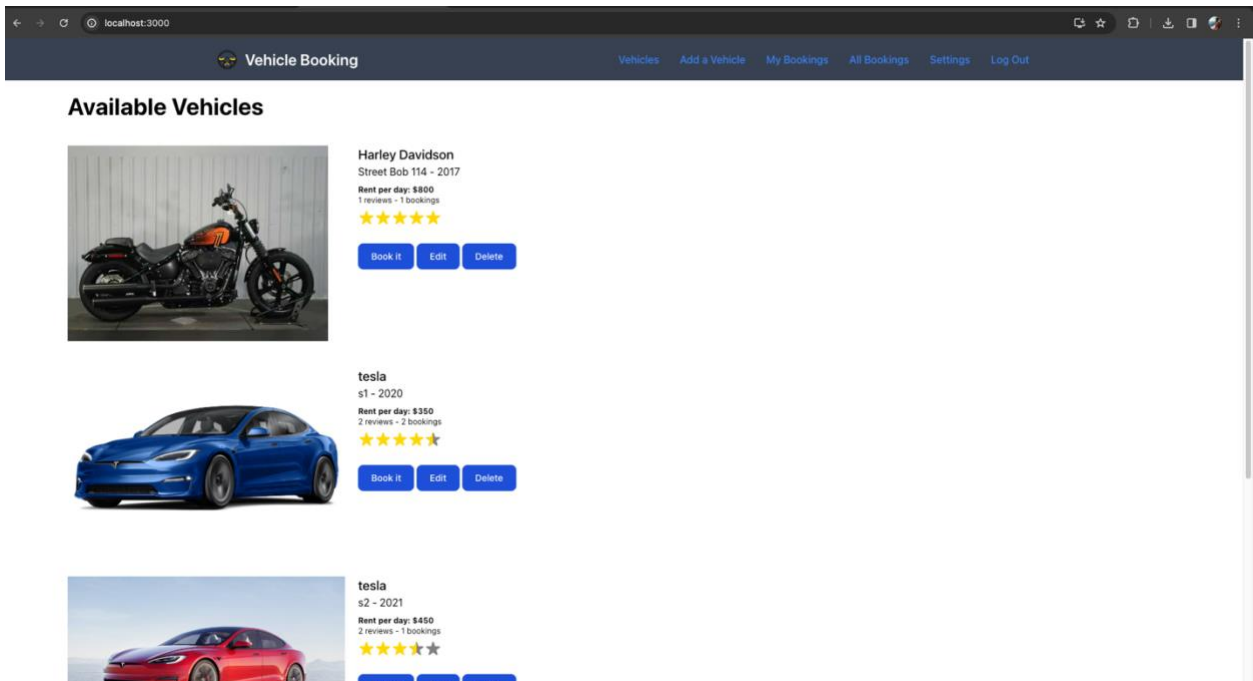


INFS 740

Project Report Vehicle Registration System



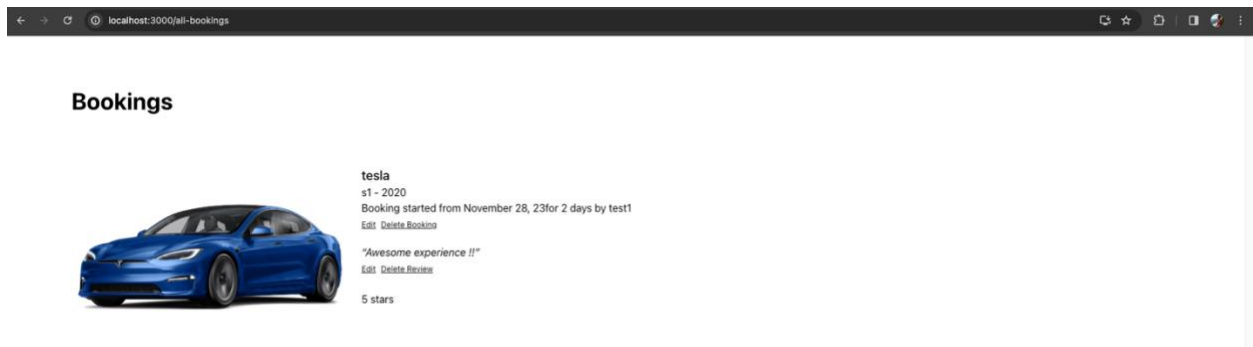
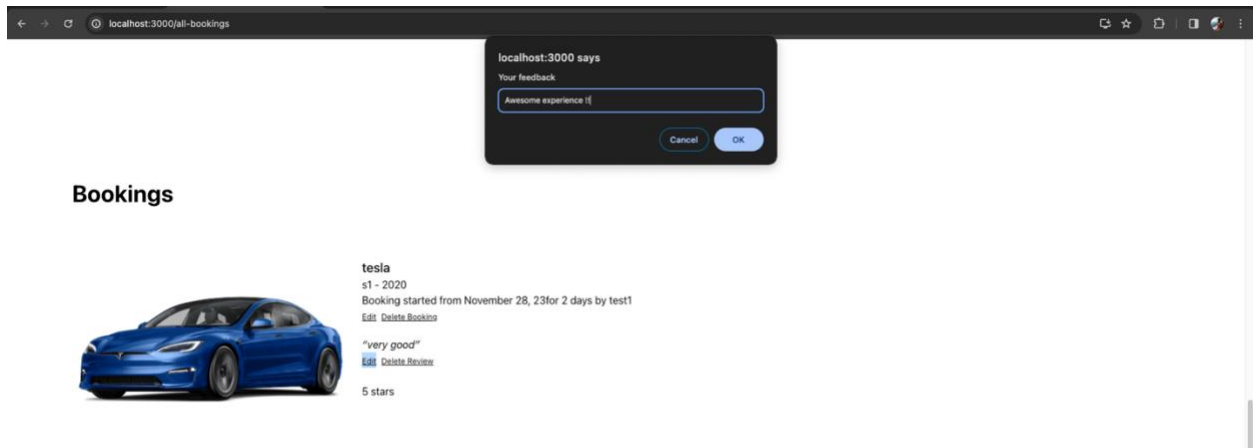
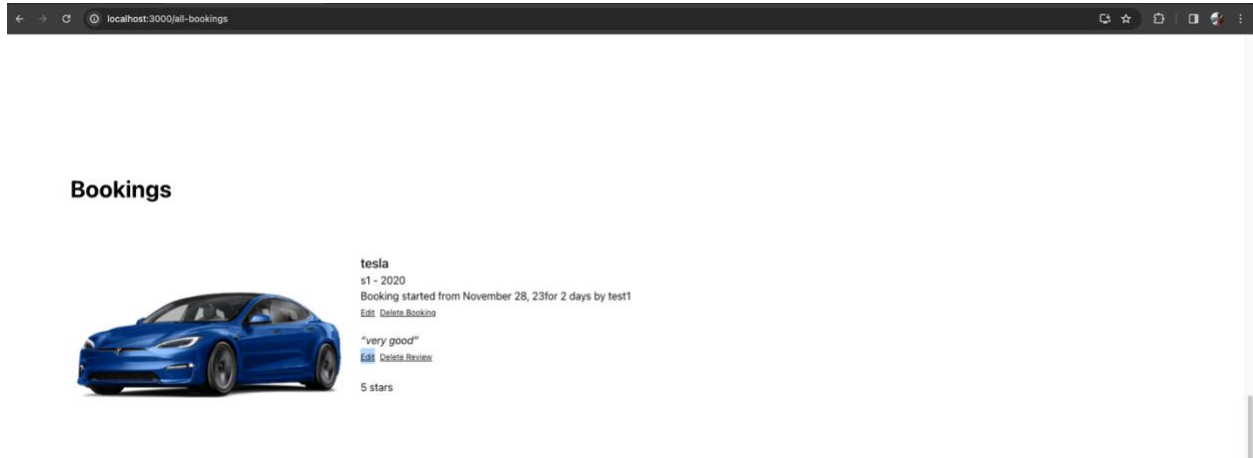
→ View/Read a review



