



Inspiring Excellence

**CSE471 : System Analysis and Design  
Project Report  
Project Title : QUICKTRIPS**

<b>Group No : 05, CSE471 Lab Section : 02, Spring 2024</b>	
<b>ID</b>	<b>Name</b>
23341112	Munawar Mahtab Ansary
23341117	Yasir Arafat Saimon
21201124	Mim Raihan
21241012	Nibir Neelim

## Table of Contents

Section No	Content	Page No
1	Introduction	03
2	Functional Requirements	03 - 05
3	User Manual	06
4	Frontend Development	07 - 22
5	Backend Development	23 - 26
6	Technology (Framework, Languages)	27
7	Github Repo Link	27
8	Individual Contribution	28

## **Introduction**

In today's fast-paced world, the demand for convenient and efficient transportation solutions is ever-growing. Car rental services play a pivotal role in meeting this demand, offering individuals and businesses the flexibility to travel on their terms. However, with the proliferation of online platforms and increasing customer expectations, it has become imperative for car rental companies to innovate and enhance their services to stay competitive in the market.

This project report delves into the comprehensive analysis and enhancement strategies undertaken by QuickTrips, a leading car rental platform renowned for its affordability and user-friendly interface. QuickTrips has recognized the evolving needs of its customers and embarked on a journey to elevate their experience while streamlining operational processes.

The report provides an in-depth exploration of QuickTrips' website architecture, encompassing key pages such as the Login Page, Home Page, Vehicles Page, Contact Us Page, Rent History Page, About Us Page, and Profile Page. Each page serves a distinct purpose in facilitating seamless navigation, efficient booking, and effective communication between the users and the company.

Furthermore, the report highlights the integration of advanced features such as password reset functionality, payment options, and user profile management to enhance convenience and security for customers. Additionally, it discusses the implementation of a responsive design framework to ensure optimal performance across various devices, thereby accommodating the diverse preferences of users.

Through meticulous data analysis and user feedback, QuickTrips has identified areas for improvement and optimization, leading to the development of tailored solutions aimed at enhancing the overall user experience. The project report outlines these initiatives and their impact on customer satisfaction, retention, and operational efficiency.

Overall, this project report offers valuable insights into the strategies employed by QuickTrips to elevate its car rental services and maintain its position as a preferred choice for customers seeking affordable and convenient transportation solutions. By embracing innovation and continuously refining its offerings, QuickTrips aims to set new benchmarks in the car rental industry while exceeding customer expectations.

## **Functional Requirements**

### **MODULE 1**

- User Log in & Log out
  - Users should be able to securely log in and log out of their accounts.
- Password Management:
  - Users should have the ability to set and update their passwords securely.
- User Profile:
  - Users should be able to view their profiles, including personal information and contact details.
- Registration:
  - New users should be able to register for an account with the system securely.
- Session Management:
  - Implement session management to control user sessions, including session expiration and session hijacking prevention

### **MODULE 2**

- Email verification:
  - Implementing email verification as a mandatory step for setting a new password enhances security by confirming the user's identity.
- User profile editing:
  - Users should be able to edit their profiles, including personal information and contact details.
- User Management
  - Admin should have the ability to manage user accounts, including suspending or banning users if necessary.
- Emergency Contact
  - Customers should have a secure way to contact the admin in case of emergencies
- Car Management:
  - Admin should be able to add new cars to the system and remove existing ones.

- Dashboard:
  - Integrating a dashboard into your platform provides users with a centralized hub for easy access to key information, analytics, and features.

## **MODULE 3**

- Car Search:
  - Customers should be able to search for available cars based on their preferences.
- Reservation:
  - Customers should be able to reserve cars for rental securely.
- Reservation Management:
  - Admin should be able to cancel reservation requests for cars and track rented cars.
- Feedback Management:
  - Admin should be able to view feedback from users and leave comments if needed.
- Vehicle History Access:
  - Customers should have access to vehicle history reports for cars listed on the platform.

## **MODULE 4**

- Transaction Management:
  - Via Bkash.
  - Through credit/debit card.
- About Us page
  - Our website offers a dynamic "About Us" page where customers can learn more about us.
- Location and Time Tracking:
  - Admin should have access to location and time details of car rentals for monitoring purposes.
- Review system
  - customer can give a review
  - admin can see all the customer's review
  - Customers have the ability to view reviews for all cars

- Rental History:
  - Allow customers to view their complete rental history, including past bookings, and payments.
  - Allow customers to view their complete past bookings and payments.
  - Allow customers to view their complete payments.
- Chatbot (AI chatbot):
  - Employing AI technology, the Chatbot feature enhances user interactions by providing real-time, engaging conversational support for inquiries and assistance.

## User Manual

### Prerequisites

- 1) Python 3.8 or newer:
  - If you don't have Python installed, download and install the latest version from the official Python website.
  - Verify the installation by opening a terminal/command prompt and typing `python --version`.
- 2) Django 3.1 or later:
  - After installing Python, you can install Django using pip, Python's package manager. Open a terminal/command prompt and type: `pip install Django`
  - Verify the installation by typing `django-admin --version`.
- 3) Other dependencies such as pip and VS Code:
  - Install any additional dependencies you might need for your project using pip.
  - Download and install Visual Studio Code from the official website.
  - Once installed, you can open your project folder in VS Code to start coding.

### Installation

- 1) Clone the repository:
  - a) `git clone https://github.com/Yasir2521/Car-Rental-System.git`
- 2) Navigate to the project directory:
  - a) `cd Car-Rental-System`
- 3) Install required packages:
  - a) `pip install -r requirements.txt`

4) Migrate the database:

a) `python manage.py migrate`

5) Run the development server:

a) `python manage.py runserver`

## Frontend Development

Our website comprises two primary interfaces: the user interface and the admin interface. Therefore, the user manual is divided into two sections to provide comprehensive guidance for both types of users.

### Chapter 1

#### 1.1 Homepage:

The homepage of QuickTrips serves as the central hub for users, providing easy access to all the features and services offered by our platform. Upon landing on the homepage, users are greeted with a clean and intuitive interface designed to streamline their navigation experience and are required to register on our website to explore further if they have never visited before. Else they can just log in with their credentials.

The user can also see some of our most popular cars which are featured in our website.

