

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.

• Technologies used

- **Frontend:** ReactJS, CSS
- **Backend:** NodeJS
- **Database:** MongoDB

• 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.

Project Report Vehicle Registration System

Update: Users can modify the information of existing vehicles.

Delete: Users can remove vehicles from the system.

→ Add a vehicle

The screenshot shows a web browser at the URL `localhost:3000/add-car`. The page title is "Vehicle Booking". A dark blue header bar contains the title and navigation links: "My Bookings", "All Bookings", "Settings", and "Log Out". A dark blue modal box is open in the center, displaying "localhost:3000 says" and "Vehicle added" with an "OK" button. Below the modal, the form "Enter Car/Bike details..." contains the following fields:

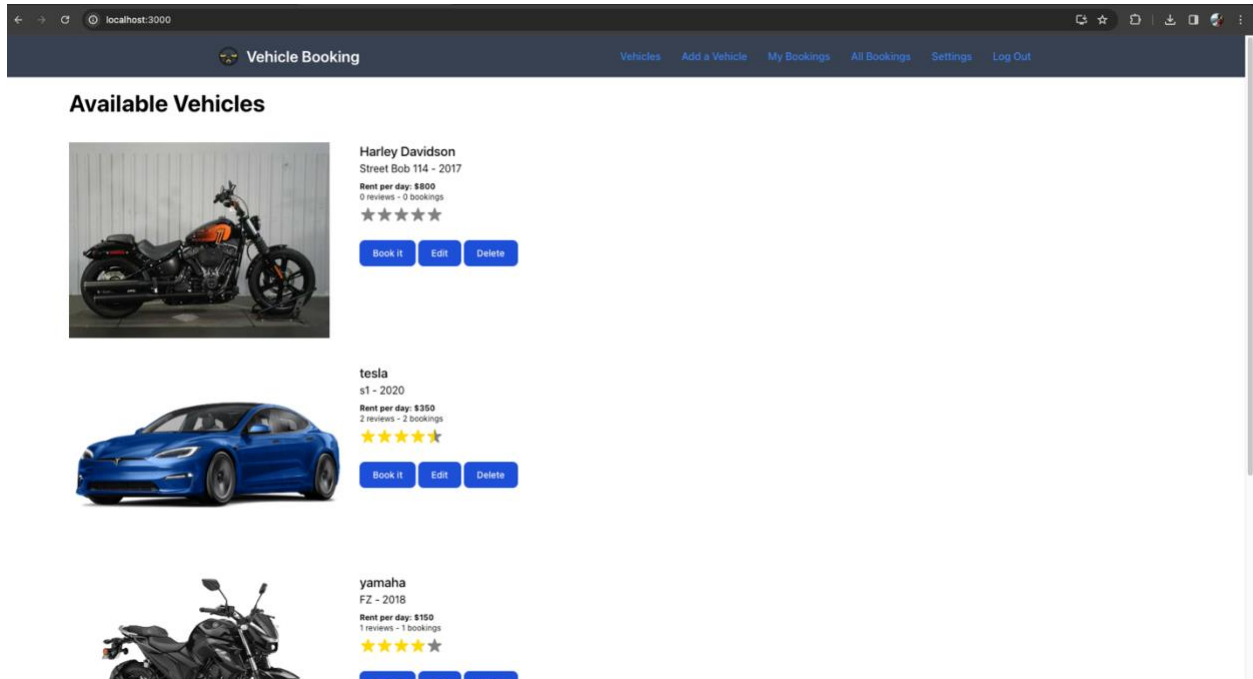
- Brand:
- Model:
- Year:
- Price:
- Image URL:

At the bottom of the form is a blue button labeled "Add Vehicle".

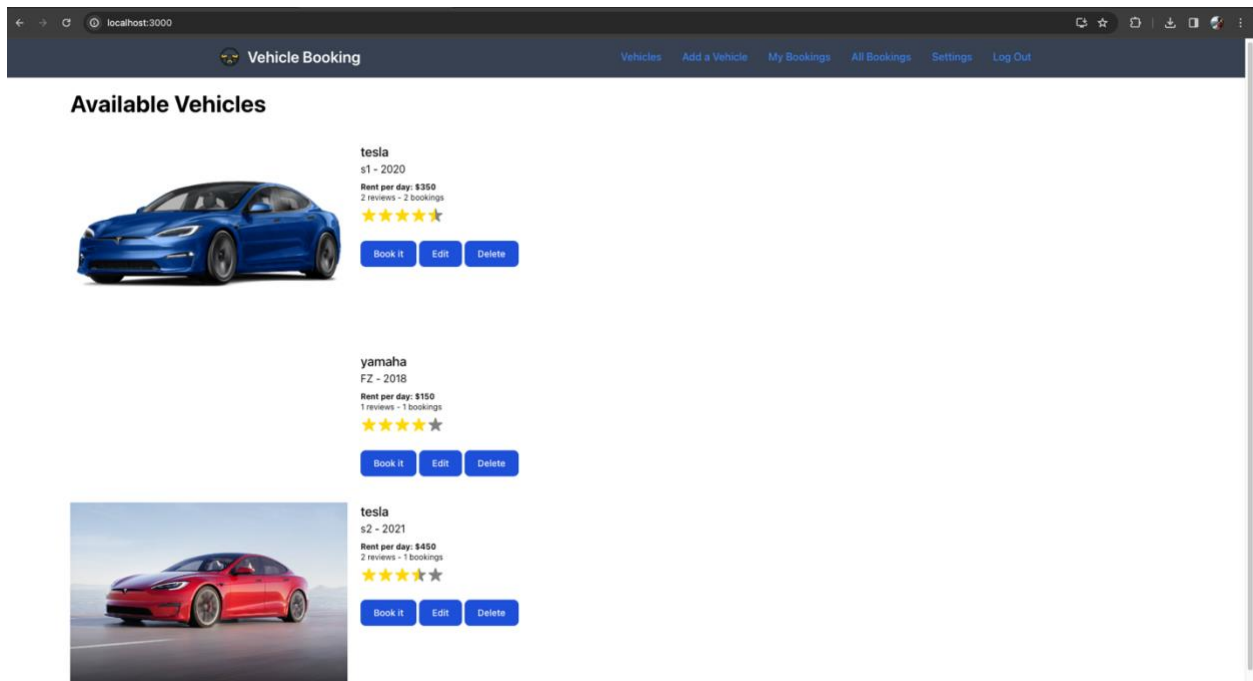
→ View/Read a vehicle

INFS 740

Project Report Vehicle Registration System



→ Update a vehicle



INFS 740

Project Report Vehicle Registration System

localhost:3000

Vehicle Booking

Vehicles Add a Vehicle 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

My Bookings All Bookings Settings Log Out

localhost:3000 says
Vehicle updated

OK

Enter Car/Bike details...