INFS 740

Project Report Vehicle Registration System

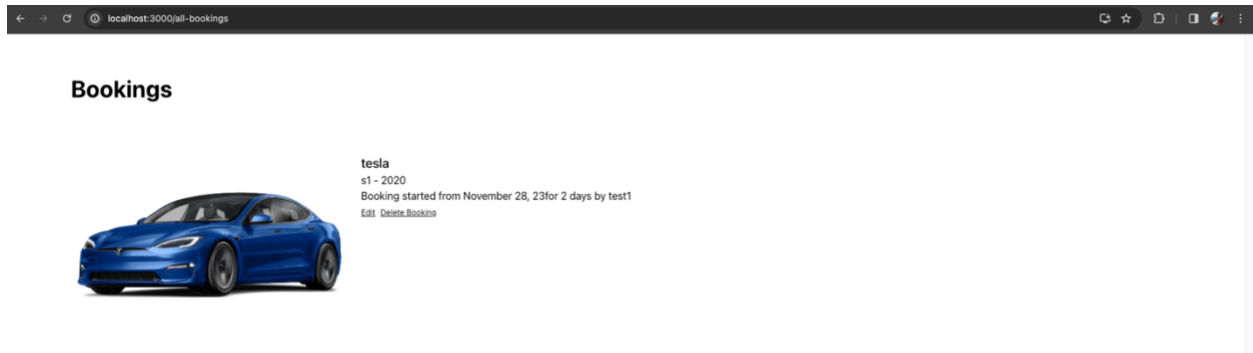
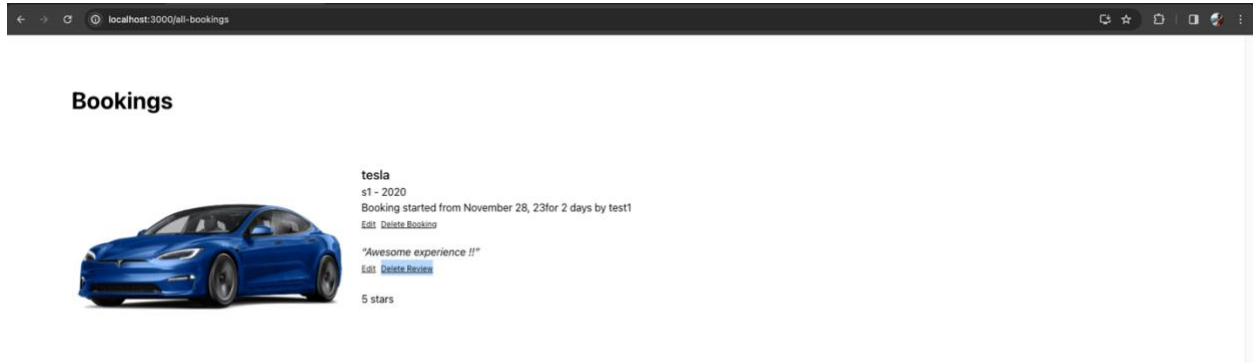
→ Update a review



INFS 740

Project Report Vehicle Registration System

→ Delete a review



c. User CRUD Operations:

Create: Users can create accounts by providing personal information.

Read: Users can view their own profiles, and administrators can view all user profiles.

