# Table of Contents

[Task A 2](#_Toc114397832)

[1.0 System Requirement Specification for the GoCheeta Cab Service. 2](#_Toc114397833)

[1.1 Purpose 2](#_Toc114397834)

[1.2 Requirements 2](#_Toc114397835)

[1.2.1 Functional Requirements 2](#_Toc114397836)

[1.2.2 Non-Functional Requirements 2](#_Toc114397837)

[Task B 3](#_Toc114397838)

[2.0 UML Diagrams for GoCheeta Online cab service system 3](#_Toc114397839)

[2.1 Use Case Diagram 3](#_Toc114397840)

[2.1.1 Customer Use Case 3](#_Toc114397841)

[2.1.2 Driver Use Case 4](#_Toc114397842)

[2.1.3 Admin Use Case 5](#_Toc114397843)

[2.2 Sequence Diagrams 6](#_Toc114397844)

[2.2.1 Customer Sequence 6](#_Toc114397845)

[2.2.2 Driver Sequence 7](#_Toc114397846)

[2.2.3 Admin Sequence 8](#_Toc114397847)

[Task C 9](#_Toc114397848)

[Design Patterns 9](#_Toc114397849)

[Creational 9](#_Toc114397850)

[Structural 10](#_Toc114397851)

[Behavioral 11](#_Toc114397852)

[Task D 12](#_Toc114397853)

[Git Hub Link: 12](#_Toc114397854)

[Task E 12](#_Toc114397855)

[Test Plan - Customers 12](#_Toc114397856)

[Test Plan – Driver 13](#_Toc114397857)

[Test Plan – Admin 13](#_Toc114397858)

# Table Of Figures

Figure 1 Customer Use Case 5

Figure 2 Driver Use Case 6

Figure 3 Admin Use Case 7

Figure 4 Customer Sequence Diagrams 8

Figure 5 Driver Sequence Diagrams 9

Figure 6 Admin Sequence Diagrams 10

Figure 7 Creational Design Pattern 12

Figure 8 Structural Design Pattern 12

Figure 9 Behavioral Design Pattern 13

Figure 10 User Login Unit Test 19

Figure 11 User Register Unit Test 20

Figure 12 User Add Booking Unit Test 20

Figure 13 Driver Accept Booking Unit Test 21

Figure 14 Driver View Past Booking Unit Test 21

Figure 15 Admin Login Unit Test 22

Figure 16 Admin Add Driver Unit Test 22

Figure 17 Admin Add Vehicle Unit Test 23

Figure 18 Admin Manage User Unit Test 23

Figure 19 Admin Manage Booking Unit Test 24

Figure 20 Admin Manage Vehicle Unit Test 24

Figure 21 User Manual Register 25

Figure 22 User Manual Navigation Bar 26

Figure 23 User Manual User Login 26

Figure 24 User Manual Customer Navigation Bar 26

Figure 25 User Manual Customer Home Page 27

Figure 26 User Manual Customer Add Booking 27

Figure 27 User Manual Customer View Booking 27

Figure 28 User Manual Driver Home Page 28

Figure 29 User Manual Driver Navigation Bar 28

Figure 30 User Manual Driver View Customer Booking 28

Figure 31 User Manual Driver Accept Booking 28

Figure 32 User Manual Driver View Accept Booking 29

Figure 33 User Manual Admin Login Page 29

Figure 34 User Manual Admin Home Page 29

Figure 35 User Manual Admin Navigation Bar 29

Figure 36 User Manual Admin Add Driver 30

Figure 37 User Manual Admin Add Vehicle 30

Figure 38 User Manual Admin Manage User 31

Figure 39 User Manual Admin Update User 31

Figure 40 User Manual Admin Delete User 32

Figure 41 User Manual Admin Manage Vehicle 32

Figure 42 User Manual Admin Update Vehicle 32

Figure 43 User Manual Admin Delete Vehicle 33

Figure 44 User Manual Admin Manage Booking 33

# Table of Tables

[Table 1 Functional Requirements 3](#_Toc114527730)

[Table 2 Test Plan - Customers 15](#_Toc114527731)

[Table 3 Test Plan – Driver 16](#_Toc114527732)

[Table 4 Test Plan – Admin 18](#_Toc114527733)

# Task A

## System Requirement Specification for the GoCheeta Cab Service.

## 1.1 **Purpose**

The Functions and the non-functions are determined by the SRS Document. SRS Document helps people who code and who don’t know to code understand the system very easily. When someone is handling a project, they check the SRS to check the functions in the system.

## 1.2 Requirements

## 1.2.1 Functional Requirements

Table 1 Functional Requirements

|  |  |  |
| --- | --- | --- |
|  | **User Requirements** |  |
| **Admin** | **Customer** | **Driver** |
| Login | Register | Login |
| Add Drivers | Login | View Hires |
| View Users | Book a Driver | Accept Hires |
| View Bookings | View the Booked Hire | View Accepted Hires |
| View Vehicles | Logout | Logout |
| Manage User (Edit/Delete) | - | - |
| Manage Bookings (Edit/Delete) | - | - |
| Manage Vehicles (Edit/Delete) | - | - |
| Add Vehicles | - | - |
| Logout | - | - |

## 1.2.2 Non-Functional Requirements

This website is made for performance and security and to make day-to-day life easy for drivers and customers the users can navigate through the site very easily the useability is very clean and there is no lag or any brakes in the system and it is very securable.

# Task B

## 2.0 UML Diagrams for GoCheeta Online cab service system

## 2.1 Use Case Diagram

### 2.1.1 Customer Use Case

Diagram

Description automatically generated

Figure 1 Customer Use Case

This UML Diagram shows all the access points of a customer as it shows the customer can see the home page and register as a customer and then log in to the system after the login the customer can Add Bookings and see his/her bookings and log out of the system.

### 2.1.2 Driver Use Case

Diagram

Description automatically generated

Figure 2 Driver Use Case

This UML Diagram shows all the access points of a driver as it shows the driver can see the home page and the driver can not register admin can add drivers and then drivers can log in view hires accept hires and then view accepted hires and lastly log out of the system.

### 2.1.3 Admin Use Case

Diagram

Description automatically generated

Figure 3 Admin Use Case

This UML Diagram shows all the access points of an admin as it shows the admin can see the home page Admin is hard coded to the system so there will be only one admin witch can log in manage users and manage vehicles and manage bookings after before that as I said driver can not register so admin will be adding the driver to the system. The admin will be adding vehicles and lastly logging out of the system.

## 2.2 Sequence Diagrams

### 2.2.1 Customer Sequence Diagrams

Table

Description automatically generated

Figure 4 Customer Sequence Diagrams

The Customer Sequence Diagram, shows all the step that a customer can do on this web site Customer needs to register using his/her details after that customer can log in to the system then only the customer can Make a booking or view the past bookings the customer has done. At last, the customer can log out of the system.

### 2.2.2 Driver Sequence Diagrams

Diagram

Description automatically generated

Figure 5 Driver Sequence Diagrams

The Driver Sequence Diagram shows all the steps that a customer can do on this web site Driver needs to log in using Email and Password. The Driver can Look for Hires Accept them and view all the past hires the driver has gone. And the Driver can logout lastly

### 2.2.3 Admin Sequence Diagrams

Graphical user interface, application, table

Description automatically generated

Figure 6 Admin Sequence Diagrams

The Admin Sequence Diagram shows all the steps that a customer can do on this web site Admin needs to log in using Email and Password. Admin can Add Drivers and Vehicles to the System and manage Users, Bookings, and Vehicle Manage = View, Update and Delete.

# Task C

## Design Patterns

There are 3 types of design patterns

1. Creational
2. Structural
3. Behavioral

A design pattern offers an all-encompassing, reusable answer to the typical issues that arise in software design. Usually, the pattern demonstrates the connections and interactions between various classes or objects.

### Creational

Creational design patterns in software engineering deal with object creation mechanisms and attempt to generate objects in a situation-appropriate way. Two main concepts make up creational design patterns. One is storing information about the concrete classes the system use. Another is to conceal the process of creating and combining instances of these concrete types.

Apply Creational Patterns when.

* A system should be unconstrained by the processes used to make its constituent parts.
* A group of associated items is intended to be utilized collectively.
* A class library or product's implementations are concealed, leaving only their interfaces visible.
* Building various representations of separate complex items.
* A class expects the object it creates to be implemented by its subclass.
* At run-time, the class instantiations are specified.
* There must be just one instance, and clients must always have access to it.
* The instance should be able to grow without changing. (Gupta, 2022)

Text

Description automatically generated

Figure 7 Creational Design Pattern

### Structural Design Pattern

A structural design pattern is a guide for combining various objects and classes to create a larger structure that can accomplish several objectives at once. The patterns in structural designs demonstrate how distinctive system components can be put together in a flexible and extensible way. Therefore, we can target and modify a specific portion of the structure with the use of a structural design pattern without altering the entire structure. Structural design patterns facilitate the design process by outlining an easy method to construct entity relationships. Different kinds of structural design patterns exist. Adaptor, bridge, composite, and proxy are among the samples.

Text

Description automatically generated

Figure 8 Structural Design Pattern

Types of Structural design patterns

These are only a few of the structural design patterns that are used commonly

* Adapter Pattern
* Bridge Pattern
* Composite Pattern
* Decorator Pattern
* Facade Pattern
* Flyweight Pattern
* Proxy Pattern

### Behavioral Design Pattern

Behavioral design patterns are patterns that recognize typical communication patterns between objects and implement them in software engineering. These patterns boost this communication's flexibility by doing so.

Text

Description automatically generated

Figure 9 Behavioral Design Pattern

# Task D

## Git Hub Link:

[emalyaditha/GoCheeta (github.com)](https://github.com/emalyaditha/GoCheeta)

To make this system I have used

* IntelliJ IDEA Ultimate Version 2022.2
* Selenium IDE (Unit Testing)
* Spring Boot
* Thymeleaf

# Task E

## Test Plan - Customers

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter your details in the register form | Email – [tenz@gmail.com](mailto:tenz@gmail.com)  First Name – Tyson  Last Name – Tenz  Password – 11111  Phone No – 1111111111 | Customers Login Page | Customers Login Page |
| Enter Correct Email and Password | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Password - 11111 | Show Customer Home page | After entering the correct email and password it shows a bootstrap link after coming back and trying again will send you to the Customer Home page. (Don’t know the reason for the error) Most probably it occurs because of version issues |
| Enter Incorrect Email and Password | Email – [kirito11@gmail.com](mailto:kirito11@gmail.com)  Password - 00000 | Load the Login page again | Load the Login page again |
| Click Customer Add Booking | - | Show Add Booking Page | Show Add Booking Page |
| Customer Add Booking | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Destination – Delkanda  Starting Location – Wijerama  Vehicle Branch – Nugegoda  Vehicle Name - BMW | Booking Successfully Added | Booking Successfully Added |
| Click My Bookings | - | Show his/her past booking. | Show his/her past booking. |

Table 2 Test Plan - Customers

## Test Plan – Driver

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter Correct Email and Password | Email – [mac@gmail.com](mailto:mac@gmail.com)  Password - 11111 | Show Driver Home page | After entering the correct email and password it shows a bootstrap link after coming back and trying again will send you to the Driver Home page. (Don’t know the reason for the error) Most probably it occurs because of version issues |
| Click View Hires | - | Show all Hires Page | Show all Hires Page |
| Click Accept a hire | Email – mac@gmail.com | Add the hire to My Hires | Add the hire to My Hires |
| Click View My Hires | - | Show My Hires page | Show My Hires page |

Table Test Plan – Driver

## Test Plan – Admin

|  |  |  |  |
| --- | --- | --- | --- |
| Test Scenario | Test Data | Expected Results | Actual Results |
| Enter Correct Email and Password | Email – [admin@gmail.com](mailto:admin@gmail.com)  Password - 11111 | Show Admin Home Page | Show Admin Home page (Admin is hard coded to the system) |
| Click Add Driver | - | Show Add Driver Page | Show Add Driver Page |
| Add a Driver | Email – [asuna@gmail.com](mailto:asuna@gmail.com)  First Name – Asuna  Last Name – Yuuki  Password – 11111  Phone No – 0000000000  Branch – Nugegoda  Car - Saab | Successfully Add Driver | Successfully Add Driver |
| Add a Vehicle | Number Plate -FAF 1111  Model – Eclipse  Name - Mitsubishi | Successfully Add Vehicle | Successfully Add Vehicle |
| Click Admin Manage User | - | View Admin Manage User Page with All User Details | View Admin Manage User Page with All User Details |
| Click the link to Update User | Email – [asuna@gmail.com](mailto:asuna@gmail.com)  First Name – Asuna  Last Name – Yuuki  Password – 00000  Phone No – 0701536969  Branch – Nugegoda  Car – Saab  Role - Driver | Successfully Updated User | Successfully Updated User |
| Click the link to Delete User | Id – U13 | Successfully Deleted User | Successfully Deleted User |
| Click Admin Manage Booking | - | View Admin Manage User Page with All Booking Details | View Admin Manage User Page with All Booking Details |
| Click the link to Update User | Email – [kirito@gmail.com](mailto:kirito@gmail.com)  Destination – Delkanda  Starting Location – Wijerama  Vehicle Branch – Nugegoda  Vehicle Name - BMW | Successfully Updated Booking | Successfully Updated Booking |
| Click the link to Delete Booking | Id - B13 | Successfully Deleted Booking | Successfully Deleted Booking |
| Click Admin Manage Vehicle | - | View Admin Manage Vehicle Page with All Vehicle Details | View Admin Manage Vehicle Page with All Vehicle Details |
| Click the link to Update Vehicle | Number Plate -FAF 0001  Model – Eclipse  Name - Mitsubishi | Successfully Updated Vehicle | Successfully Updated Vehicle |
| Click the link to Delete Vehicle | Id - 13 | Successfully Deleted Vehicle | Successfully Deleted Vehicle |
| Click logout | - | Send back to the index | Send back to the index |