Brand
yamaha

Model
FZ

Year
2018

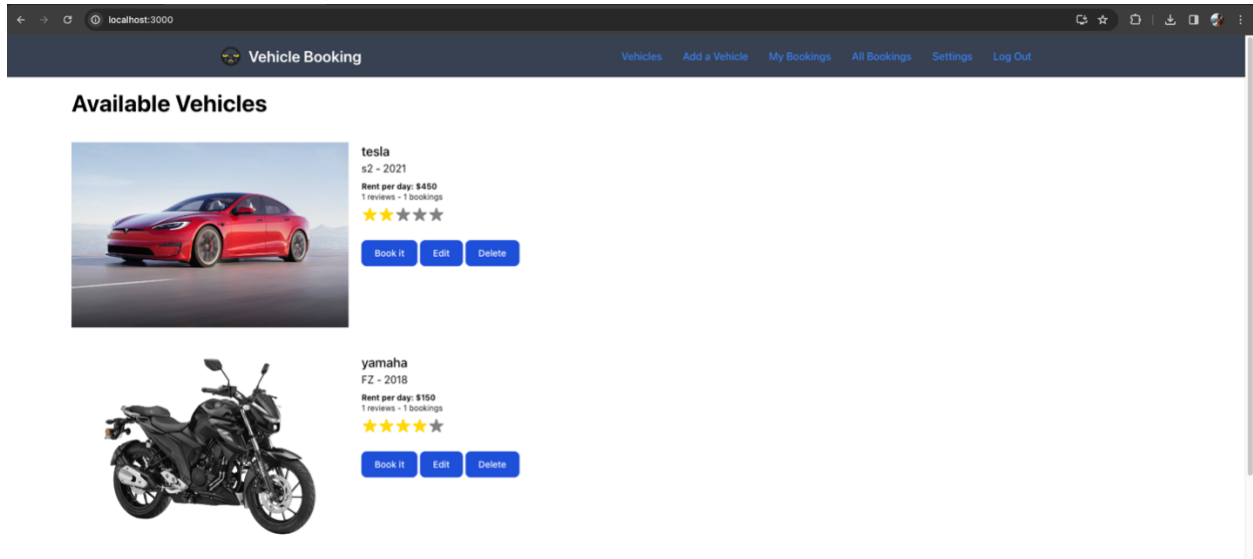
Price
150

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

Update Vehicle

INFS 740

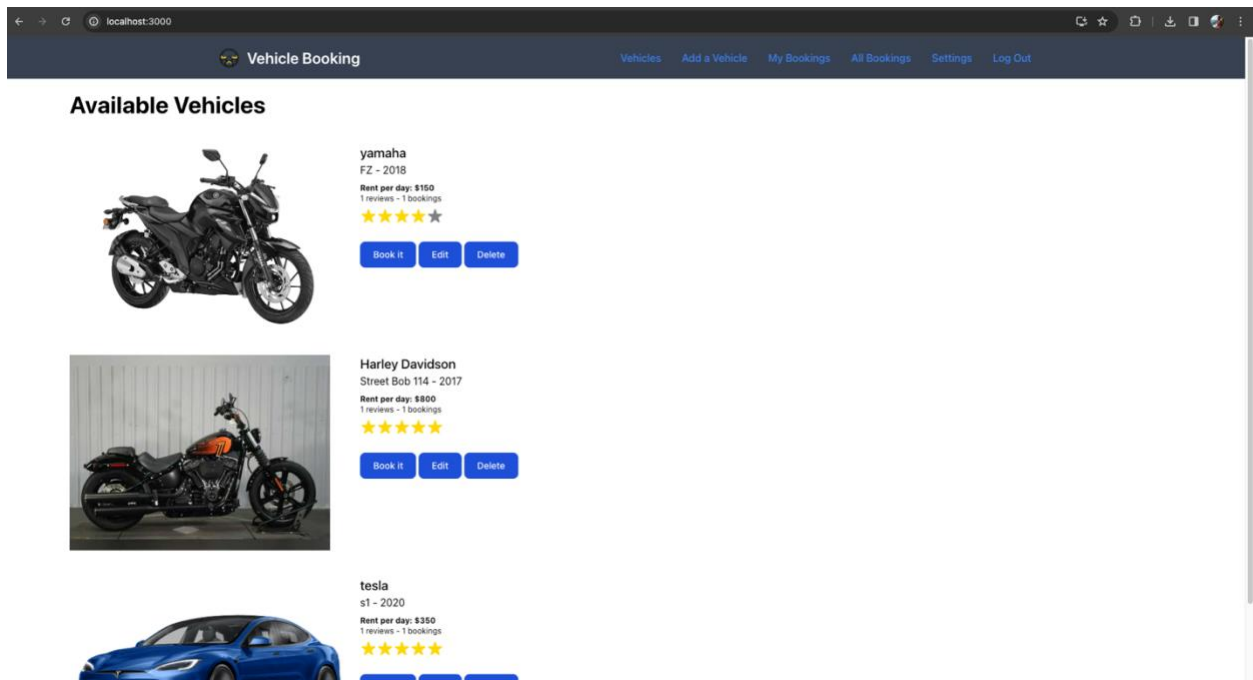
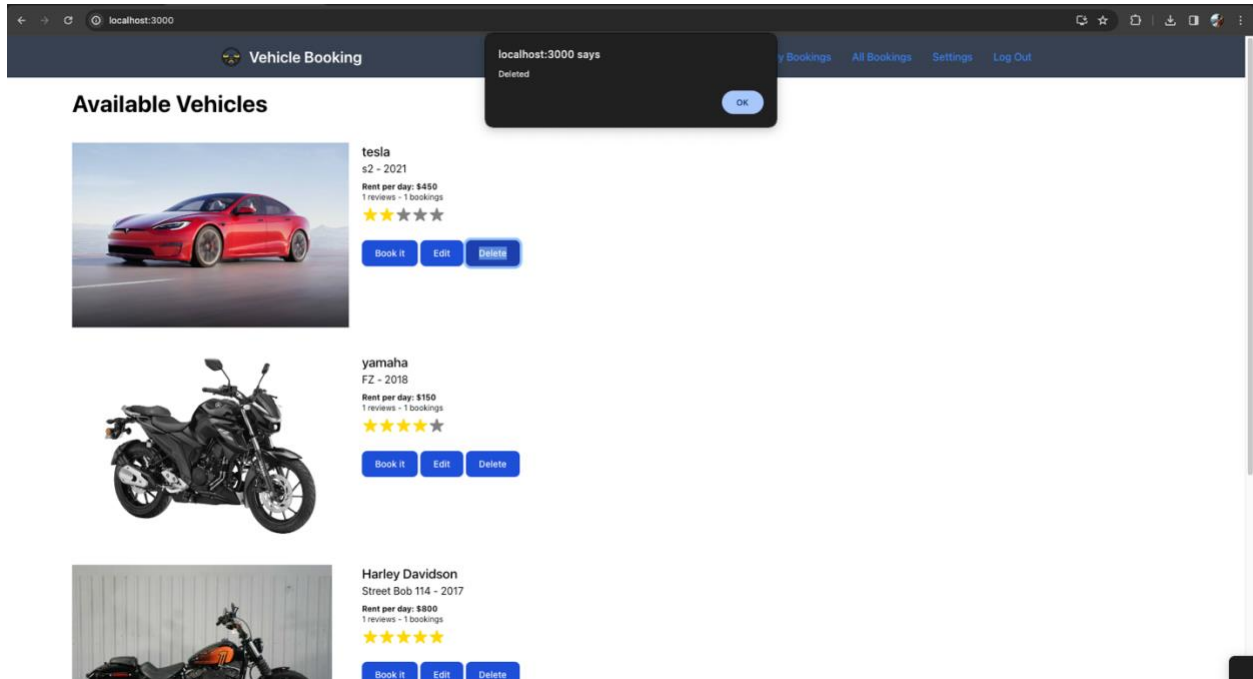
Project Report Vehicle Registration System



→ Delete a vehicle

INFS 740

Project Report Vehicle Registration System



b. Review CRUD Operations:

Create: Users can submit reviews for specific vehicles.