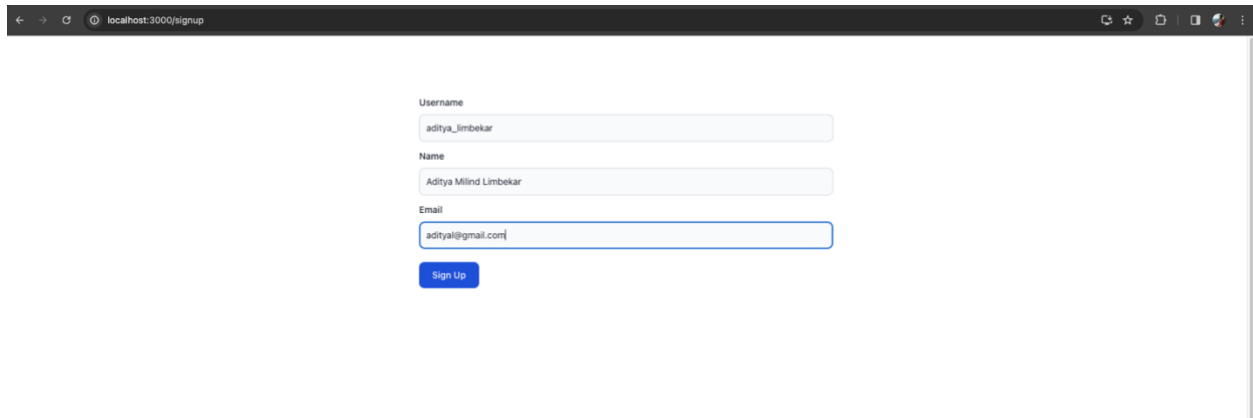
Update: Users can update their own profiles, and administrators can update any user profile.

Delete: Users can delete their own accounts, and administrators can delete any user account.

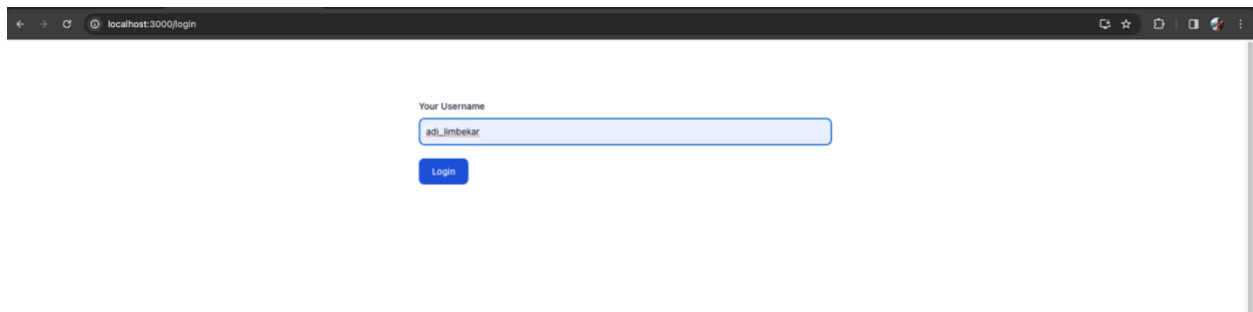
INFS 740

Project Report Vehicle Registration System

→ Add a user

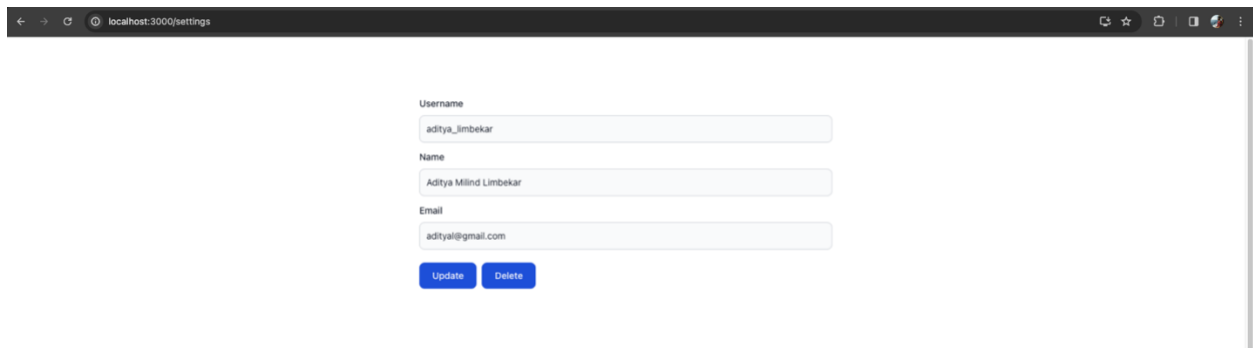


A screenshot of a web browser window showing a signup form. The browser's address bar displays 'localhost:3000/signup'. The form contains three input fields: 'Username' with the value 'aditya_limbekar', 'Name' with the value 'Aditya Milind Limbekar', and 'Email' with the value 'adityai@gmail.com'. Below these fields is a blue 'Sign Up' button.



A screenshot of a web browser window showing a login form. The browser's address bar displays 'localhost:3000/login'. The form contains one input field labeled 'Your Username' with the value 'ad_limbekar'. Below the field is a blue 'Login' button.