Table 4 Test Plan – Admin

## Unit Testing

## User Login Test

Graphical user interface, text, application, email

Description automatically generated

Figure User Login Unit Test

## User Register Test

Graphical user interface, application

Description automatically generated

Figure User Register Unit Test

## User Add Booking

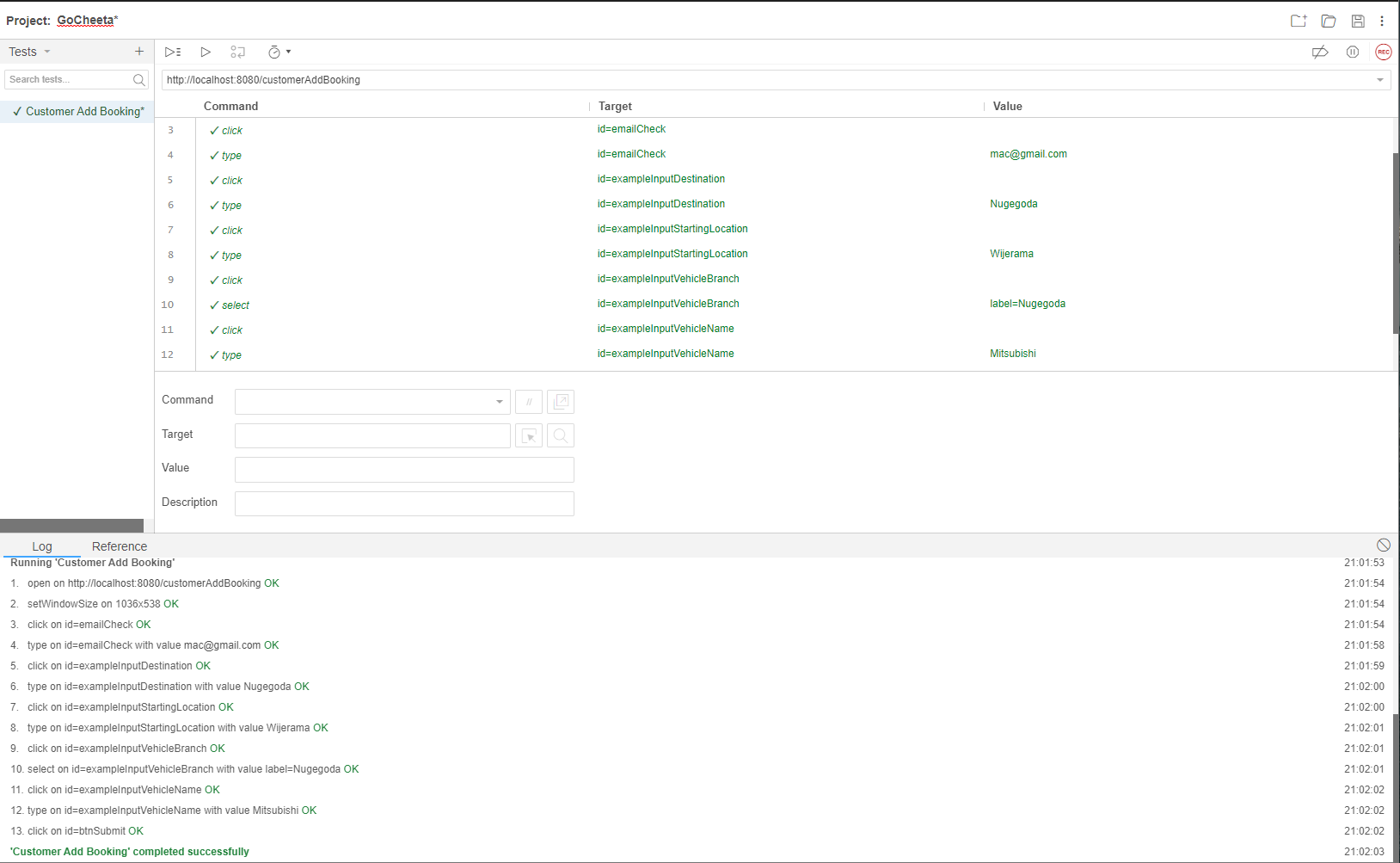


Figure User Add Booking Unit Test

## Driver Accept Booking

Graphical user interface, application

Description automatically generated

Figure Driver Accept Booking Unit Test

## Driver View Past Booking

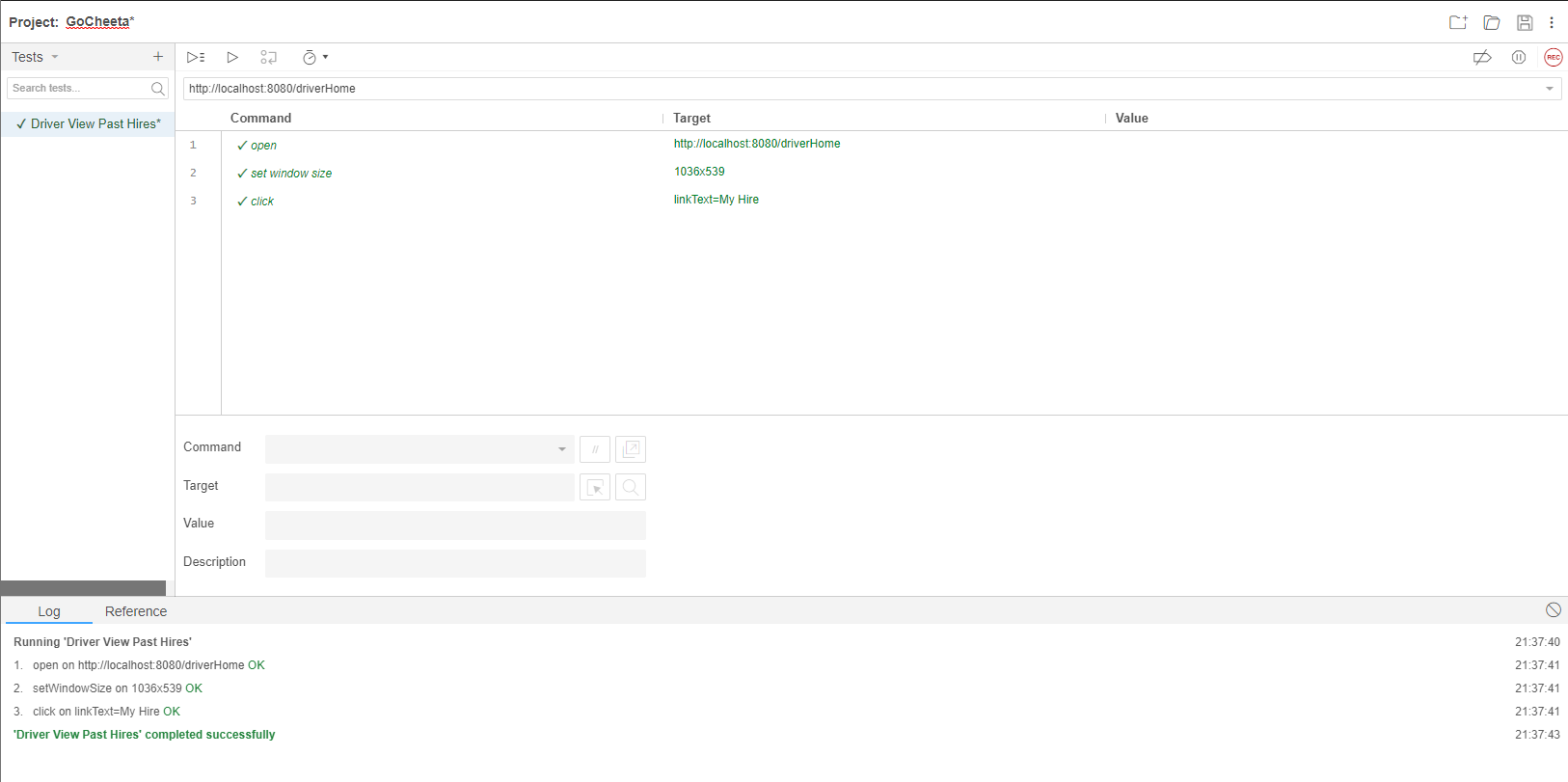


Figure Driver View Past Booking Unit Test

## Admin Login

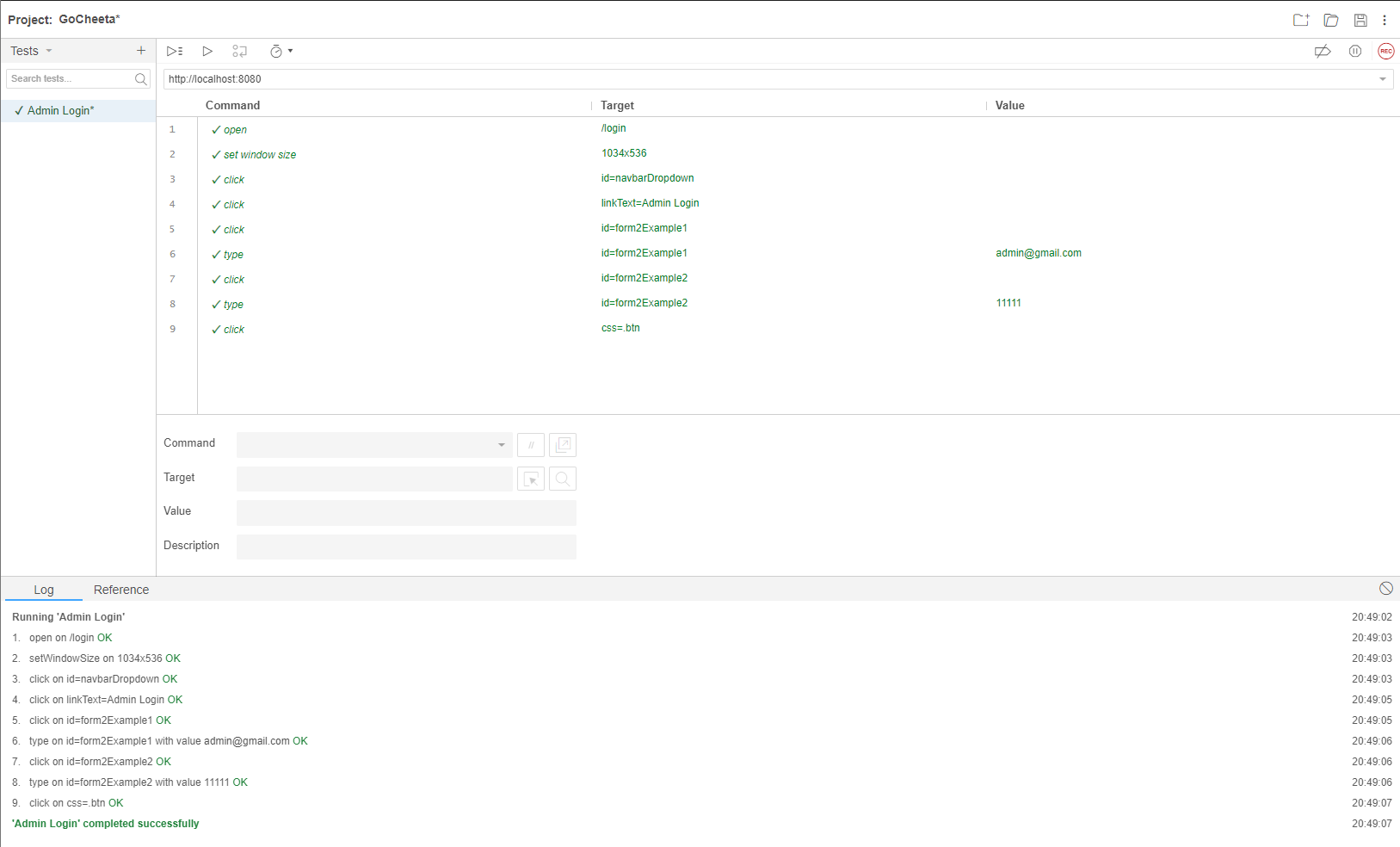


Figure Admin Login Unit Test

## Admin Add Driver

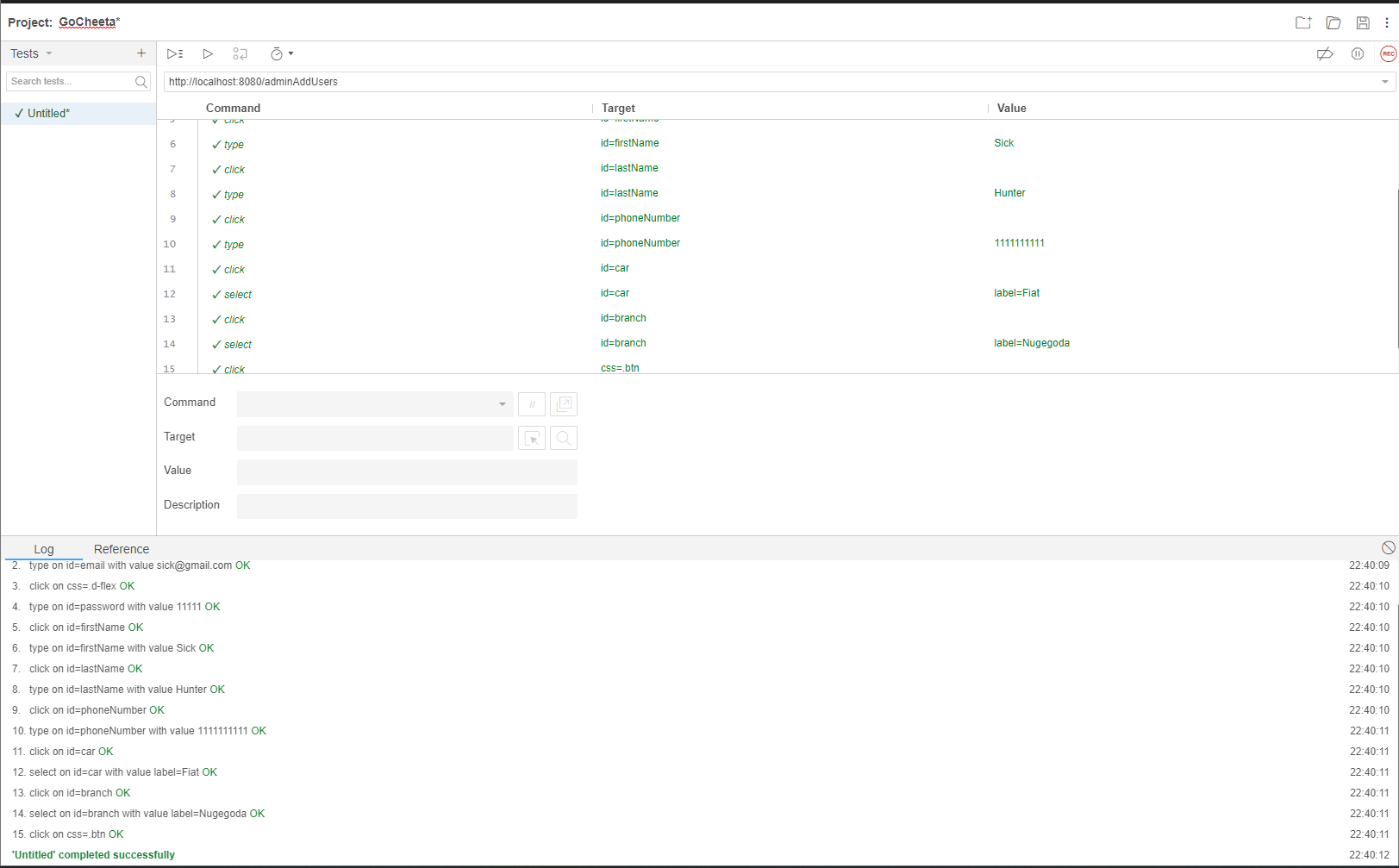


