

Chapter 1

Introduction

Car rental or car service providing agencies provide short time leasing vehicles for a specified time with a fee to their customers. Car rental service increasingly becomes the preferred option for most people, especially among students in campuses and universities or for the daily officials who wants to travel at ease. Besides, the raising taxi fares and inconsistent bus arrivals in Bangladesh continue to discourage people from taking up the public transport. Many organizations used web-based system because most people often used mobile phone that gives convenience to the users who are familiar with web technology.

1.1 Background

There are very few online car rental services in Bangladesh. So, most people are not familiar with them. As a result of inadequate and proper advertisement most people are not aware of this easy and feasible online car rental and services facility. Apart from that it has become very difficult for the regular people who often remain busy at their work to book a car manually when they needed. Moreover, those manual booking systems are also super expensive and highly unsafe as well. Therefore, with a view to solve this greater problem, this project is accomplished which will eventually make people's day to day life much smoother by providing a convenient, regular, safe & feasible way of hiring a car and will give a complete solution to all the problems of their cars by providing reliable services .

1.2 Motivation

The motivation is to promote availability of the cars to the general mass people. Due to technological revolution, getting a car through a website has become a mandatory part. Besides, the smooth providence of services to the cars through a dynamic online channel is also a great part of this system. So, overall this project intends to make everyone's day to day to life easier through a complete solution for their adorable vehicle - "CARS".

1.3 Objective

- ✓ Presenting an efficient system of car rental and service process.
- ✓ Providing instant support to the customers on the basis of their need.
- ✓ Ensuring easy booking policy and online payment reliable methods.

- ✓ To allow users to maintain their own account by tracking their booking history.
- ✓ Completing all steps online including filtering, booking, tracking and payment.

1.4 Significance of the project

The project, “**GARI LAGBE**” is designed to help people utilize transport effectively. In recent times cars have become most convenient modes of transportation. Our Car rental system helps in making this an easier, hassle-free and enjoyable experience to acquire and use a car as per ones needs. The rental system traverses from designing a database to understanding business concept and above all to make this an easy to adapt system for various travelling needs. Apart from this, we also provide services for cars. One can easily book a service and we will do the rest part. So, to get a complete car solution there is no other better options than “**GARI LAGBE!**”

Chapter 2

Project Description

The In our whole project, our goal was to create a website for Car rental and services, that is mostly user friendly and easy to navigate. For using this site, firstly a user must have an account which can be done through registration. After completion of registration the user can login and book a car or service. Before booking one user can filter car based on availability on a particular date range. Moreover, the user can also take services and choose his/her suitable date range for drop off and pick up time of the vehicle. After all of this the user can complete the payment online through card. The users can also keep the track record of his/her booked cars and services.

Required Technology

In this project mainly MERN stack is used. MERN stands for MongoDB, Express, React, Node with four key technologies that make up the stack. The key contributions of these are:

- MongoDB : Document database
- Express(.js) : Node.js web framework
- React(.js): a client-side JavaScript framework
- Node(.js): the premier JavaScript web server

Express and Node make up the middle (application) tier. Express.js is a server side web framework, and Node.js is the popular and powerful JavaScript server platform. Regardless of which variant you choose, ME(RVA)N is the ideal approach to working with JavaScript and JSON, all the way through.

Now, let's see a demonstration how does MERN stack worked in this project:-

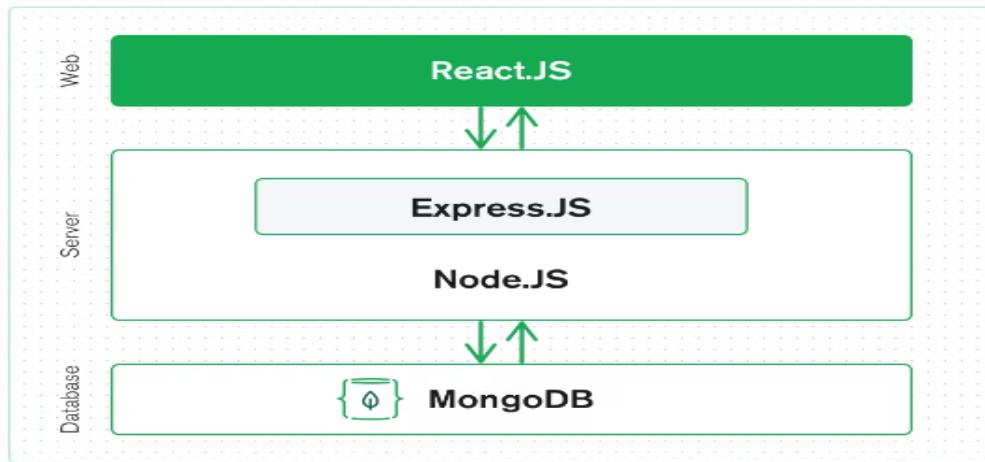


Figure No 2.1: Working procedure of MERN

React.js (front end) : The top tier of the MERN stack is React.js, the declarative JavaScript framework for creating dynamic client-side applications in HTML. React lets us build up complex interfaces through simple components, connect them to data on your back-end server, and render them as HTML.

Express.js and Node.js (server tier): The next level down is the Express.js server-side framework, running inside a Node.js server. Express.js bills itself as a “fast, opinionated, minimalist web framework for Node.js,” and that is indeed exactly what it is. Express.js has powerful models for URL routing (matching an incoming URL with a server function), and handling HTTP requests and responses.

MongoDB (database tier): JSON documents created in React.js front end can be sent to the Express.js server, where they can be processed and (assuming they’re valid) stored directly in MongoDB for later retrieval.

CSS: To specify how a page is presented a markup language like HTML uses a style sheet language called Cascading Style Sheets (CSS). CSS is mainly used to styling the components of the web browser.

Bootstrap 5: Bootstrap 5 is the newest version of Bootstrap, which is the most popular HTML, CSS, and Java Script framework for creating responsive, mobile-first websites.

Chapter 3

User Interface

3.1 Homepage

This is the homepage of the website. A user can access login and register from here, can see car demo, service demo, our gallery, about us and footer with contact details.



Figure No 3.1.1: Homepage

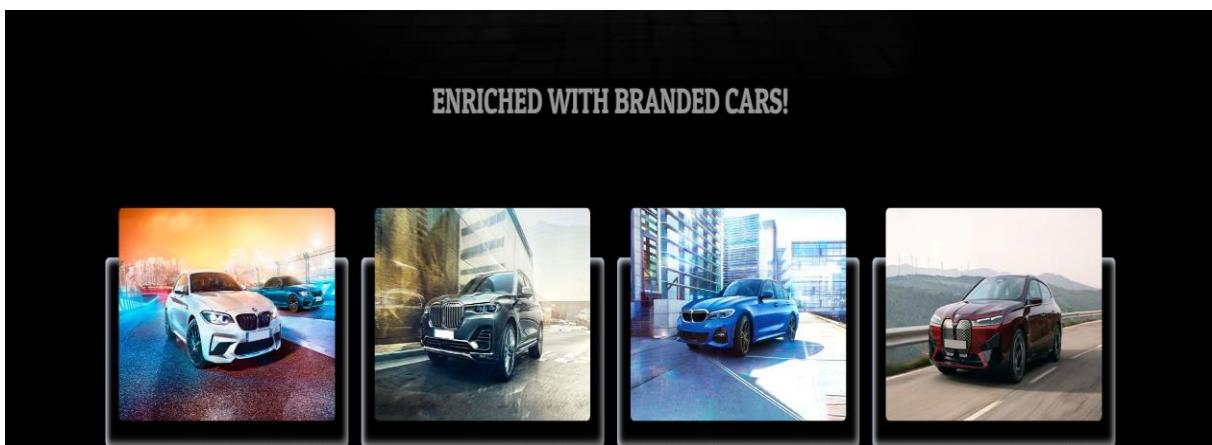


Figure No 3.1.2: Homepage

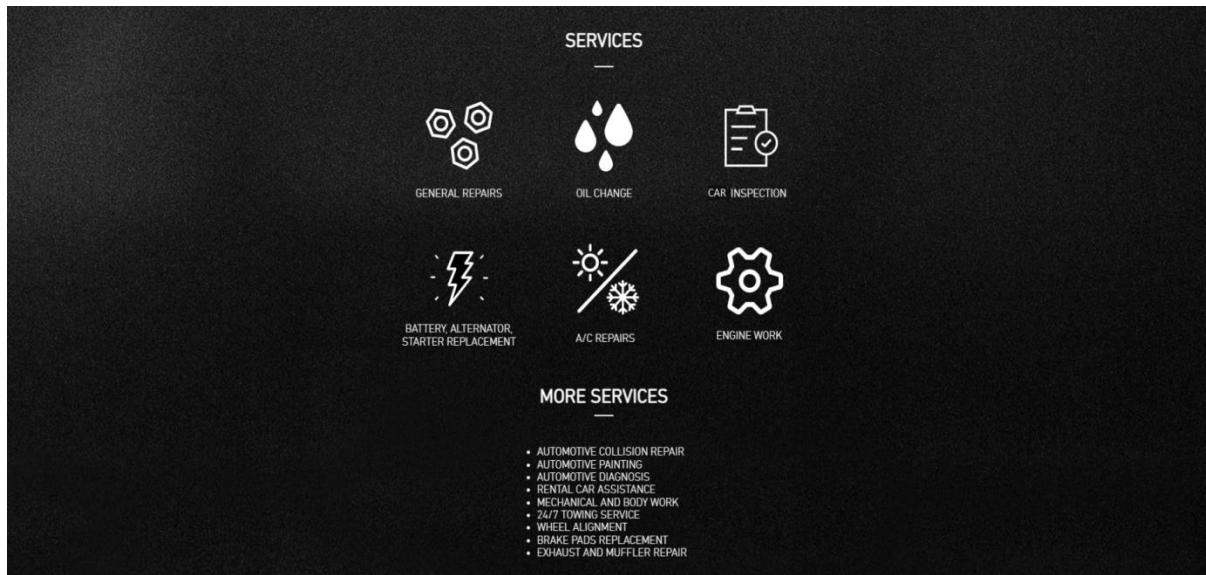


Figure No 3.1.3: Homepage



Figure No 3.1.4: Homepage



Figure No 3.1.5: Homepage



Figure No 3.1.6: Homepage

3.2 Registration Page:

By providing necessary information a user can register and create an account for renting cars or taking service. A user should provide email address, username and password.

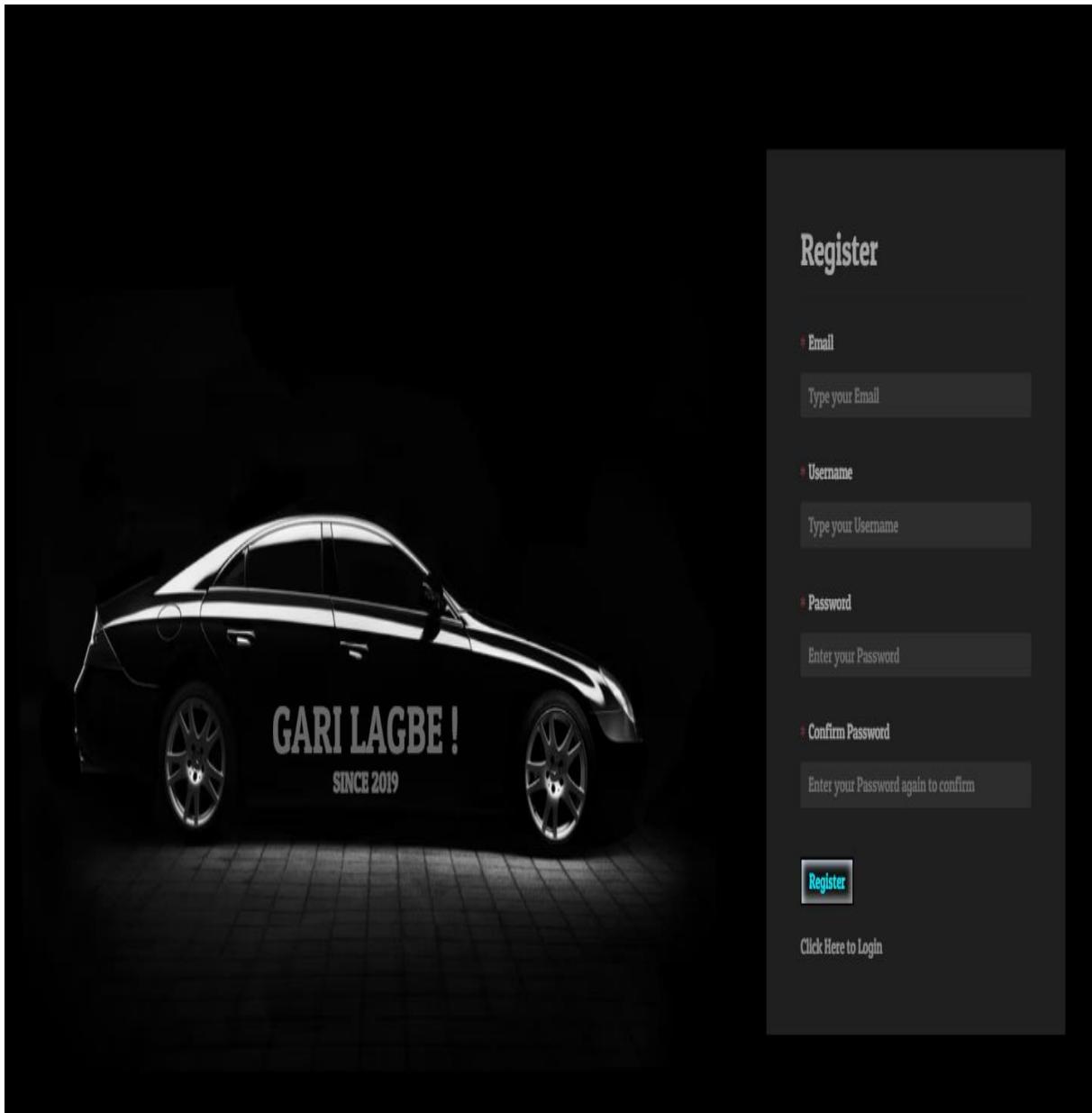


Figure No 3.2: Registration Page

3.3 Login Page

If a user already has an account then it's possible to enter to the website by giving email address and password.

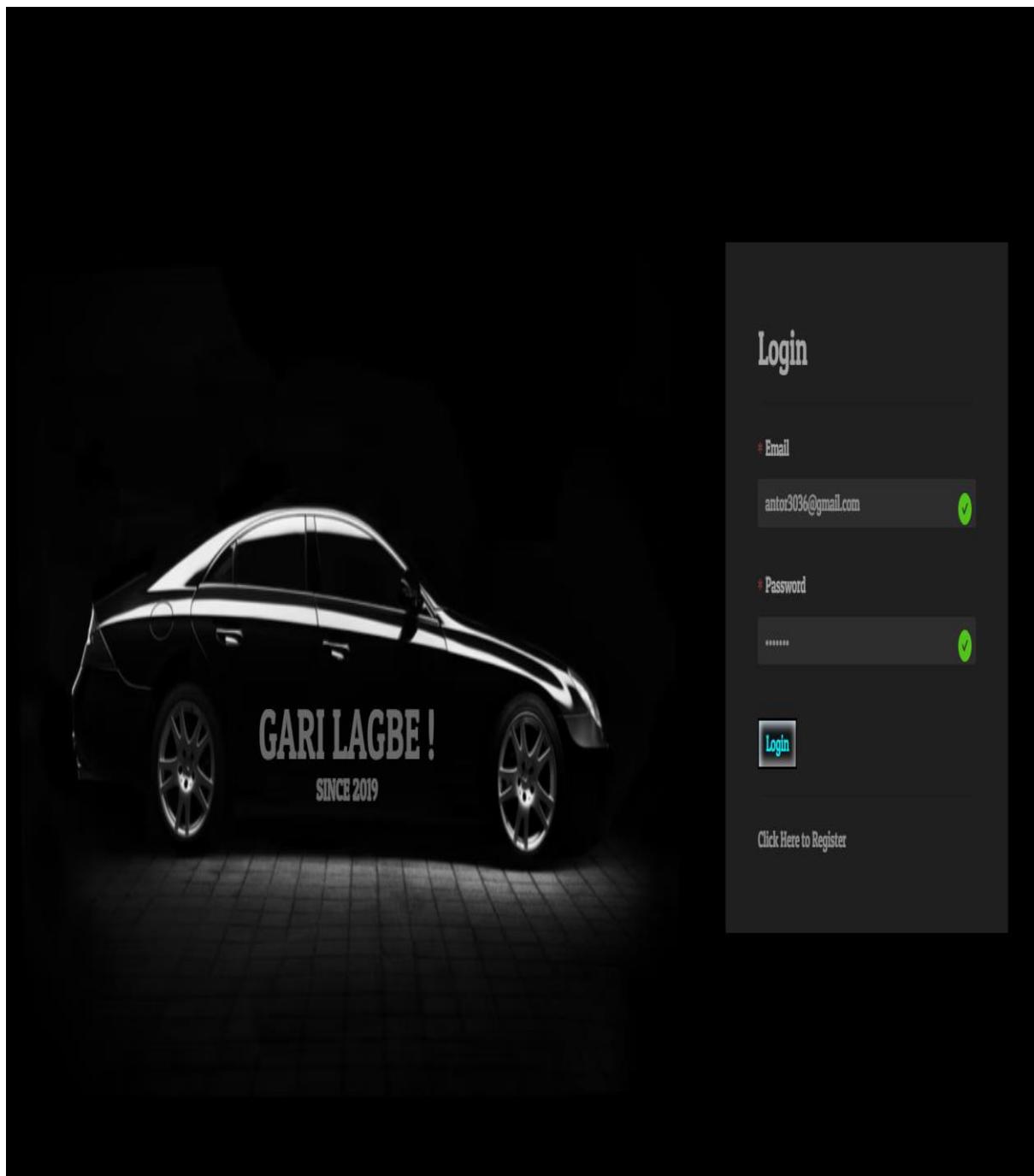


Figure No 3.3: Login Page

3.4 Rent Car Page:

In this page, all the cars available in the website is shown. The available cars in a time range can be filtered by Datepicker. After pointing the cursor on a car image, the name of that car, rent per hour and booking button will be shown.