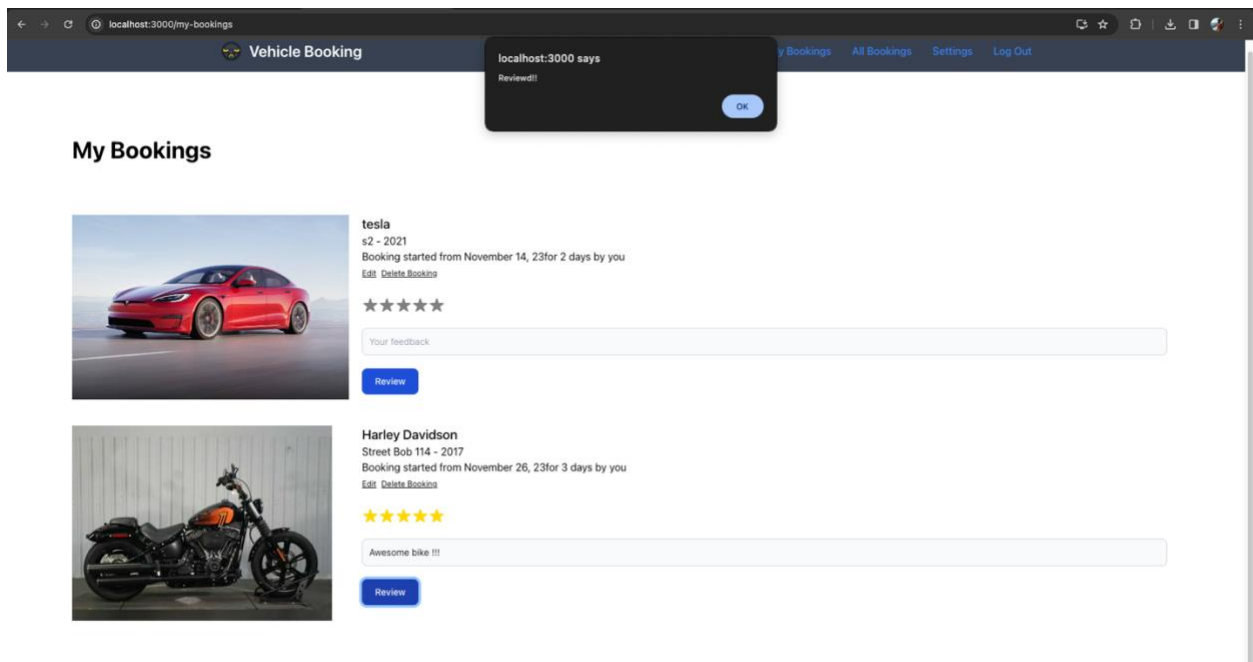
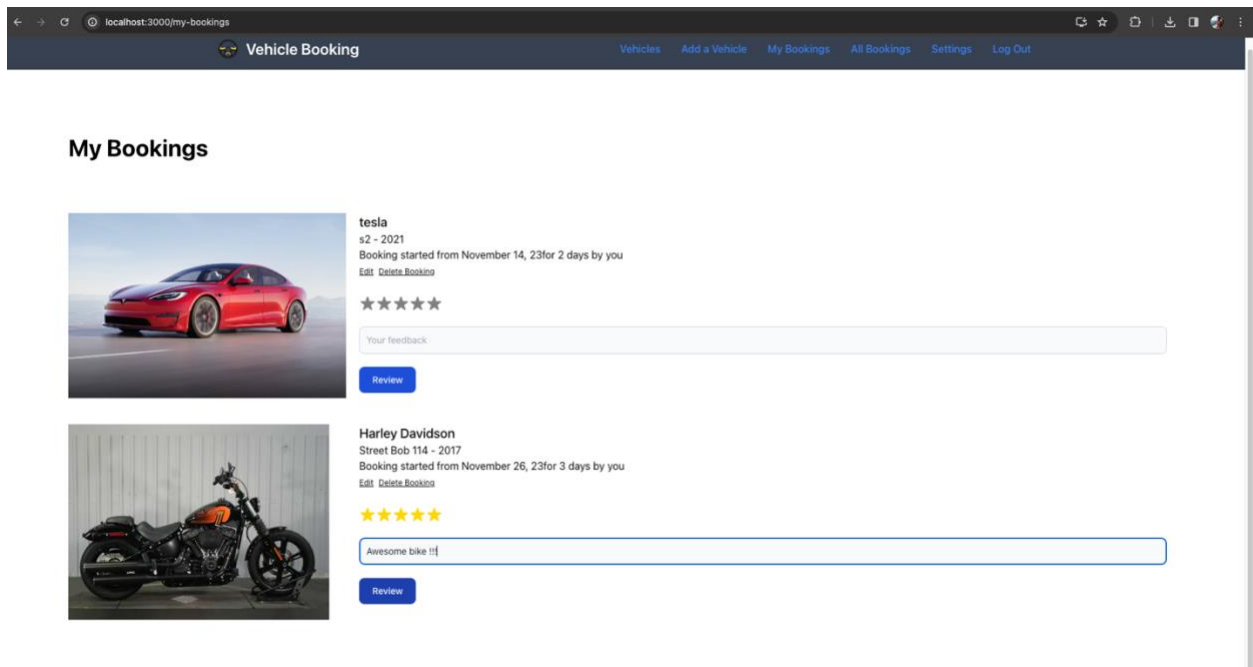
Project Report Vehicle Registration System

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.