Figure Admin Add Driver Unit Test

## Admin Add Vehicle

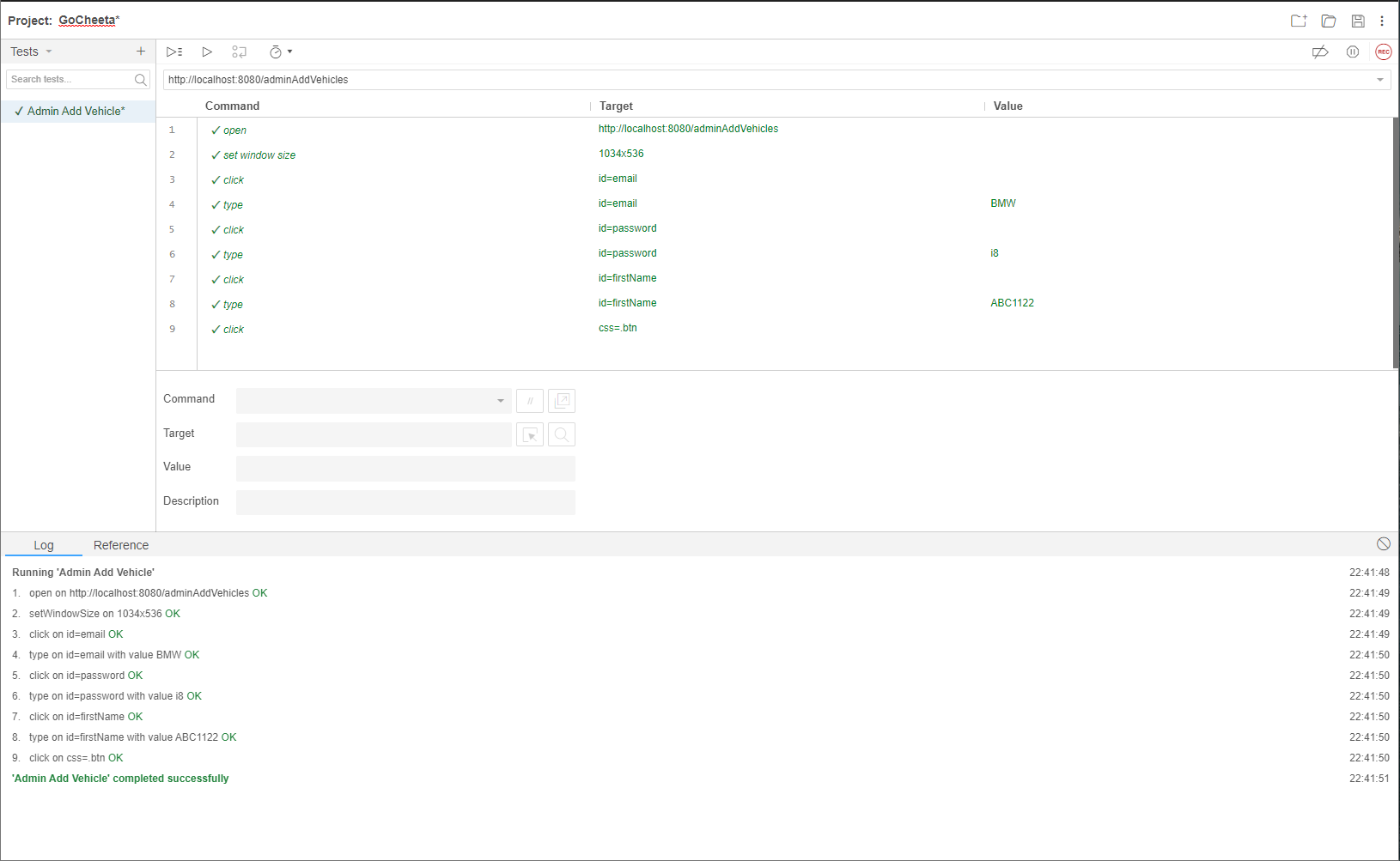


Figure Admin Add Vehicle Unit Test

## Admin Update User

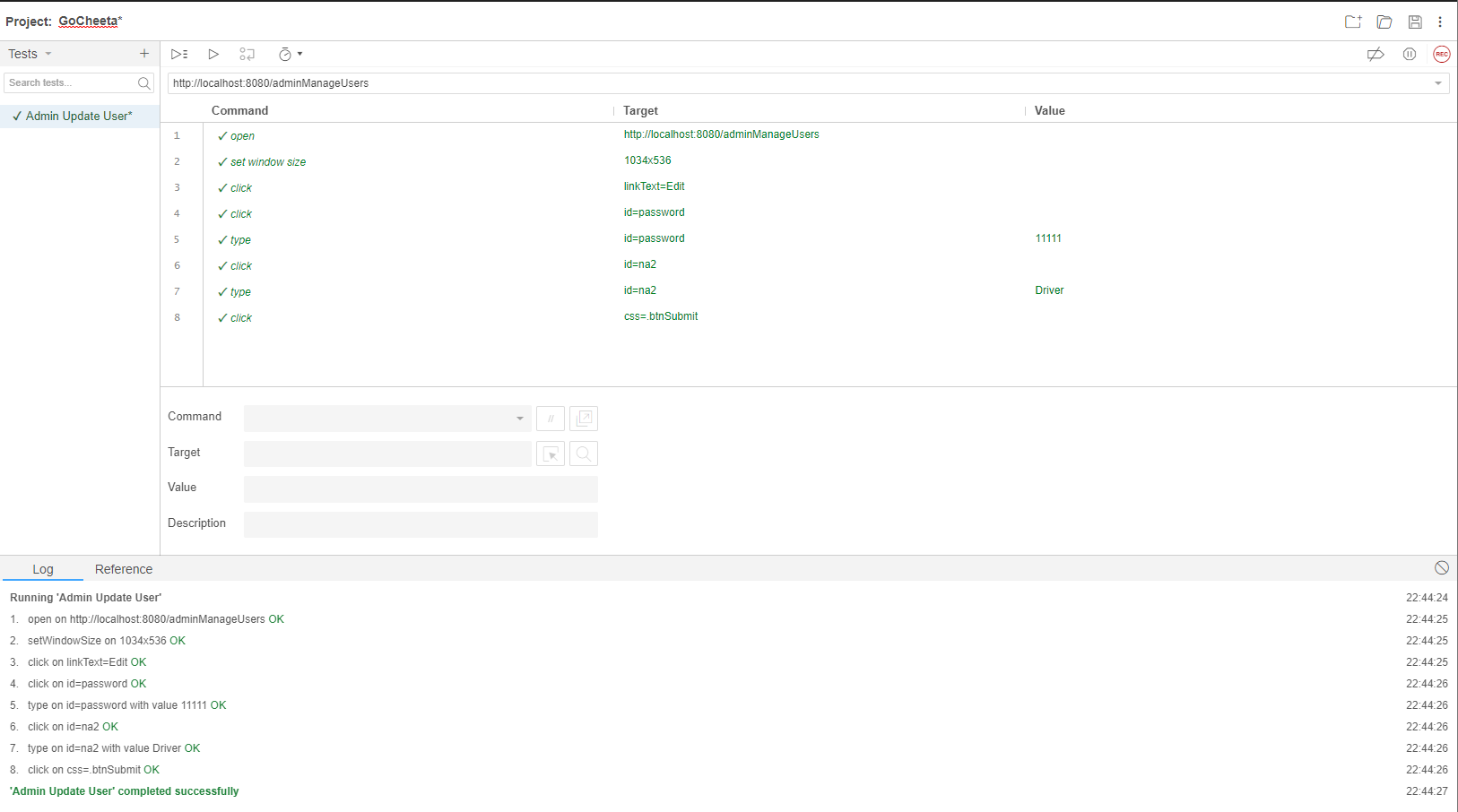


Figure Admin Manage User Unit Test

## Admin Update Booking

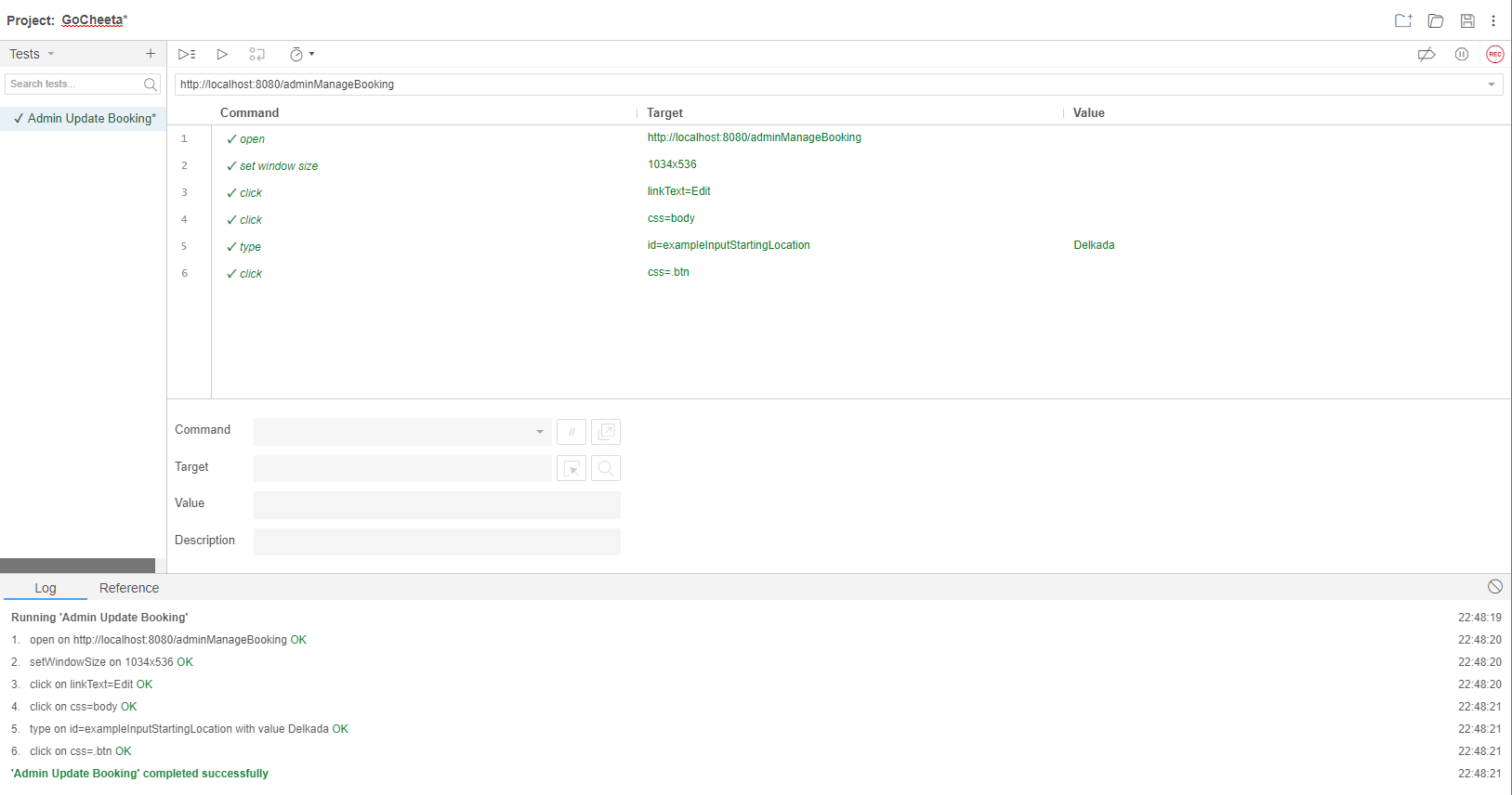


Figure Admin Manage Booking Unit Test

## Admin Update Vehicle

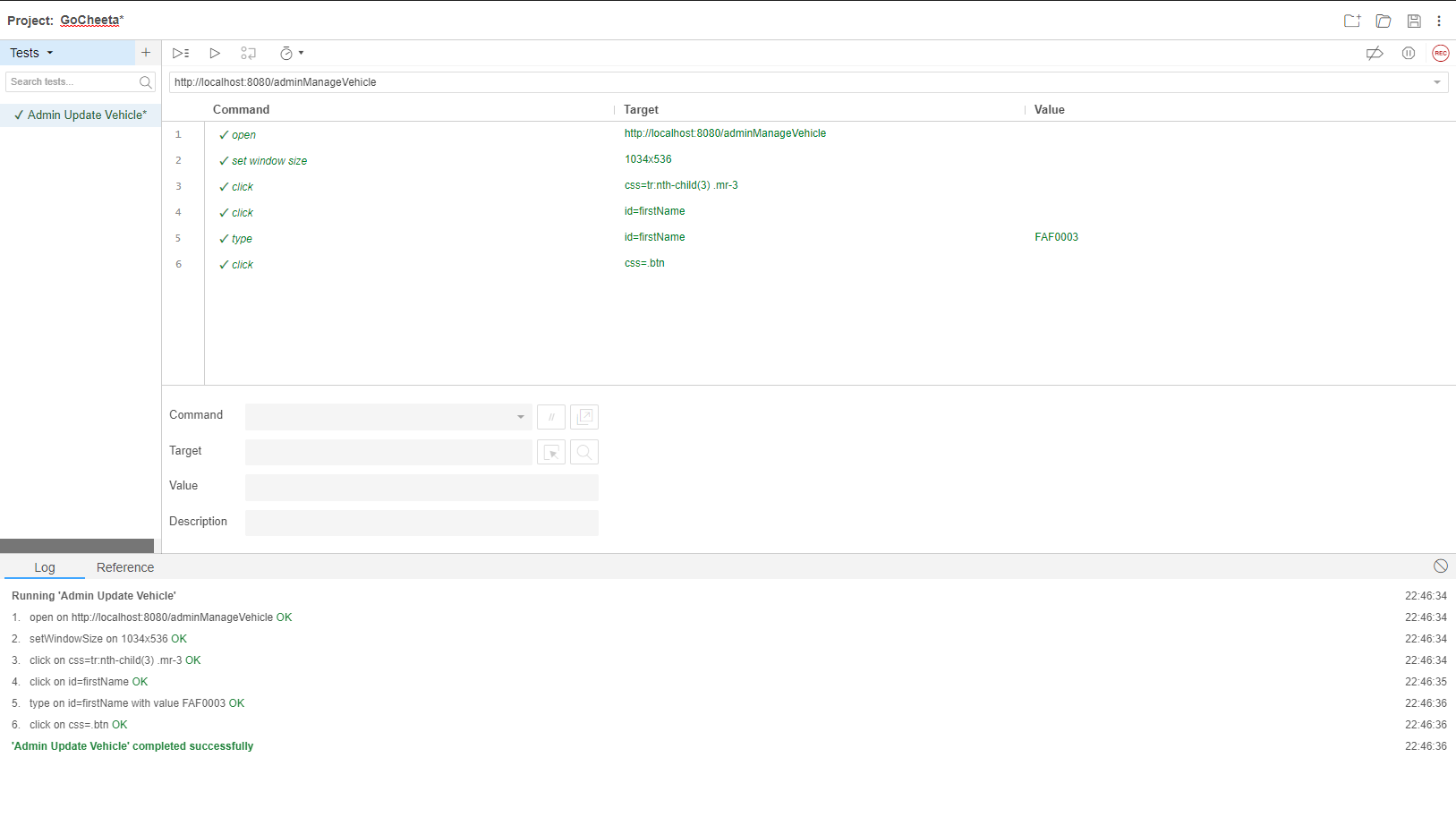


Figure Admin Manage Vehicle Unit Test

## Logout

Graphical user interface, application

Description automatically generated

# Task F

## User Manual

### Register

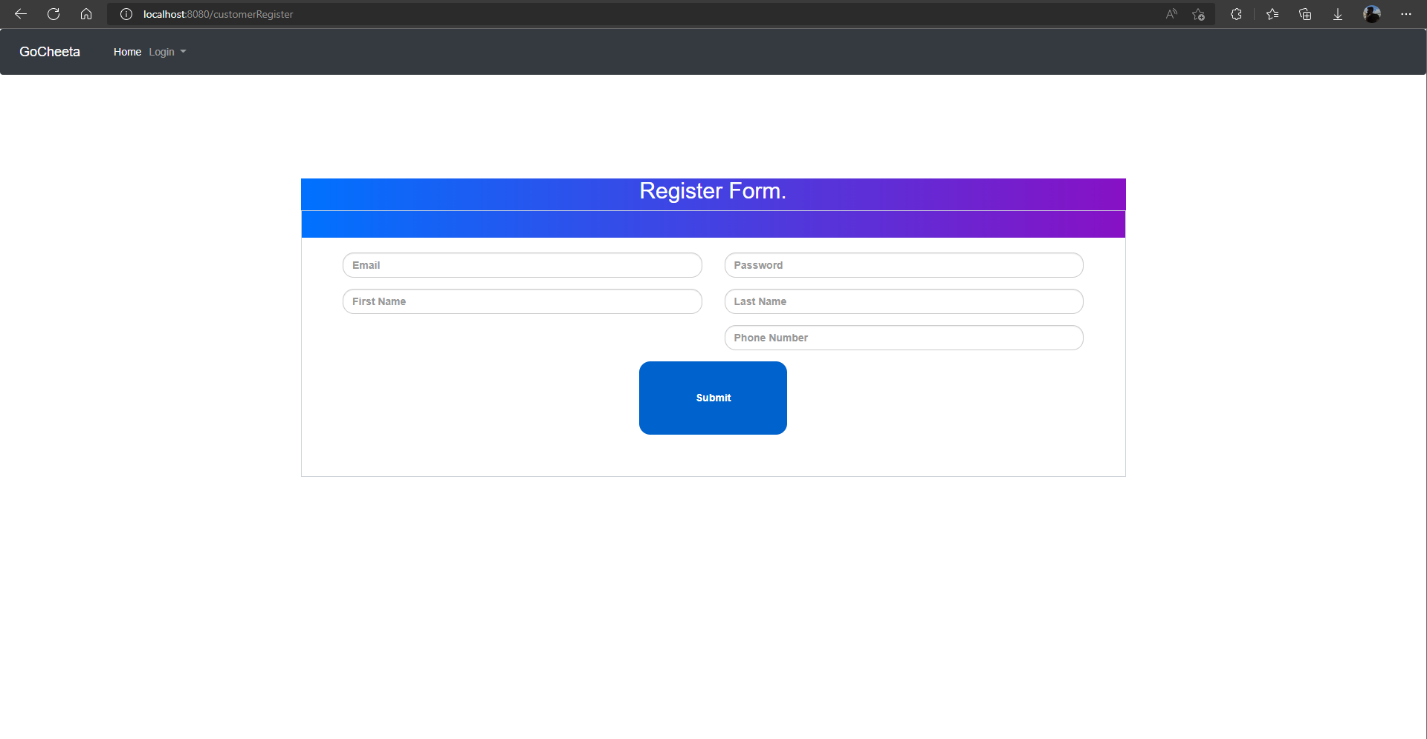


Figure User Manual Register

This Is the registration form that customers register them selfs to the system and then get access to log in.

### User Navigation Bar



Figure User Manual Navigation Bar