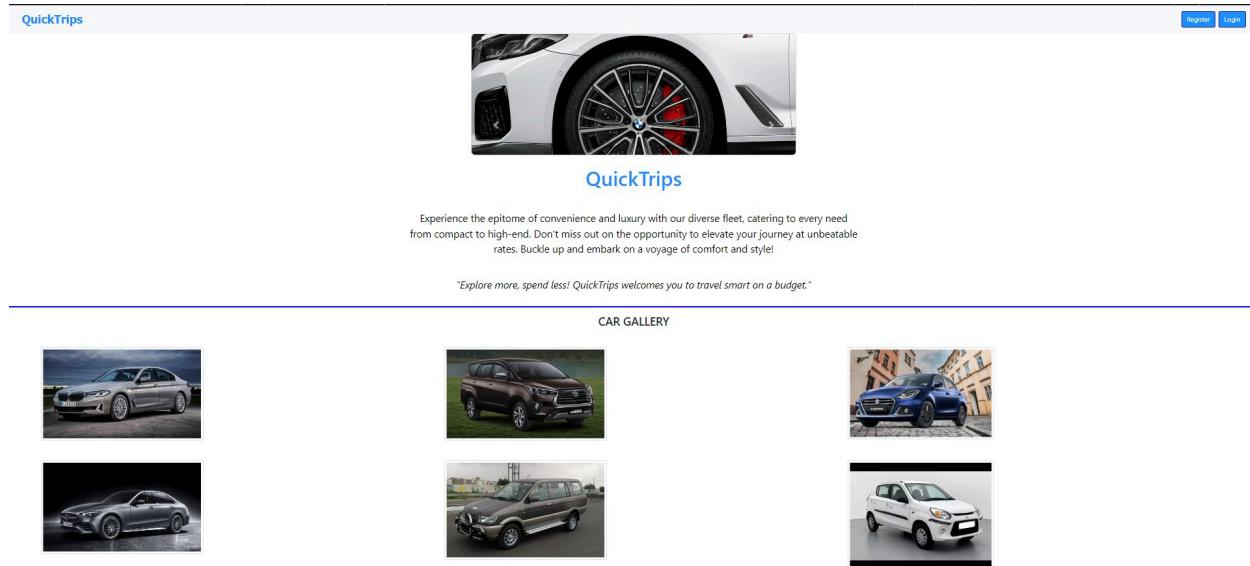


Figure : HomePage

## 1.2 User Login/Signup:

### 1.2.1. User Signup:

- To sign up, we need to fill out the form with your First Name, Last Name, Mobile Number, Email address, and precious password, and confirm the password to recheck.

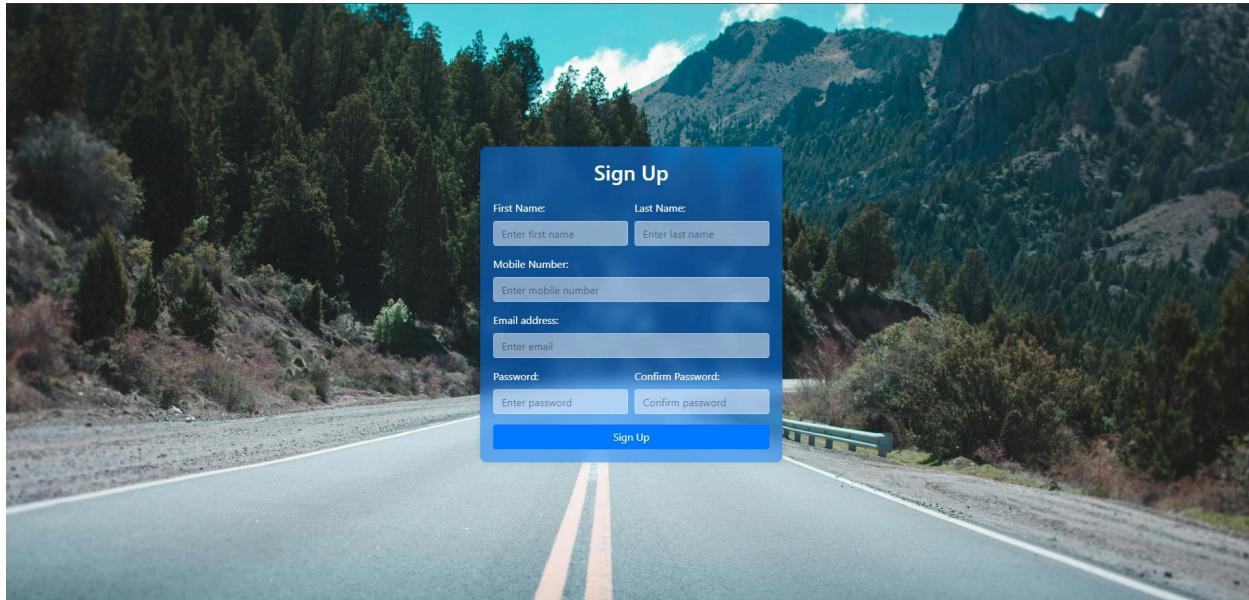


Figure : Sign up page

### 1.2.2 Authentication:

- The user cannot use numbers in the First Name and Last Name boxes
- User cannot use any phone number which is not 11 digits
- The user has to use a legit email. If the email address does not match with the basic format such as: if there is no @mail.com in the mail, then the user cannot sign up.
- The password needs to have at least 8 characters, including letters and numbers so that the user does not have to use any easy-to-guess password.

### 1.2.3 User Login:

- The user then logs in to the website using the email address and password.
- To err is human. If the user uses the wrong password, then the system will not let the user log in and prompt an invalid password message.
- If the user tries to use any sort of wrong credentials while logging in, such as the wrong email, then the website won't let the user log in.

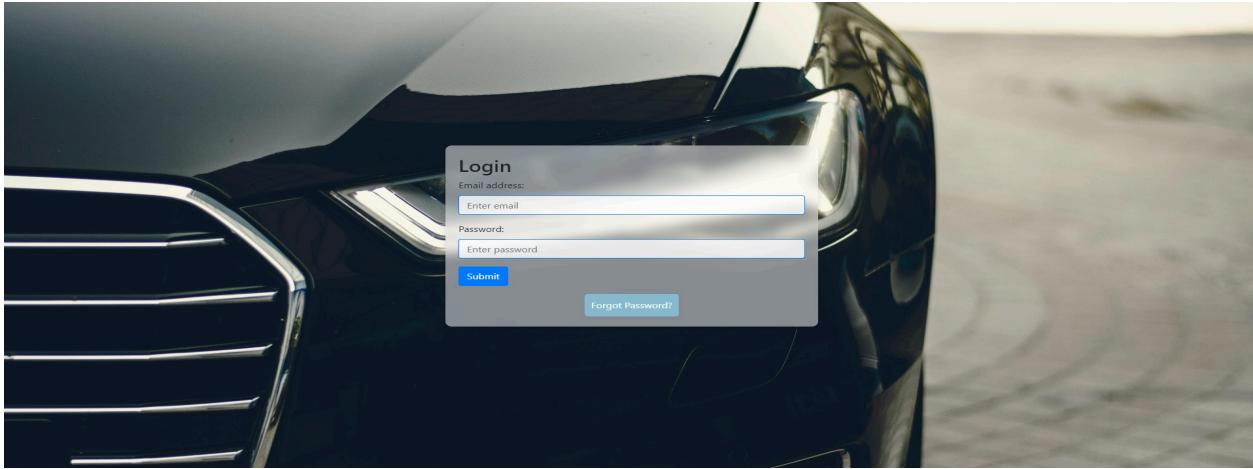


Figure : Login page

#### 1.2.4 Password Recovery:

- People are forgetful. Let's say the user forgot the password of his account. We made it very simple and easy for users to recover their passwords.