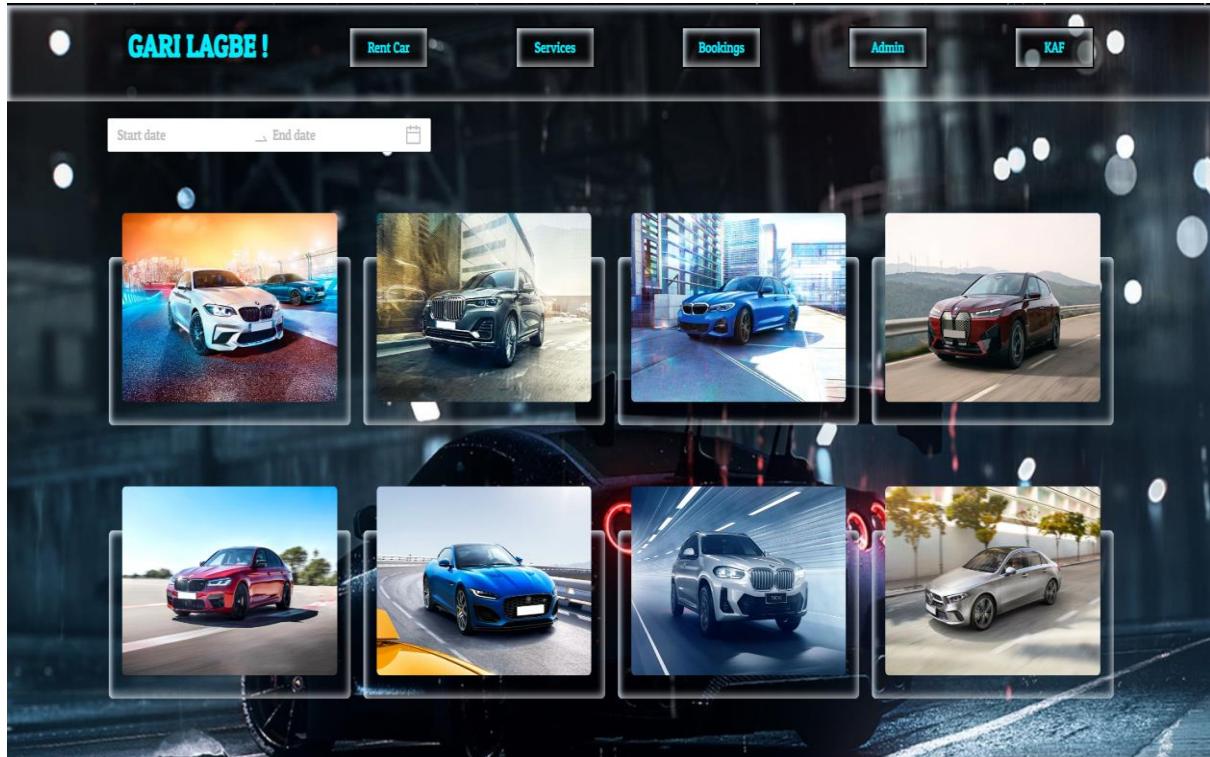


Figure No 3.4.1: Rent Car Page

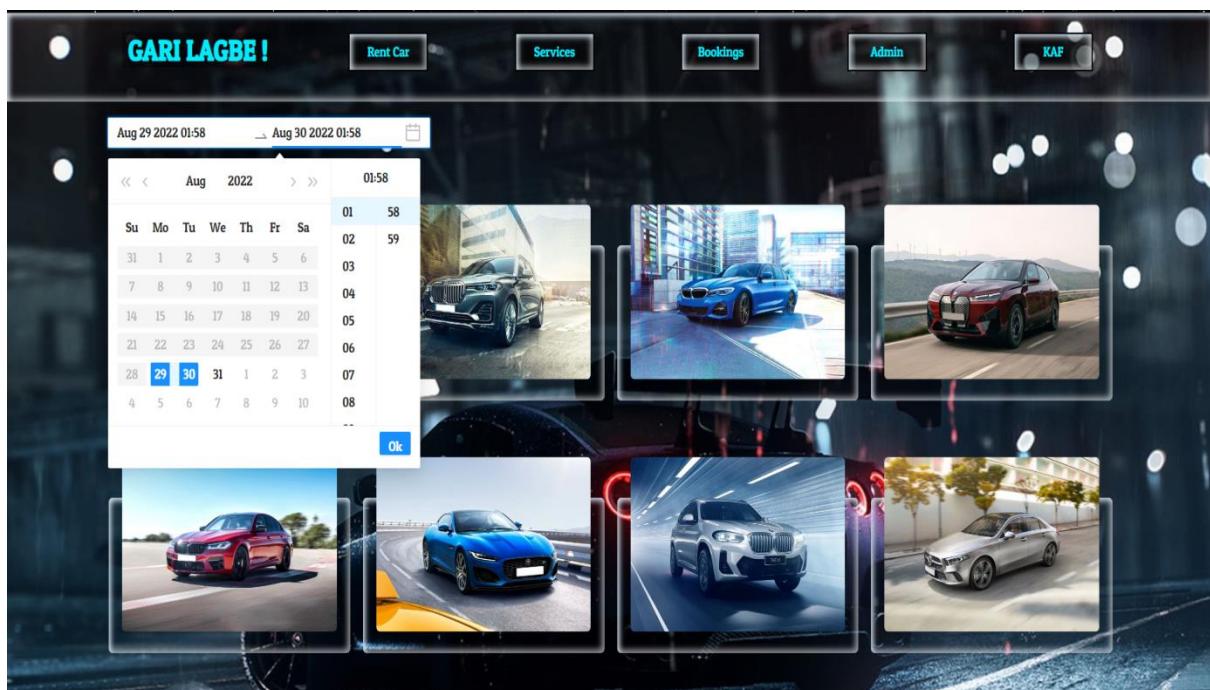


Figure No 3.4.2: Rent Car Page (filtering with date picker)

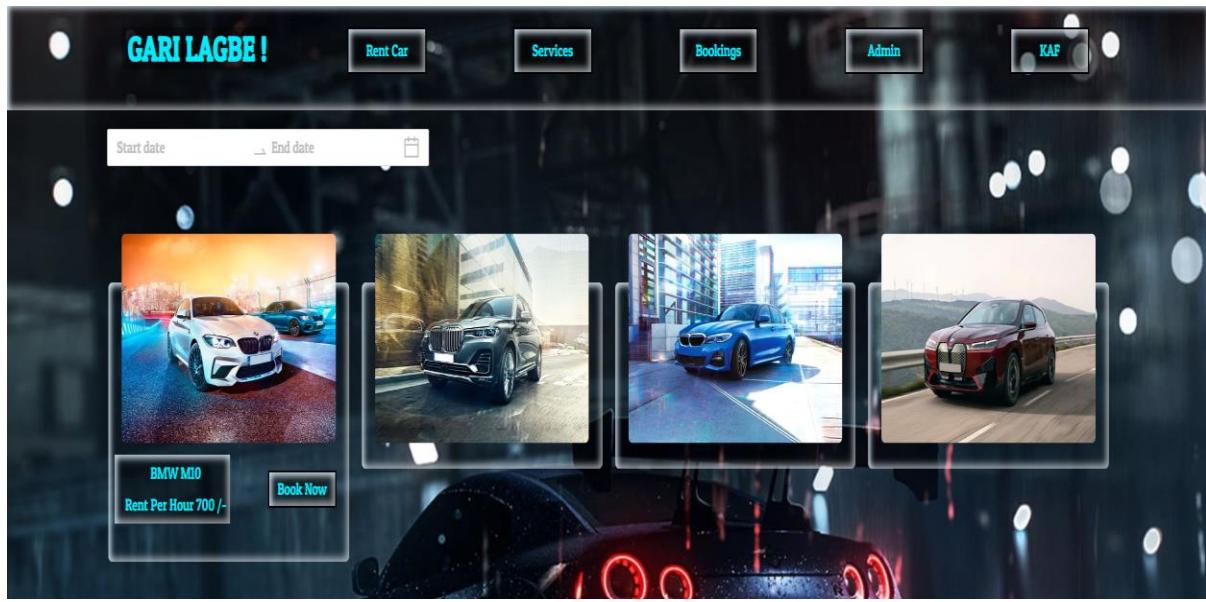


Figure No 3.4.3: Rent Car Page (showing information of a car)

3.5 Car Booking Page:

This page shows all the details of selected cars. The time duration of taking a car can also be selected by the user with date picker. By clicking ‘See Booked Slots’ button, user can see the booking history of that car. Considering all the condition the total payments can be calculated. Payment can be completed with Visa card.



Figure No 5.1: Car Booking Page

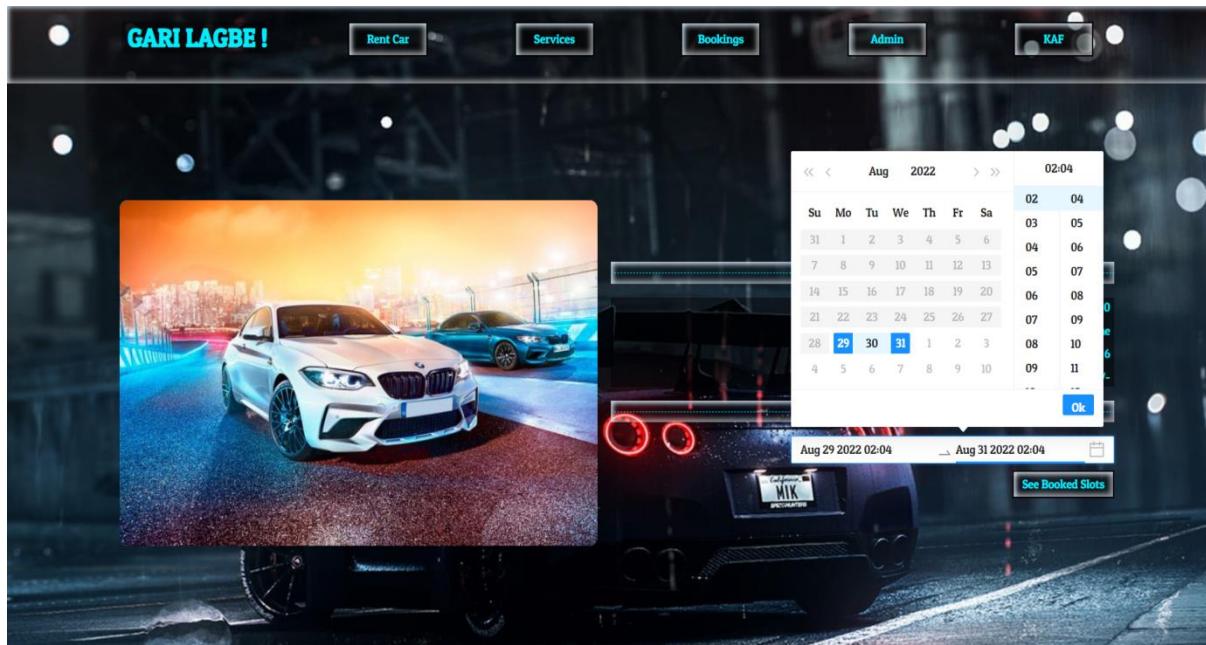


Figure No 3.5.2: Selecting Booking Time Slots

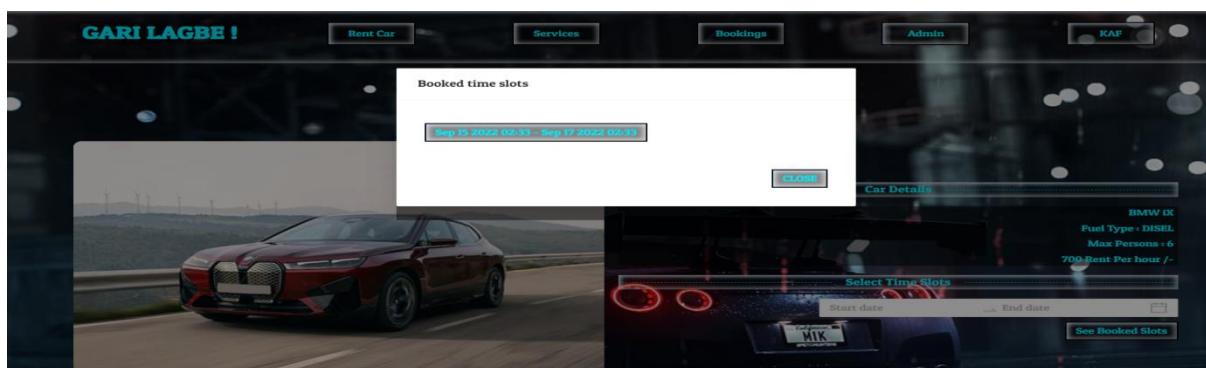


Figure No 3.5.3: Checking Booked Time Slots

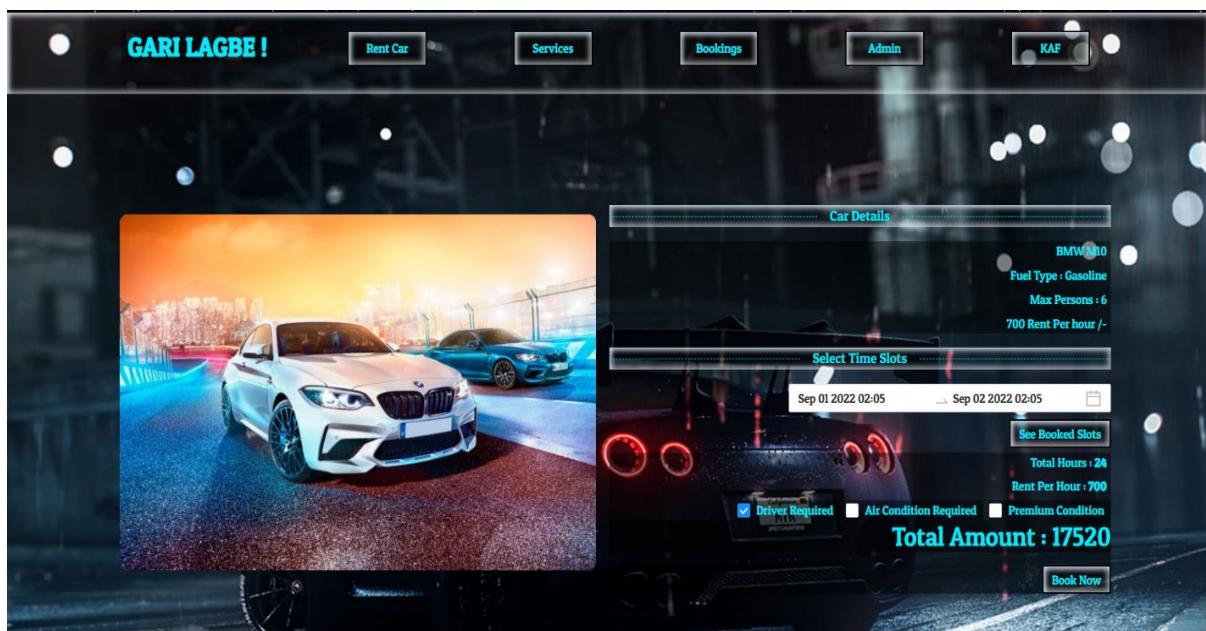


Figure No 3.5.4: Total Payments

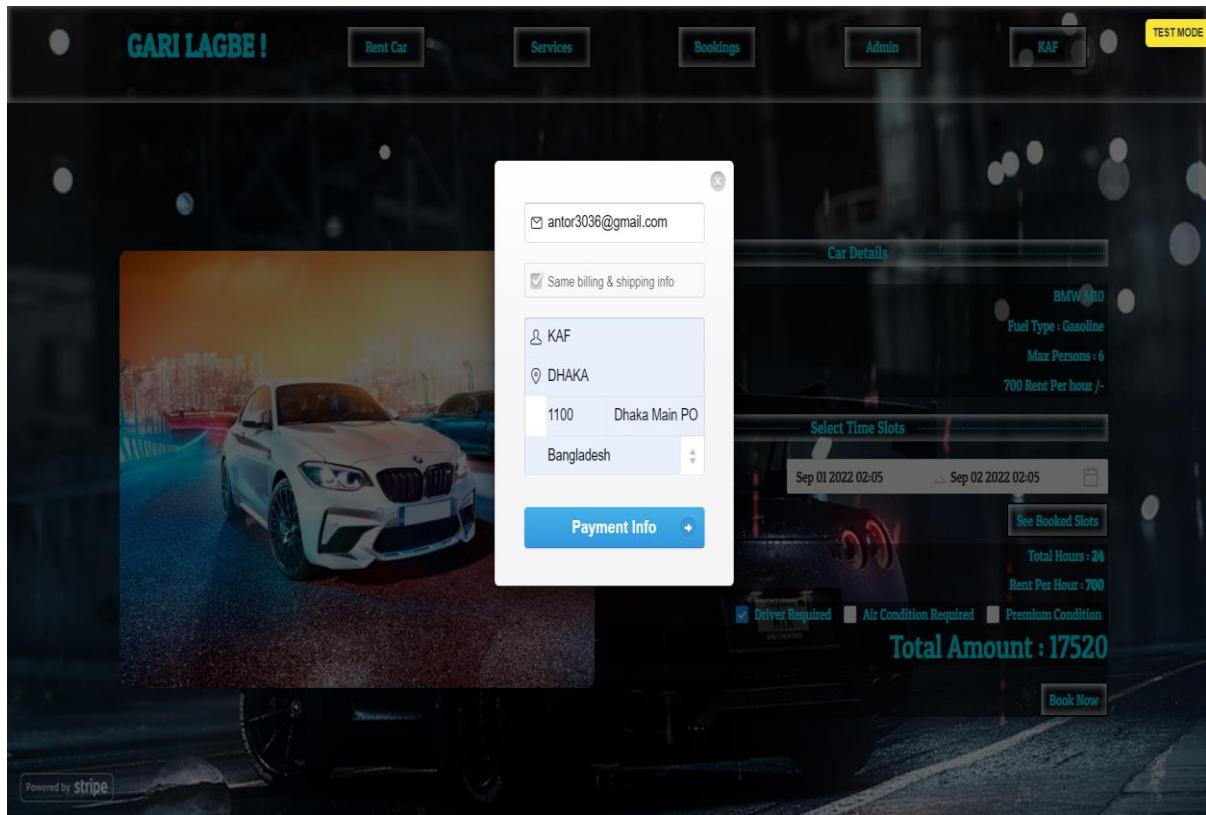


Figure No 3.5.5: Car Payment Gateway (1)

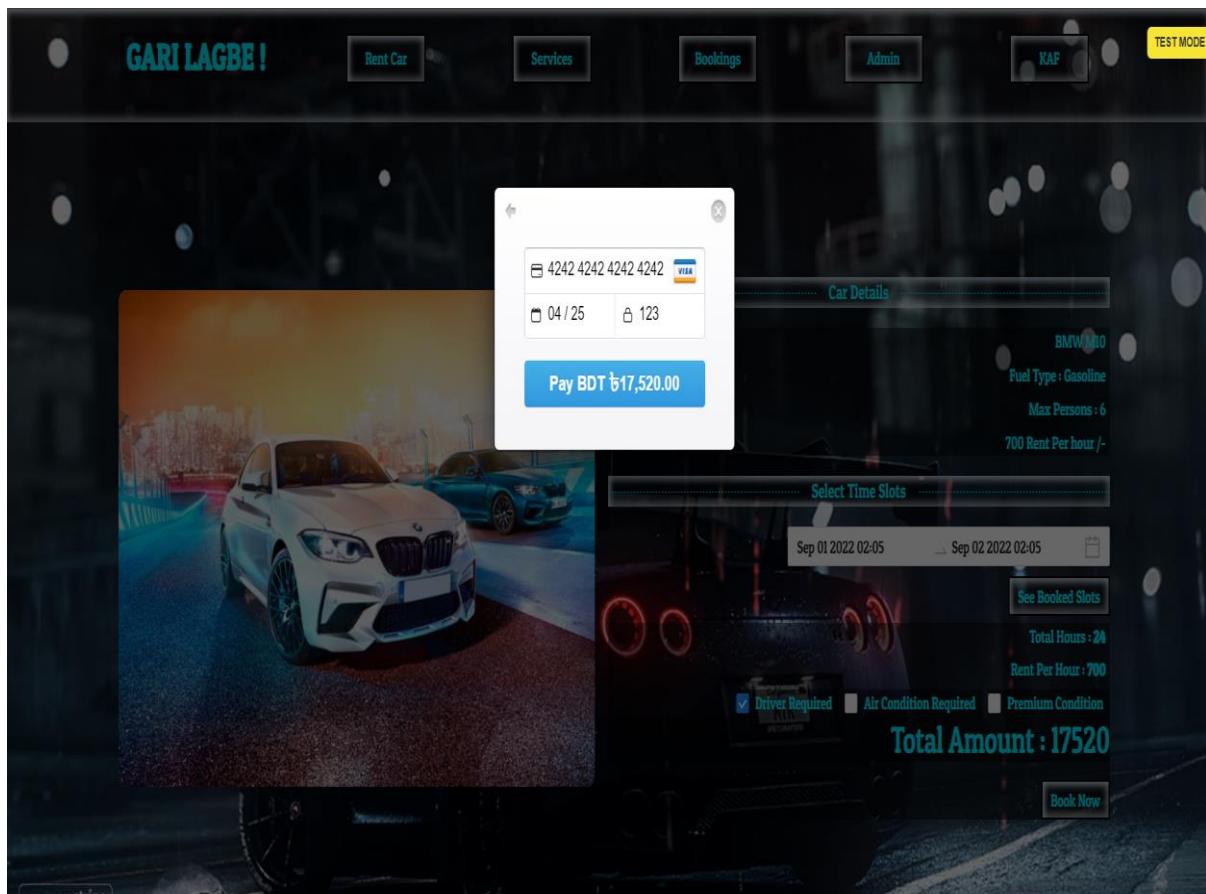


Figure No 3.5.6: Car Payment Gateway (2)

3.6 Booked Car Page:

By clicking Booking button another two option Cars and Services are shown. They represent Cars Booking and Services Booking. In the Car Booking option all the booked car are listed by the user.