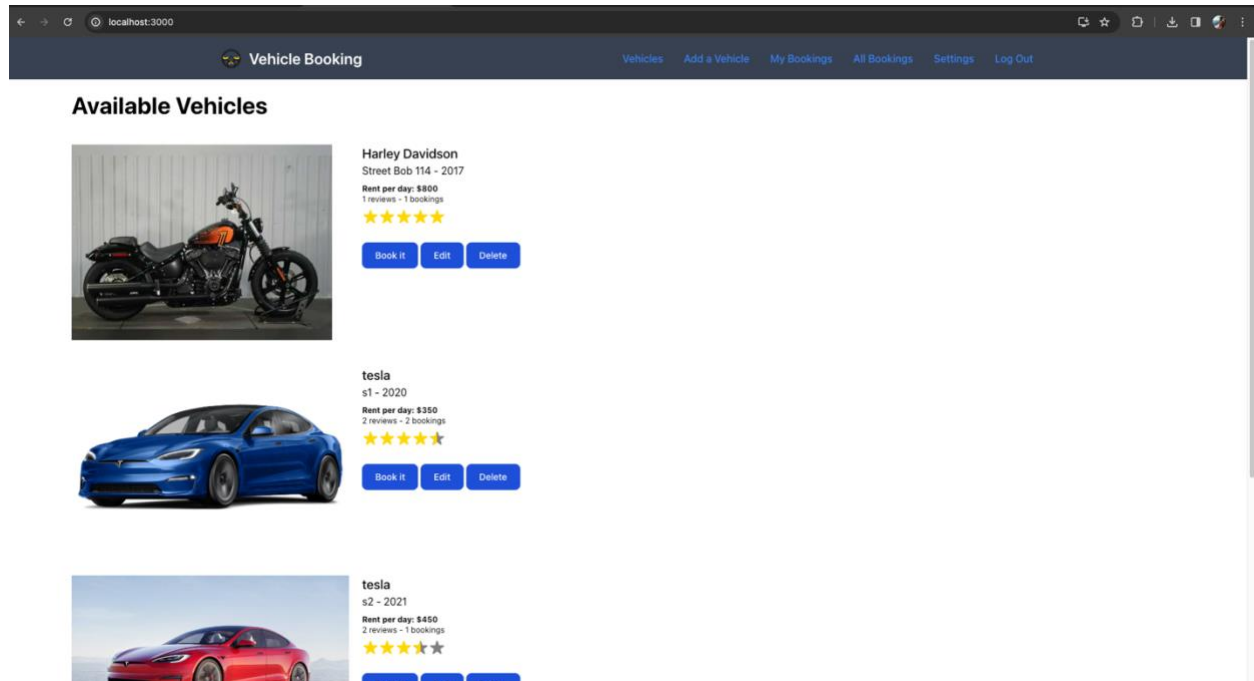
→ Add a review



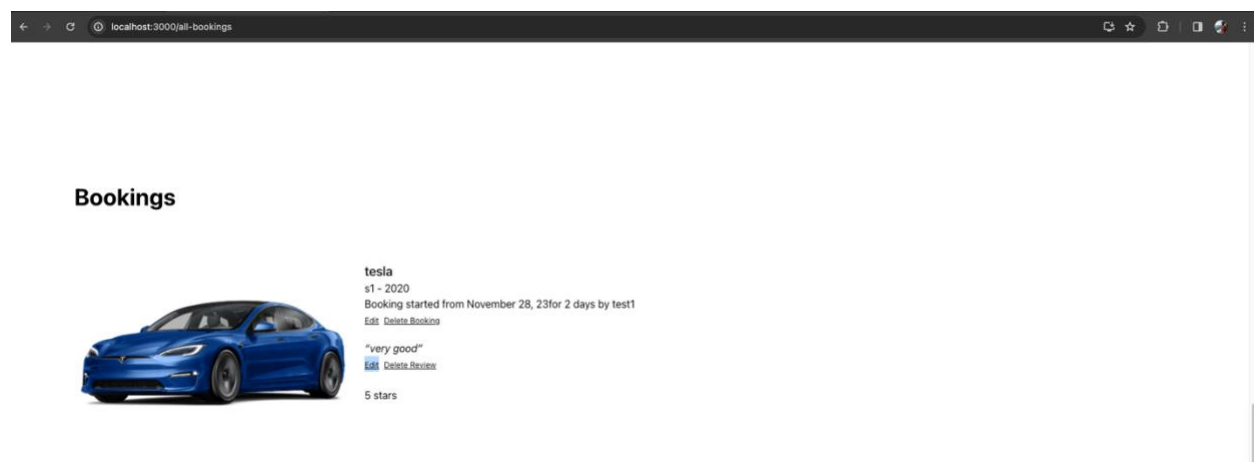
INFS 740

Project Report Vehicle Registration System

→ View/Read a review

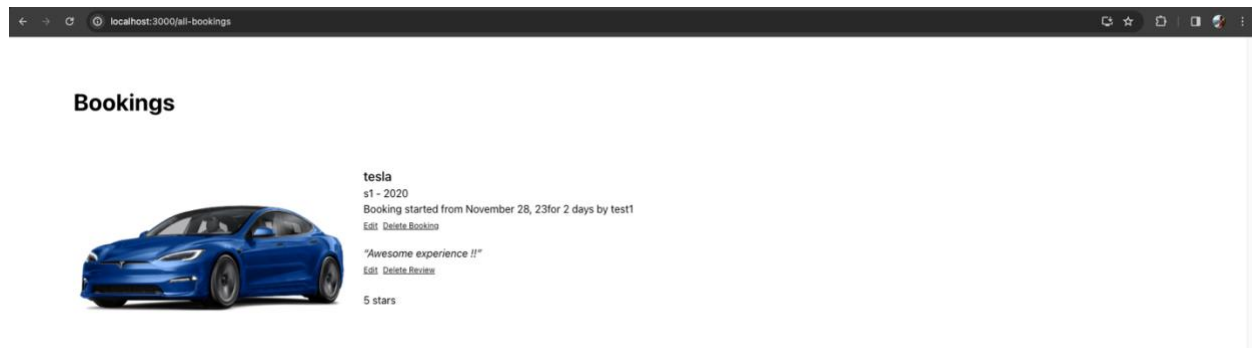
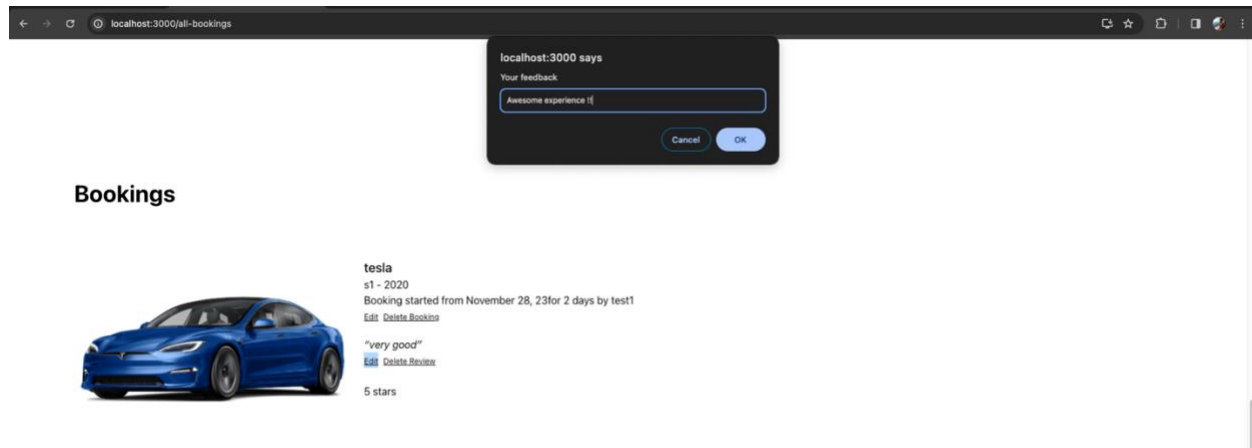