The composite image illustrates the password recovery process:

- Email Step:** An email from "quicktrips178@gmail.com" is shown in the inbox, with the subject line "Reset your password".
- Confirmation Step:** A screenshot of a web page titled "Reset Password" shows the message "An email has been sent to your given Gmail account. Use the mail to reset your password." with a "Visit Gmail" button.
- Reset Form Step:** A screenshot of a "Set New Password" form. It includes a "New password:" field, a list of password requirements (e.g., "Your password can't be too similar to your other personal information"), and a "New password confirmation:" field. A "Reset Password" button is at the bottom.
- Success Step:** A screenshot of a red box titled "Password Reset Complete" with the message "The password has been reset. Please visit the login page." and a "Return to Login" button.

Figure : Password Recovery through mail

After logging in, at the top of the homepage, users will find a navigation bar featuring essential buttons:-

#### 1.3 Homepage button features:

- QuickTrips: Leads back to the homepage
- Vehicles: Directs to the Vehicles Page
- Contact: Navigate to the Contact Us Page,
- About Us: Provides information about QuickTrips

- Rent History: Accesses the Rent History Page
  - Bhororsha: An AI bot, your personal Car rental Helper.
- \*\*\*A dropdown menu offers convenient access to user profile settings and the option to log out.

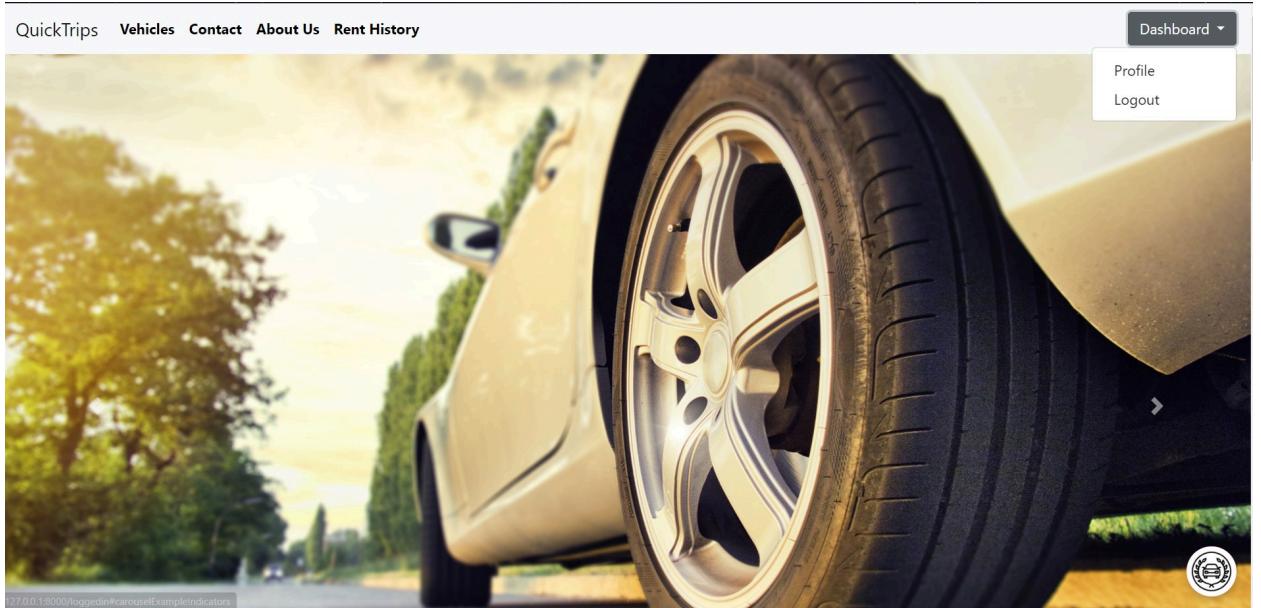


Figure 5: After logging, a successful message and Home page (After logging)

#### 1.4 3D Car Showcase:

- Our homepage shows 3 Dimensional Car models.
  - The user can zoom in and out to see the inside parts of the car so that anyone can see the condition of the car and see if they want to buy it or not.
  - Users can check the condition and aesthetics of the car using this feature.

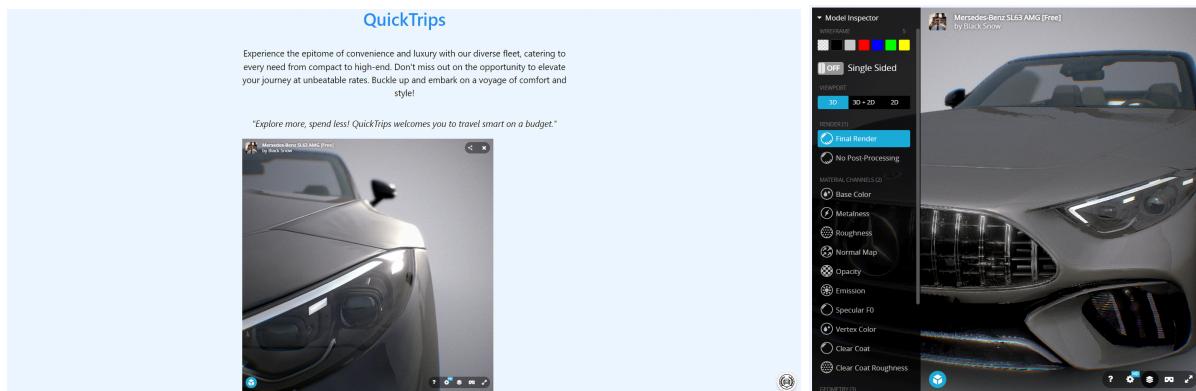


Figure :3D Car Model

## 1.5 Vehicles:

After clicking on the vehicles page, the user can see what sort of cars are being offered in our website. Let's say our database contains information about the cars' brand, model name, pictures, pricing, etc. This information is shown here so that the user can choose easily which car they want to rent and for how many hours considering the condition of the car. If any car is already booked, it does not show up on this page.

### 1.5.1 Search Bar:

- The user can search for a car by writing the name of the brand they want to rent.
- This search bar works as a filter, it does not show any other car of other brand other than the one it's searched on
- If a car is booked, it's not shown in search results.