This is the Navigation Bar that is shown to the users who have not logged in to the system.

### User Login

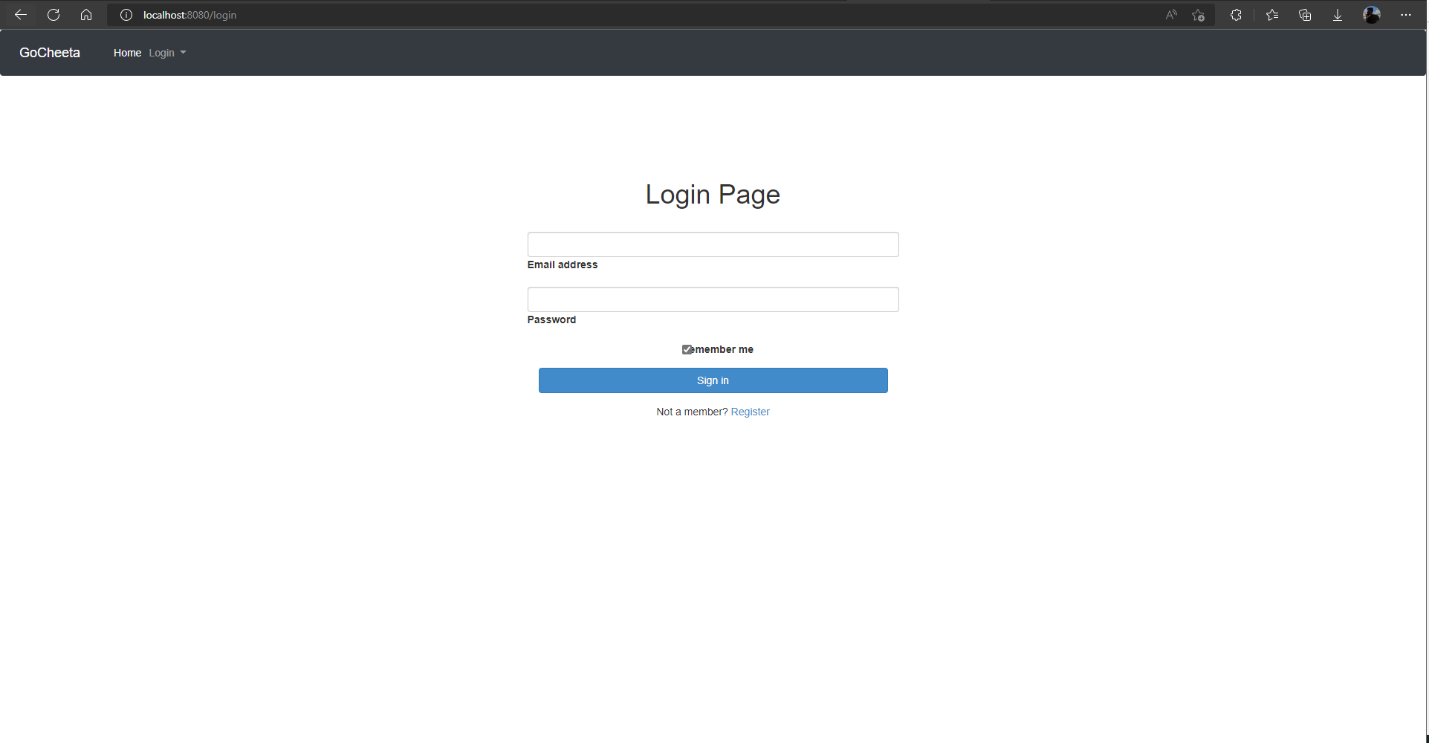


Figure User Manual User Login

This is the login page where Customers and Drivers Login to the System.

### Customer Navigation Bar

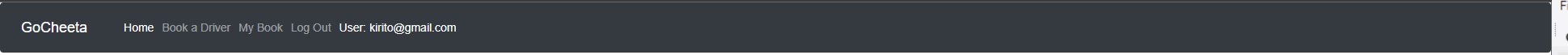


Figure User Manual Customer Navigation Bar

This is the Navigation Bar that is used by the Customers after logging in.

### Customer Home Page

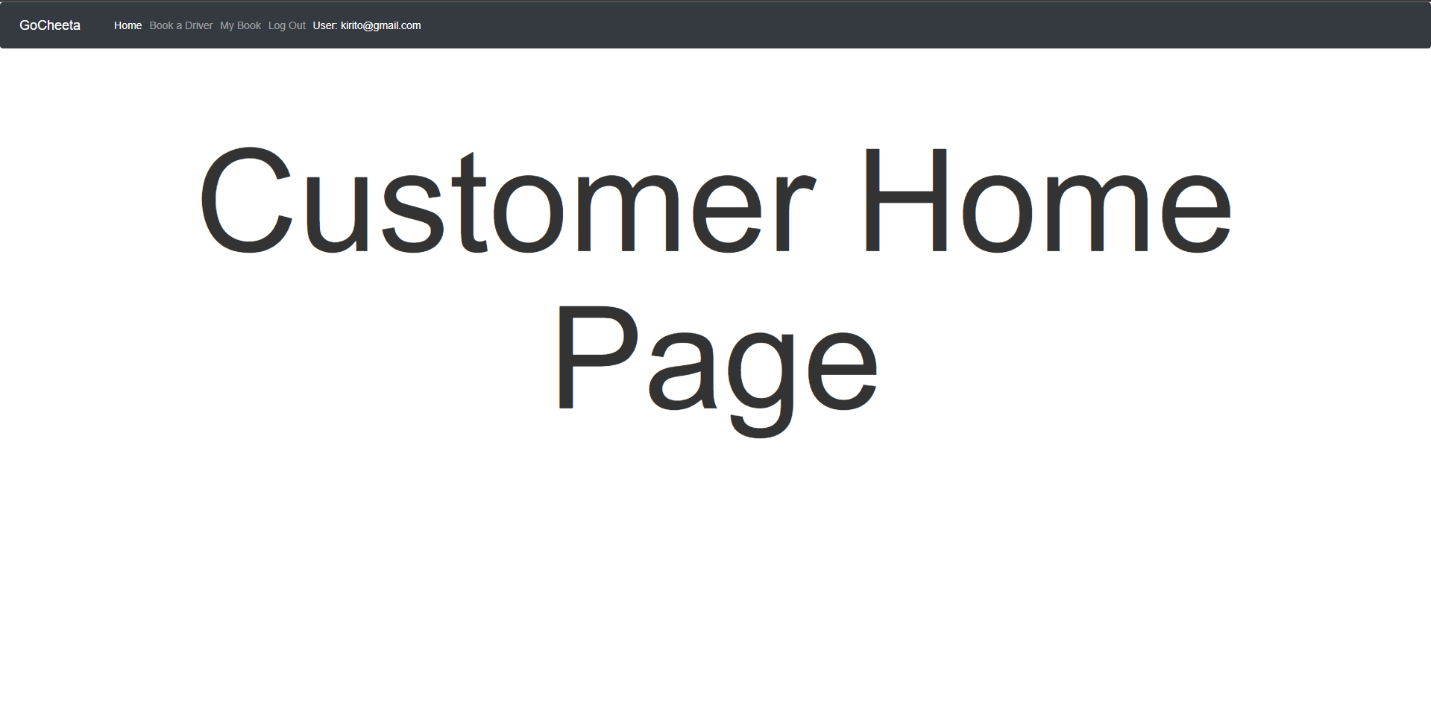


Figure User Manual Customer Home Page

After Logging in as a customer this is the home page that is redirected as the Customer Home page.

### Customer Add Booking Page

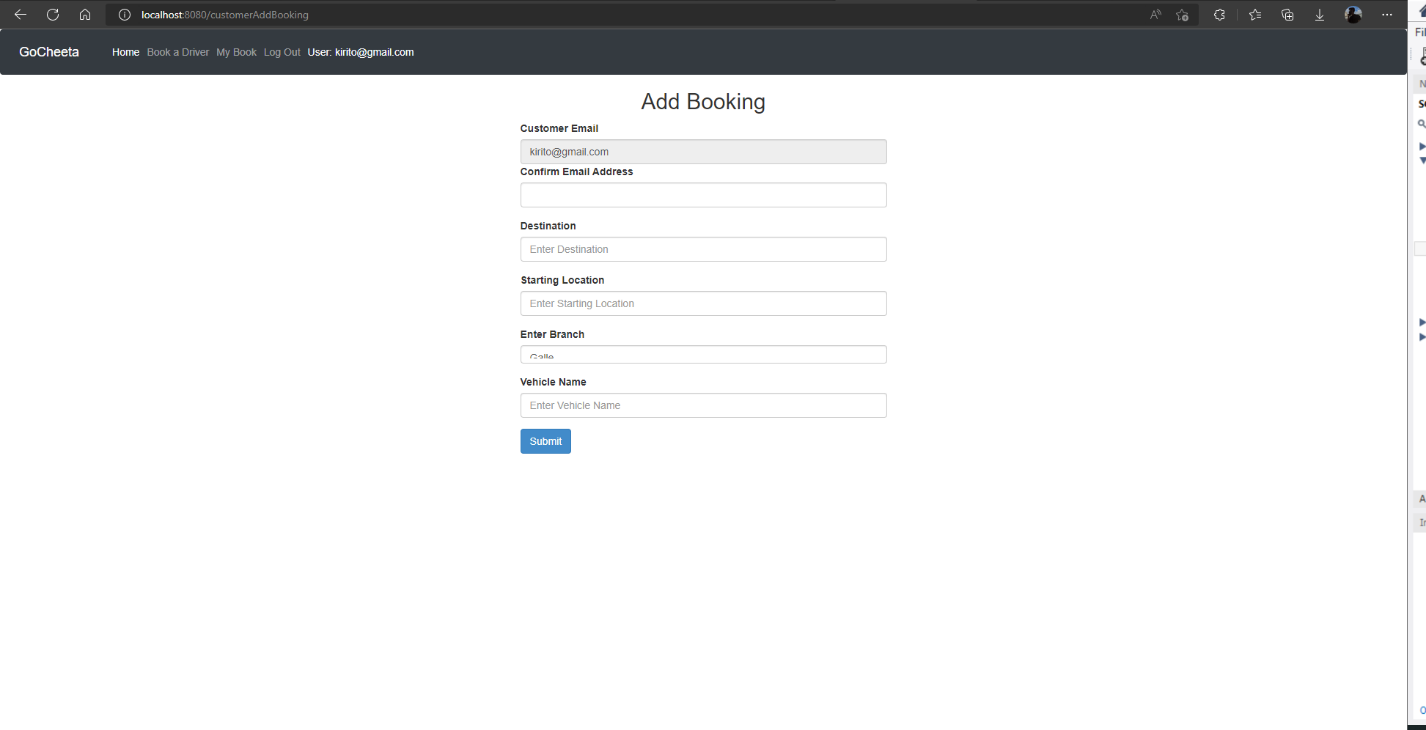


Figure User Manual Customer Add Booking

After the Customer log in this is the main function of the system where the customer adds a booking.