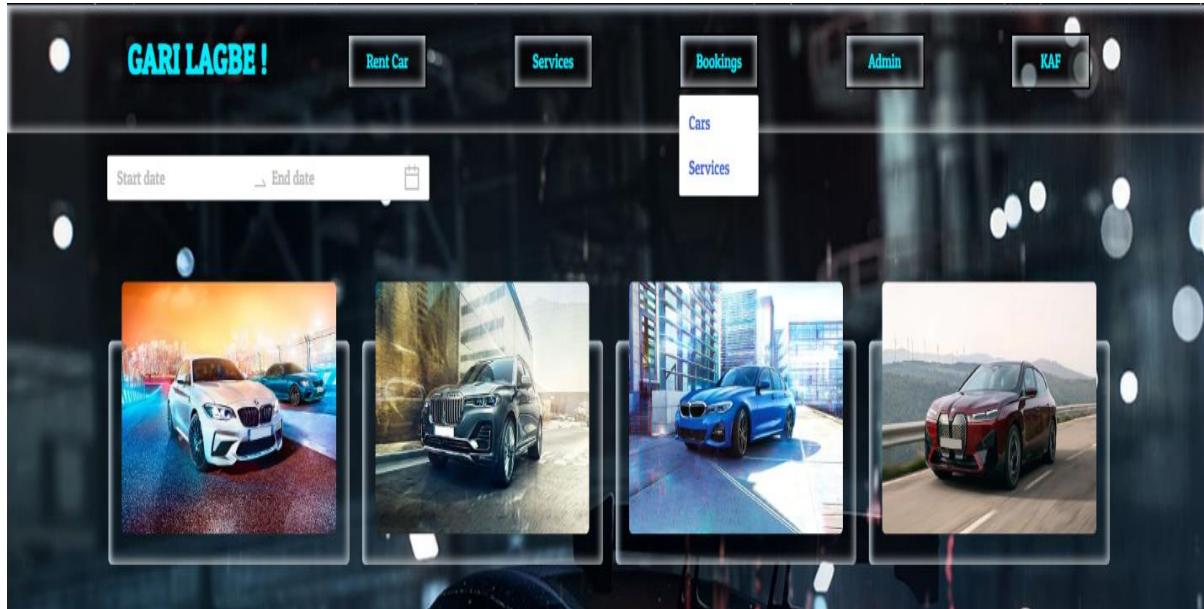


Figure No 3.6.1: Booking Button of Home page

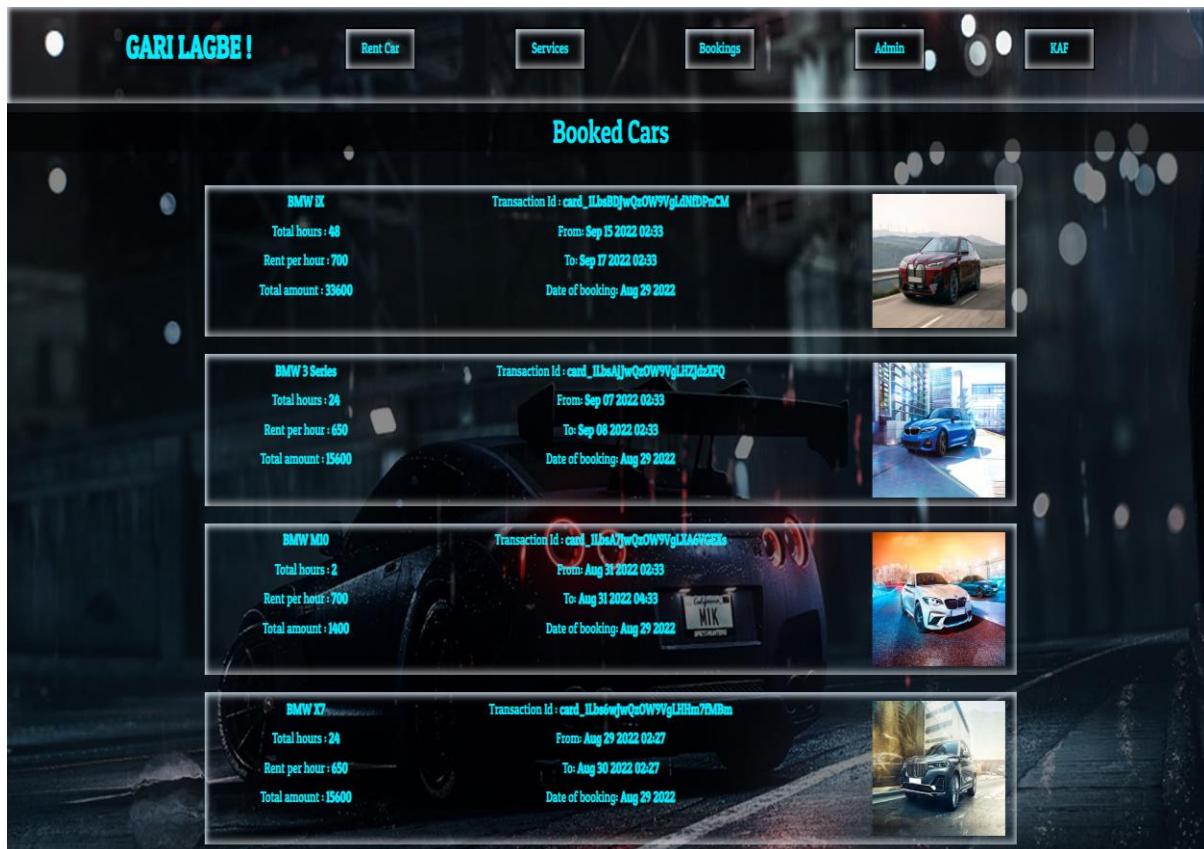


Figure No 3.6.2: Booked Cars

3.7 Service Page:

All the available services are shown in this page. The details of the services can be shown by putting the cursor on a service.

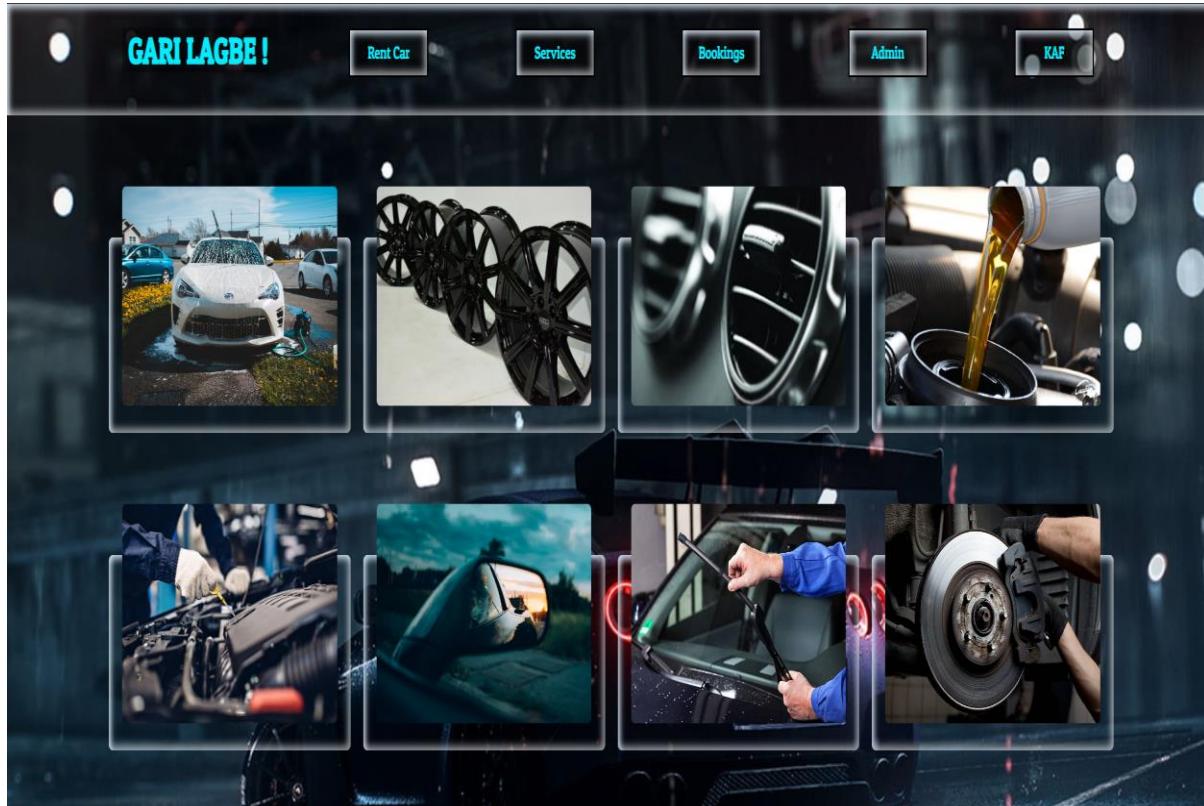


Figure No 3.7.1: Service Page

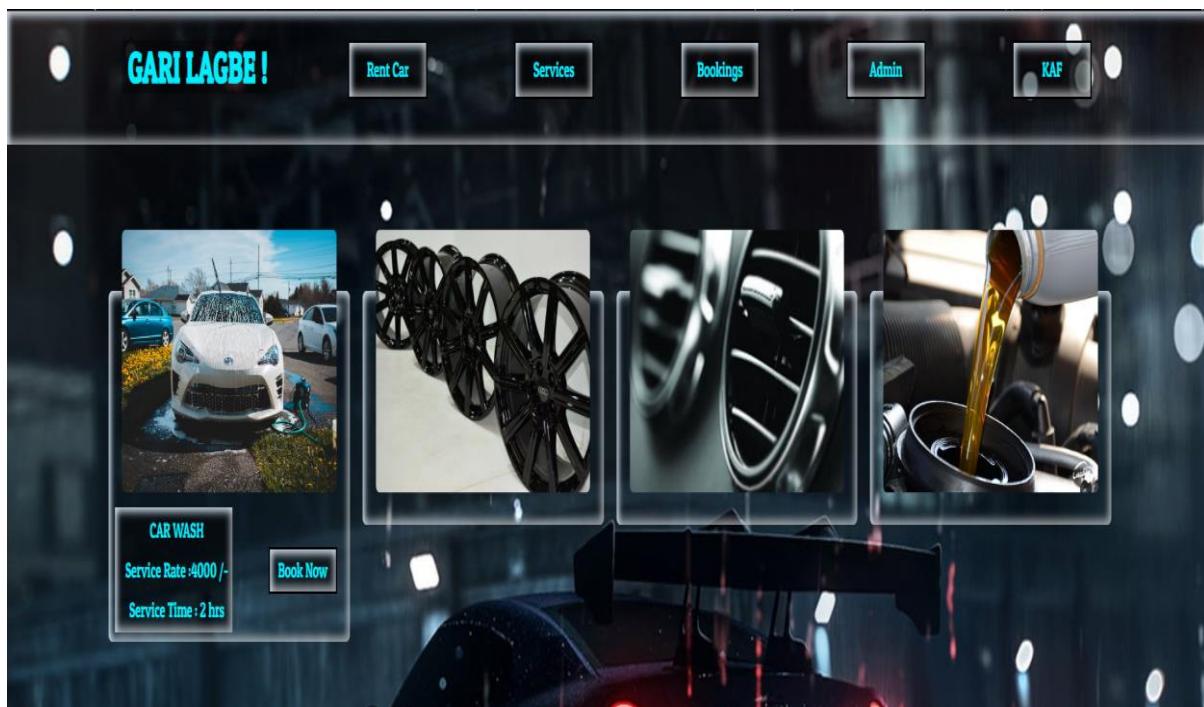


Figure No 3.7.2: Service Page with details

3.8 Service Booking Page:

This page shows all the details of selected service. The time duration of taking service can also be selected by the user with date picker. By clicking ‘See Booked Slots’ button, user can see the booking history of that service. Considering all the condition the total payments can be seen. Payment can be completed with Visa card.



Figure No 3.8.1: Service Booking Page

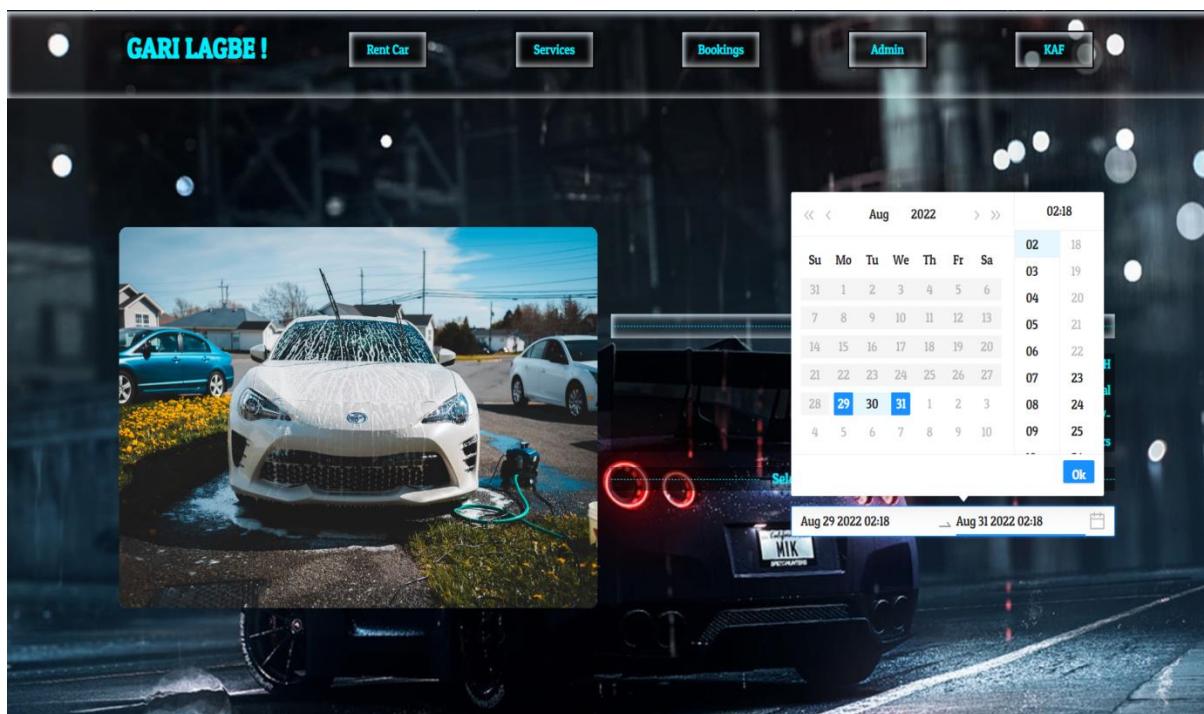


Figure No 3.8.2: Selecting Booking Time Slots (Service)

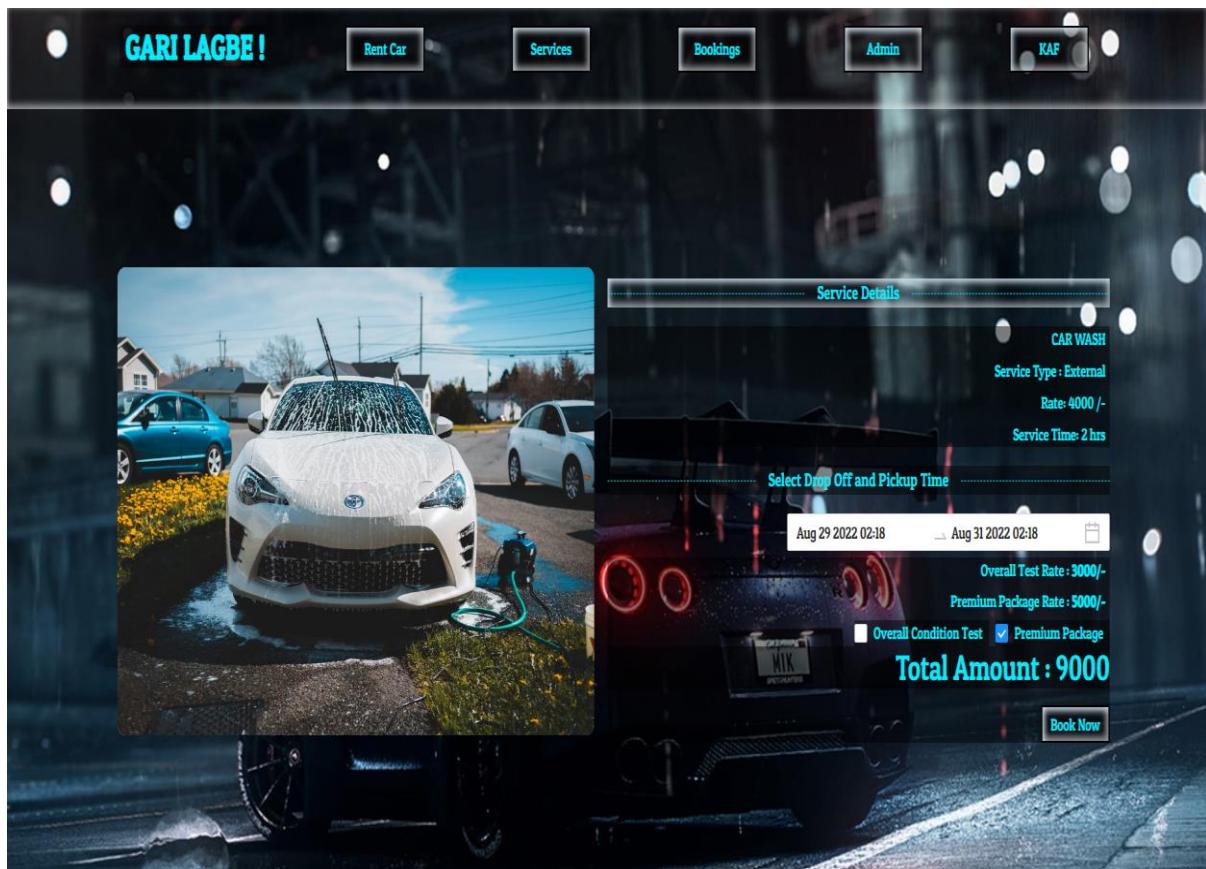


Figure No 3.8.3: Total Payments (Service)

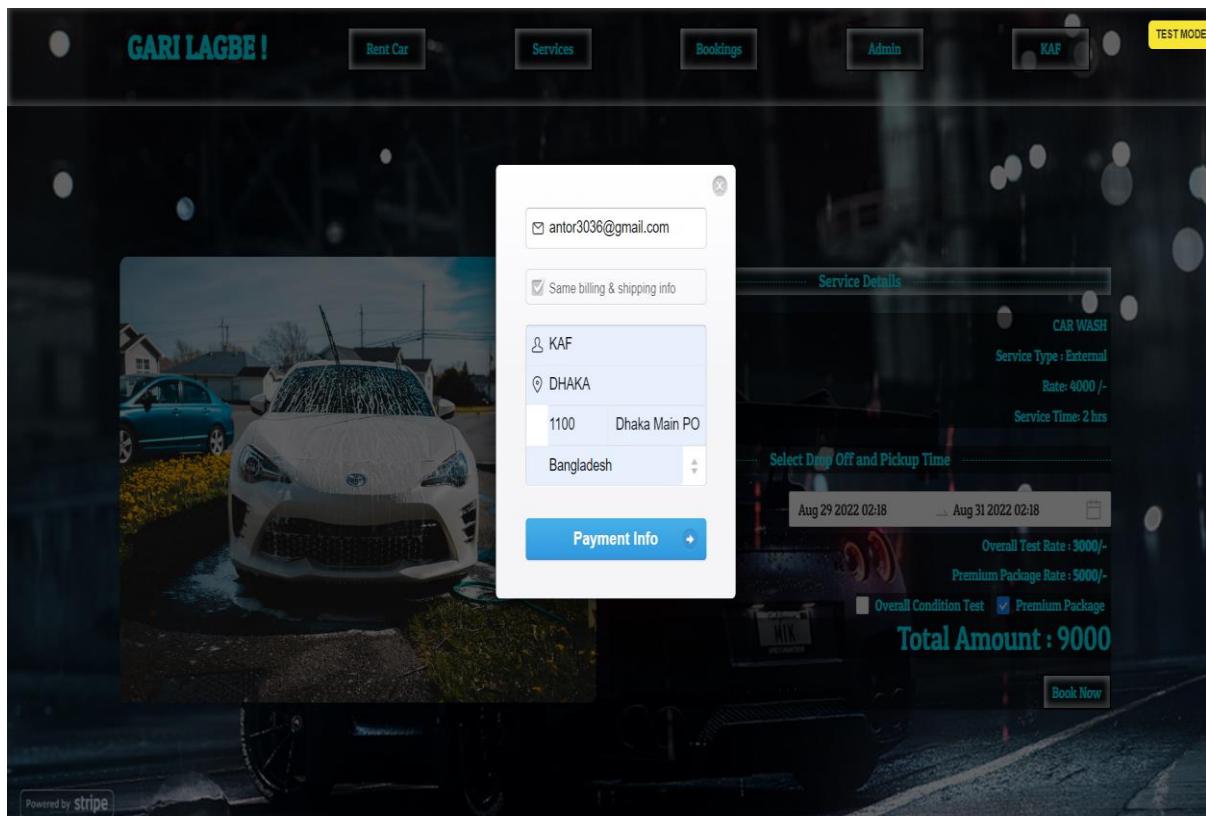


Figure No 3.8.4: Service Payment Gateway (1)

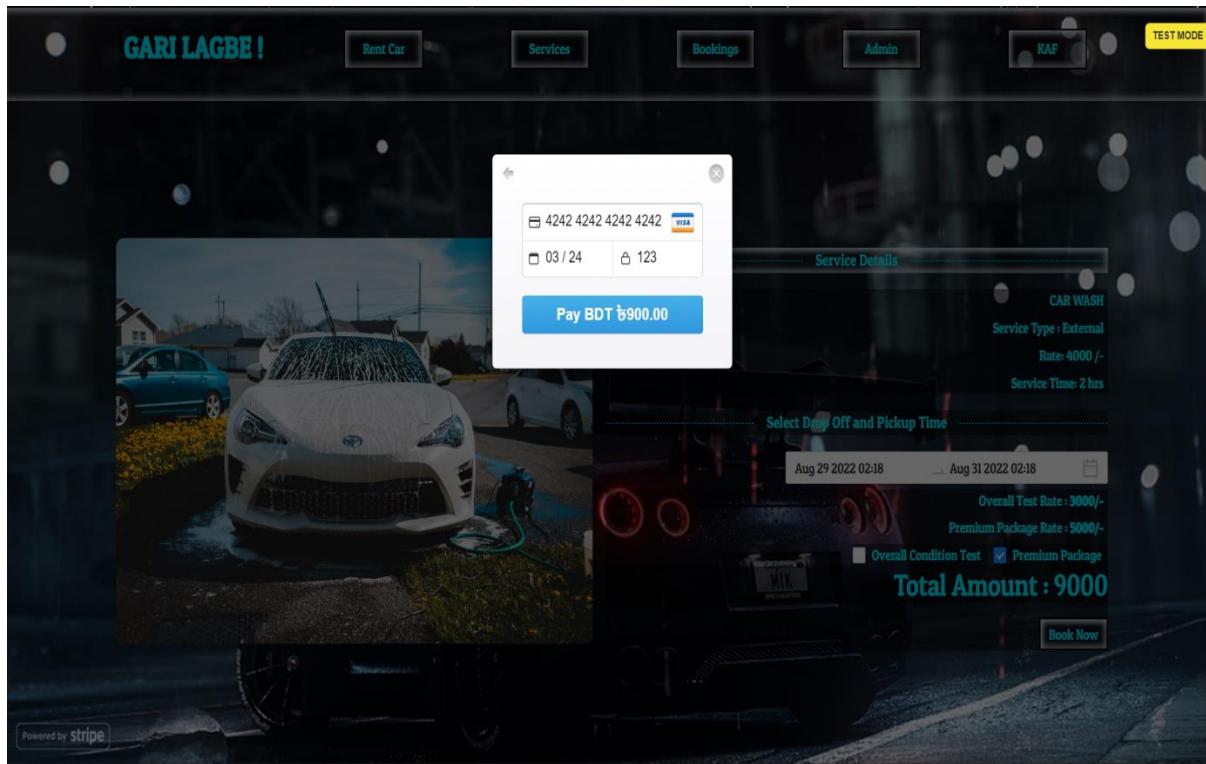


Figure No 3.8.5: Service Payment Gateway (2)

3.9 Booked Service

In the Service Booking option all the booked services are listed by the user.