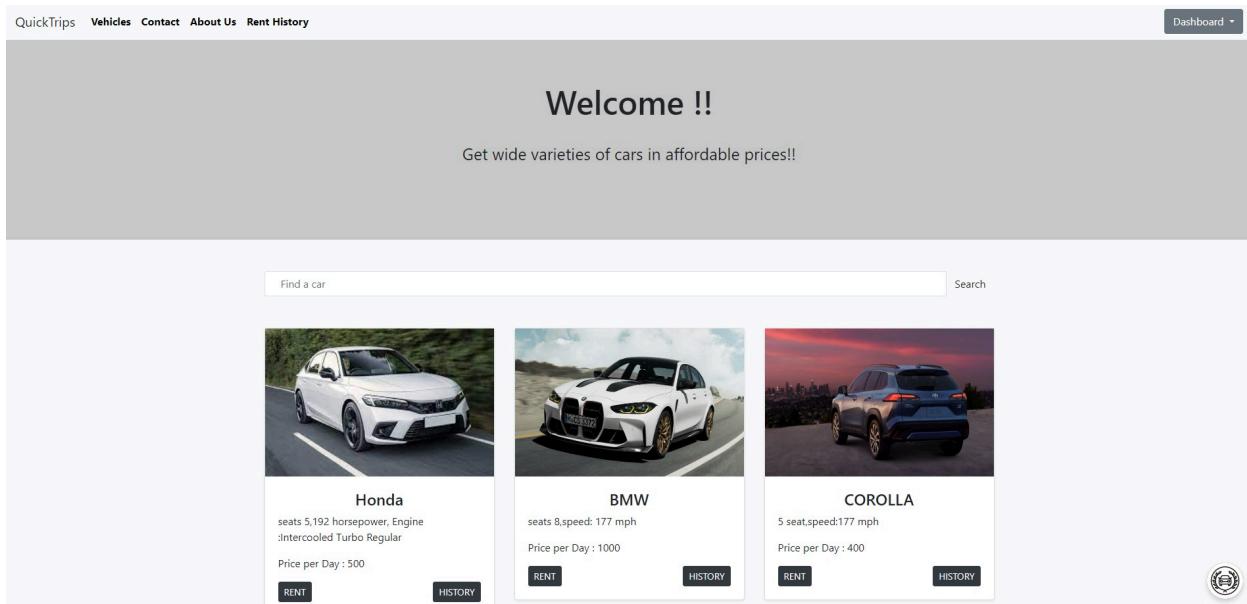


Figure :Car search button

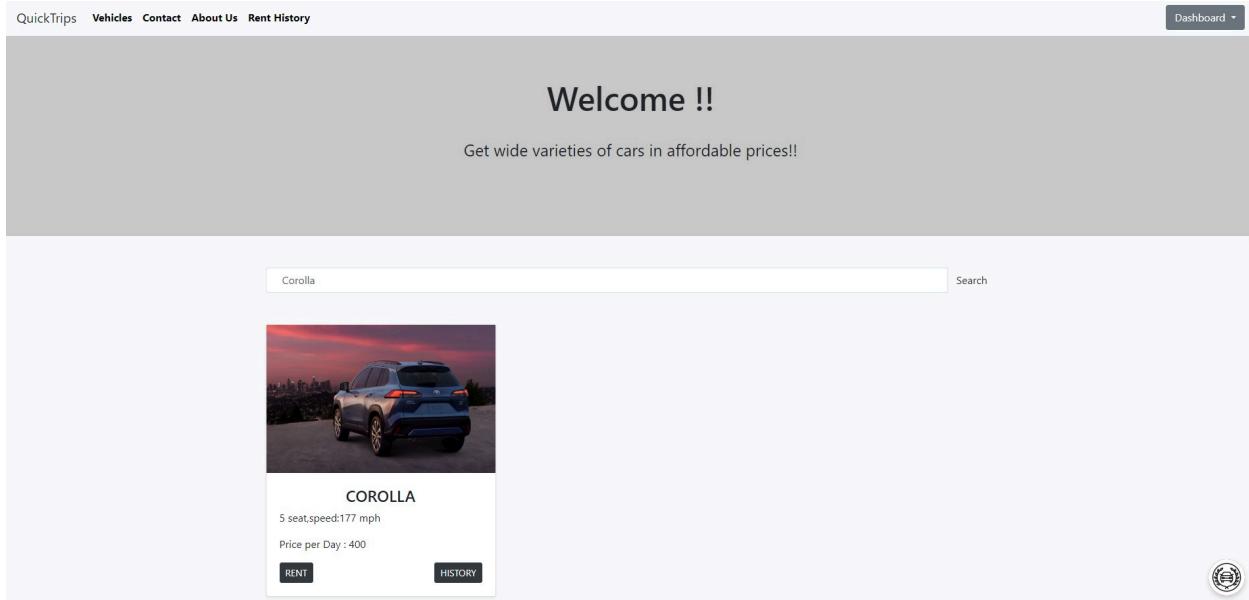


Figure :Car Search Result

### 1.5.2 Rent and history buttons:

- The user can click on the rent button to rent a car
- A car is not always in the best condition and our precious users are important to us which is why we let them see if any of our cars has anything dismissed. If the engine was changed or if the car had ever faced an accident, we keep them in our database. These things are evaluated by the experts and we never let any unqualified car to be enlisted on our website because we want to gain the trust of the users.

**Car History Information**

**Car ID: 2**

**Manufacturing Date:** March 22, 2024

**Damages:** No damages reported

**Latest Safety Rating:** Safe for use

[Back to Vehicles](#)



**Car Image**



Figure :After clicking car history button

### 1.5.3 User reviews:

- After any trip, we prompt the user to leave us with a review about the car they rented.

- These reviews are shown at the footer of our website in this page. These reviews include both good and bad reviews because we don't want to hide any information from the user. They deserve to know everything about the cars.

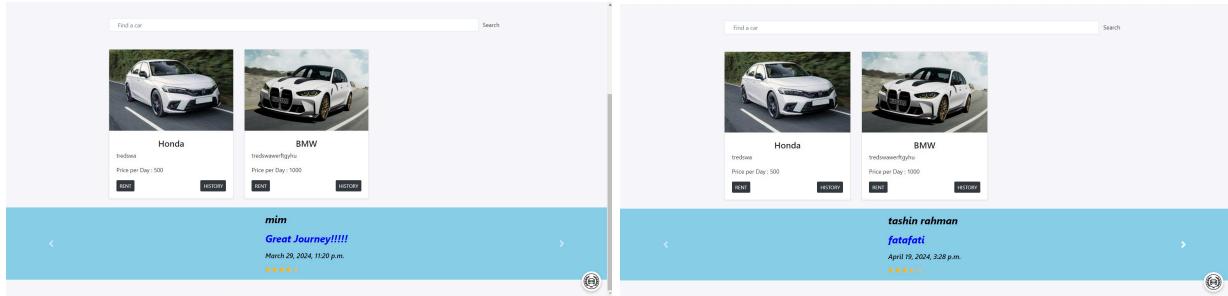


Figure: Customer review slideshow

## 1.6 Rent a car:

The user can click on the Rent button of the chosen car for renting a car in the vehicles page. Process:

- It prompts the user to agree to the terms and conditions of QuickTrips.
  - Authentication: If the user doesn't agree to the terms and conditions, he/she cannot proceed further with the process of renting a car.