### Customers view their Bookings

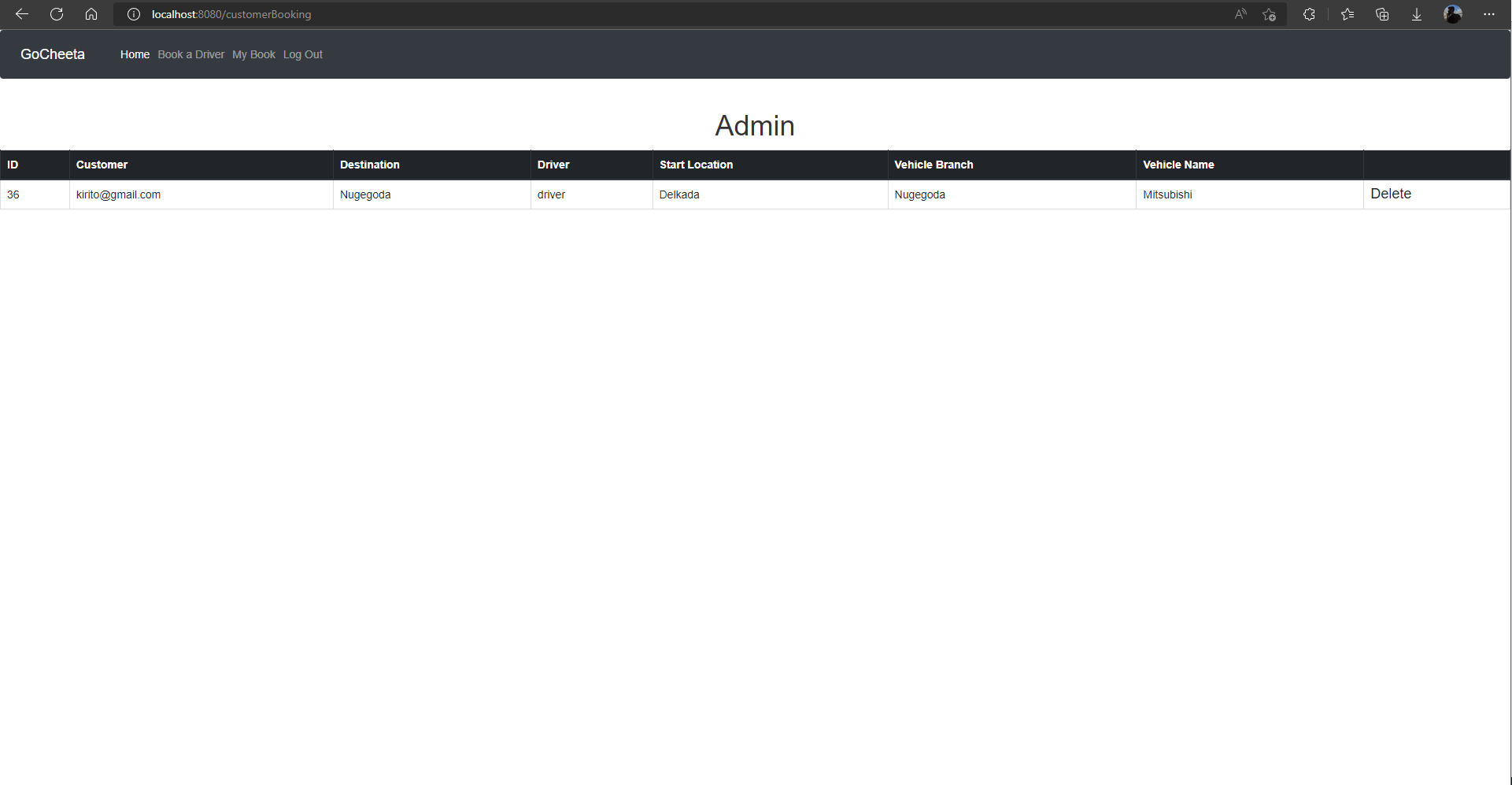


Figure User Manual Customer View Booking

The customer can view all his pending and past bookings from this view.

### Driver Home Page

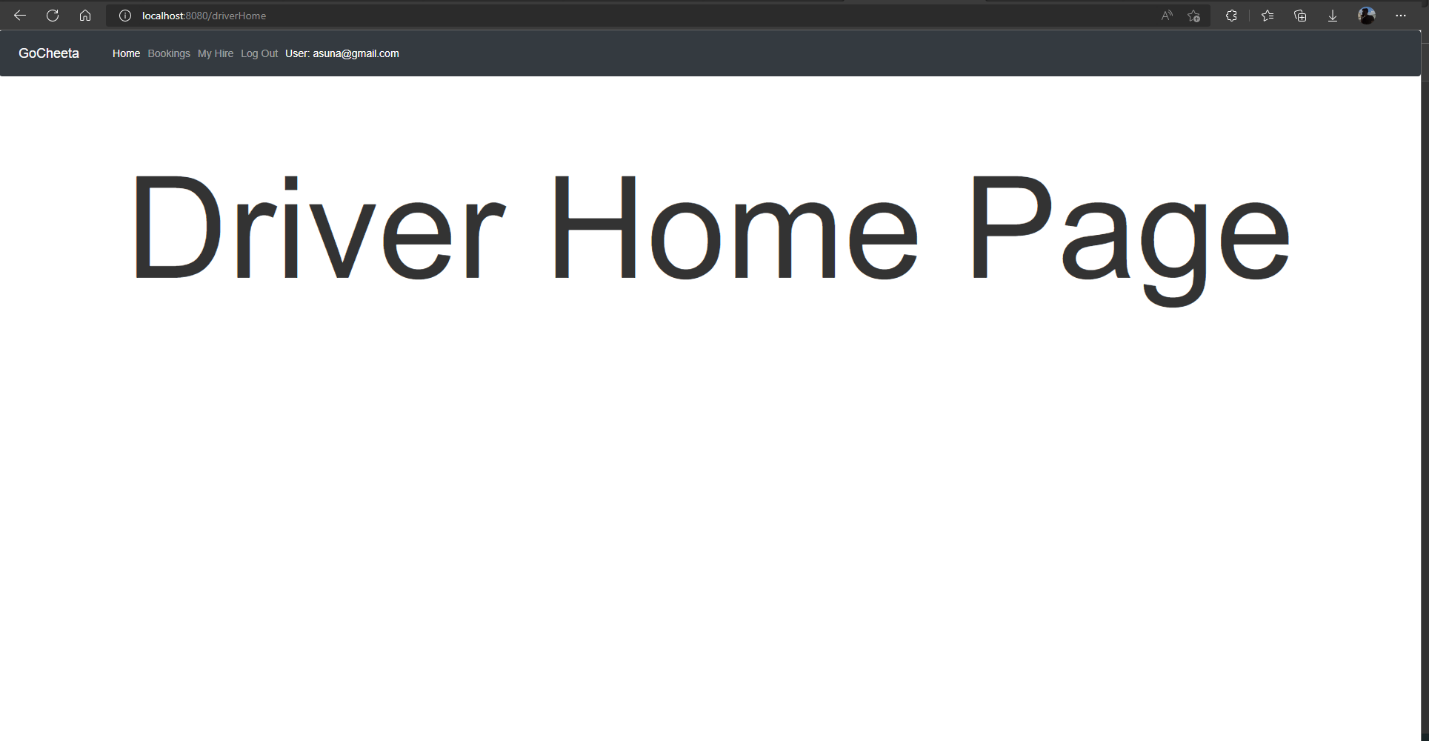


Figure User Manual Driver Home Page

This is the page that is shown when you are logged in as the Driver.

### Driver Navigation Bar

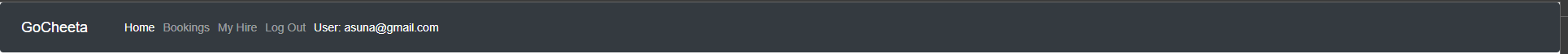


Figure User Manual Driver Navigation Bar

This is the Driver Navigation bar where the driver controls the system from the driver’s side.

### Driver View Customer Booking

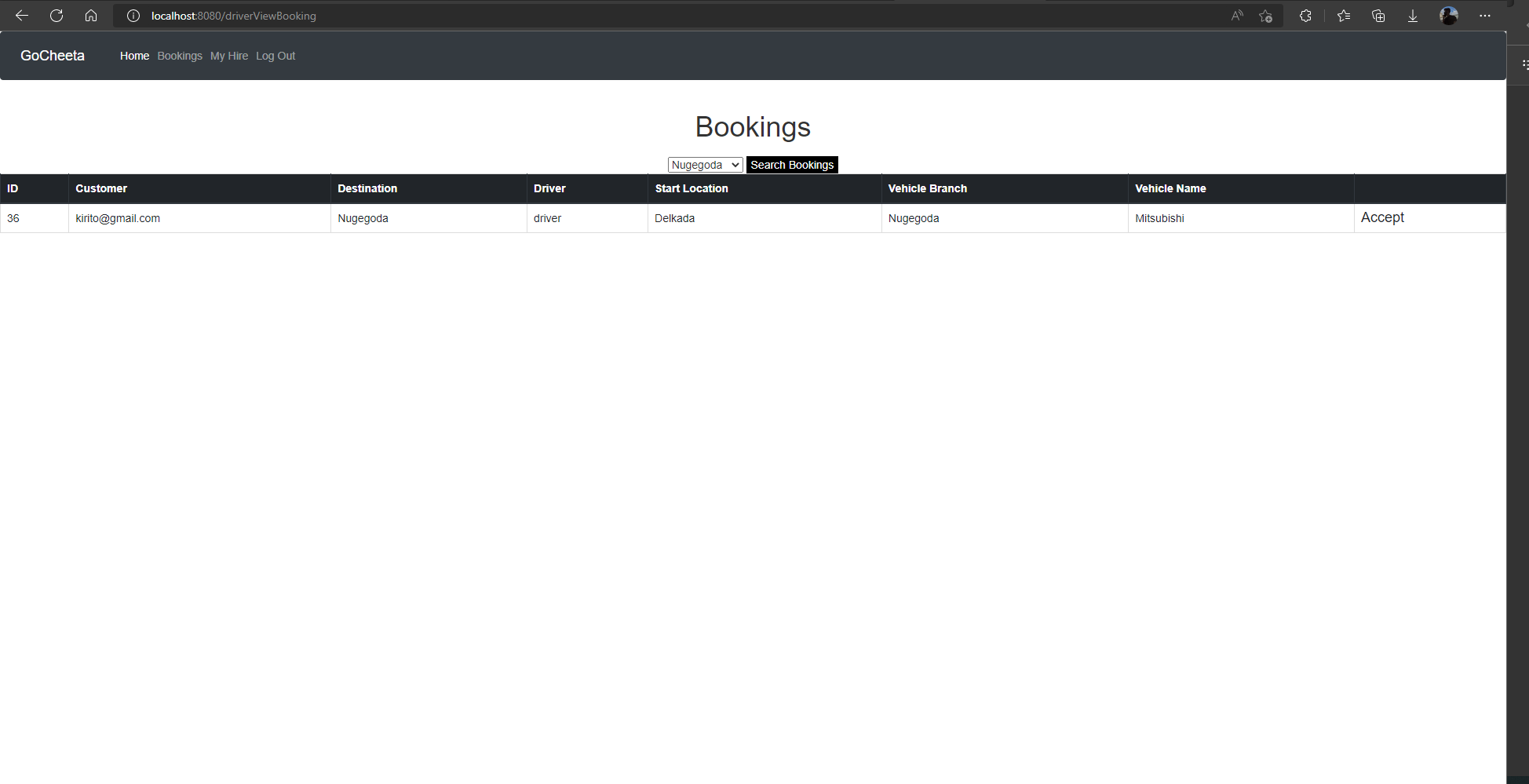


Figure User Manual Driver View Customer Booking

On this page the drivers can use the drop-down to select branches and there is a link on the right side so the driver can accept the booking if he needs.

### Driver Accept Booking

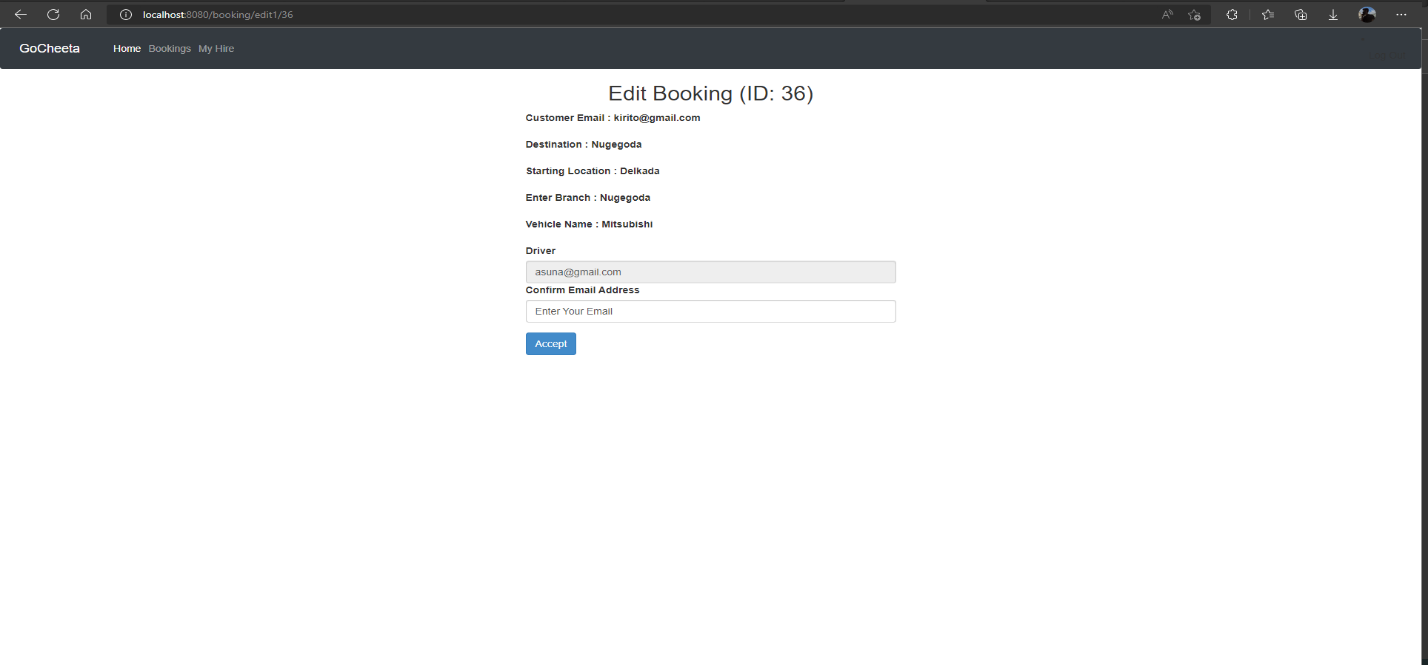


Figure User Manual Driver Accept Booking

After Clicking the accept link in driver view customer booking this is the page that is been loaded the driver needs to reenter the email to accept the login.