Figure No 3.9: Booked Services

3.10 Update Profile

Here users can update their profile information

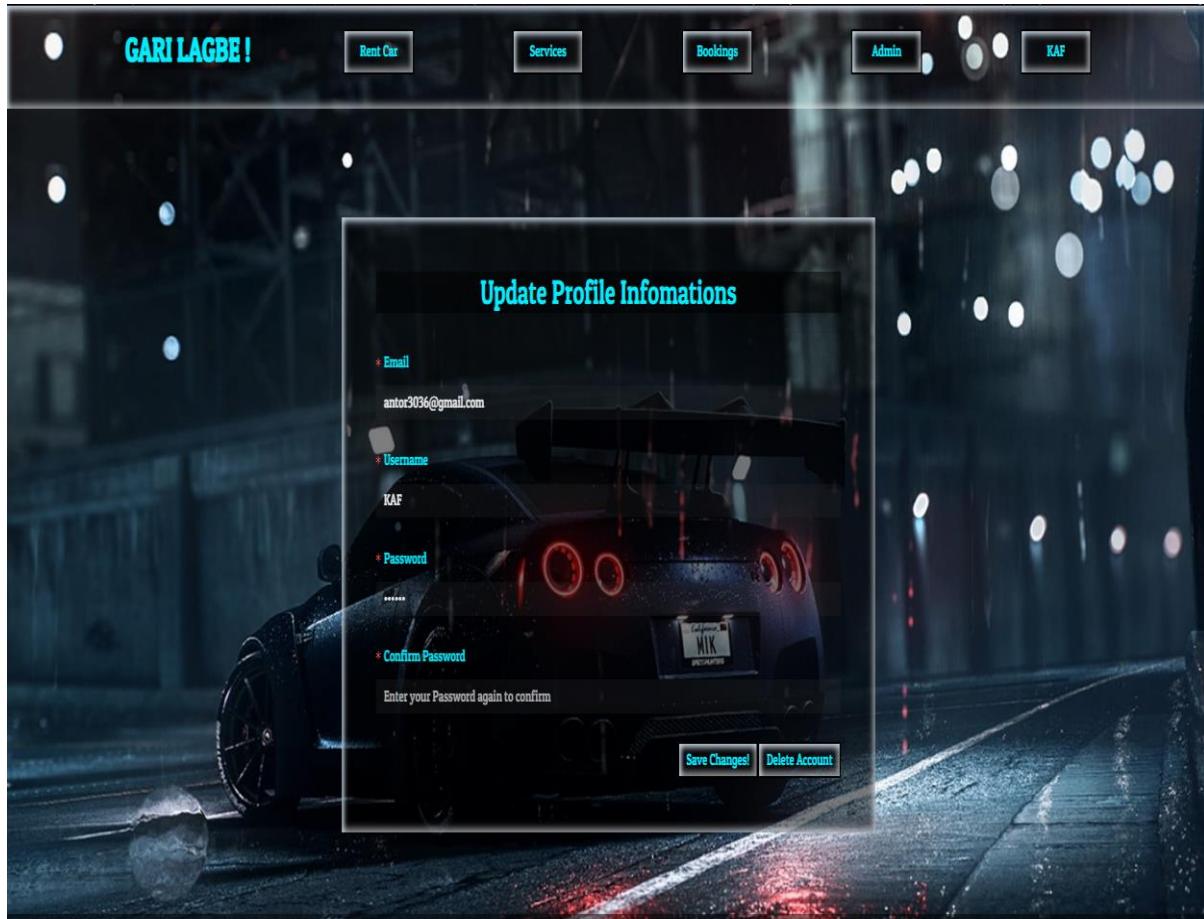


Figure No 3.10: Update Profile

3.11 Admin Panel

Admin is a higher level user. Car admin panel can add and delete car from the database.

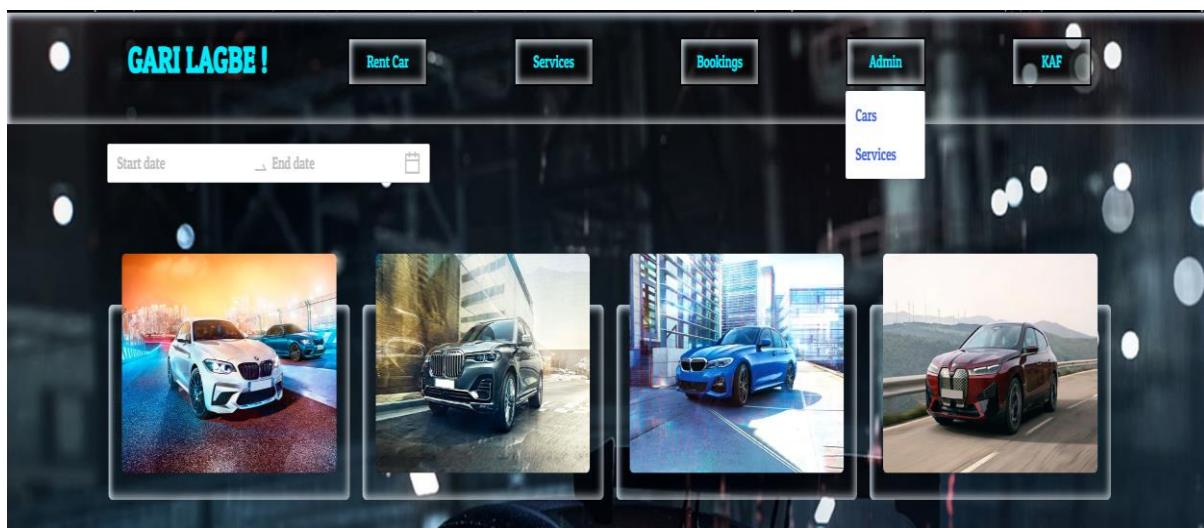


Figure No 3.11.1: Admin Button

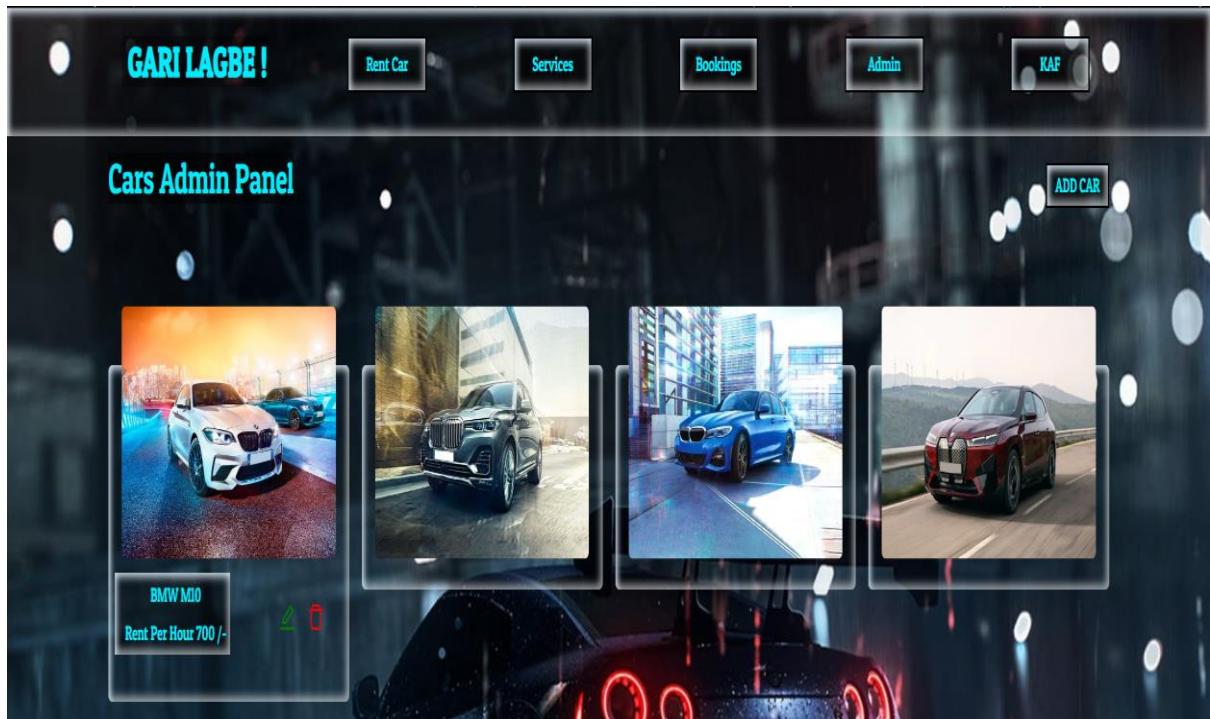


Figure No 3.11.2: Car Admin Panel with Details

3.12 Add New Car Page:

This page is reserved for only Admin to add the car.

Figure No 3.12: Add New Car

3.13 Edit New Car Page:

The information of the Cars can be changed with edit car page. Admin can edit car.

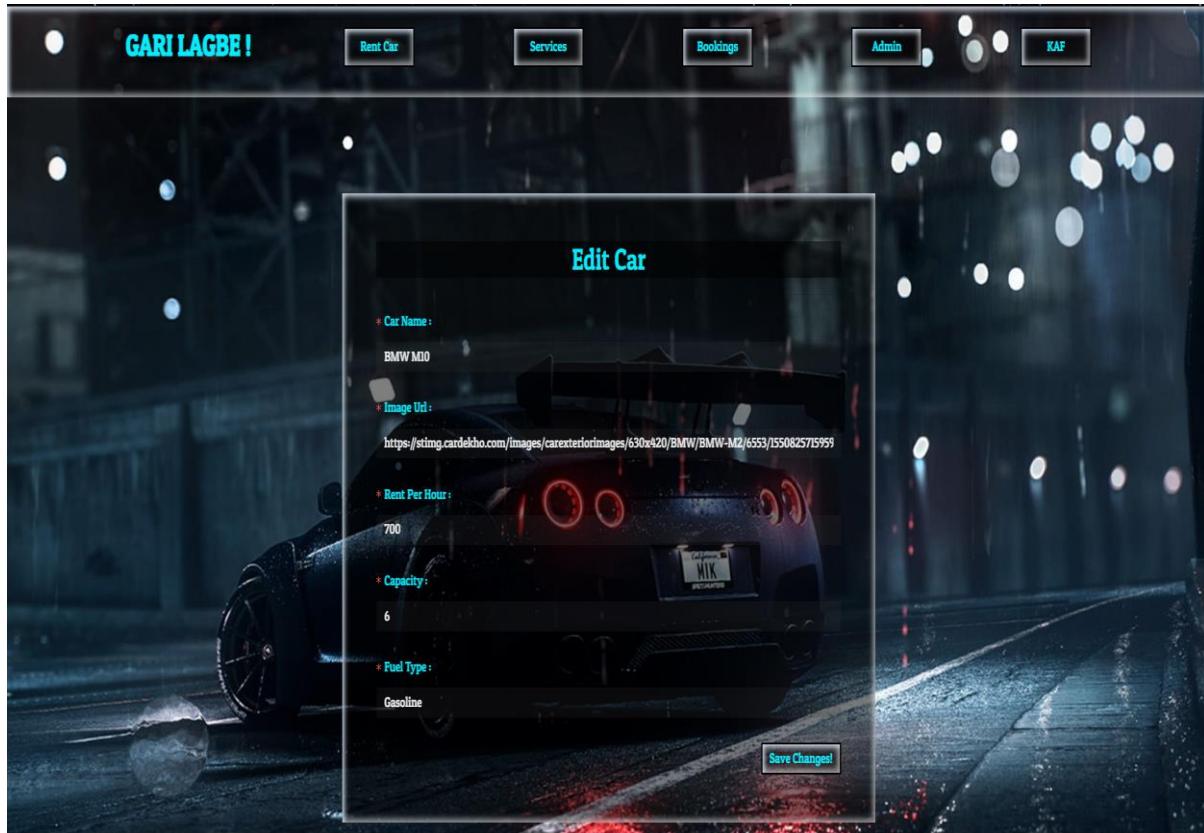


Figure No 3.13: Edit Car

3.14 Service Admin Page

Car admin panel can add and delete car from the database.

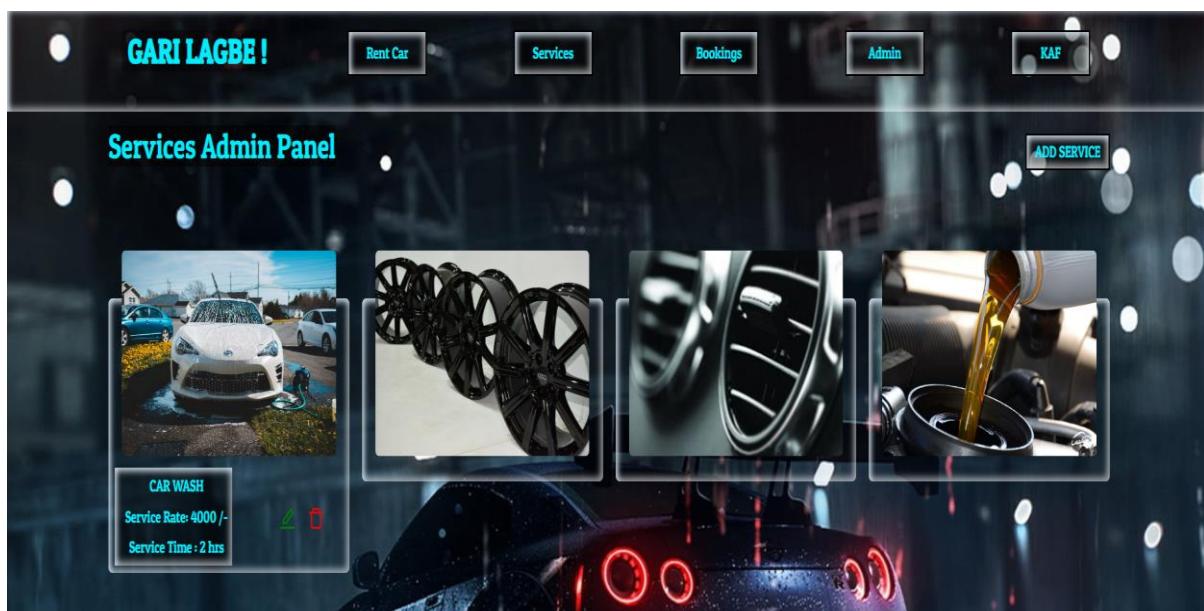


Figure No 3.14: Service Admin Page

3.15 Add New Service Page:

This page is reserved for only Admin to add the service.

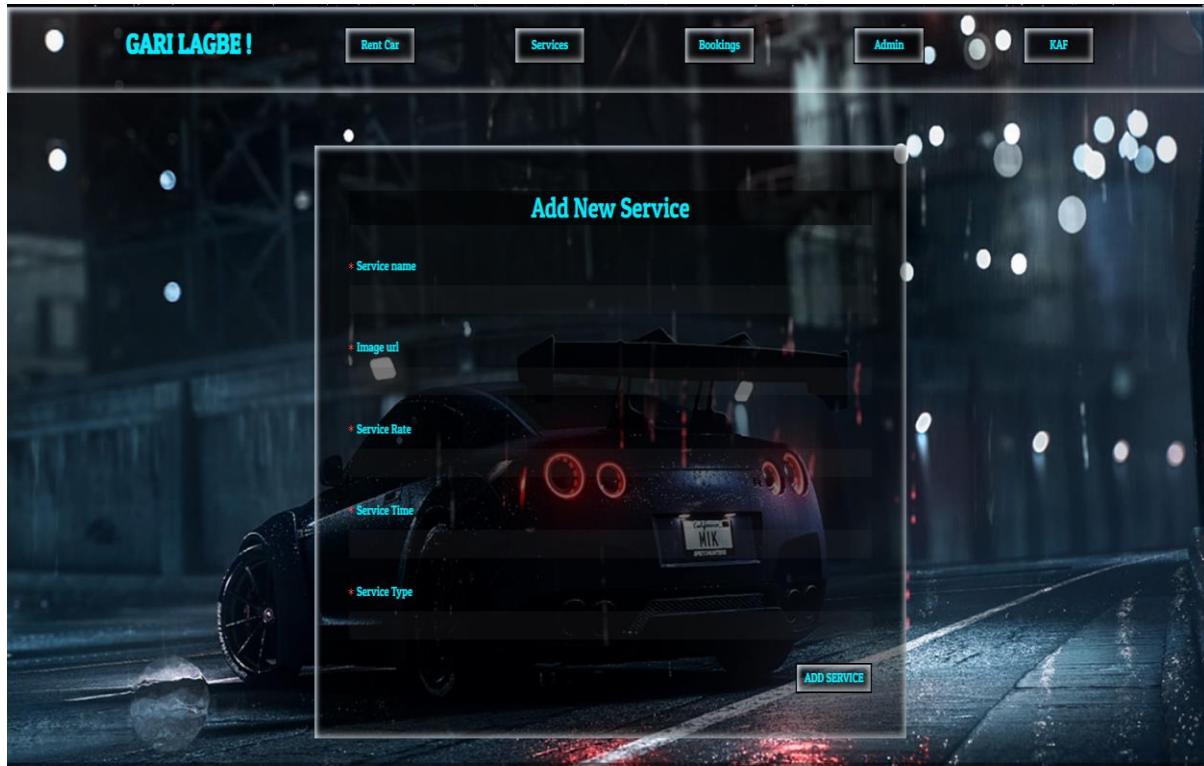


Figure No 3.15: Add Service Page

3.16 Edit New Service Page:

The information of the service can be changed with edit service page. Admin can edit service.

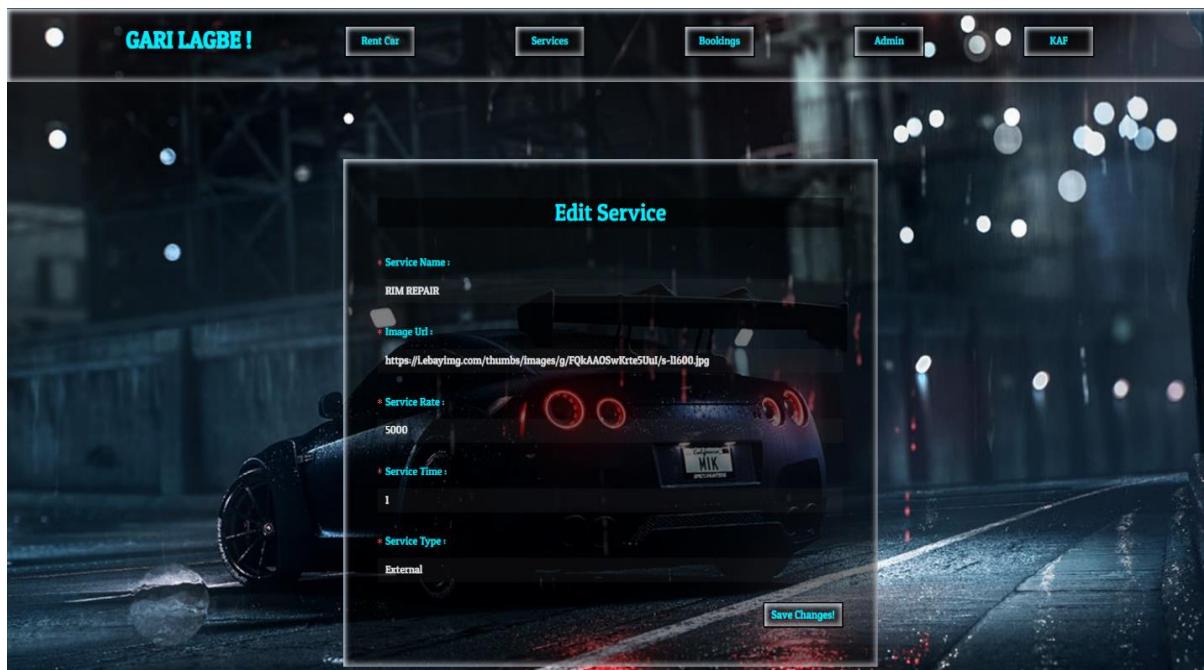


Figure No 3.16: Edit Service Page

Chapter 4

Backend

4.1 GARI_LAGBE Database

The whole database of our project is called GARI_LAGBE. We collected our data and stored in Mongodb.