→ Update a review

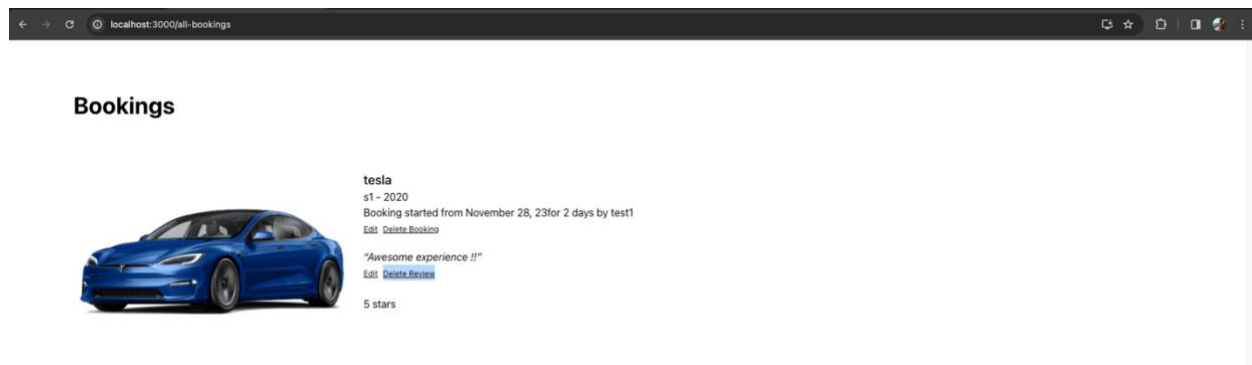


INFS 740

Project Report Vehicle Registration System

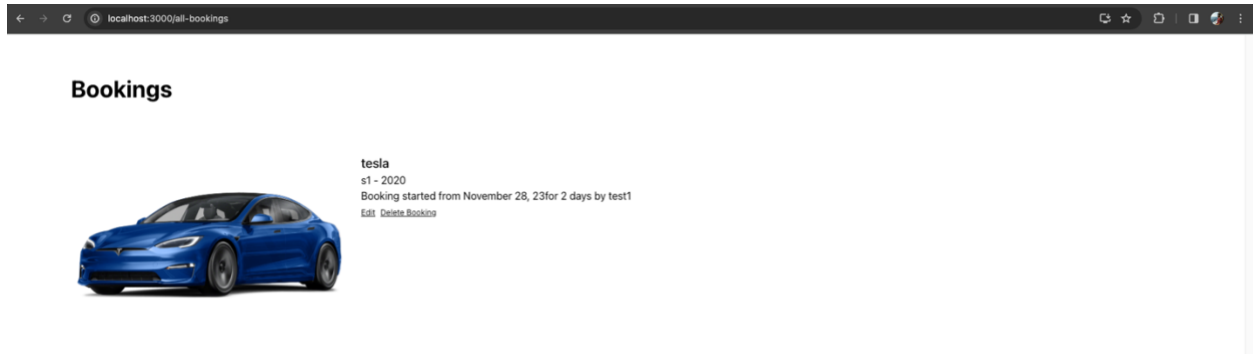


→ Delete a review



INFS 740

Project Report Vehicle Registration System



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.

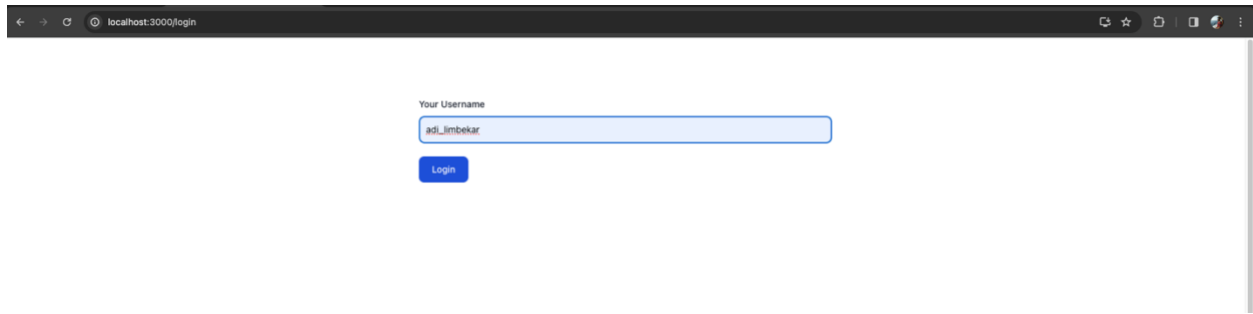
Delete: Users can delete their own accounts, and administrators can delete any user account.

→ Add a user

A screenshot of a web browser showing a user registration page. The browser's address bar indicates the URL is localhost:3000/signup. The page contains a registration form with 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 the email field is a blue 'Sign Up' button.