### Driver View Accept Booking

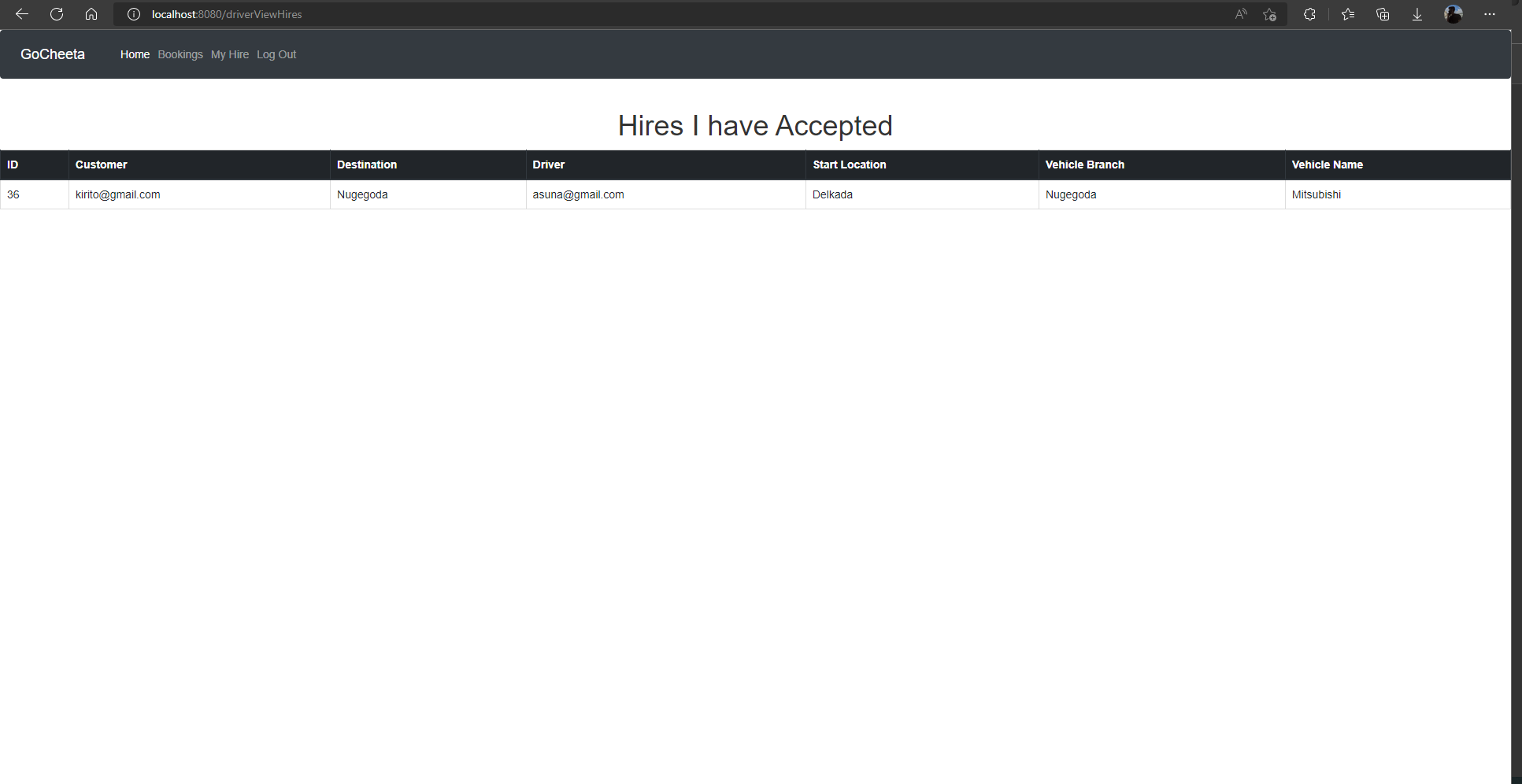


Figure User Manual Driver View Accept Booking

This page shows all the past and present bookings that the logged-in driver accepted.

### Admin Login Page

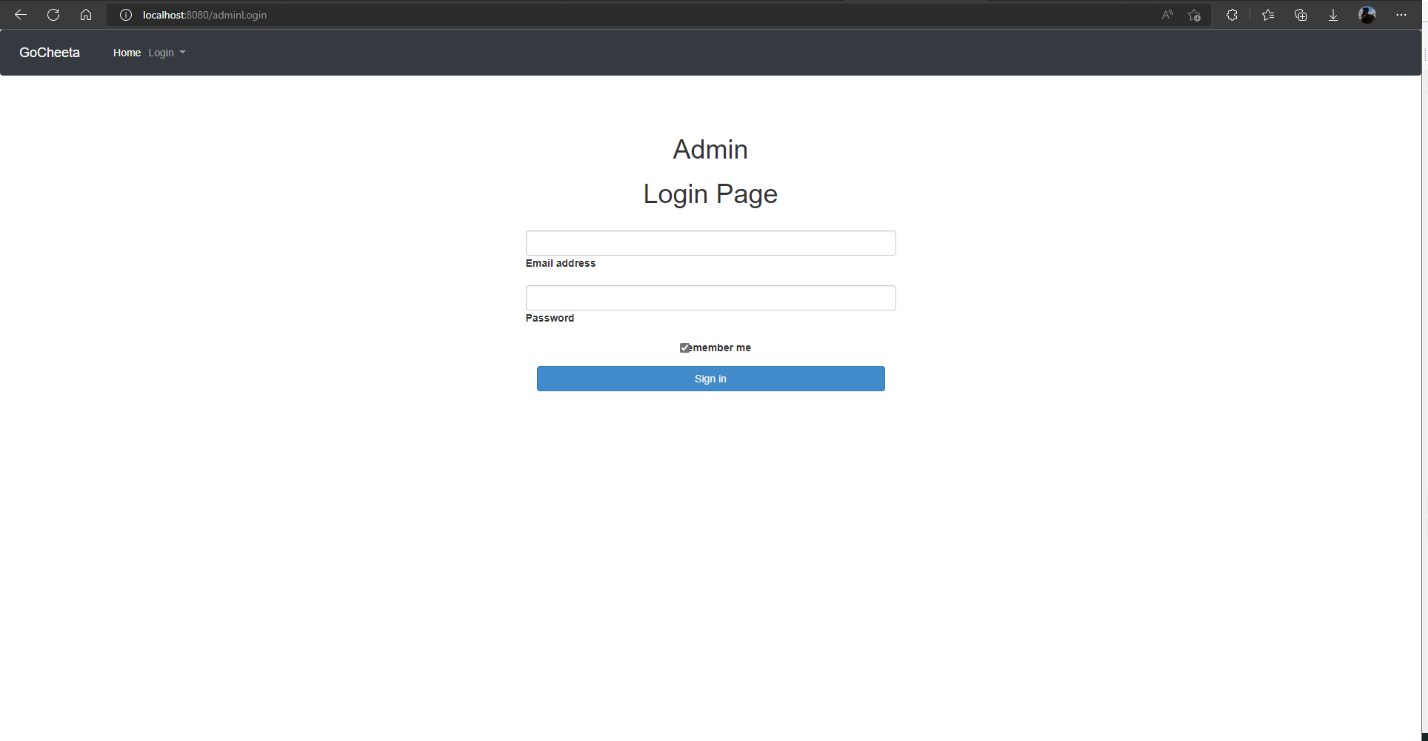


Figure User Manual Admin Login Page

This is the admin login page where the admin login to the system. Since the admin is hardcoded, I have used two logins one for the customer and driver and the other for the admin.

### Admin Home Page

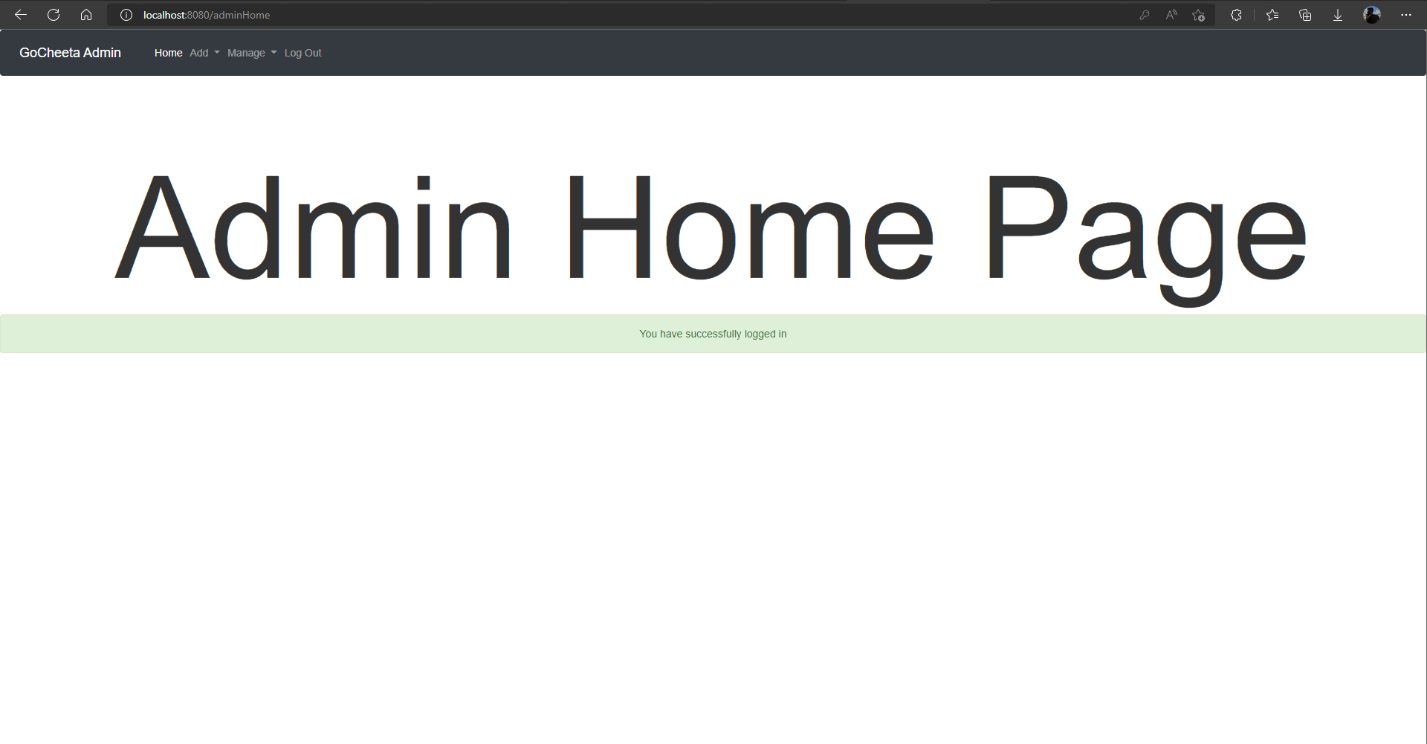


Figure User Manual Admin Home Page

After Admin Login to the system this is the home page that is been loaded.

### Admin Nav Bar



Figure User Manual Admin Navigation Bar

This is the Admin Navigation bar.

### Admin Add Driver

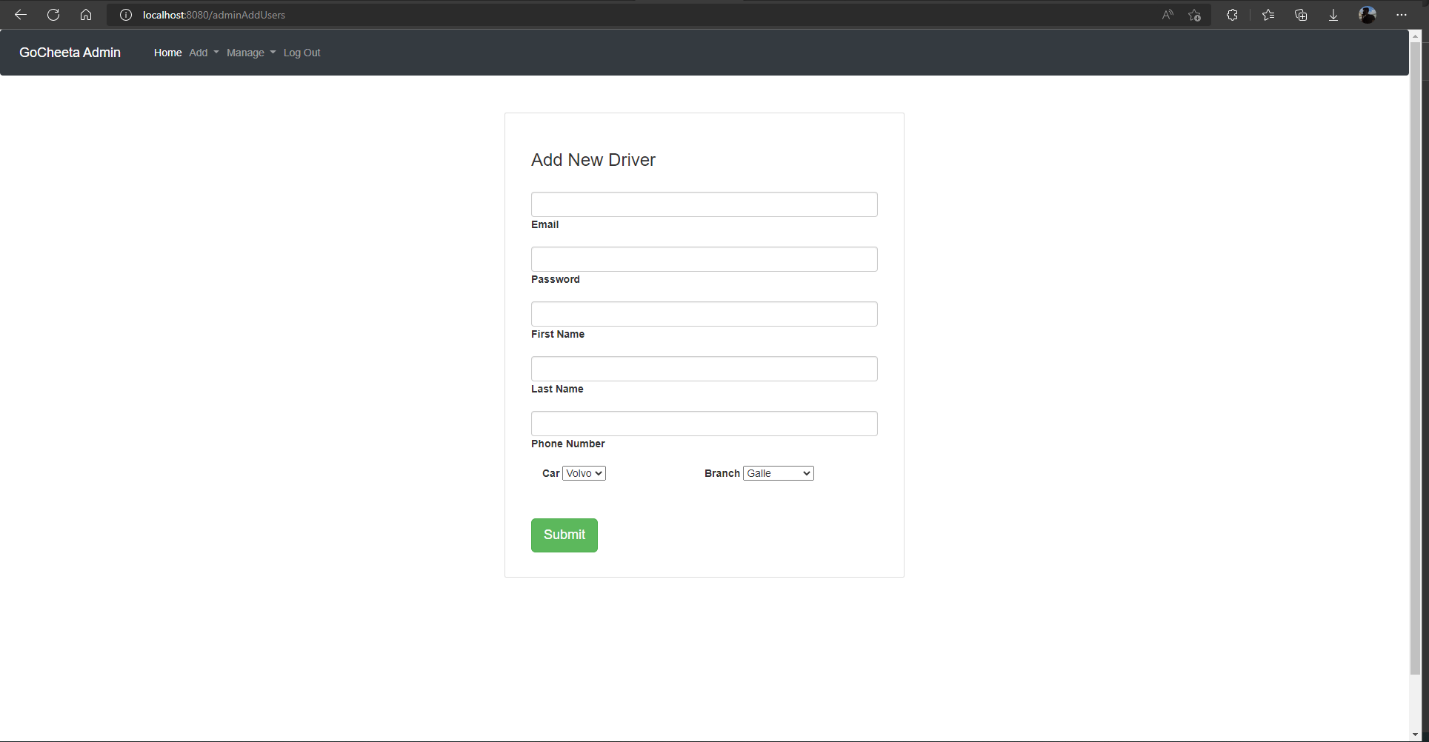


Figure User Manual Admin Add Driver

This Page is used by the admin to add Drivers to the system since the drivers can not register to the system.