→ View/Read a user

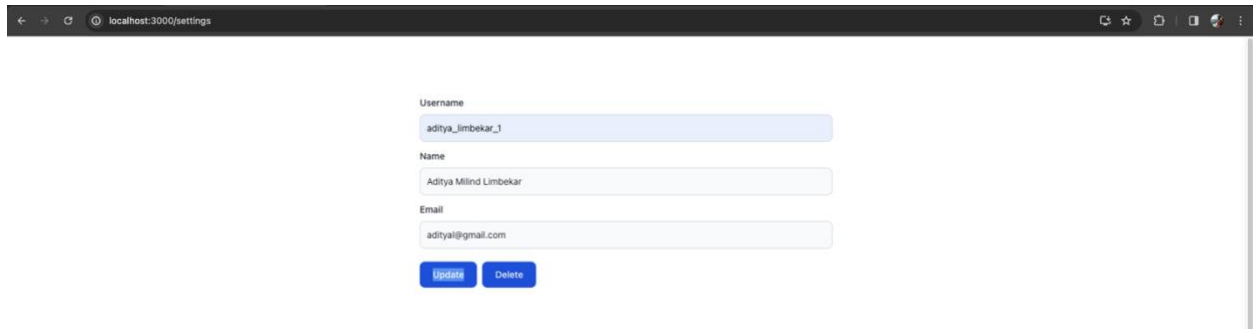


A screenshot of a web browser window showing a user settings or edit profile page. The browser's address bar displays 'localhost:3000/settings'. The page shows the same three input fields as the signup page: 'Username' (aditya_limbekar), 'Name' (Aditya Milind Limbekar), and 'Email' (adityai@gmail.com). Below these fields are two blue buttons: 'Update' and 'Delete'.