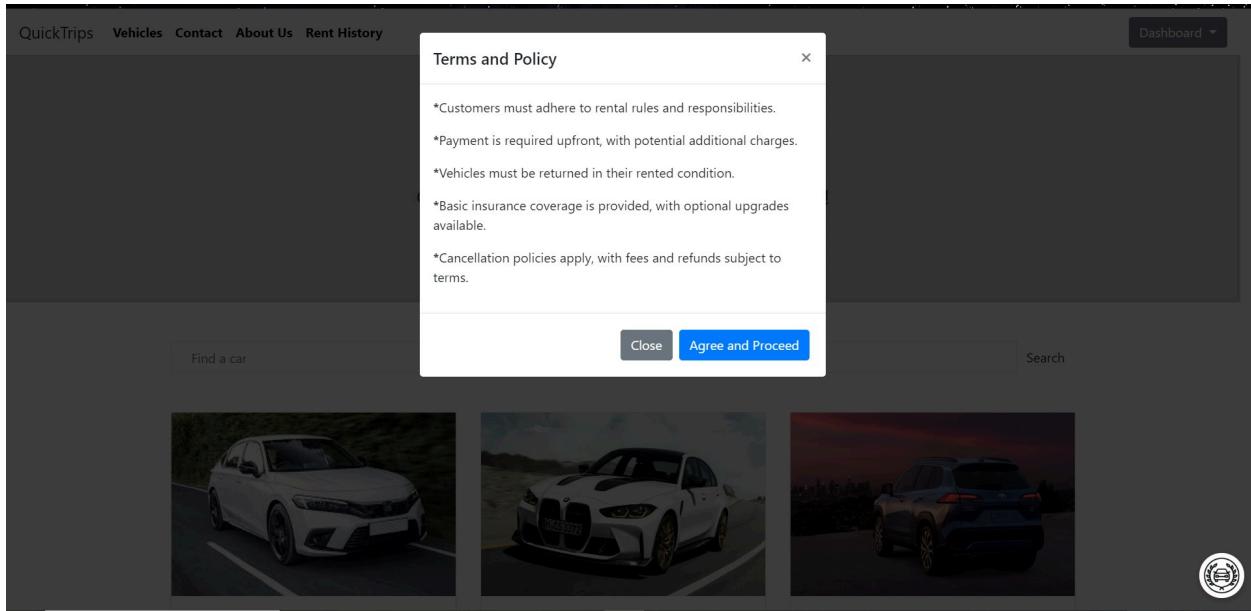


Figure: Terms and policy pop-up

- The user fills in the required blanks related to personal information on the page after agreeing to the terms and conditions.
  - The user writes his/her full name, mailing address, phone number, address
- Then the user fills in the car details that he wants to rent. This shows all the cars in our database. Then he can choose that car along with the color that he wants.
- The user needs to choose the starting date and for how many days the user wants to rent the car. This will be needed for calculating the fare.
- The user can choose the starting location and destination of the car which the user rented.

**Booking Details**

Full Name:

Email:  name@gmail.com Phone Number:

Address:

**Car Details**

Choose...

No. of days:  mm/dd/yyyy:

From Location:  To Location:

**Total Bill**

Total Rent: BDT **0**

**Display Bill** **Confirm Booking**

Figure:Booking Form details

After finishing filling up the form, the user proceeds to the payment phase

### 1.7 Booking:

#### 1.7.1 Displaying the bill:

- The user can click the display bill to get a summary of the amount of money the user needs to pay to our company.
- It gives a small glimpse of the whole payment amount.

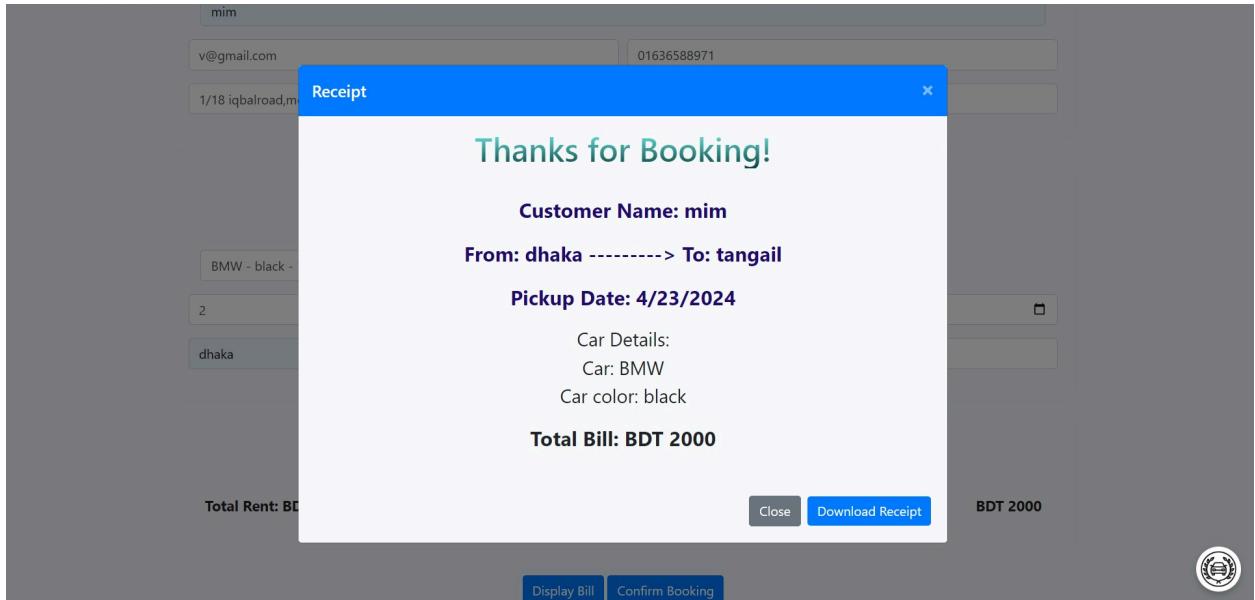


Figure: Displaying the bill

### 1.7.2 Download receipt:

- The user can keep the receipt for offline use by using the download receipt option.
- By clicking the download receipt button, the receipt is downloaded to the downloads file of one's PC in PDF format.