### Admin Add Vehicle

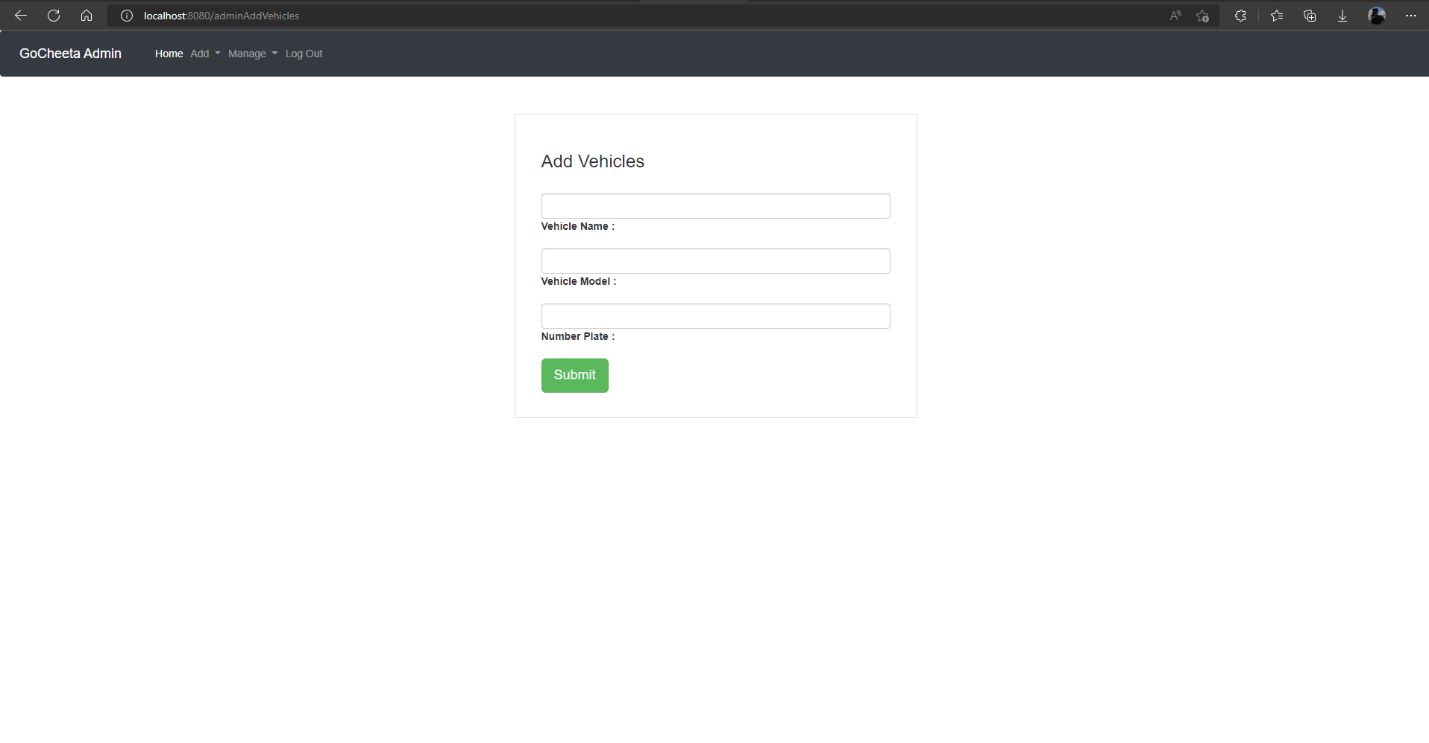


Figure User Manual Admin Add Vehicle

This Page is used by the admin to add vehicles to the system.

### Admin Manage User

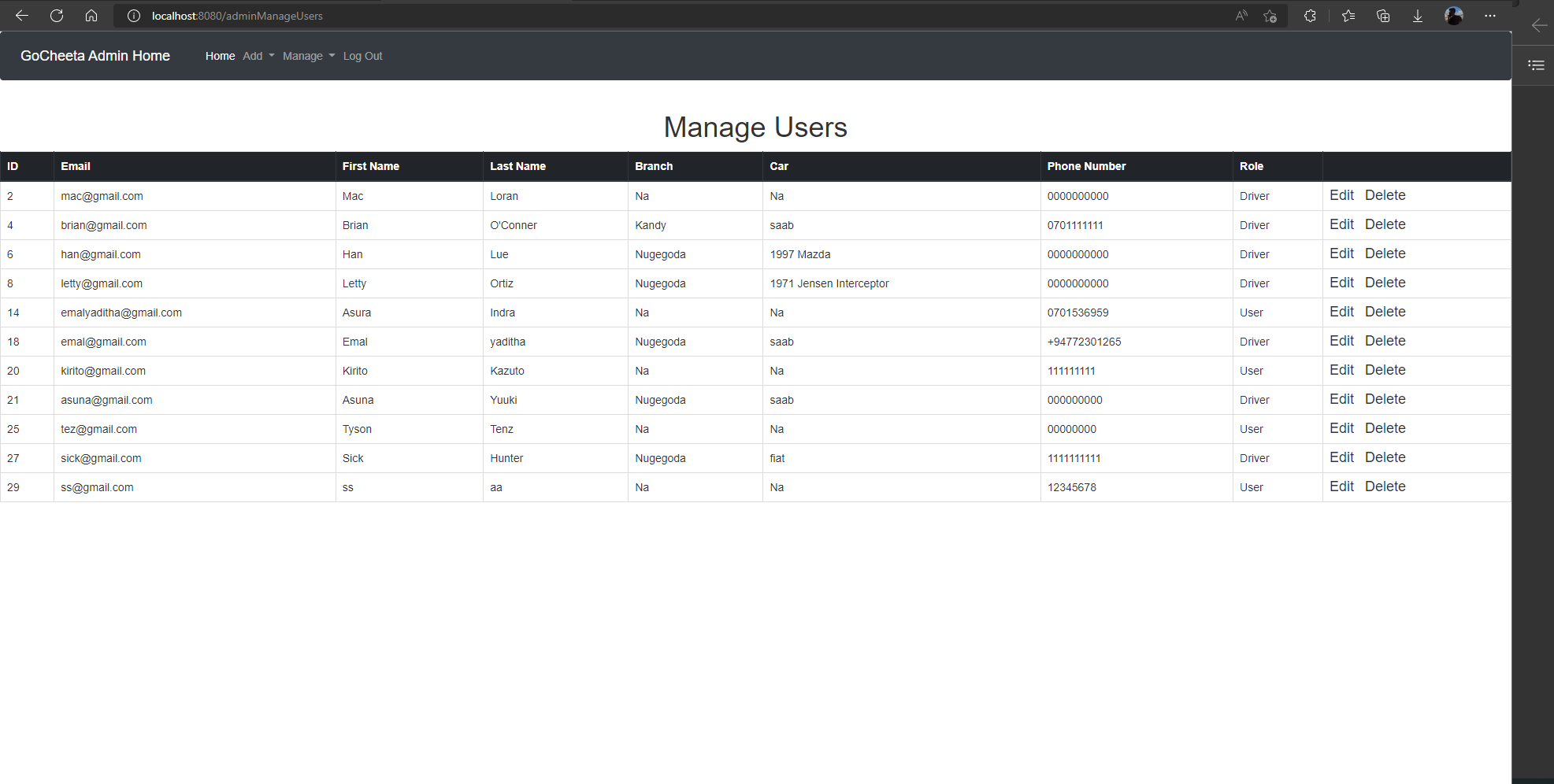


Figure User Manual Admin Manage User

Admin can view all the users from the system and use the links on the right side to update users and delete the user.

### Admin Update User

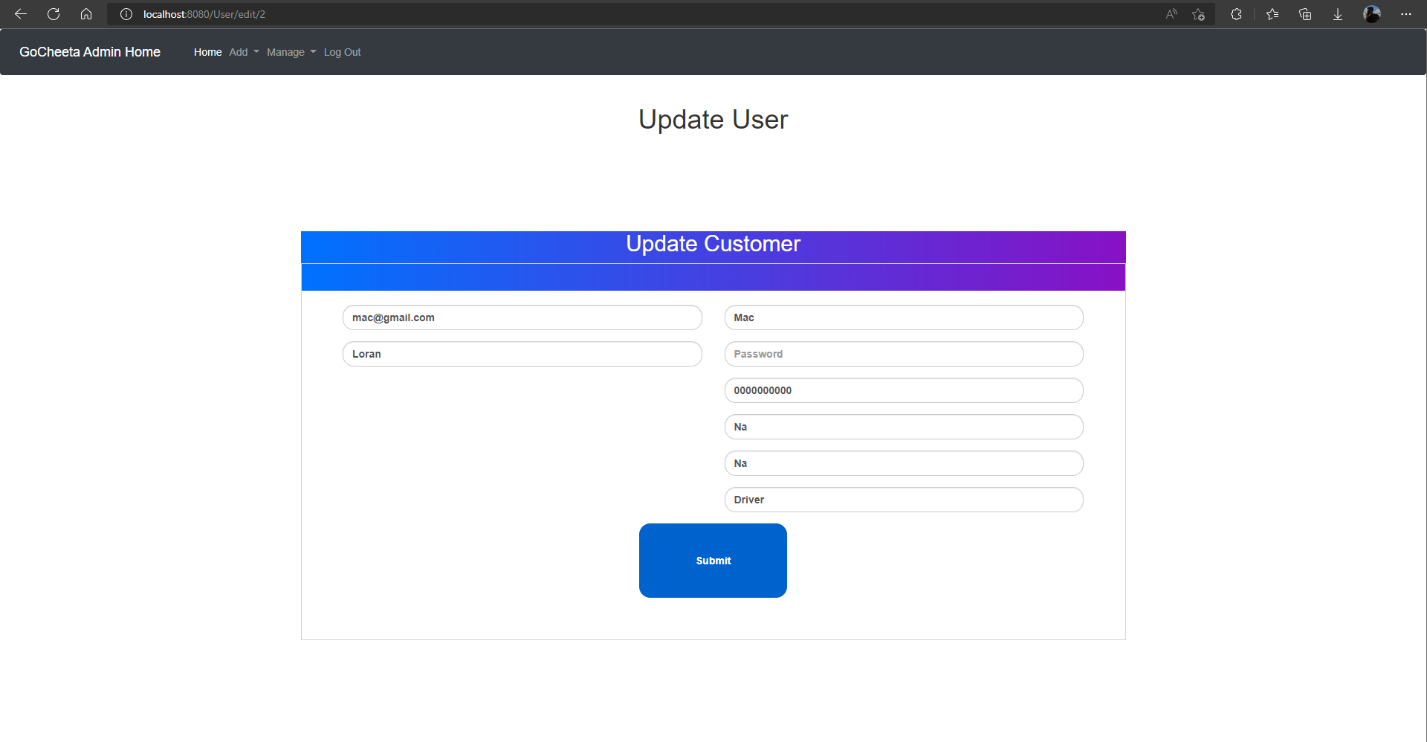


Figure User Manual Admin Update User

This form is loaded when the update link from the priviest page is clicked and after that admin can update user from the system.

### Admin Delete User

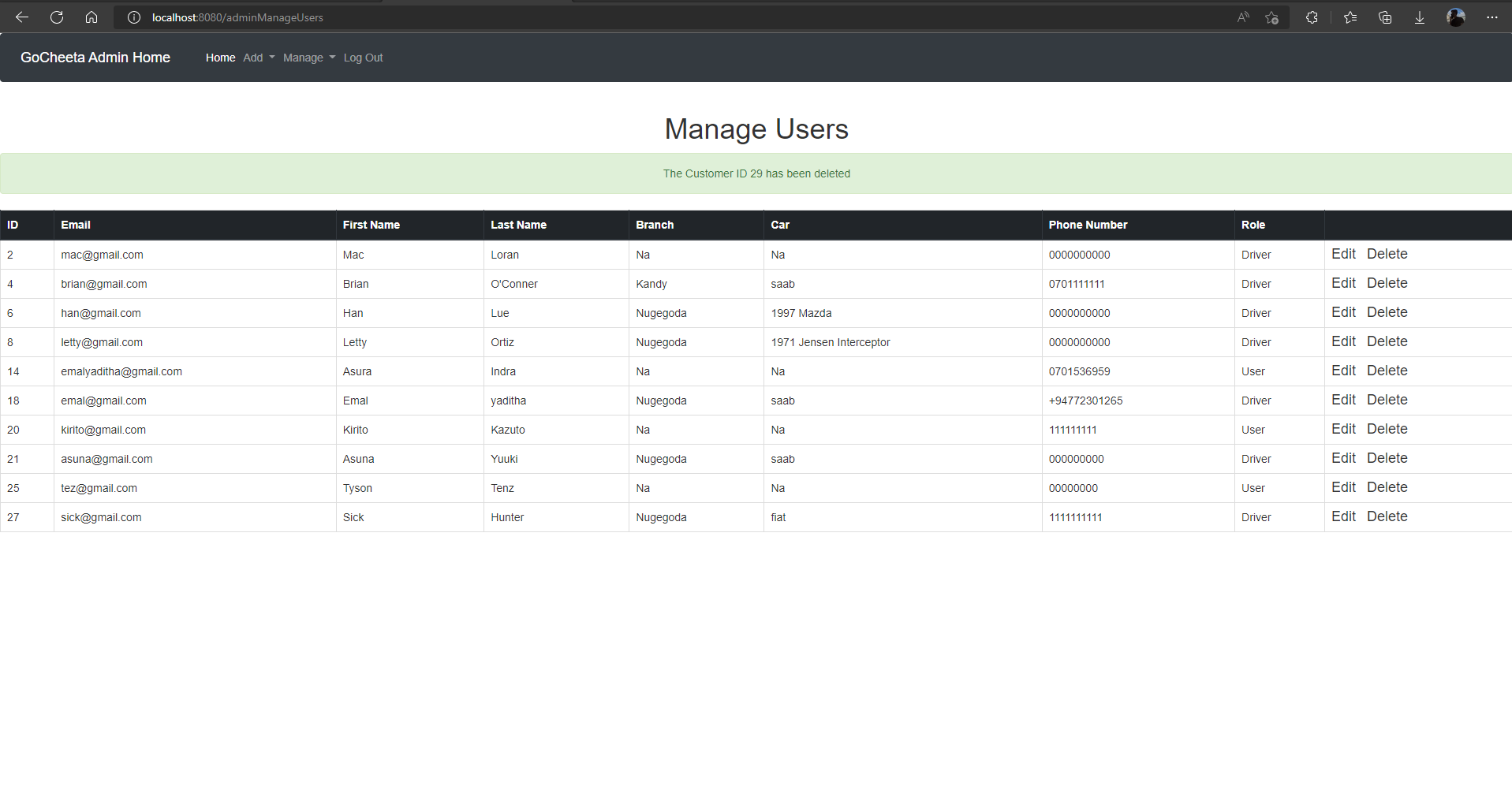


Figure User Manual Admin Delete User

This is what happens when the delete link is clicked by the admin.