MongoDB Compass - cluster0.i85ysj7.mongodb.net/GARI_LAGBE

Collections

- bookings**
Storage size: 20.48 kB | Documents: 5 | Avg. document size: 319.00 B | Indexes: 1 | Total index size: 36.86 kB
- cars**
Storage size: 24.56 kB | Documents: 51 | Avg. document size: 270.00 B | Indexes: 1 | Total index size: 36.86 kB
- registrations**
Storage size: 20.48 kB | Documents: 4 | Avg. document size: 104.00 B | Indexes: 1 | Total index size: 36.86 kB
- services**
Storage size: 20.48 kB | Documents: 9 | Avg. document size: 340.00 B | Indexes: 1 | Total index size: 36.86 kB
- servicesbookings**
Storage size: 20.48 kB | Documents: 4 | Avg. document size: 314.00 B | Indexes: 1 | Total index size: 36.86 kB

Figure No 4.1: GARI_LAGBE Database

4.2 Registration Table

In this table we hold the data of the user when they register for their account.

MongoDB Compass - cluster0.i85ysj7.mongodb.net/GARI_LAGBE registrations

GARI_LAGBE.registrations

4 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET REFRESH

Displaying documents 1 - 4 of 4

_id	email	username	password	_v
<code>ObjectId('62db0fe95f5232fd862f673a')</code>	"ul804035@student.cuet.ac.bd"	"#kf"	"123456"	0
<code>ObjectId('62efef4453ee0ab850d249ef4')</code>	"suhanaibr99@gmail.com"	"Suhana"	"Suhana"	0
<code>ObjectId('630110b5895a2c4870700db3')</code>	"roni3036@gmail.com"	"Roni"	"123456"	0
<code>ObjectId('630bb21326fb730109e4e6f4')</code>	"antor3036@gmail.com"	"Xar"	"123456"	0

Figure No 4.2: Registration Table

4.3 Cars Table

The information of the cars are stored in this database

The screenshot shows the MongoDB Compass interface with the 'cluster0.i85sj7.mongodb.net/GARI_LAGBE' connection selected. The left sidebar shows databases like 'GARI_LAGBE', 'bookings', 'cars', 'registrations', 'services', 'servicesbookings', and 'users'. The main area displays the 'GARI_LAGBE.cars' collection with 51 documents. A single document is expanded, showing fields such as _id, name, image, rentPerHour, fuelType, bookedTimeSlots, capacity, and updatedAt. The document ID is ObjectId('63009cefef5fc1000a6dd52').

```
_id: ObjectId('63009cefef5fc1000a6dd52')
name: "BMW M10"
image: "https://stimg.cardekho.com/images/carexteriorimages/630x420/BMW/BMW-M2_"
rentPerHour: 700
fuelType: "Gasoline"
bookedTimeSlots: Array
capacity: 6
__v: 2
updatedAt: 2022-08-28T20:33:02.116+00:00

_id: ObjectId('63009cefef5fc1000a6dd54')
name: "BMW X7"
image: "https://stimg.cardekho.com/images/carexteriorimages/630x420/BMW/BMW-X7_"
rentPerHour: 650
fuelType: "Gasoline"
bookedTimeSlots: Array
capacity: 5
__v: 1
updatedAt: 2022-08-28T20:29:46.782+00:00

_id: ObjectId('63009cefef5fc1000a6dd53')
name: "BMW 3 Series"
image: "https://stimg.cardekho.com/images/carexteriorimages/630x420/BMW/3-Seri..."
rentPerHour: 650
fuelType: "Gasoline"
```

Figure No 4.3: Cars Table

4.4 Booking Table

The information about a booked car is stored in this database.

The screenshot shows the MongoDB Compass interface with the 'cluster0.i85sj7.mongodb.net/GARI_LAGBE' connection selected. The left sidebar shows databases like 'GARI_LAGBE', 'bookings', 'cars', 'registrations', 'services', 'servicesbookings', and 'users'. The main area displays the 'GARI_LAGBE.bookings' collection with 5 documents. A single document is expanded, showing fields such as _id, car, user, bookedTimeSlots, totalHours, totalAmount, transactionId, driverRequired, acRequired, premiumConditionRequired, createdAt, and updatedAt. The document ID is ObjectId('630b9cbef1846f62d29911eb').

```
_id: ObjectId('630b9cbef1846f62d29911eb')
car: ObjectId('63009cefef5fc1000a6dd52')
user: ObjectId('62db0f555f5232fd862f6731')
bookedTimeSlots: Object
totalHours: 48
totalAmount: 33600
transactionId: "card_11bk9nJwQzOM9VgLiNuPbrm"
driverRequired: false
acRequired: false
premiumConditionRequired: false
createdAt: 2022-08-28T12:00:11.132+00:00
updatedAt: 2022-08-28T12:00:11.132+00:00
__v: 0

_id: ObjectId('630b9cbef1846f62d29911eb')
car: ObjectId('63009cefef5fc1000a6dd54')
user: ObjectId('630b21326fb730109e4ef54')
bookedTimeSlots: Object
totalHours: 24
totalAmount: 15600
transactionId: "card_11bk9nJwQzOM9VgLiNuPbrm"
driverRequired: false
acRequired: false
premiumConditionRequired: false
createdAt: 2022-08-28T20:29:46.651+00:00
updatedAt: 2022-08-28T20:29:46.651+00:00
```

Figure No 4.4: Booking Cars Table

4.5 Service Table

The information about the available services are stacked in this database

The screenshot shows the MongoDB Compass interface with the database 'cluster0.i85yj7.mongodb.net' and collection 'GARI_LAGBE.services'. The interface includes a sidebar with 'My Queries', 'Databases', and a 'services' section. The main area displays three service documents:

```
_id: ObjectId("63001e6bb2f934bacbc67b97")
name: "RIM REPAIR"
image: "https://i.ebayimg.com/thumbs/images/g/fQKAQSwMrte5Ui/a-11600.jpg"
serviceType: "External"
> bookedTimeSlots: Array
rate: 5000
createdAt: 2022-08-19T23:36:11.424+00:00
updatedAt: 2022-08-28T20:36:49.538+00:00
__v: 1
servicetime: 1

_id: ObjectId("6302f88e83f5bd6cc04d2660")
name: "AC REPAIR"
image: "https://go.carolinassaaa.com/wp-content/uploads/2020/06/Air_Conditioni..."
serviceType: "Internal"
> bookedTimeSlots: Array
rate: 4000
createdAt: 2022-08-19T23:35:34.845+00:00
updatedAt: 2022-08-26T11:53:12.191+00:00
servicetime: 3

_id: ObjectId("6302ff5983f5bd6cc04d266f")
name: "OIL CHANGE"
image: "https://saadify.com/wp-content/uploads/2022/05/Types-of-Engine-Oil-Gas..."
```

Figure No 4.5: Service Table

4.6 Service Booking Table

The information about booked services are stored in this database.

The screenshot shows the MongoDB Compass interface with the database 'cluster0.i85yj7.mongodb.net' and collection 'GARI_LAGBE.servicesbookings'. The interface includes a sidebar with 'My Queries', 'Databases', and a 'servicesbookings' section. The main area displays two booking documents:

```
totalAmount: 4000
transactionId: "card_11kb02Jwgr0W9VgL4dlPGVXr"
overallTestRequired: false
premiumServicesRequired: false
createdAt: 2022-08-28T20:36:02.798+00:00
updatedAt: 2022-08-28T20:36:02.798+00:00
__v: 0

_id: ObjectId("630bd1e116014afb26b822b0")
service: ObjectId("63001e6bb2f934bacbc67b97")
user: ObjectId("630bd1326fb730109e4e6f4")
> bookedTimeSlots: Object
totalHours: 24
totalAmount: 5000
transactionId: "card_11kb0nNwqz0W9VgLcHToOtc"
overallTestRequired: false
premiumServicesRequired: false
createdAt: 2022-08-28T20:36:49.430+00:00
updatedAt: 2022-08-28T20:36:49.430+00:00
__v: 0

_id: ObjectId("630bd20216014afb26b822b8")
service: ObjectId("6303097283f5bd6cc04d2672")
user: ObjectId("630bd1326fb730109e4e6f4")
> bookedTimeSlots: Object
totalHours: 24
```