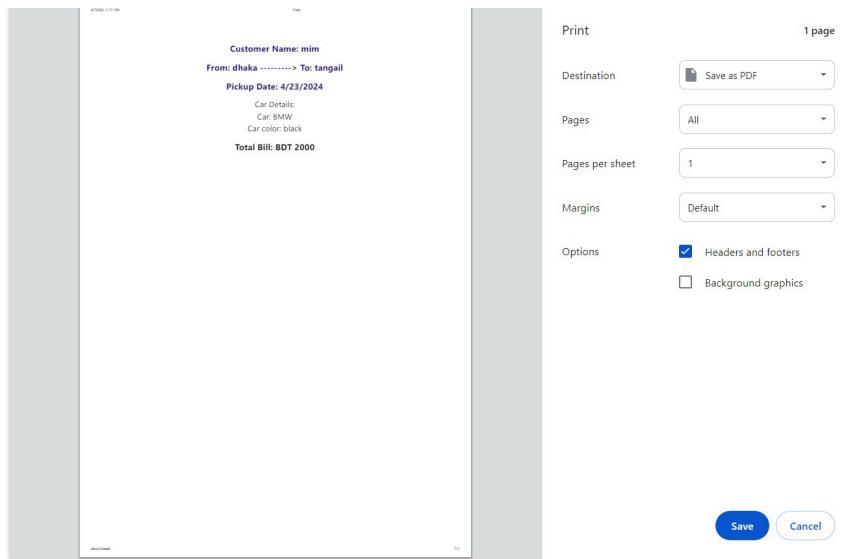


Figure: Download/print the bill

### 1.7.3 The Confirmation of Booking:

- The confirm booking button is then clicked and the booking confirmation is done and all the information related to it is saved in the database of our company. We

can use it for later use. After that, you return to the homepage again so that you can use our other useful features.

## 1.8 Payment and rent history:

### 1.8.1 Previous instances of renting a car:

- The rental history button on the homepage shows all the cars one has booked throughout the time.
- No other user's history will be shown in one's user id.

Rent History											Actions
Date	Car	From Location	To Location	Days Rented	Address	Phone	\$/day	Total	Status		
2024-04-25	BMW	dhaka	tangail	2	1/18,19 iqbalroad,dhaka-1207	01636788222	1000.00	2000.00	Pending	<button>Pay Now</button>	<button>Cancel X</button>

Figure: Rental history showing of each customer

### 1.8.2 Pay now button:

- The user can pay the amount of money needed only after the trip ends.
- After clicking the Pay Now button, the user is taken to the payment gateway of our company.

Select Payment Method



Bkash



VISA

Total Payable Amount

৳. 6000.0

Figure: Payment methods

### 1.8.3 Cancel button:

- Customers can cancel their reservation by clicking the cancel button.

The image shows a 'Rent History' table and a 'Confirm Cancellation' dialog box. The table has columns: Date, Car, From Location, To Location, Days Rented, Address, Phone, \$/day, Total, Status, and Actions. A row in the table shows a rental from 2024-04-25 to tangail for 2 days at 1/18, 19 iqbal road, dhaka-1207. The 'Actions' column contains 'Pay Now' and 'Cancel X' buttons. A modal window titled 'Confirm Cancellation' asks 'Do you really want to cancel your reservation for this order?' with 'No' and 'Yes, cancel my order' buttons.

Figure: Canceling Reservation

### 1.8.4 Payment Gateway:

- The user can pay the required amount of fare either through a Bkash/Visa card
- If the user pays through Bkash:
  - The system checks for a legit 11-digit number
  - It also asks for a transaction ID so that the user has a trusted gateway to pay.
- If the user pays through a Visa Card:
  - The system checks for a legit Card number
  - It also asks for the name on that card to verify the user
  - The user is asked about the Expiry date along with CVV so that the website can confirm if the Card user's account is valid and the card doesn't bounce.

The image shows a 'Payment Page' with the following fields:

- Pay to 01846044512 ("Send Money")
- Phone Number: 08879797979
- Transaction ID: X-HSJGDJ

A blue 'Submit' button is at the bottom.

Figure:Payment Gateway (bkash)

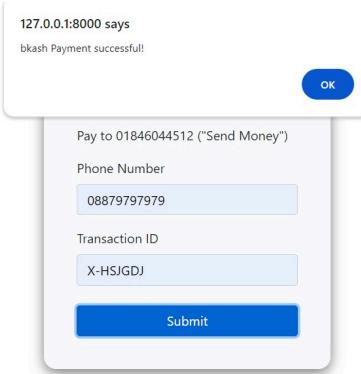


Figure:Payment gateway(card)

#### 1.8.5 Verification:

- The user is restricted to giving any sort of non legitimate card number less than 16 digits

#### 1.9 Review System:

After the journey and the payment, the user is asked to give a review and also for a rating out of 5 stars.

- Feedback box: The user needs to fill in the user's name and a description of the journey one had in the description box.
- Star Rating: The star rating is taken basically for marketing purposes.
- Skip button: According to our research, we analyzed that many users don't want to give a review. For them, we created a skip button as we value their time and not waste it by making it a mandatory step for the next ones.