INFS 740

Project Report Vehicle Registration System

→ Update a user



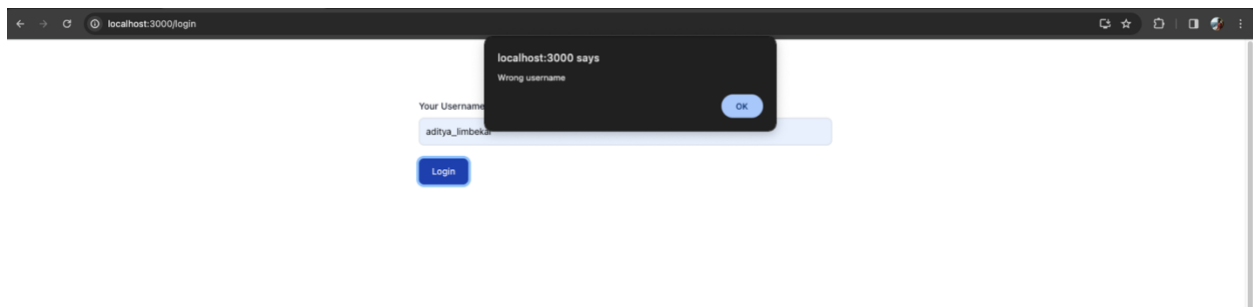
localhost:3000/settings

Username
aditya_limbekar_1

Name
Aditya Milind Limbekar

Email
adityai@gmail.com

Update Delete



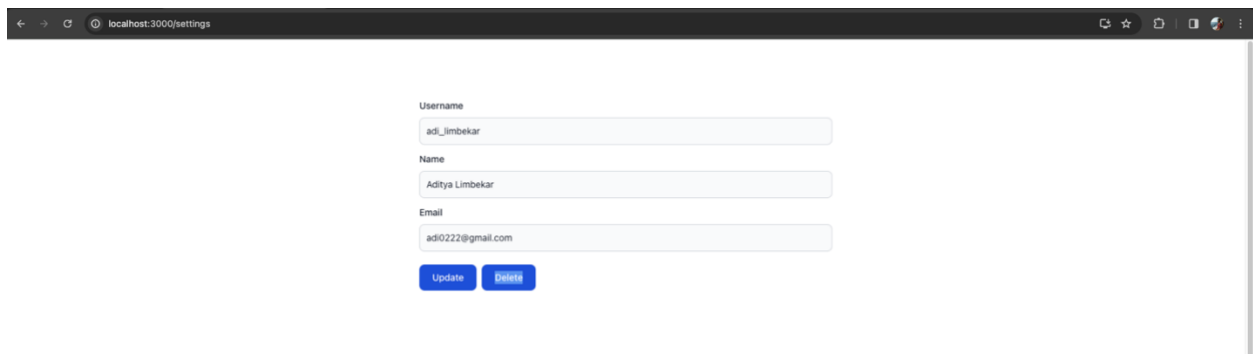
localhost:3000/login

localhost:3000 says
Wrong username

Your Username
aditya_limbekar_1

Login

→ Delete a user



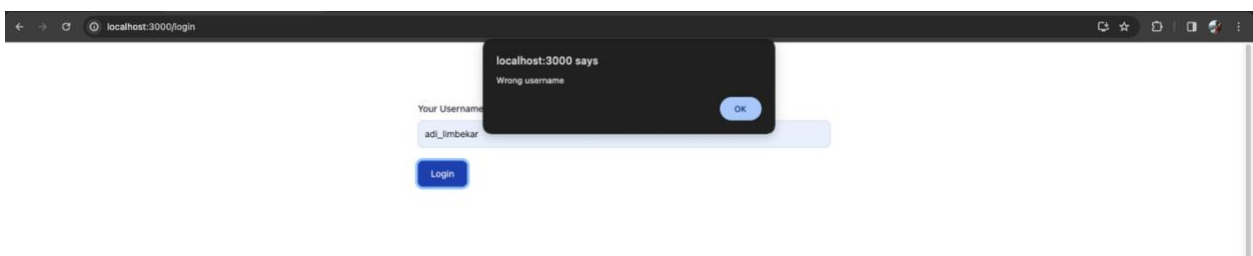
localhost:3000/settings

Username
ad_limbekar

Name
Aditya Limbekar

Email
adi0222@gmail.com

Update Delete



localhost:3000/login

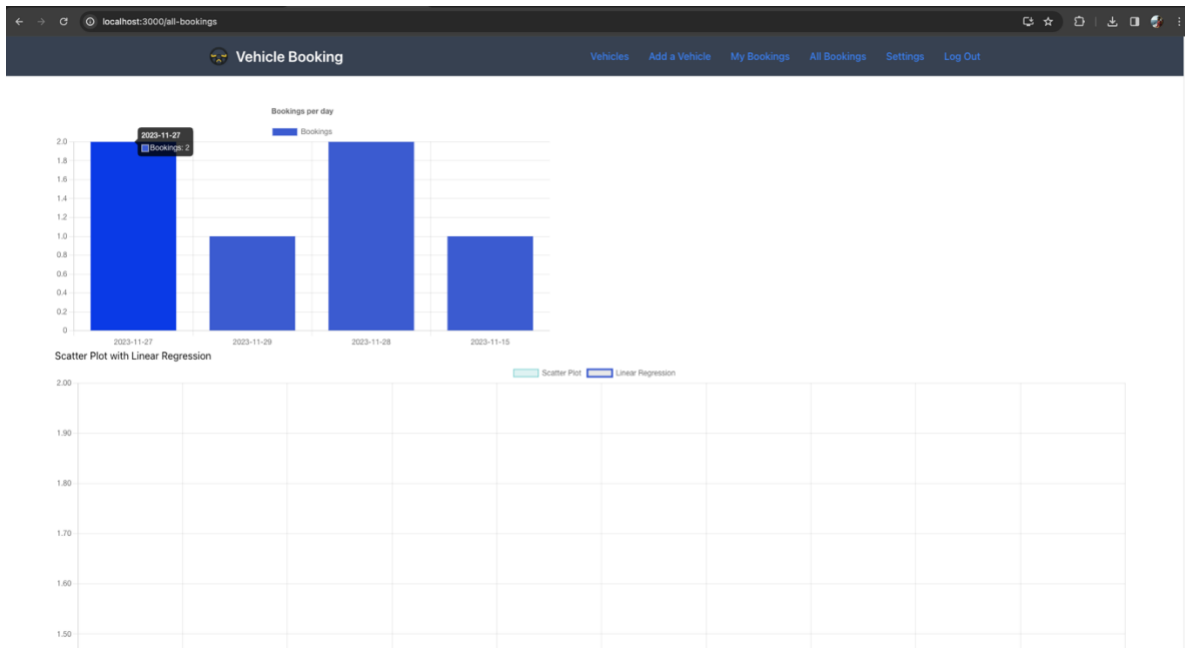
localhost:3000 says
Wrong username

Your Username
ad_limbekar

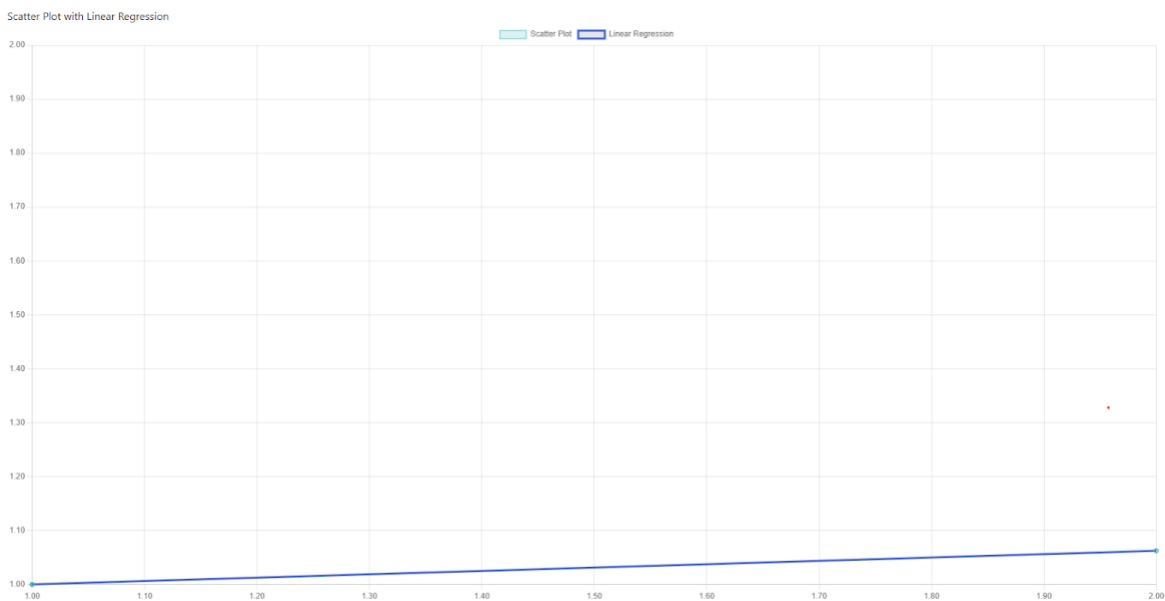
Login

Project Report Vehicle Registration System

- **Implementation of visualization and linear regression using machine learning**
 - Data visualization has been done for the numbers cars that have been booked for the number of days.