Figure No 4.4: Booking Service Table

Chapter 5

User Actions

5.1 Successful Registration

The Successful message shown after valid Registration

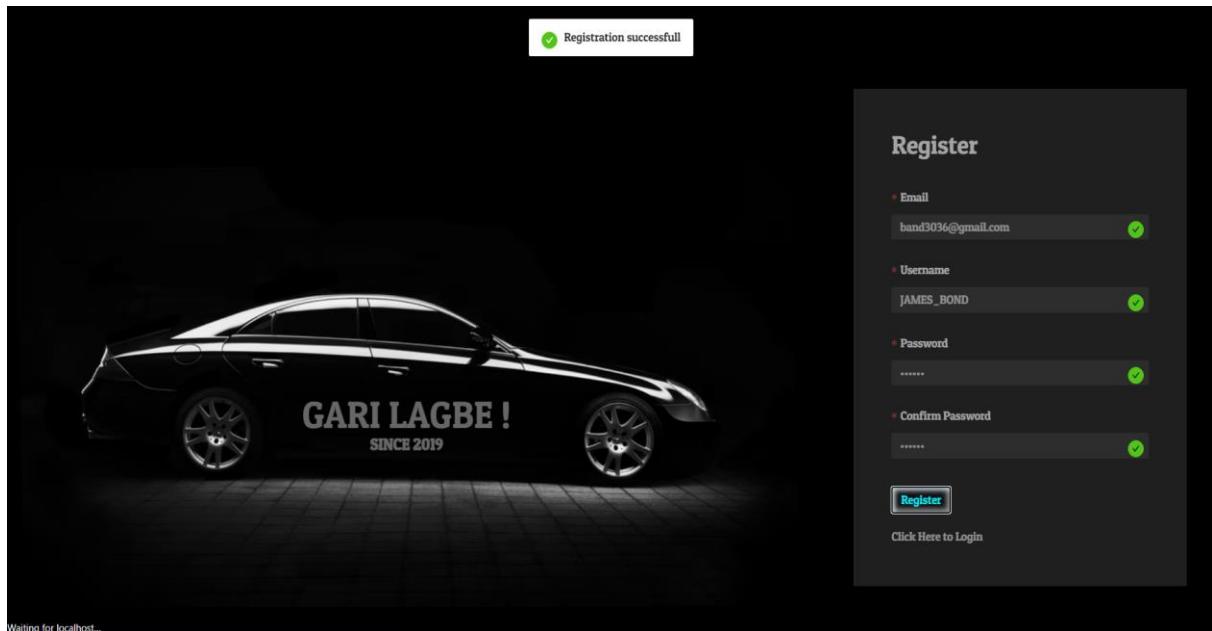


Figure No 5.1: Registration Successful

5.2 Successful LogIn

The Successful message shown after valid LogIn

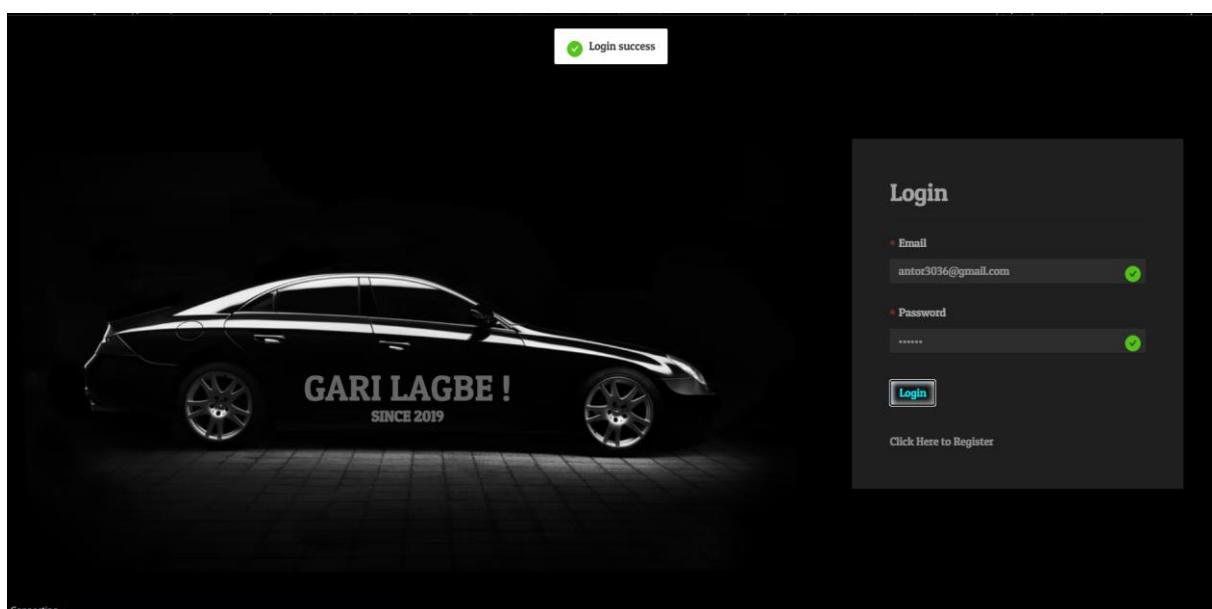


Figure No 5.2: LogIn Successful

5.3 Before Filtering Cars

Filtering cars based on availability

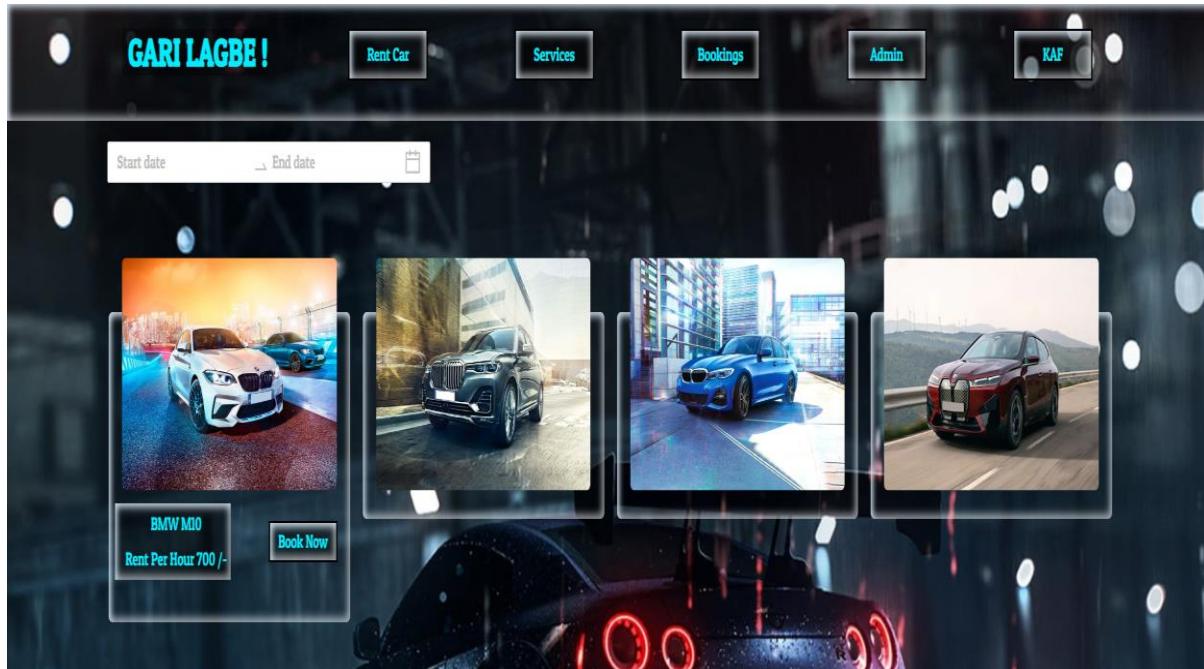


Figure No 5.3: Before Filtering Cars

5.4 After Filtering Cars

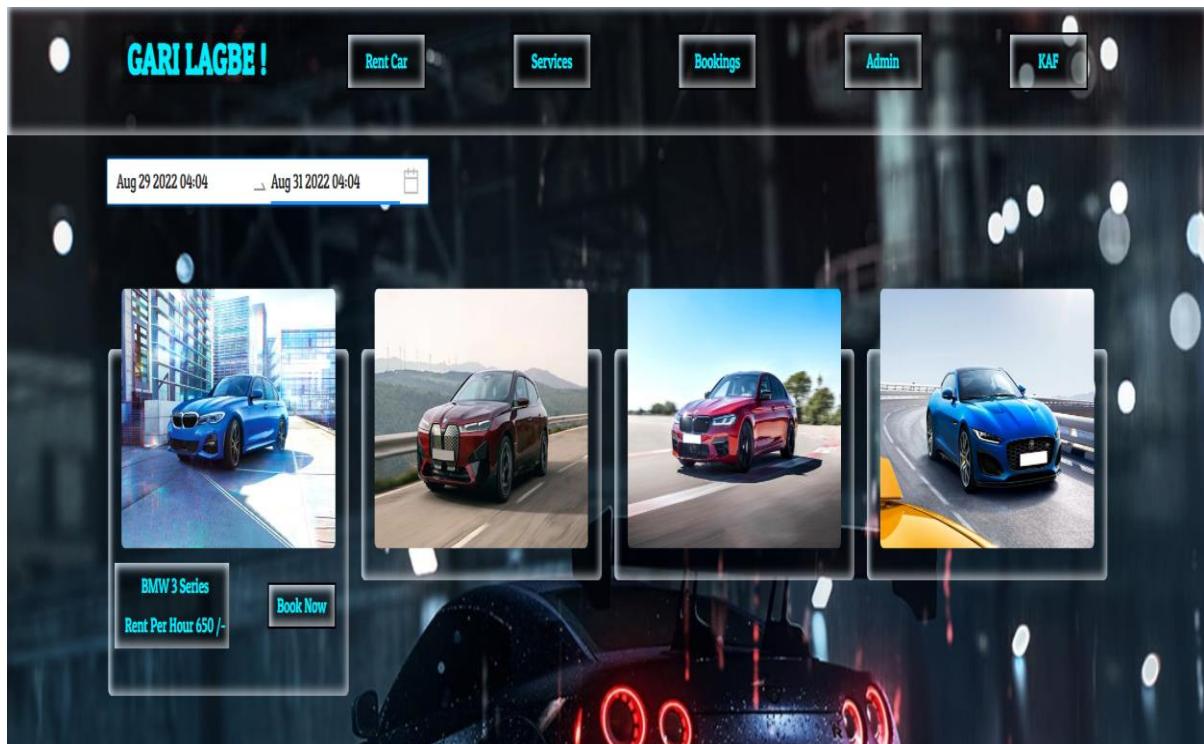


Figure No 5.4: After Filtering Cars

5.5 After Successful Bookings of Cars

Successful message is shown after completion of bookings of cars

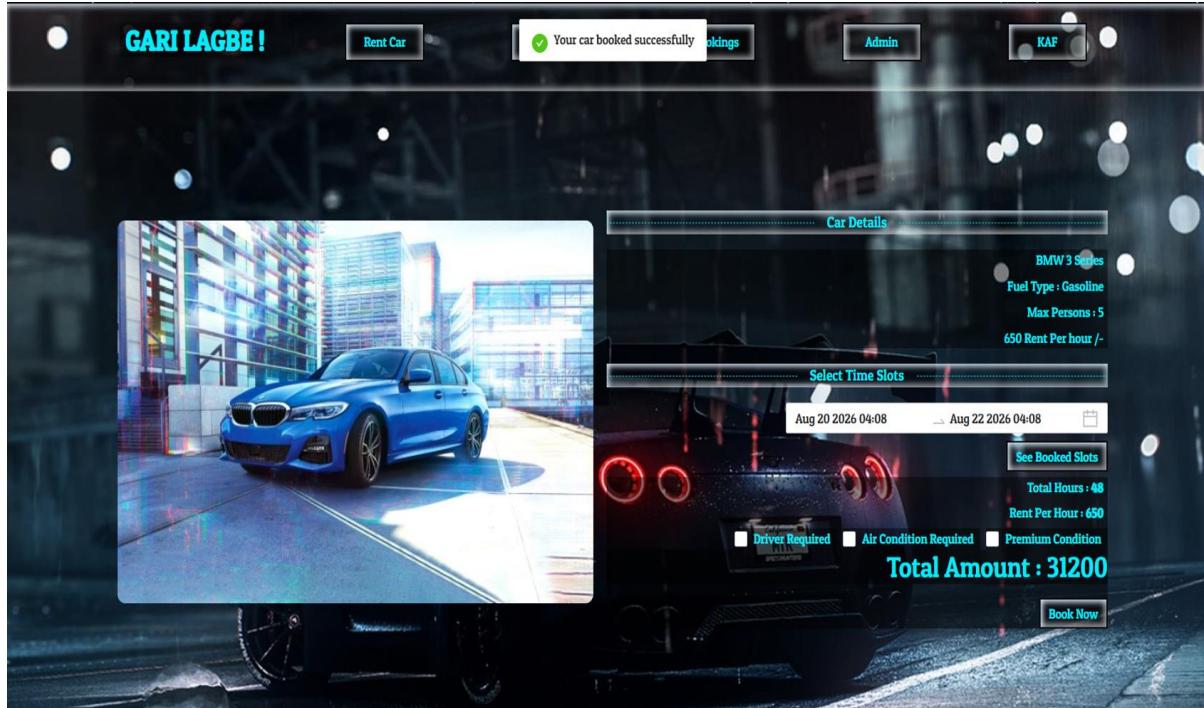


Figure No 5.5: After Successful Bookings of Cars

5.6 After Successful Bookings of Service

Successful message is shown after completion of bookings of services

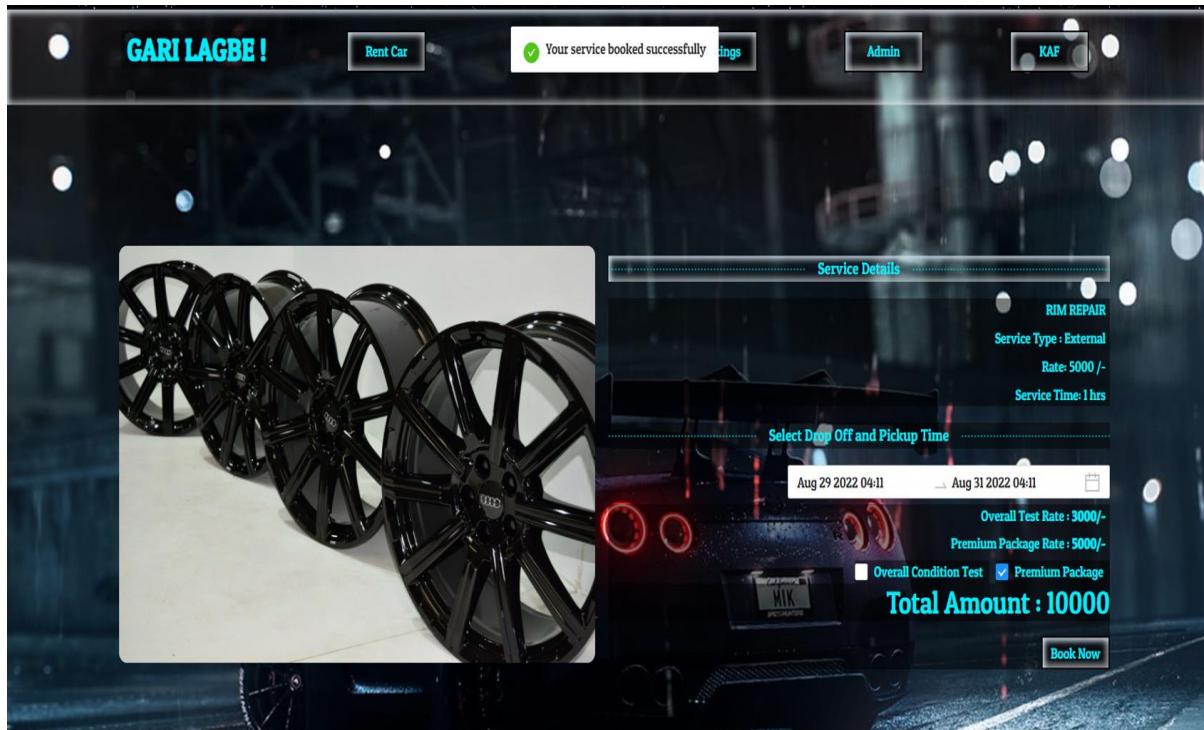


Figure No 5.6: After Successful Bookings of Service

5.7 After Adding a Car

After successfully adding a new car

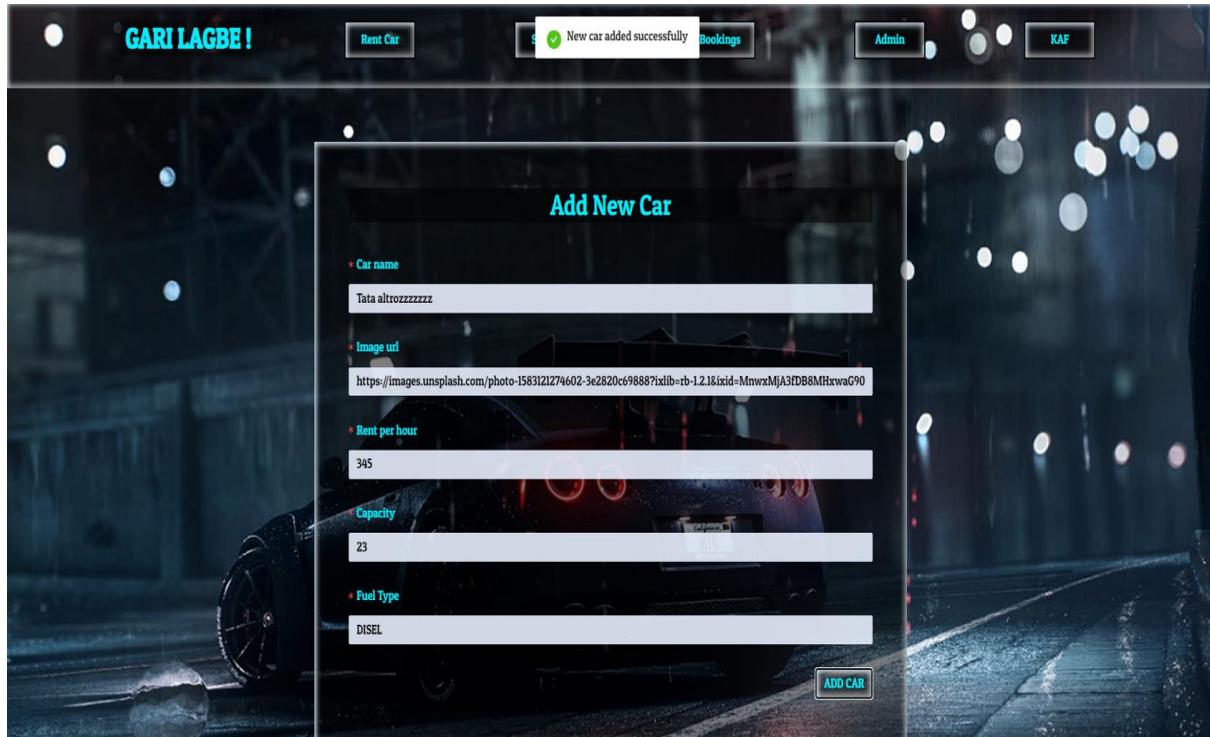


Figure No 5.7: After Adding a Car

5.8 After Successful Edition

After successfully editing a car

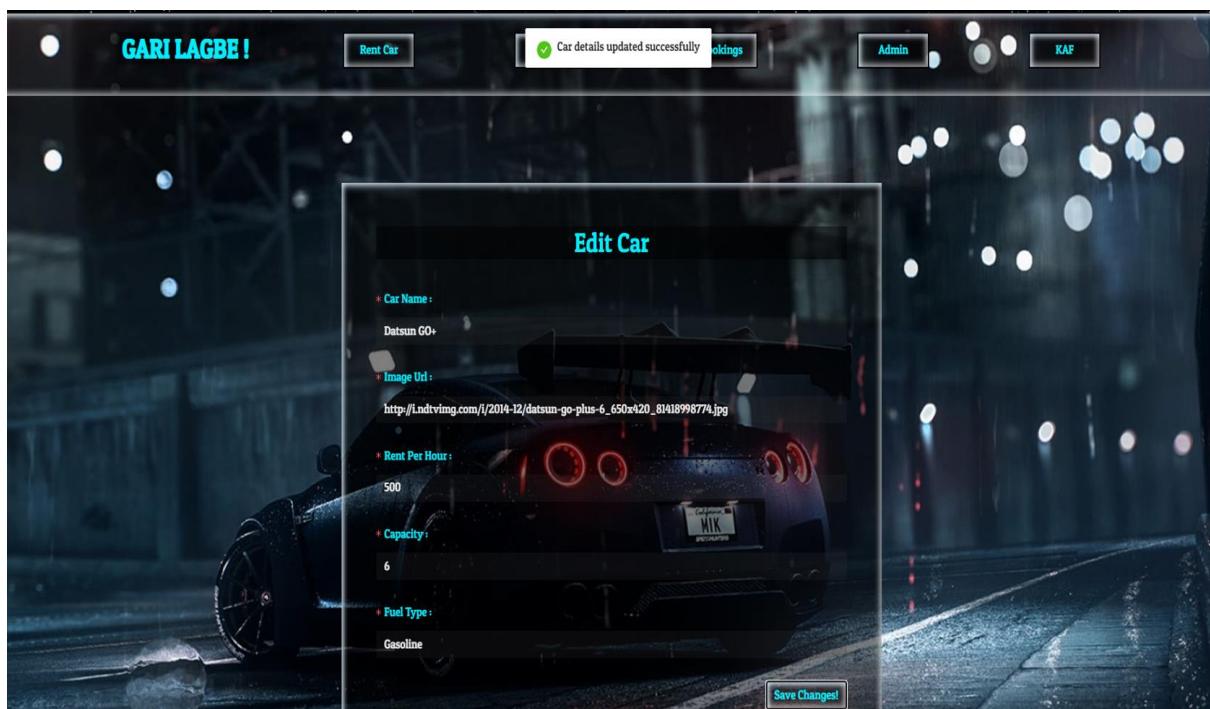


Figure No 5.8: After Successful Edition

5.9 Before Deleting Car

Pop Up confirming deletion of a car.

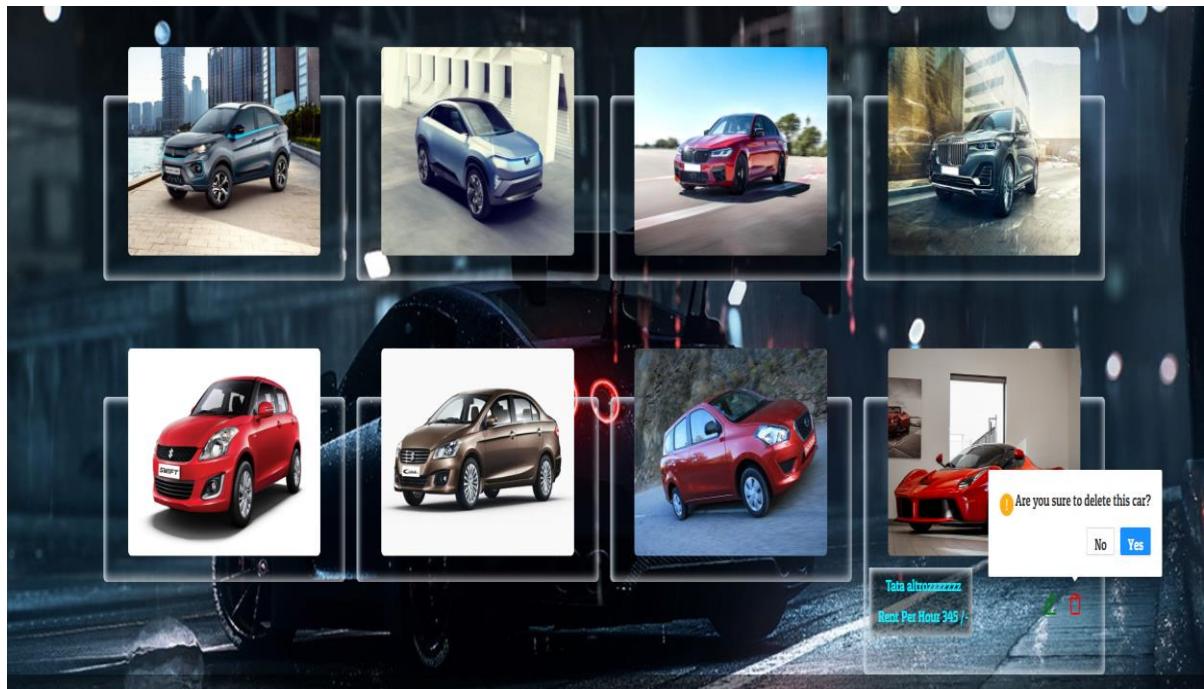


Figure No 5.9: Before Deleting Car

5.10 After Successful Deleting Car

Successful message after deleting a car

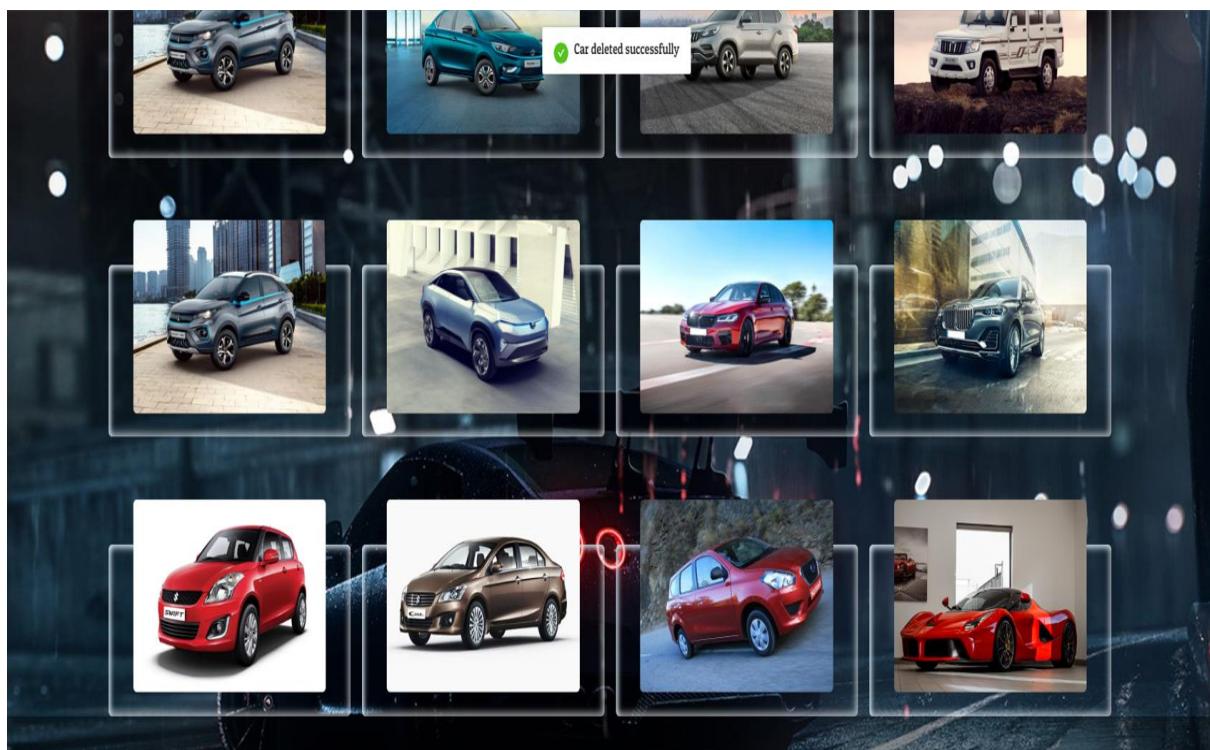


Figure No 5.9: After Successful Deleting Car

5.11 After Successful Adding Service

Successful message after adding a service

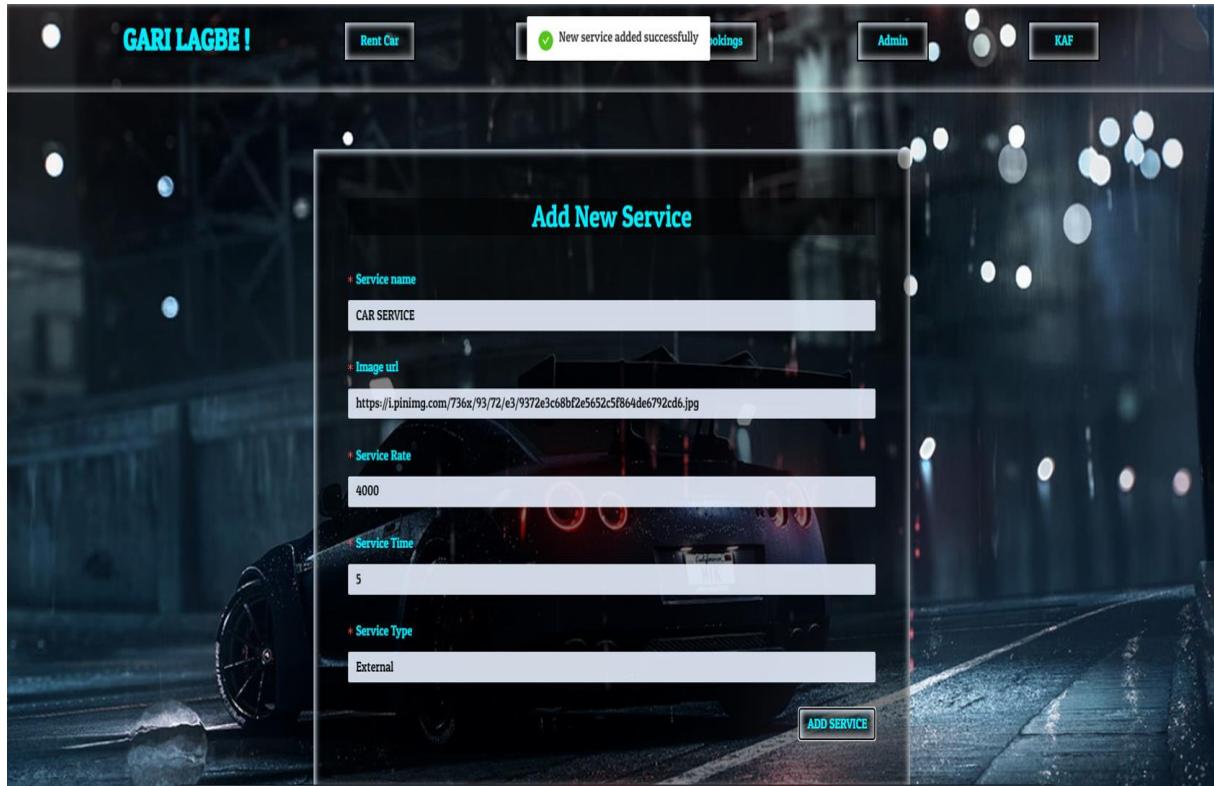


Figure No 5.11: After Successful Adding Service

5.12 Before Deleting Service

Pop Up confirming deletion of a service.