### Admin Manage Vehicle

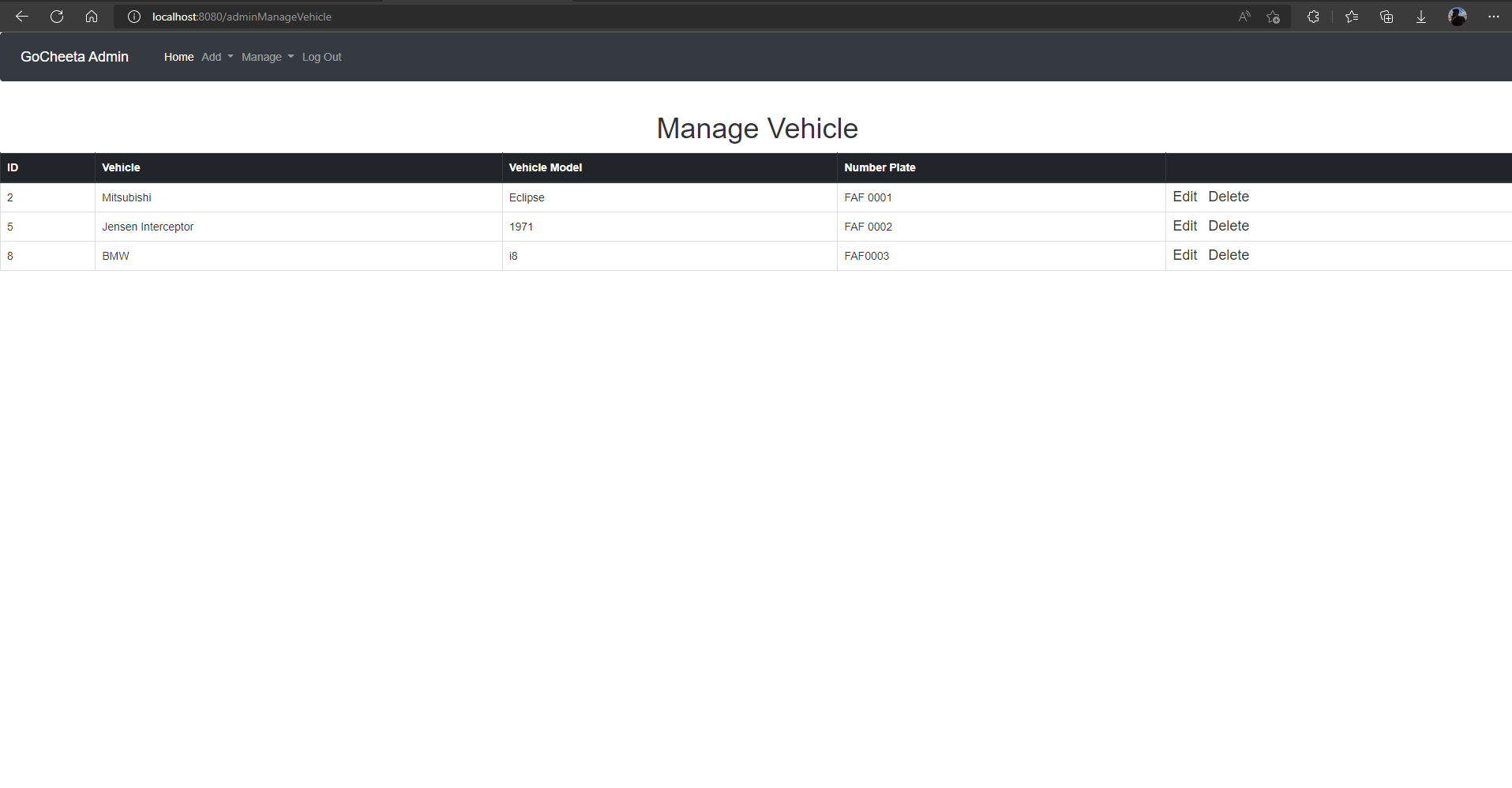


Figure User Manual Admin Manage Vehicle

Admin can view all the vehicle from the system and use the links on the right side to update vehicle and delete the vehicle.

### Admin Update Vehicle

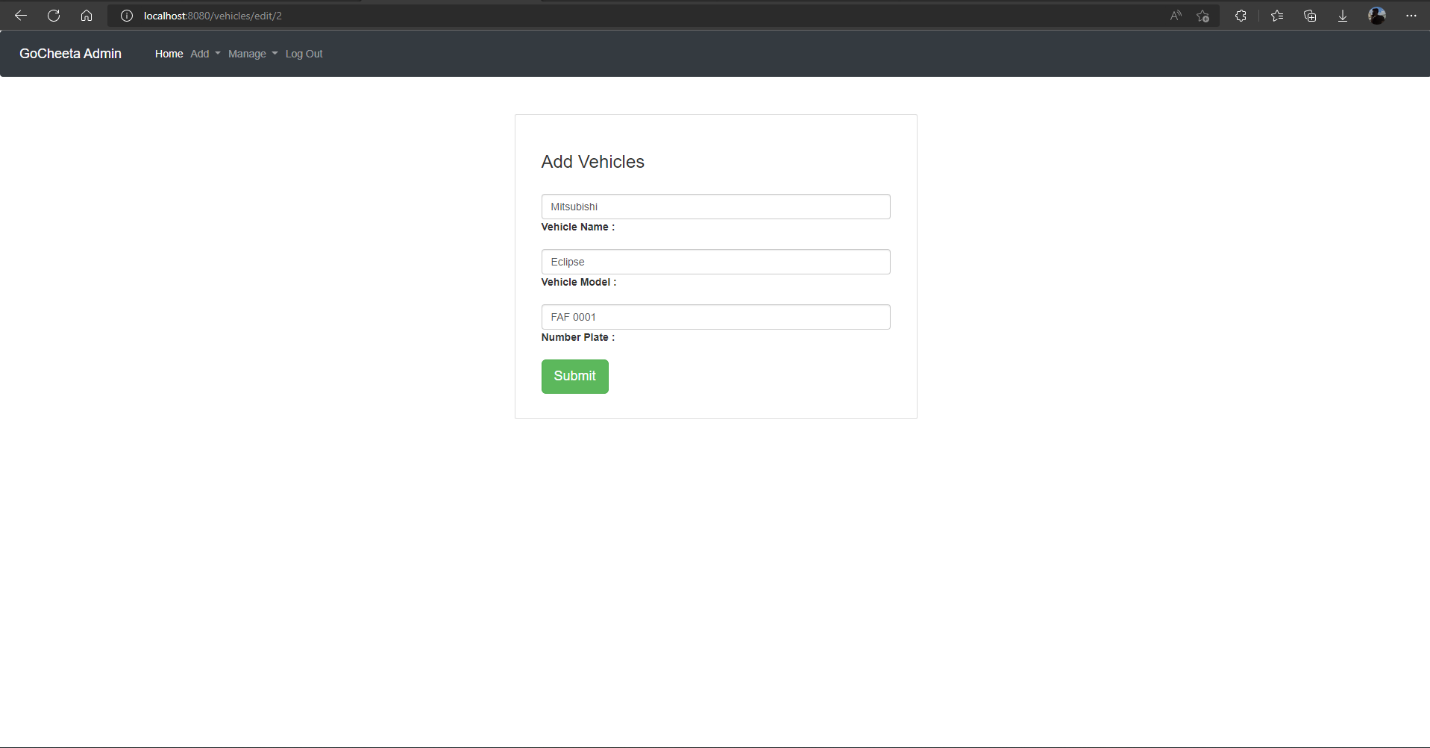


Figure User Manual Admin Update Vehicle

This form is loaded when the update link from the priviest page is clicked and after that admin can update vehicle from the system.

### Admin Delete User

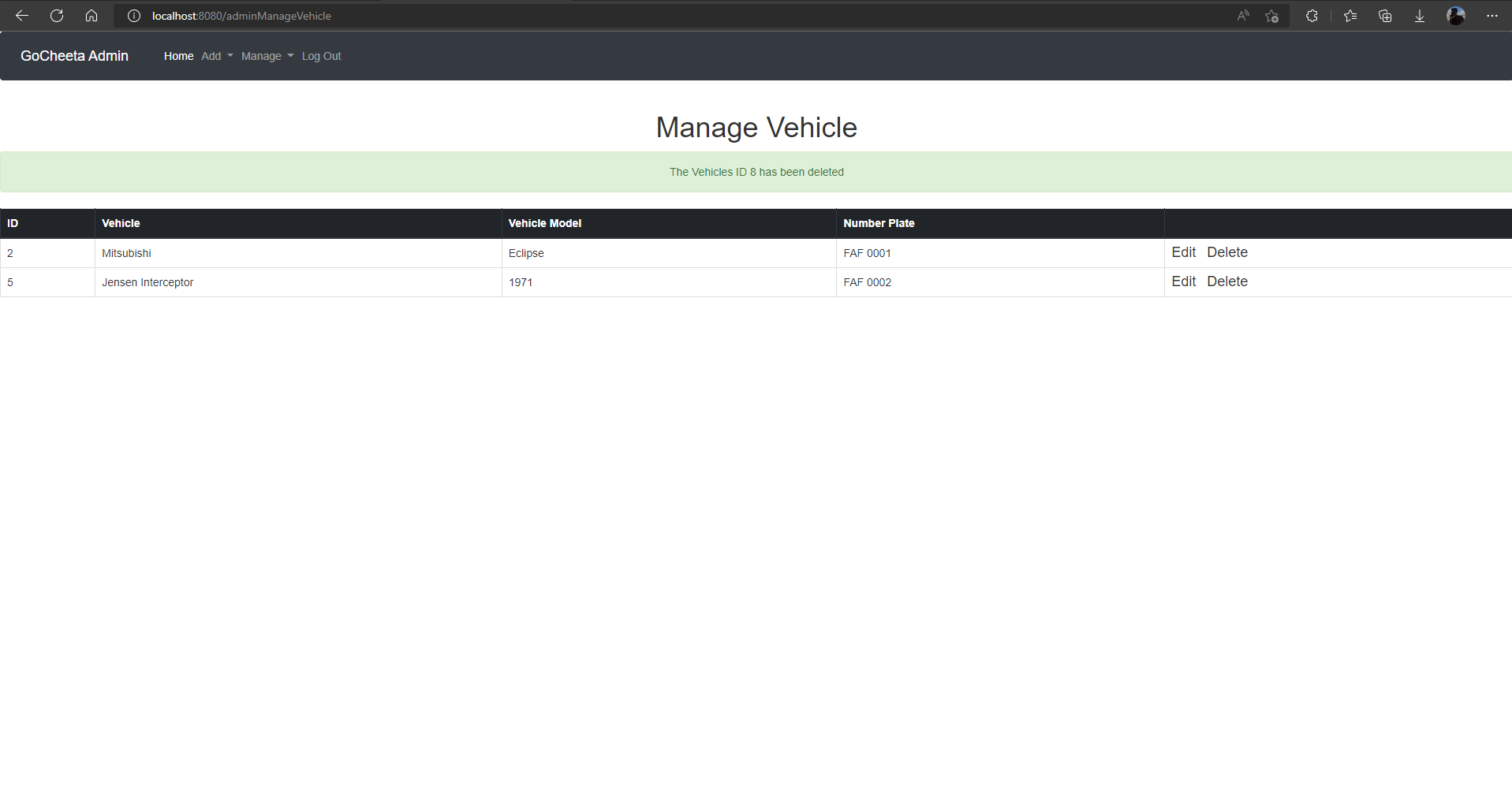


Figure User Manual Admin Delete Vehicle

This is what happens when the delete link is clicked by the admin.

### Admin Manage Booking

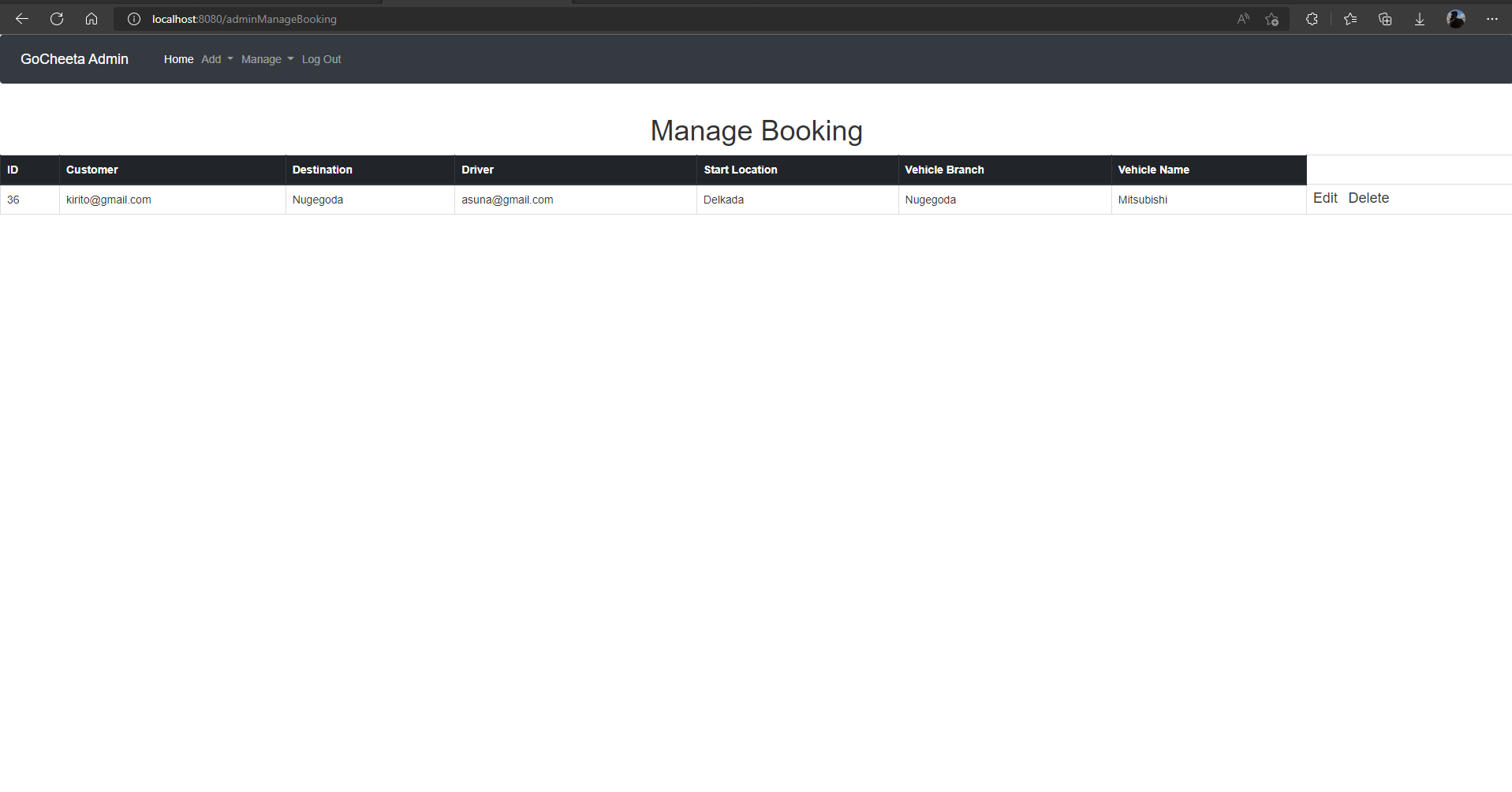


Figure User Manual Admin Manage Booking

Admin can view all the booking from the system and use the links on the right side to update booking and delete the booking.

# Conclusion

When I started this project, I only knew HND level Java and some things that Professor Tharik Kanaka thought I started doing the System with spring boot because I thought it will be easy but as I stated I finished All the CRUD in the system in two days but when it came to making a dual login which I have done in JAVA, C++, C# and PHP which is easy but in spring boot it was very hard and it took me a week to figure out as a slow Lerner still the system login has errors some time pop up I think that’s because of version mismatch. As I was doing the project because my pc hard got corrupted because of an application my brother installed and all my progression was 0 I thought I had to do it again but I remembered committing the project to the git hub after making the Login come so in later if I accidentally delete a line in the login code while editing I can check from git hub so I took the repository and downloaded it and started from there it took only 7 house to come to where I left off that day onwards I always save my project in git hub and the assignment as well this is the first time I have created a 3- tier architecture project. So this project has thought me More knowledge in Java and I learned Spring Boot, MySQL, and thyme leaf. As I still do not work and don’t have experience in industry work now, I have a pretty good idea about the stranded of a project at the industry level.

# References

Gupta, L., 2022. *Creational Patterns.* [Online]   
Available at: https://howtodoinjava.com/design-patterns/creational/