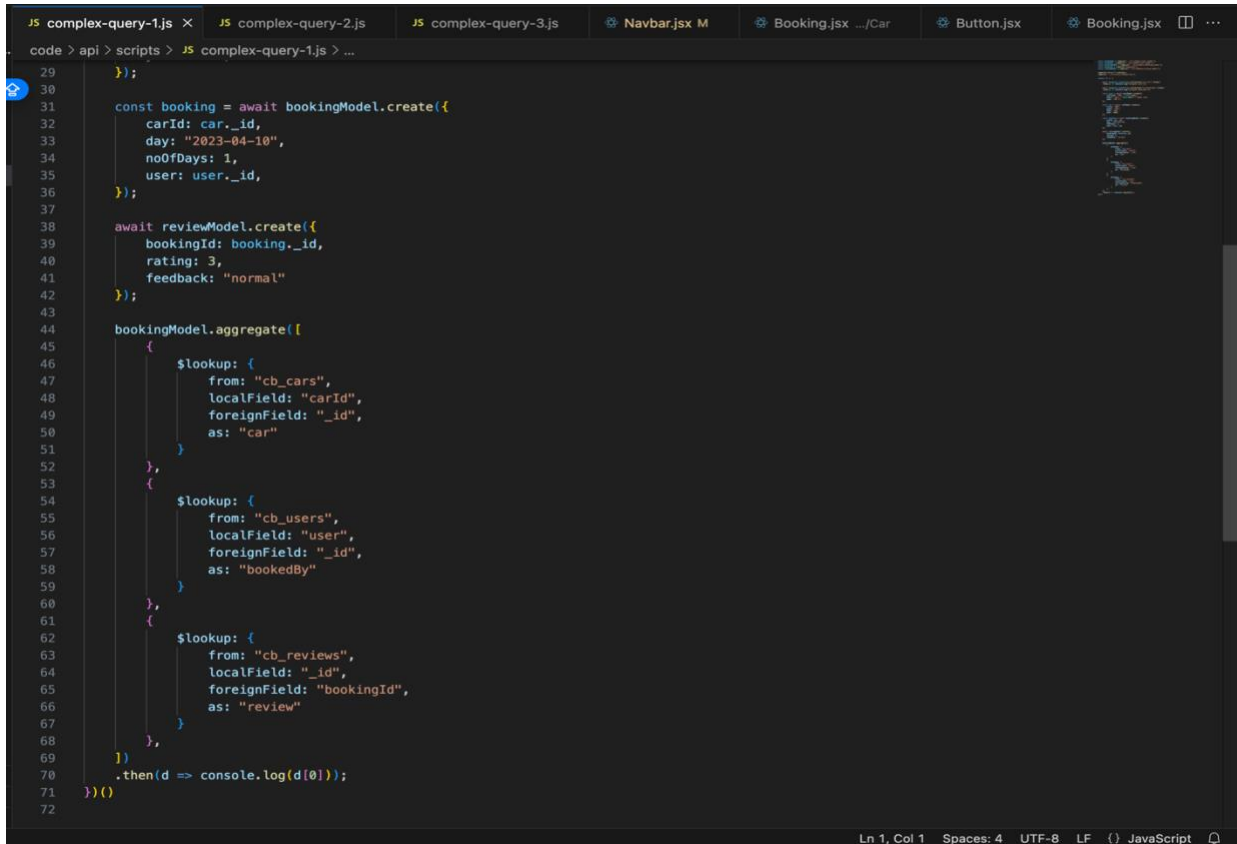
- On the same data the Linear regression has been performed and show how good the system is doing and how they can improve the booking rate.



Project Report Vehicle Registration System

- **Complex queries**

We have implemented 3 complex queries.



```
29  });
30
31  const booking = await bookingModel.create({
32    carId: car._id,
33    day: "2023-04-10",
34    noOfDays: 1,
35    user: user._id,
36  });
37
38  await reviewModel.create({
39    bookingId: booking._id,
40    rating: 3,
41    feedback: "normal"
42  });
43
44  bookingModel.aggregate([
45    {
46      $lookup: {
47        from: "cb_cars",
48        localField: "carId",
49        foreignField: "_id",
50        as: "car"
51      },
52    },
53    {
54      $lookup: {
55        from: "cb_users",
56        localField: "user",
57        foreignField: "_id",
58        as: "bookedBy"
59      },
60    },
61    {
62      $lookup: {
63        from: "cb_reviews",
64        localField: "_id",
65        foreignField: "bookingId",
66        as: "review"
67      },
68    },
69  ])
70  .then(d => console.log(d[0]));
71  })()
72
```