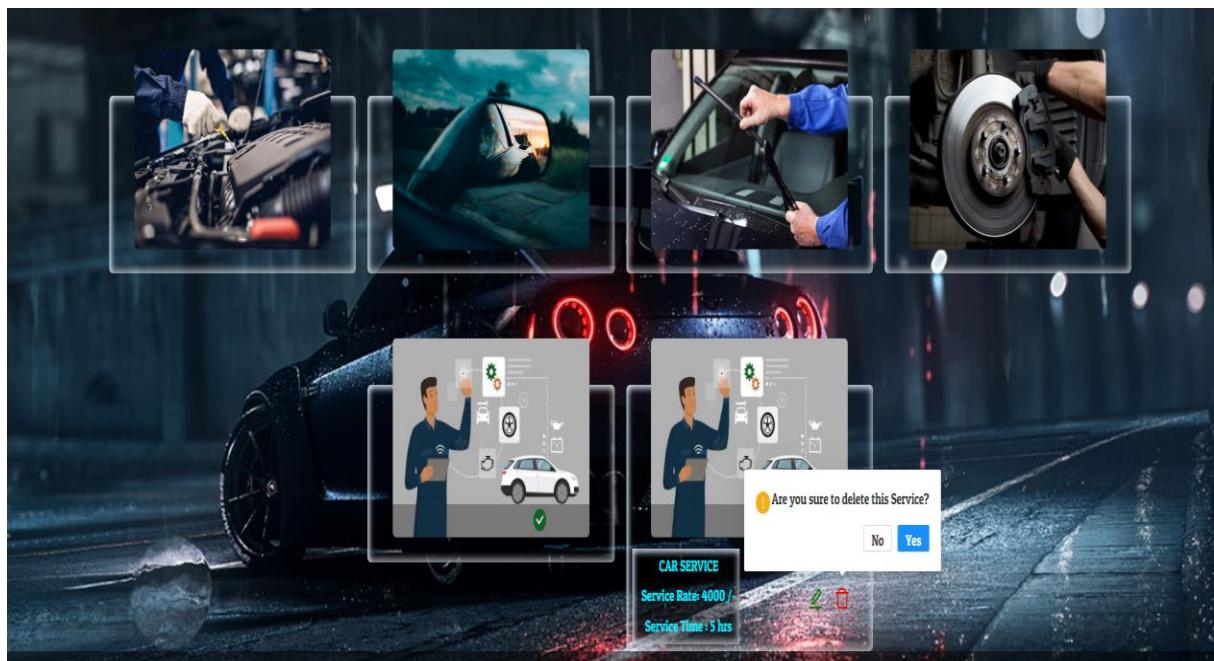


Figure No 5.12: Before Deleting Service

5.13 After Deleting Service

Successful message after deleting a service



Figure No 5.13: After Deleting Service

5.14 After Successful Updating of User

Successful message after updating user information

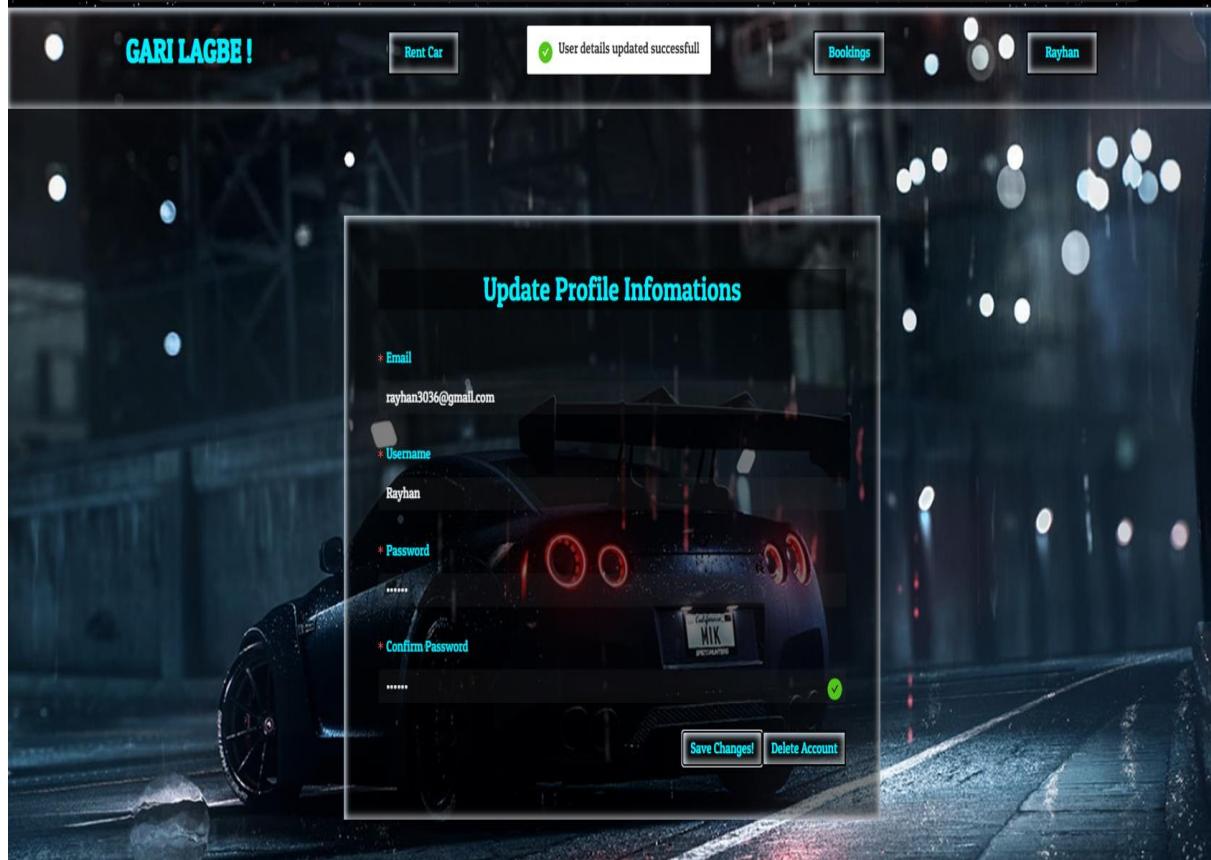


Figure No 5.14: After Successful Updating of User

5.15 After Successful Deletion of User

Successful message after deleting a user

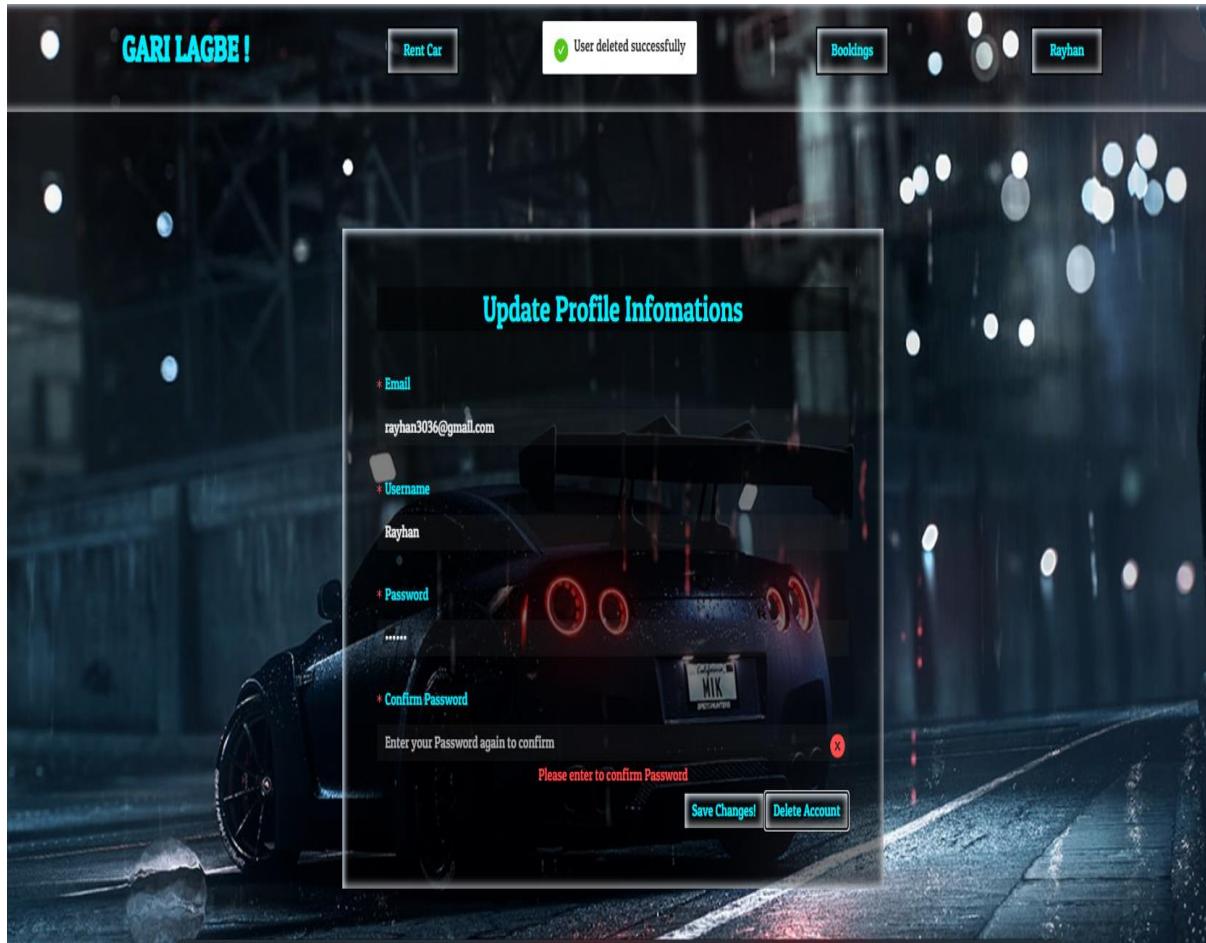


Figure No 5.15: After Successful Deletion of User

Chapter 6

Testing

6.1 Register Error 1

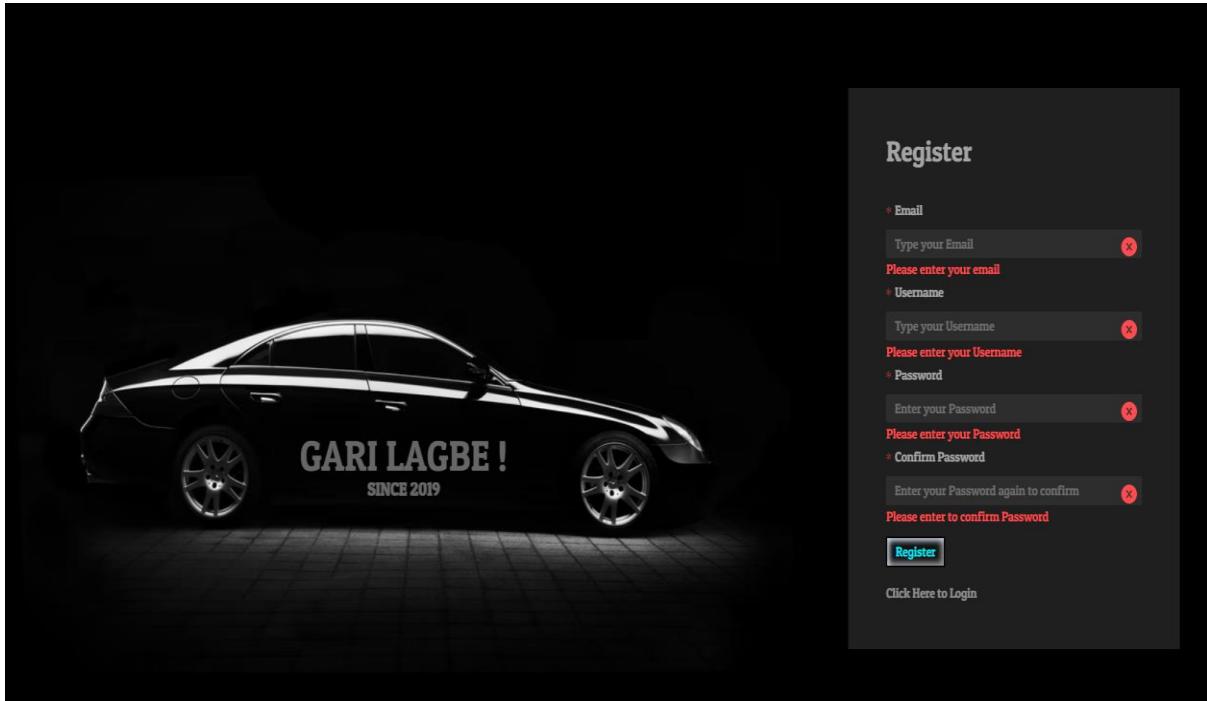


Figure No 6.1: Error shown due to null input.

6.2 Register Error 2

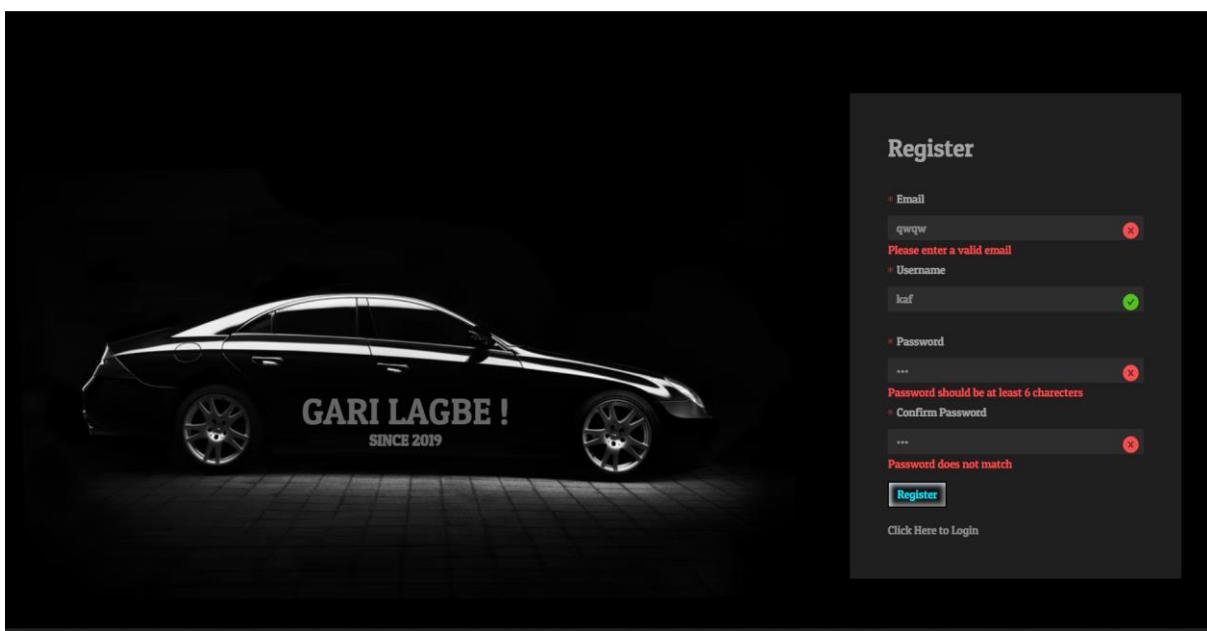


Figure No 6.2: Error shown due to invalid input

6.3 Login Error 1

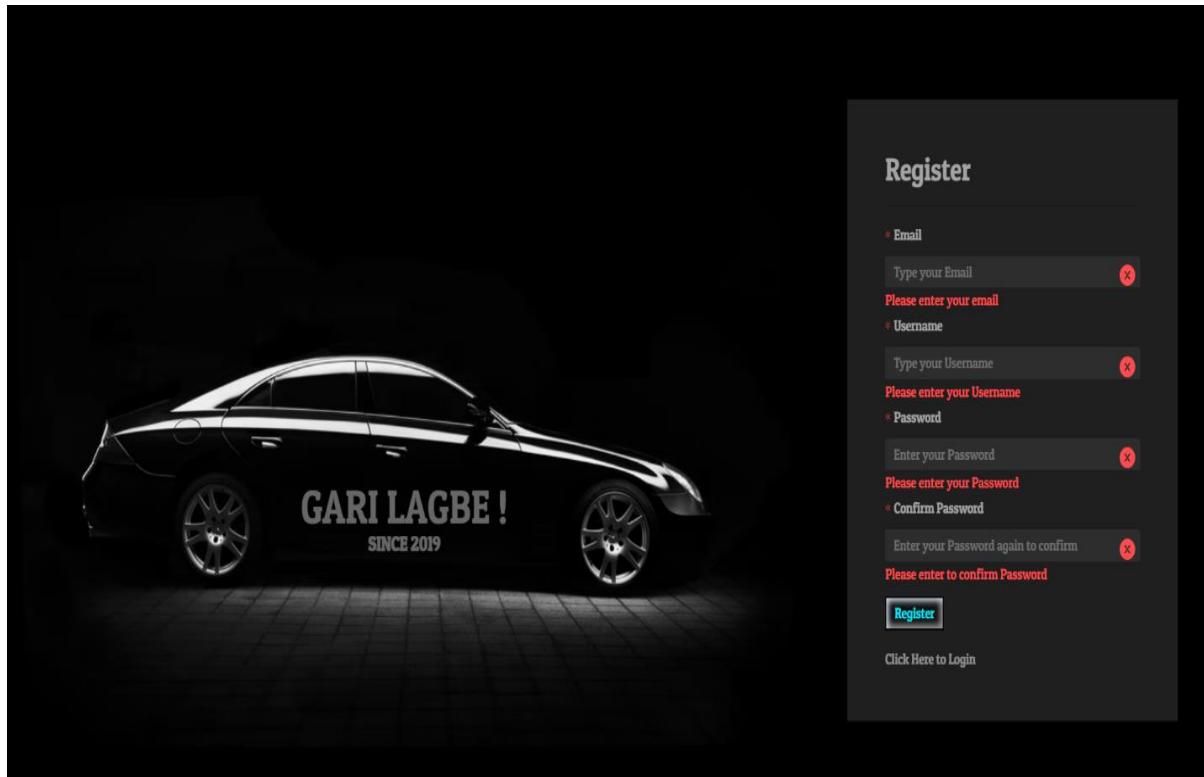


Figure No 6.3: Error shown due to null input.

6.4 Login Error 2

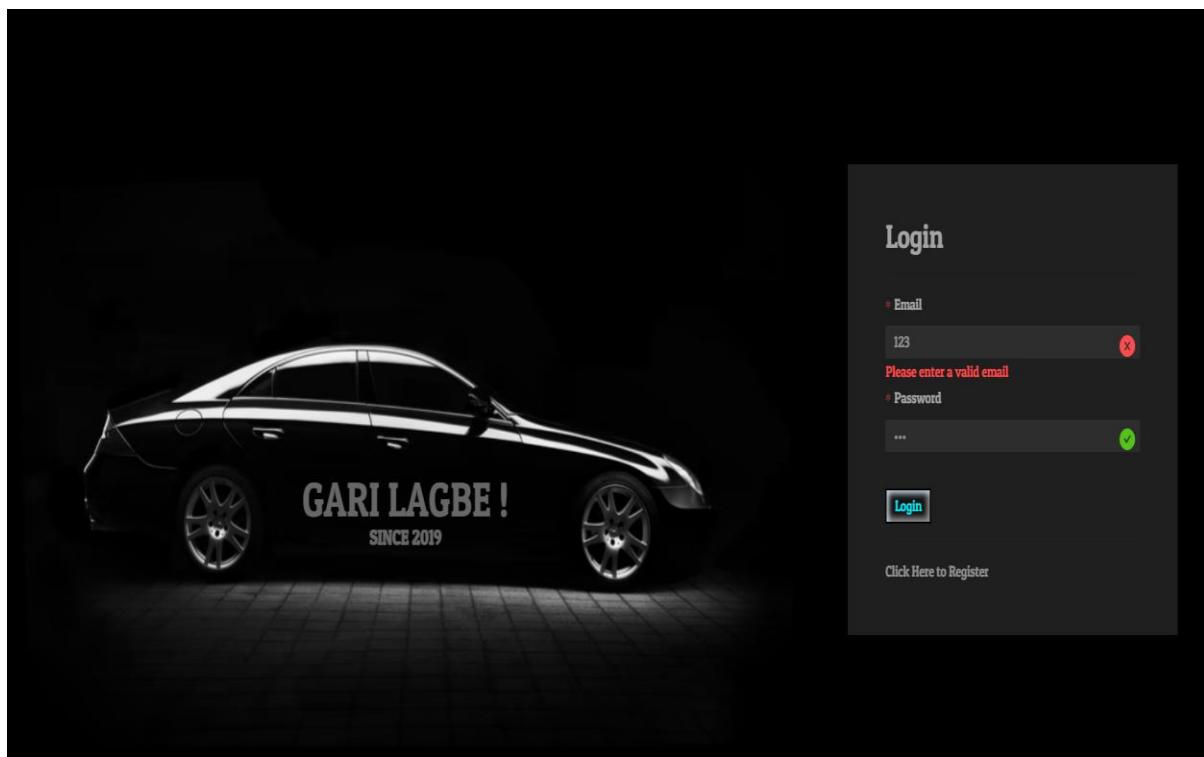


Figure No 6.4: Error shown due to invalid input.