The user is taken back to the homepage after hitting submit

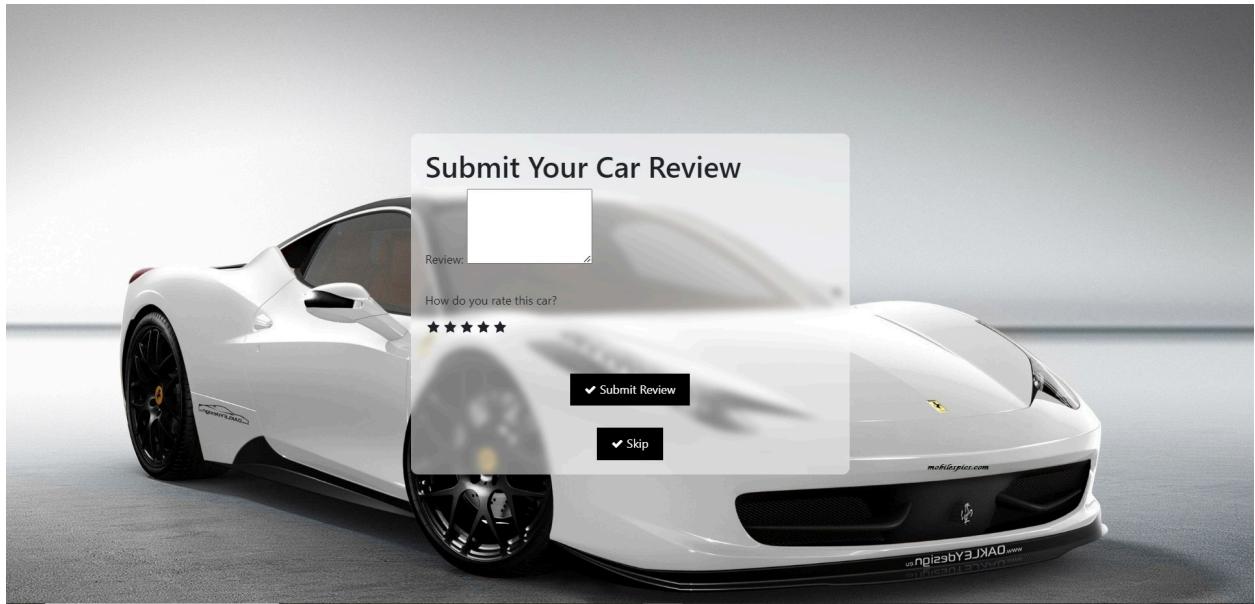


Figure: Review system

### 1.10 Contact Us:

We have created a portal for anyone who wants to get help from our service. They can fill up the form and all the other details. These will be saved in the database through which we can take action regarding the requests.

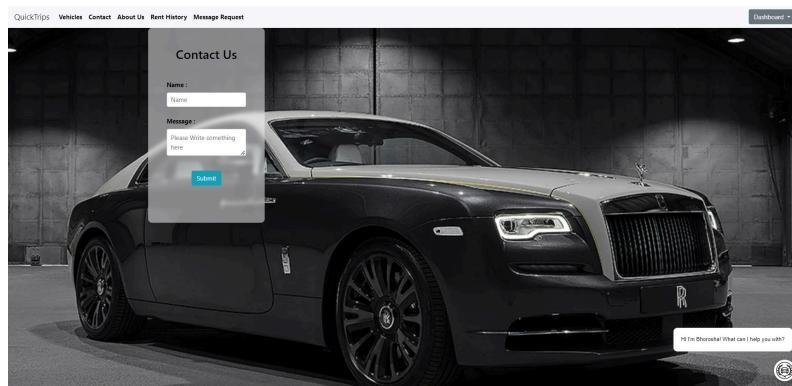


Figure:Contact us page

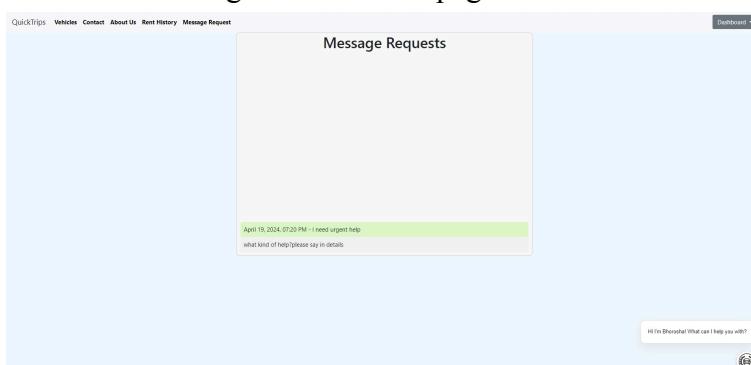


Figure:Admin can direct talk to customer in emergency cases

## 1.11 About Us:

### 1.11.1 Social Media:

QuickTrips has a mission and vision which we wrote in the documentation here along with our Facebook, Instagram, and Twitter page for marketing purposes and having more followers.

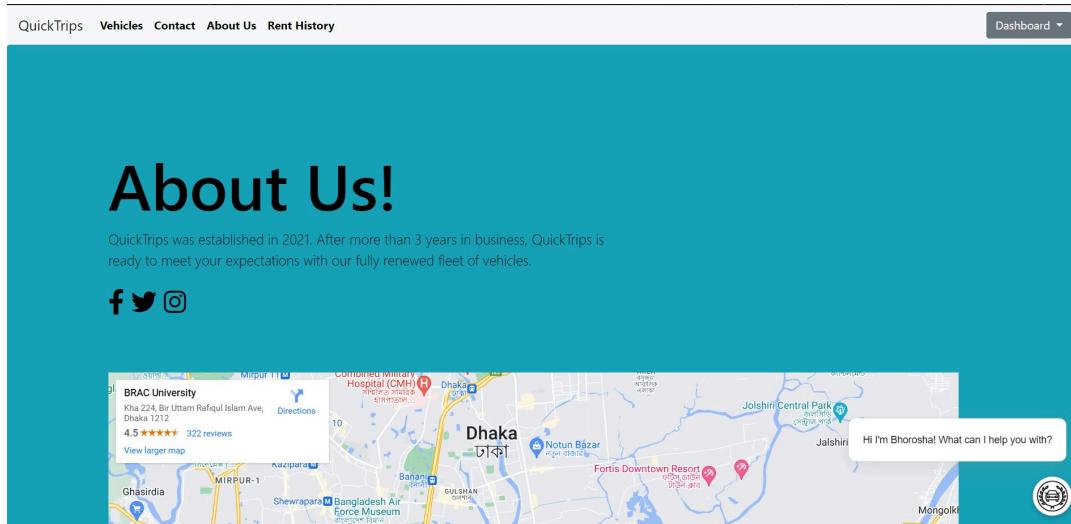


Figure: About us page with real-time map

### 1.11.2 Real-Time Google Maps:

It shows our office and headquarters so that people can easily locate our company.

## 1.12 Dashboard:

### 1.12.1 Profile

- User can see his detailed information here

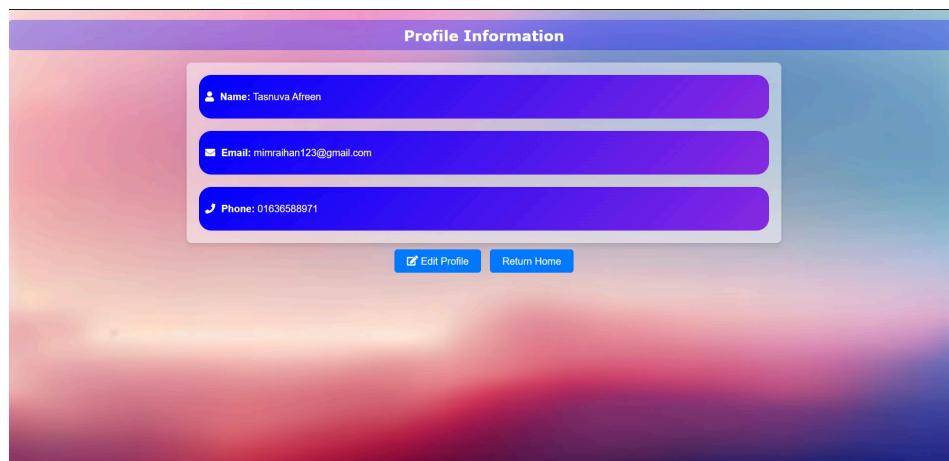


Figure: User detail information

- The user can change any of his personal information if he wants.