INFS 740

Project Report Vehicle Registration System

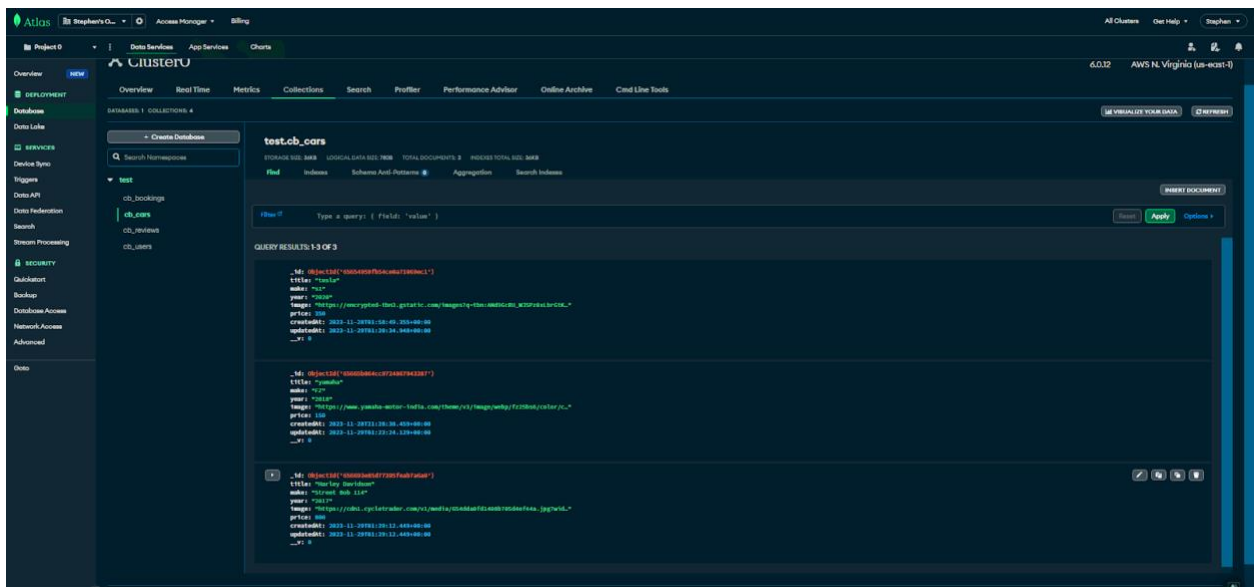
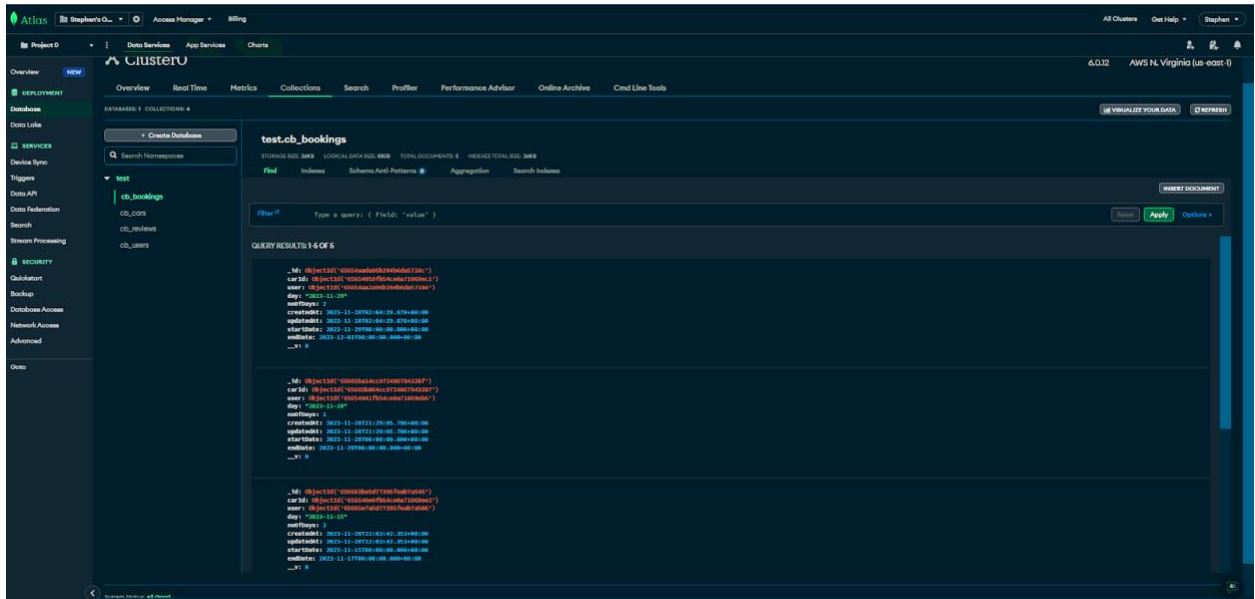
```
JS complex-query-1.js JS complex-query-2.js X JS complex-query-3.js NavBar.jsx M Booking.jsx .../Car Button.jsx Booking.jsx ...
code > api > scripts > JS complex-query-2.js > ...
59   rating: 3,
60   feedback: "normal"
61 });
62
63
64   carModel.aggregate([
65     {
66       $lookup: {
67         from: "cb_bookings",
68         localField: "_id",
69         foreignField: "carId",
70         as: "bookings",
71       },
72     },
73     {
74       $unwind: {
75         path: "$bookings",
76         preserveNullAndEmptyArrays: true
77       },
78     },
79     {
80       $lookup: {
81         from: "cb_reviews",
82         localField: "bookings._id",
83         foreignField: "bookingId",
84         as: "reviews"
85       },
86     },
87     {
88       $group: {
89         _id: {
90           _id: "$_id",
91           title: "$title",
92           make: "$make",
93           year: "$year",
94           image: "$image",
95         },
96         bookings: { $addToSet: "$bookings" },
97         reviews: { $addToSet: "$reviews" },
98       },
99     },
100   ])
101   .then(d => console.log(JSON.stringify(d)))
102 })
103
```

```
JS complex-query-1.js JS complex-query-2.js JS complex-query-3.js X NavBar.jsx M Booking.jsx .../Car Button.jsx Booking.jsx ...
~/Library/CloudStorage/OneDrive-GeorgeMasonUniversity-O365Production/ -3.js > ...
Sem_3/INFS 740/Project/
INFS740_Project_Vehicle-Registration-System/
code/api/package.json
51   feedback: "normal"
52 });
53
54
55   userModel.aggregate([
56     {
57       $lookup: {
58         from: "cb_bookings",
59         localField: "_id",
60         foreignField: "user",
61         as: "bookings"
62       },
63     },
64     {
65       $unwind: {
66         path: "$bookings",
67         preserveNullAndEmptyArrays: true,
68       },
69     },
70     {
71       $lookup: {
72         from: "cb_cars",
73         localField: "bookings.carId",
74         foreignField: "_id",
75         as: "cars",
76       },
77     },
78     {
79       $group: {
80         _id: {
81           id: "$_id",
82           username: "$username",
83           email: "$email",
84           name: "$name",
85         },
86         bookings: { $addToSet: "$bookings" },
87         cars: { $addToSet: "$cars" },
88       },
89     },
90   ])
91   .then(d => console.log(JSON.stringify(d)))
92 })
93
```