INFS 740

Project Report Vehicle Registration System



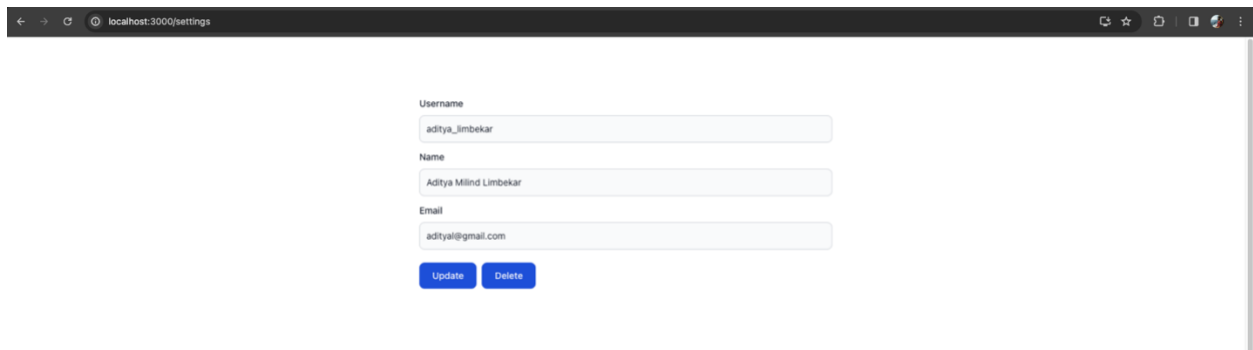
localhost:3000/login

Your Username

adi_limbekar

Login

→ View/Read a user



localhost:3000/settings

Username

aditya_limbekar

Name

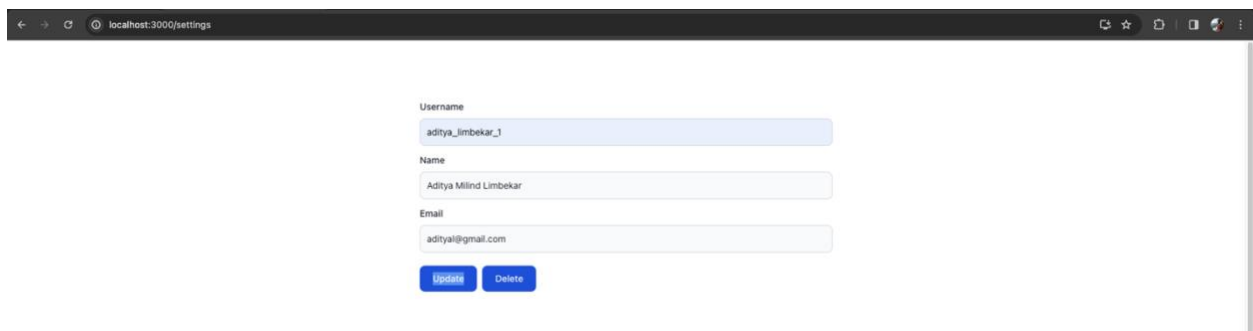
Aditya Milind Limbekar

Email

aditya@gmail.com

Update Delete

→ Update a user



localhost:3000/settings

Username

aditya_limbekar_1

Name

Aditya Milind Limbekar

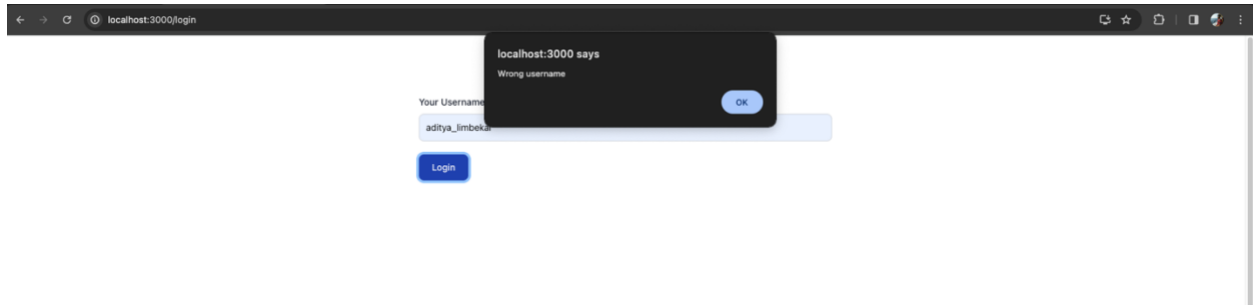
Email

aditya@gmail.com

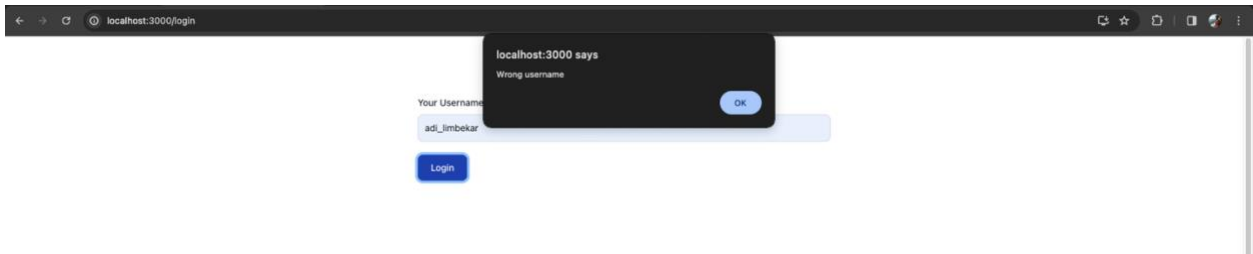
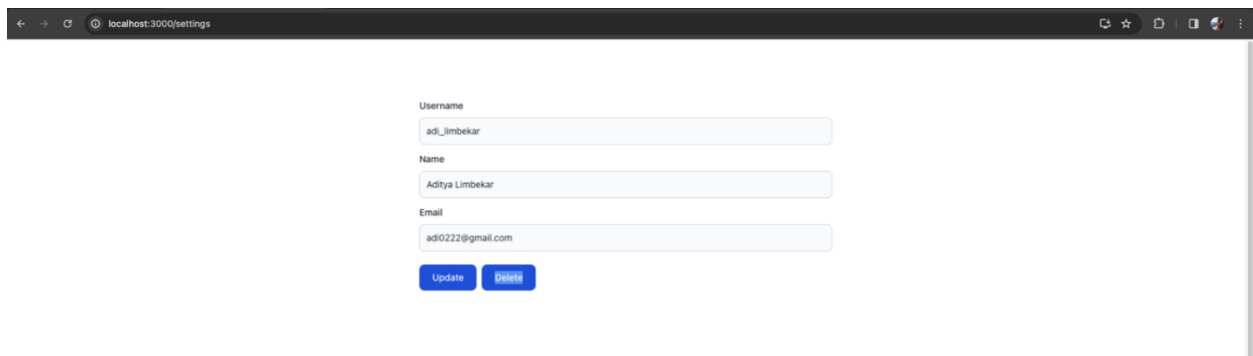
Update Delete