6.5 Booking Car Error 1

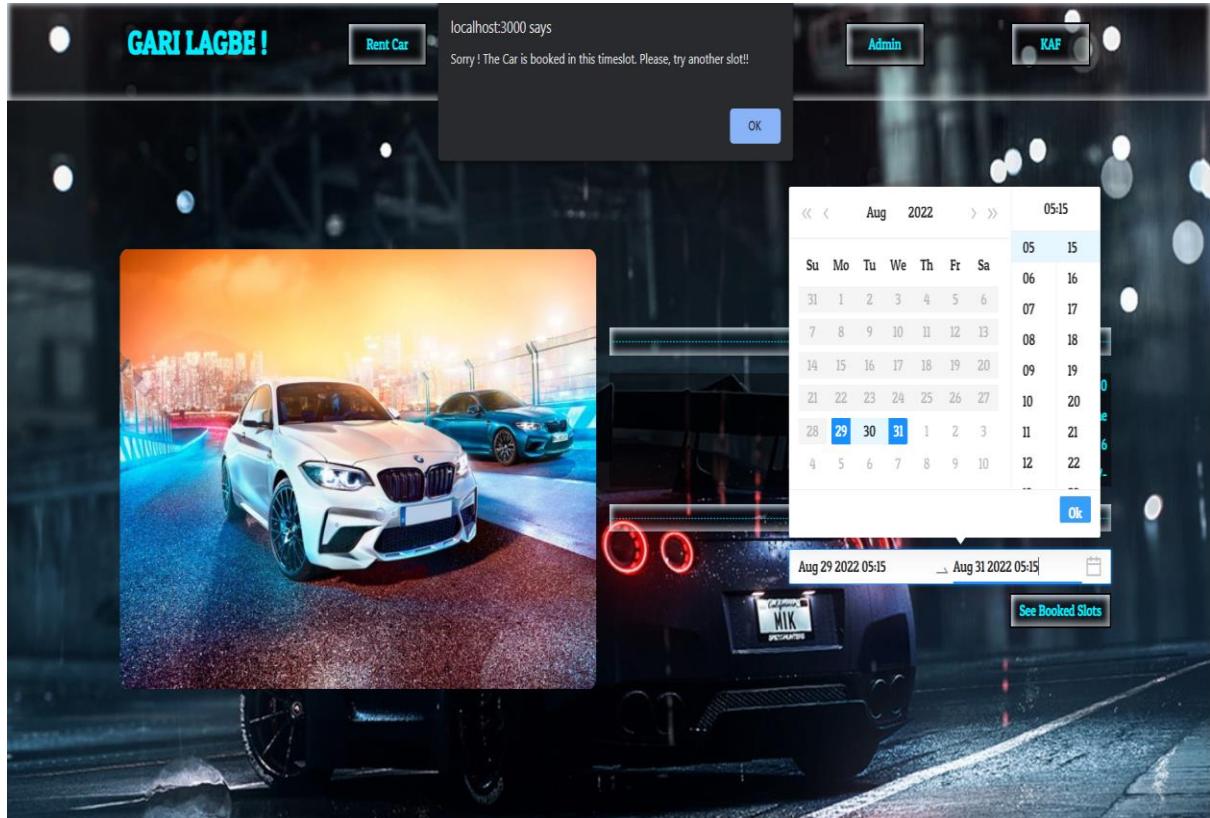


Figure No 6.5: Error shown due to trying to book in booked time slots

6.6 Booking Car Error 2

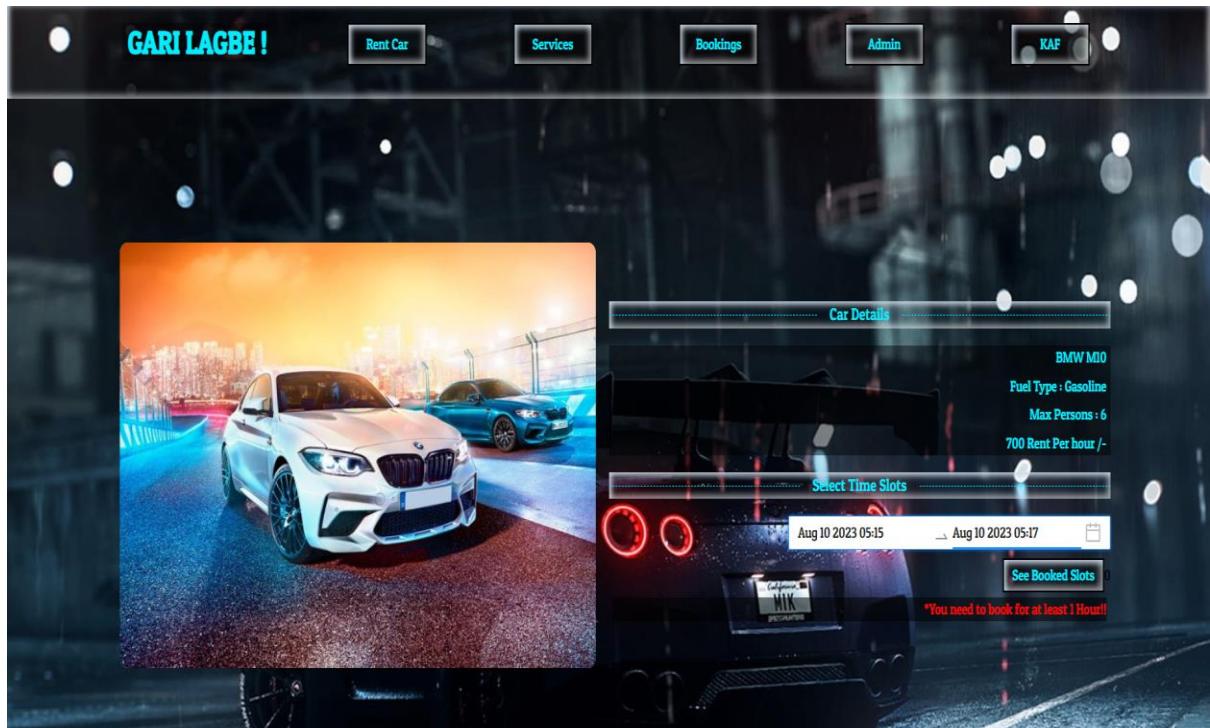


Figure No 6.6: Error shown due to trying to book for less than 1 hour

6.7 Service Error

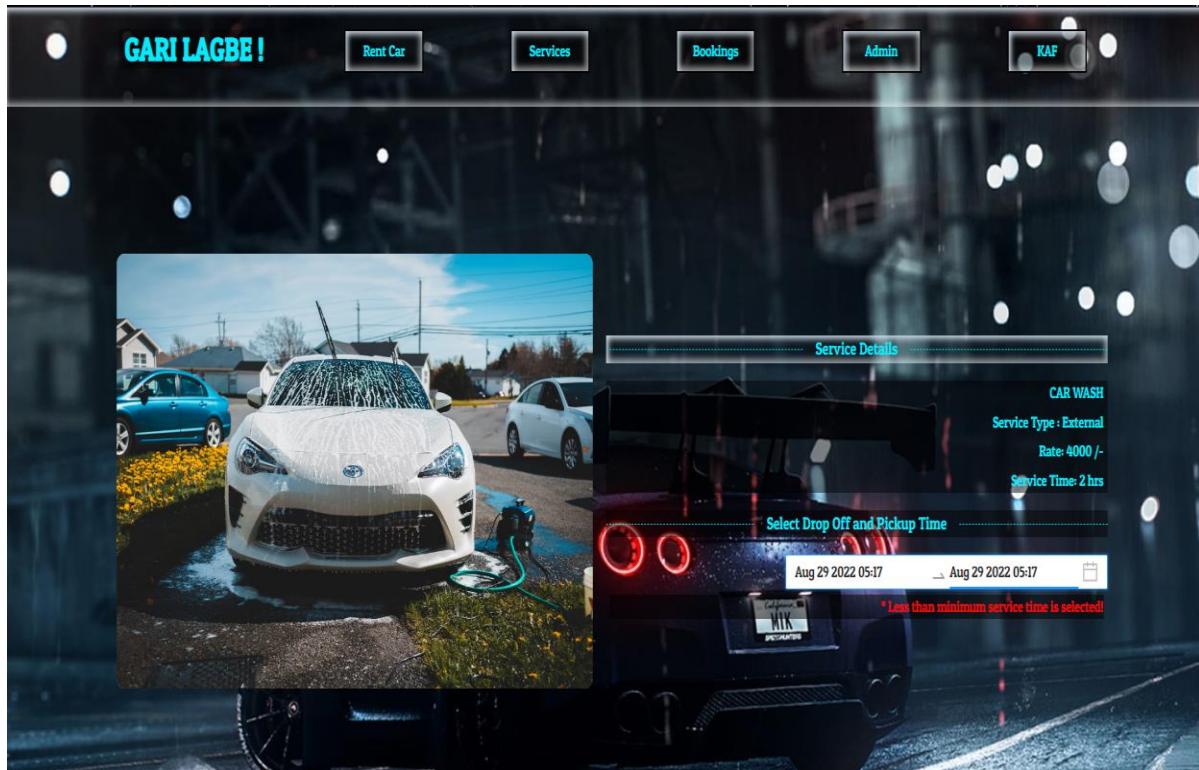


Figure No 6.7: Error shown due to trying to book for less than minimum service time

6.8 Car Admin Error 1

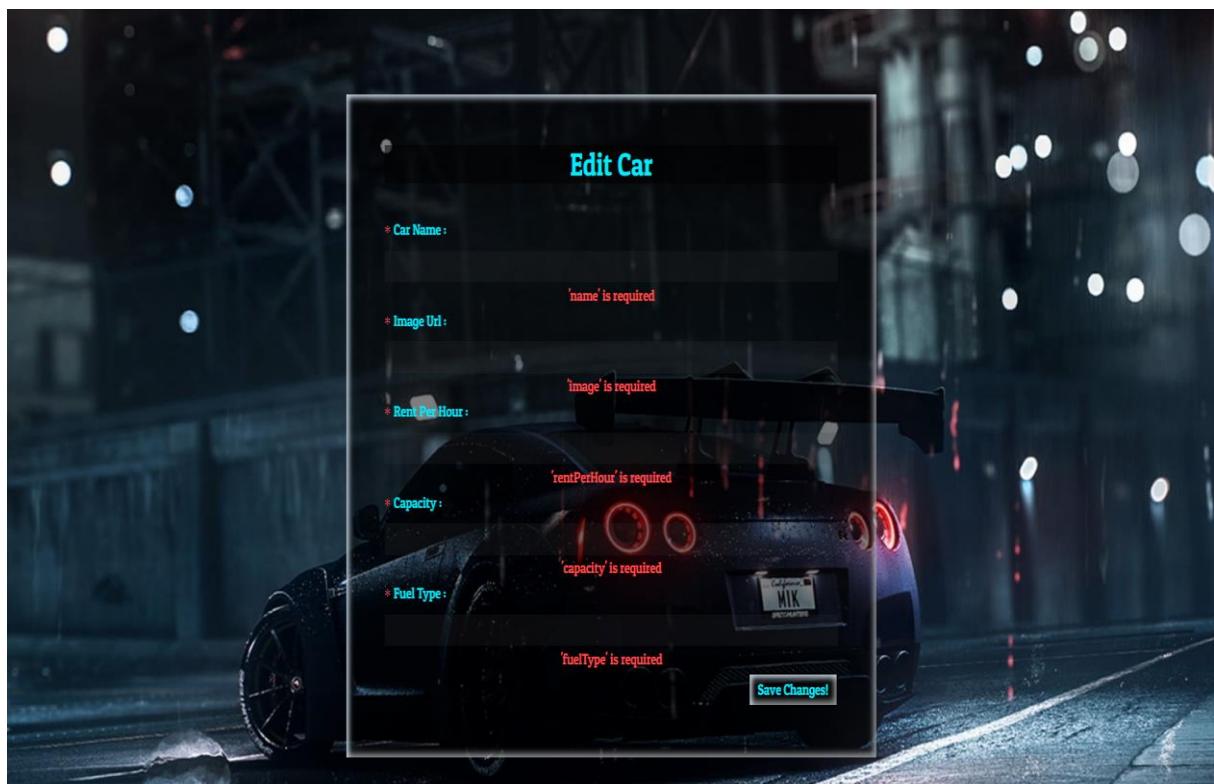


Figure No 6.8: Error shown due to null input

6.9 Car Admin Error 2

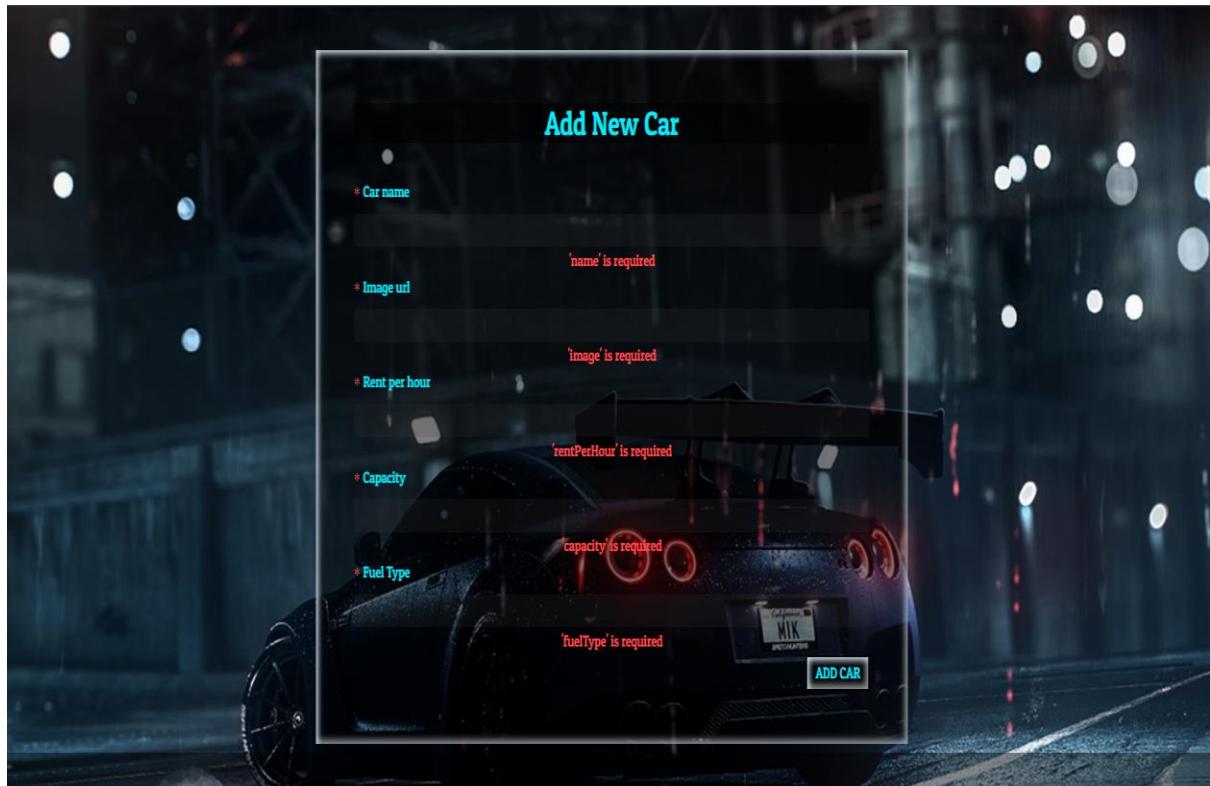


Figure No 6.9: Error shown due to null input

6.10 Service Error 1

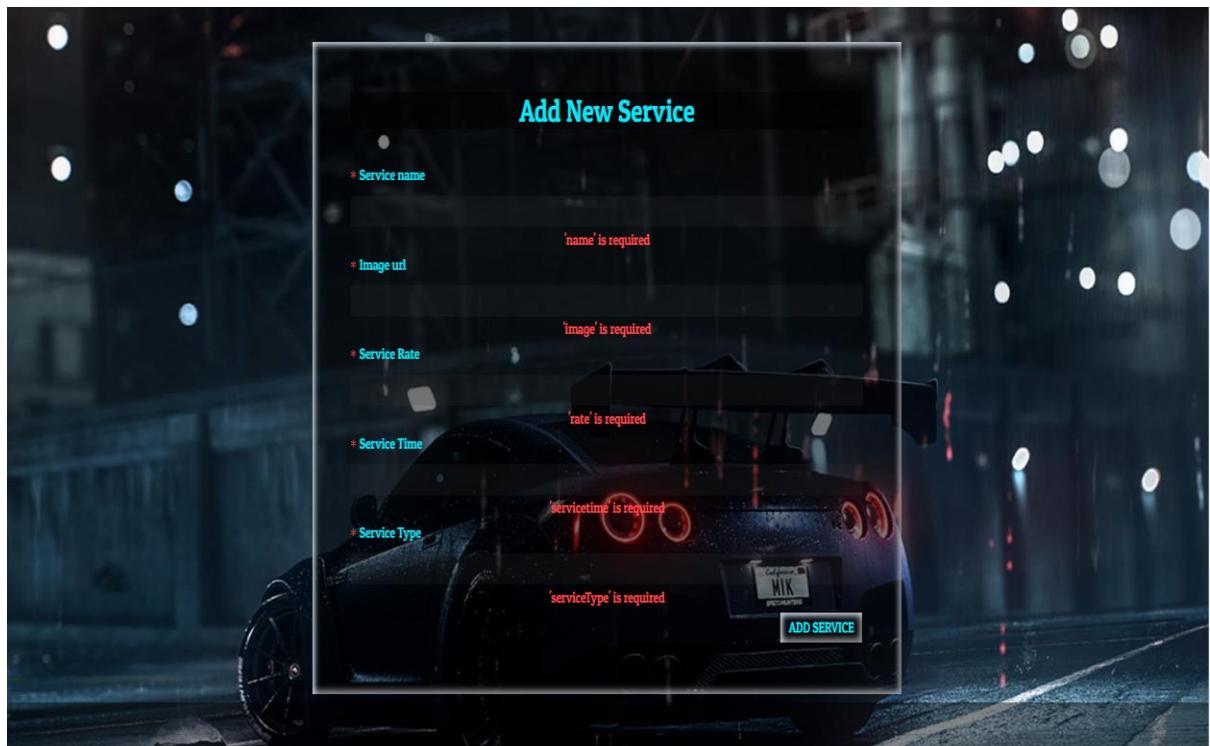


Figure No 6.10: Error shown due to null input

6.11 Service Error 2

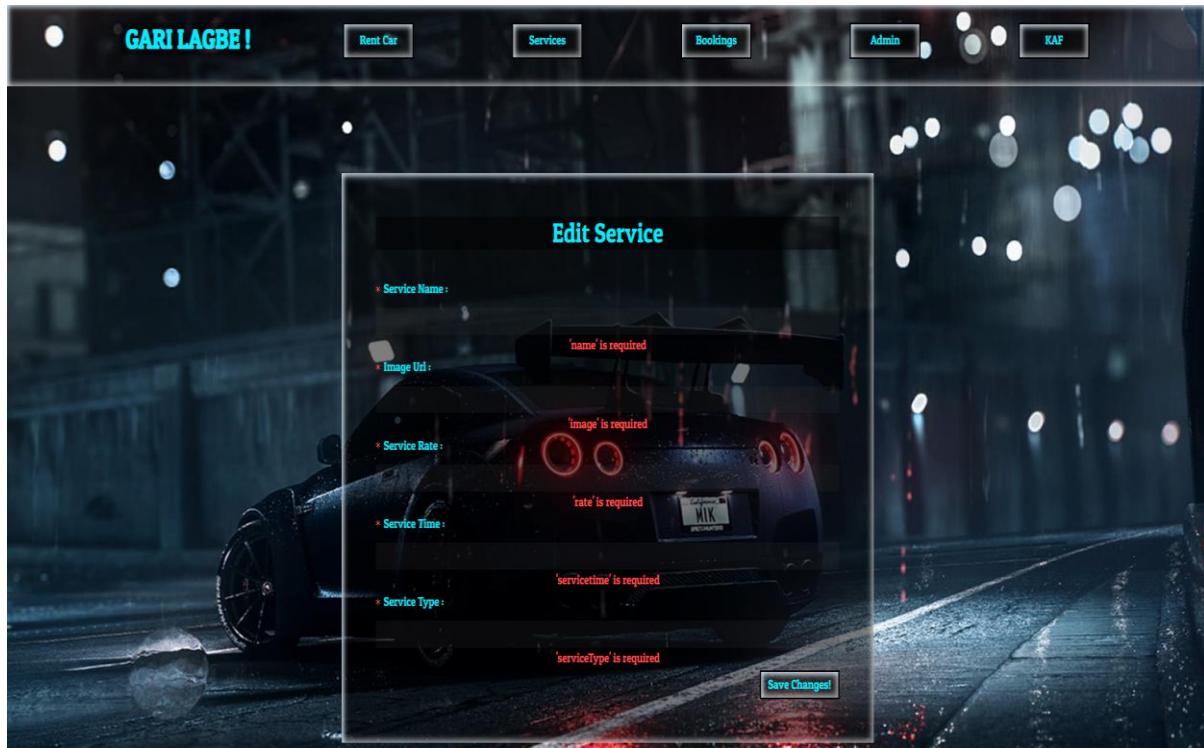


Figure No 6.11: Error shown due to null input

6.12 Update Profile Error

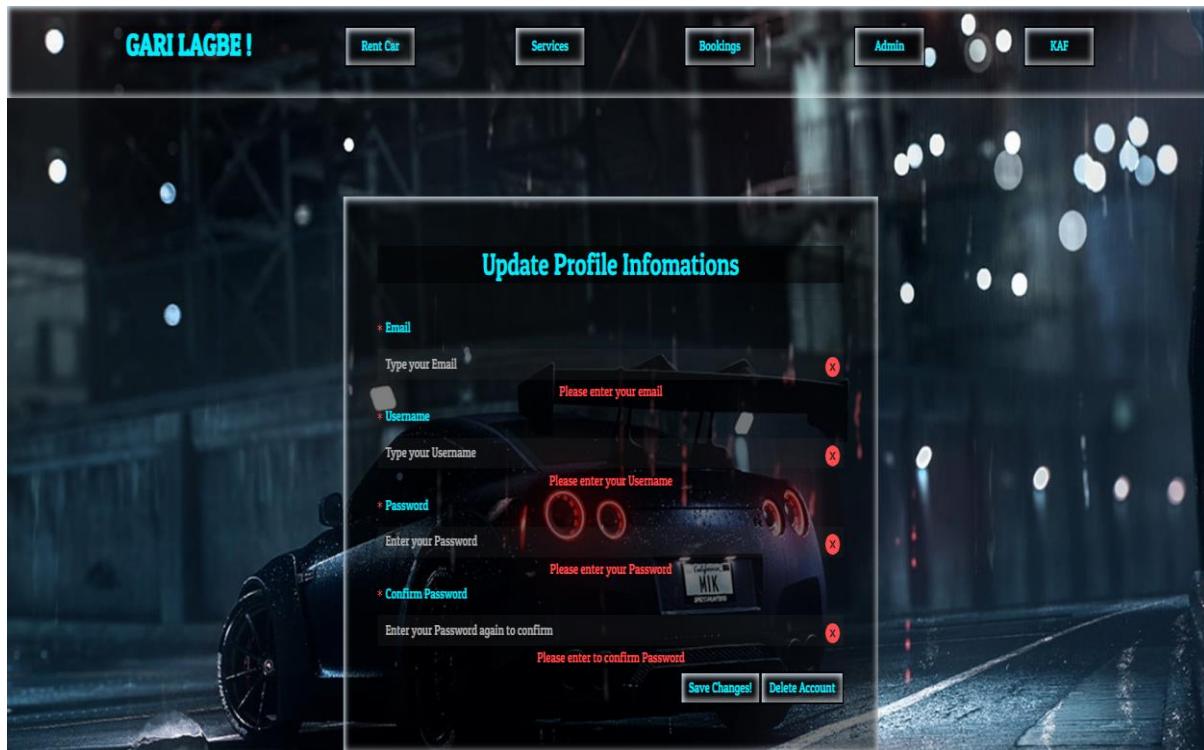


Figure No 6.12: Error shown due to null input

Chapter 7

Conclusion

With the pace of modern technological revolution it has become a mandatory option to automate every system. Car rental and servicing system is also not exceptional from that. To keep harmony with the massively rising population and their demands, automation in car rental and servicing system has become a compulsory task for Bangladesh. Though still some online car rental system exists, our goal is to provide more solid, developed and efficient system for our countrymen. This system will not only enhance our mobility but also will make our day to day life easier, simpler and smoother.