INFS 740

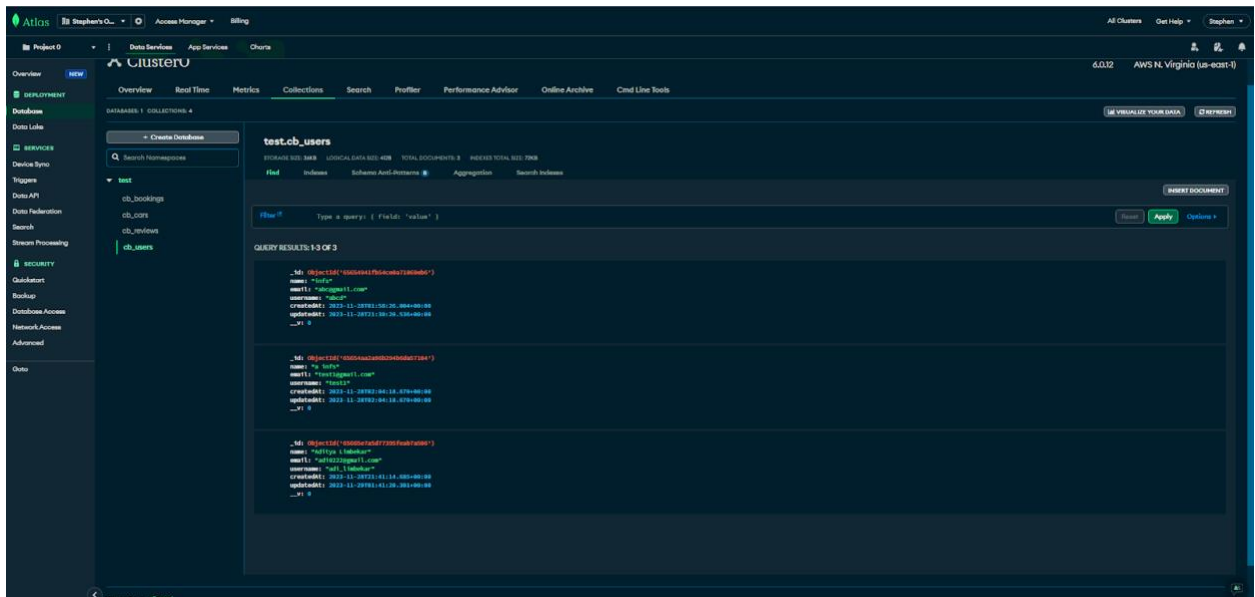
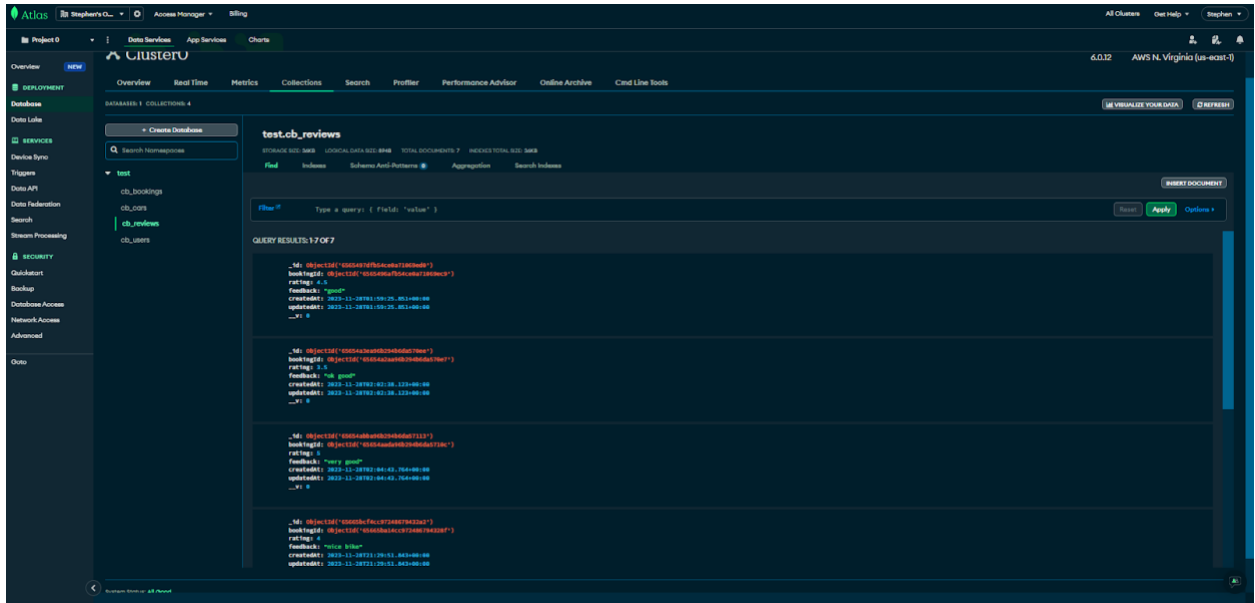
Project Report Vehicle Registration System

- MongoDB Atlas



INFS 740

Project Report Vehicle Registration System



- **Contributions**

→ **“Vehicles”, “Add a Vehicle”, “Login”, “Signup”** components (UI as well as the respective backend functionality along with queries) – **Aditya Milind Limbekar**

→ **“My Bookings”, “All Bookings”, “Settings”, “Log Out”** components (UI as well as the respective backend functionality along with queries) – **Stephen Dias**