INFS 740

Project Report Vehicle Registration System



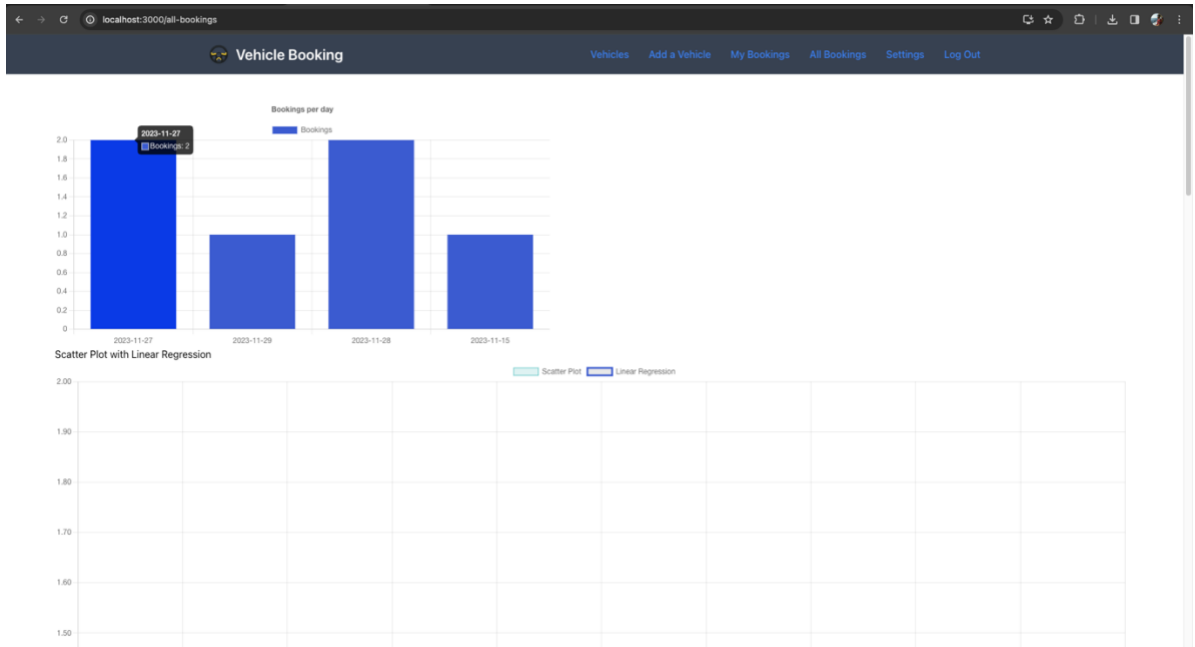
→ Delete a user



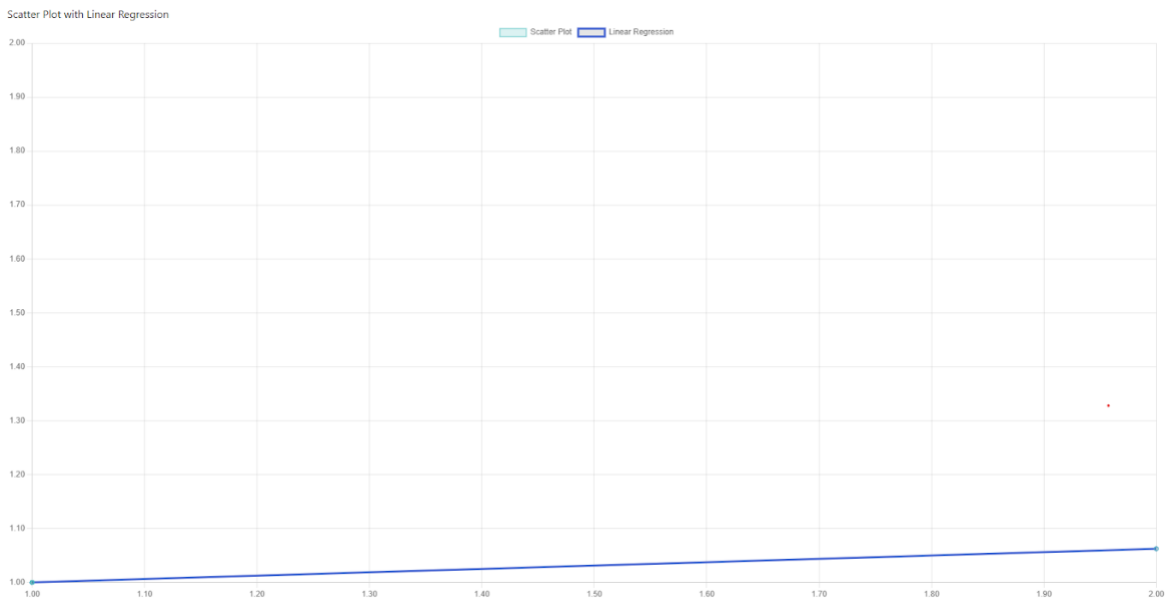
- **Implementation of visualization and linear regression using machine learning**
 - Data visualization has been done for the numbers cars that have been booked for the number of days.

INFS 740

Project Report Vehicle Registration System



- 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.



- Complex queries**
We have implemented 3 complex queries.

Project Report Vehicle Registration System

```

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

```

Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

```

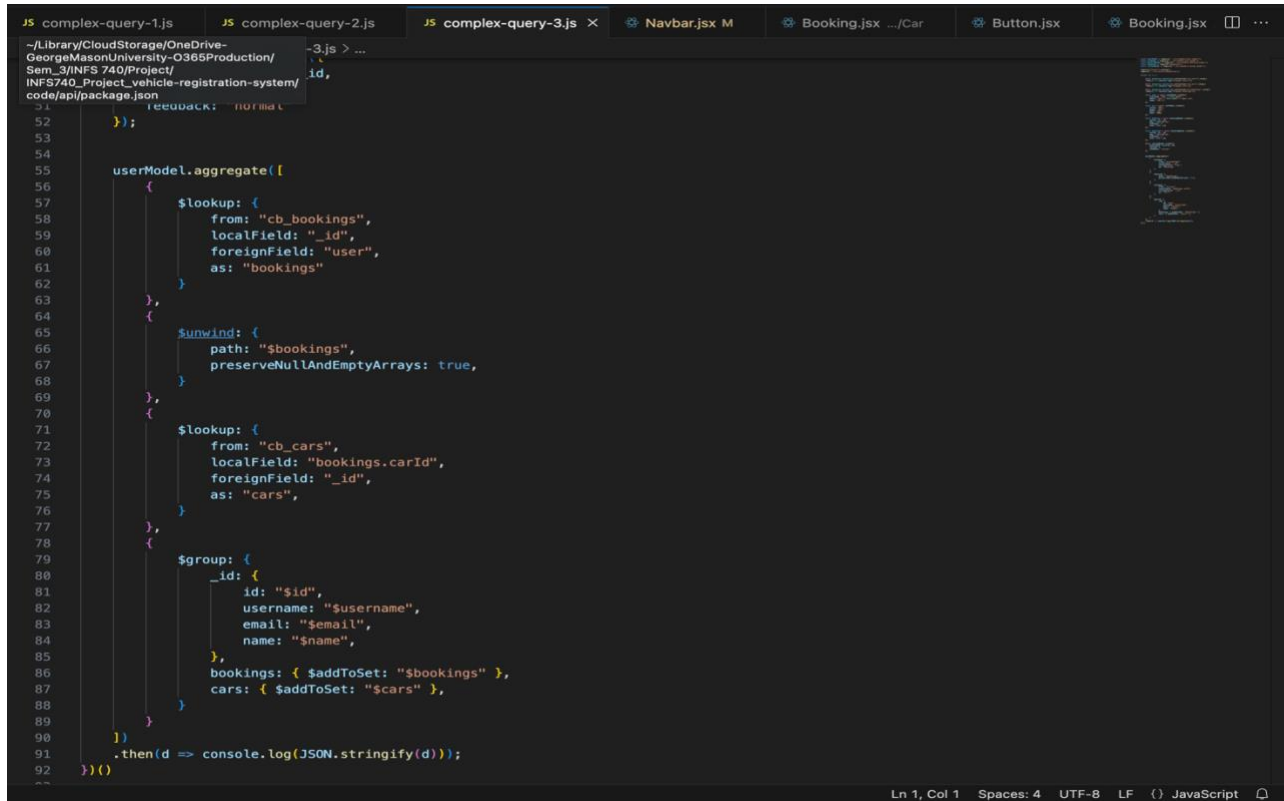
59  rating: 3,
60  feedback: "normal"
61  });
62
63
64  carModel.aggregate([
65    {
66      $lookup: {
67        from: "cb_bookings",
68        localField: "_id",
69        foreignField: "carId",
70        as: "bookings",
71      },
72    },
73    {
74      $unwind: {
75        path: "$bookings",
76        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  })()

```

Ln 1, Col 1 Spaces: 4 UTF-8 LF {} JavaScript

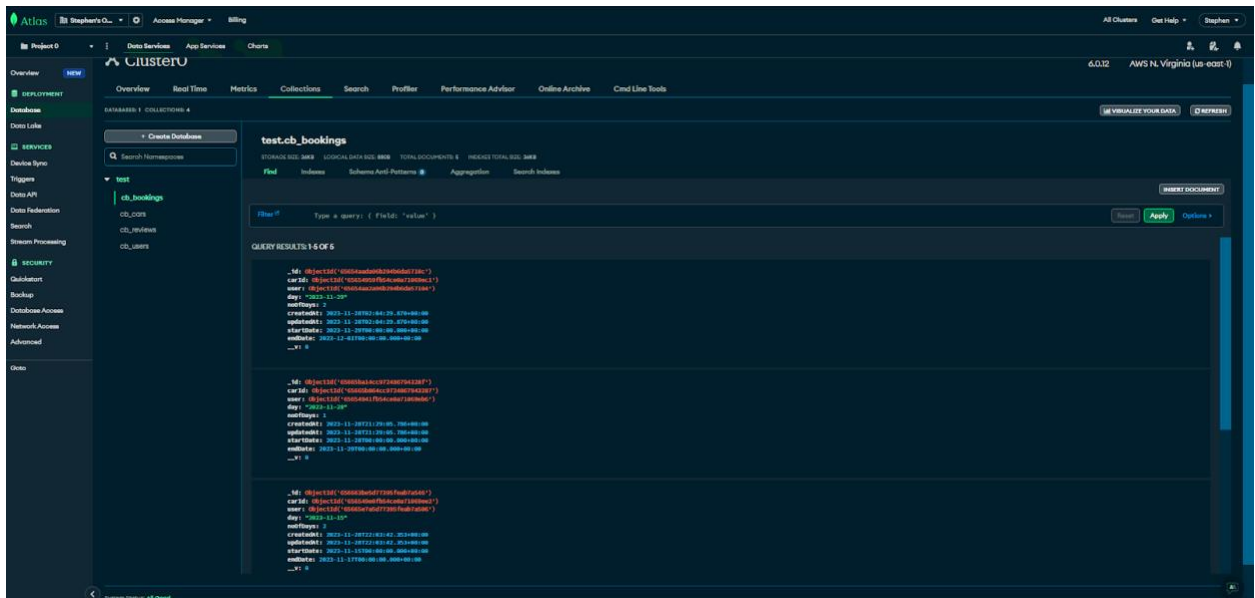
INFS 740

Project Report Vehicle Registration System



```
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/Sem_3/INFS 740/Project/INFS740_Project_vehicle-registration-system/code/api/package.json
52 feedback: "not like
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 })()
```

- MongoDB Atlas



INFS 740

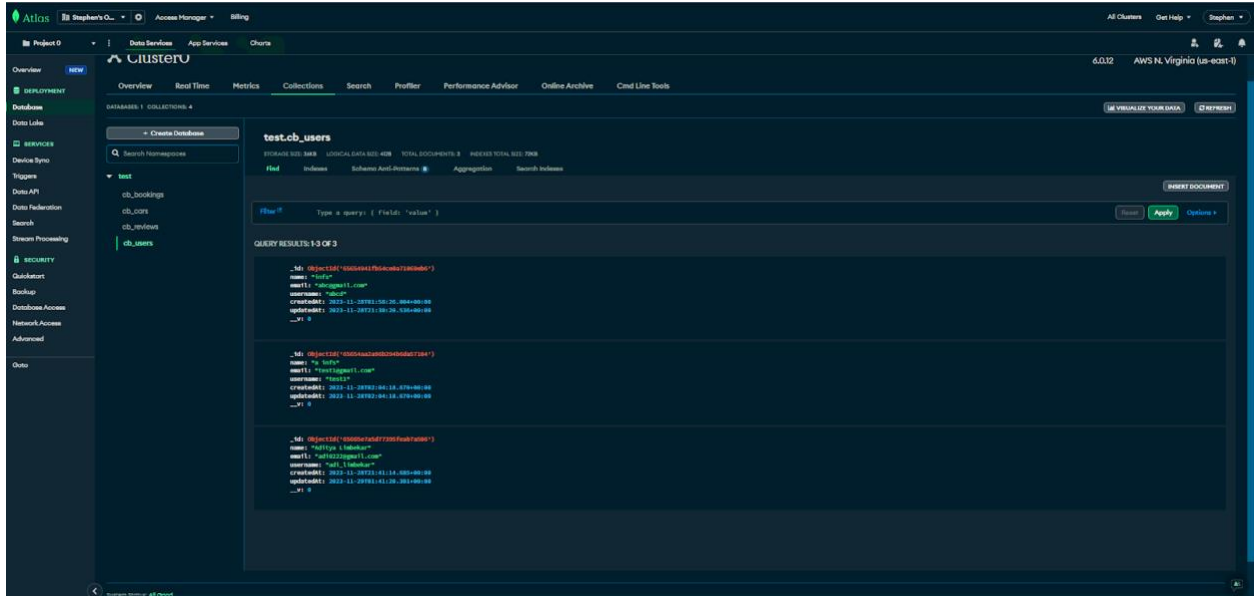
Project Report Vehicle Registration System

The screenshot displays the Atlas ClusterIO interface for a project named "Project 0". The left sidebar contains navigation options: Overview, Deployment, Database, Data Lake, Services, Device Type, Triggers, Data API, Data Federation, Search, Stream Processing, Security, Quickstart, Backup, Database Access, Network Access, and Advanced. The main panel shows the "test.cb_cars" collection. It includes a search bar, a "Filter" button, and a "Type a query" input. The query results are displayed in a table format, showing fields like "_id", "title", "year", "price", "created_at", and "updated_at". The interface also features a "Visualize Your Data" button and a "Refresh" button.

The screenshot displays the Atlas ClusterIO interface for a project named "Project 0". The left sidebar contains navigation options: Overview, Deployment, Database, Data Lake, Services, Device Type, Triggers, Data API, Data Federation, Search, Stream Processing, Security, Quickstart, Backup, Database Access, Network Access, and Advanced. The main panel shows the "test.cb_reviews" collection. It includes a search bar, a "Filter" button, and a "Type a query" input. The query results are displayed in a table format, showing fields like "_id", "book_id", "rating", "feedback", "created_at", and "updated_at". The interface also features a "Visualize Your Data" button and a "Refresh" button.

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**