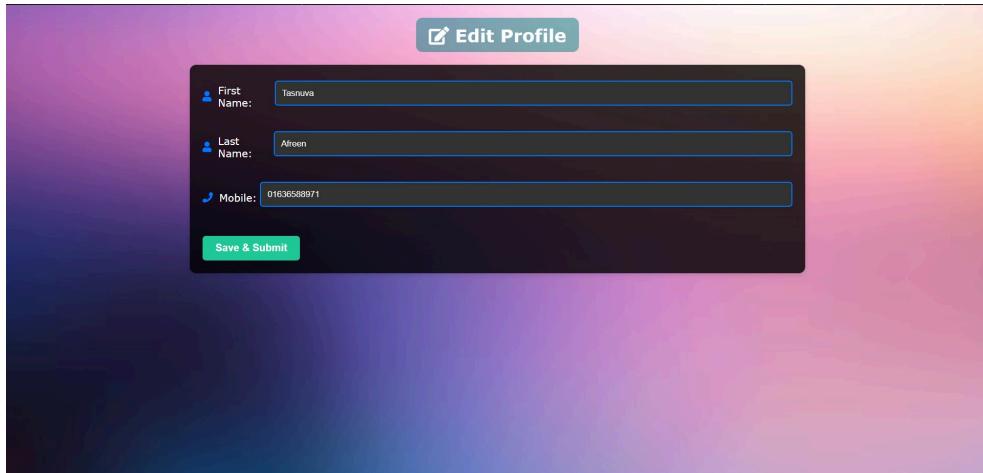


Figure:Profile information editing(including name, mobile number)

### 1.13 Bhorosha:

#### 1.13.1 Description:

- Bhorosha is our personally trained AI model.
  - It can help the user to book a car by explaining in detail about the whole website
  - It can assist our valued customers in suggesting how the cars they want to rent from us be rented.

#### 1.13.2 Very user friendly:

- Bhorosha is very simple to use.
- It doesn't answer anything other than questions related to our company.

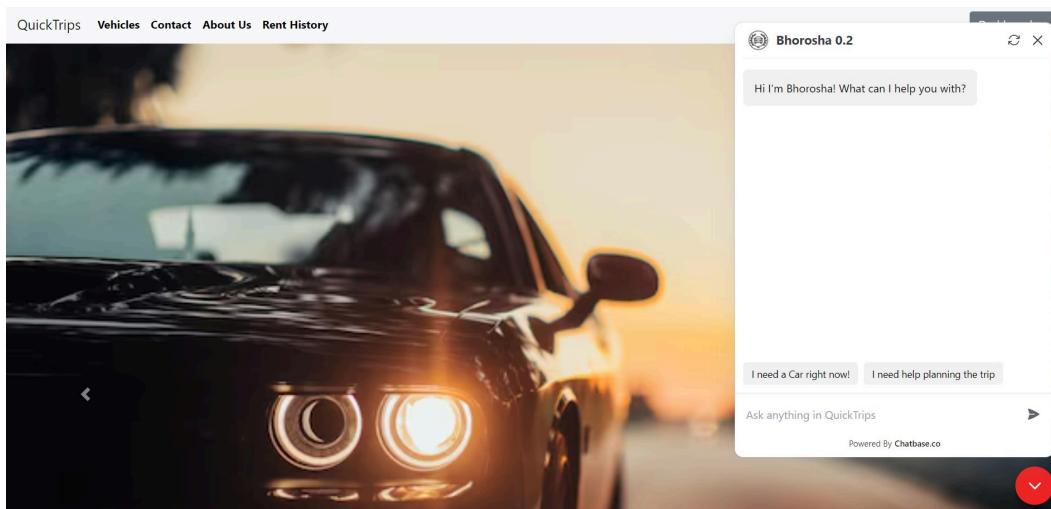


Figure:Personally trained AI model bhorosha

## Chapter 2

## 2.1 Add cars:

- The admin can add cars after evaluating the car and the driver of the car on the website.
- Admin can add pictures and label every car with a category

The screenshot shows the 'Add car' form. It has fields for Car name, Car color (set to Black), Car desc, Price (0), Image (choose file, no file chosen), Manufacturing date (Today), Damages (empty text area), Car status (Available), and Latest safety rating (Safe for use). At the bottom are buttons for Save, Save and add another, and Save and continue editing.

Figure: Adding cars by admin

## 2.2 Contacts:

- Admin can see what all the users sent in the Contact Us form.
- He can email them after seeing the user's mail as the admin has all the access to the website and can see them anytime

The screenshot shows the Django admin 'Contacts' list view and a detailed view of a contact message. The list view shows multiple entries. The detailed view shows a message from 'Techno Neelam' asking for urgent help, with fields for Time (April 19, 2018, 03:00 PM), Email (techno.neelam@gmail.com), Name (Techno Neelam), Message (I need urgent help), and Reply (what kind of help/please say in details). At the bottom are buttons for Save, Save and add another, and Save and continue editing.

Figure: Admin can see all the messages and reply to each of the messages.

## 2.3 Orders:

- The order table shows all the booked cars along with other information related to the Payment.

The screenshot shows the Django admin 'Orders' list view and a detailed view of an order. The list view shows multiple order entries. The detailed view shows an order for 'Nilesh Neelam' with various fields filled in, including User email (nilesh.neelam@gmail.com), Name (Nilesh Neelam), Email (nilesh.neelam@gmail.com), Phone (09898989898), Address (Gangotri, Badli), Selected car ID (2), Car color (Black), Date for rent (2018-04-30), Date (2018-04-30), Set time (01:00), Set location (near Noida), Rent price per day (1000.00), Total rent (2000.00), and Payment status (paid). At the bottom are buttons for Save, Save and add another, and Save and continue editing.

Figure: Admin can see all the order details

## 2.4 Real-time activity:

- The admin can see which user is active right at that moment and also see if anyone's trying to breach our website.

The screenshot displays two side-by-side panels from the Django Admin interface. The left panel shows the 'User Management' section, where a user named 'helloworld2024@gmail.com' is being edited. It includes fields for Username, First name, Last name, Email address, and various permissions like 'Is active', 'Is staff', and 'Is superuser'. The right panel shows the 'User permissions' section, where the user's specific permissions are listed, such as 'Can add log entry', 'Can change log entry', and 'Can delete log entry'. Below these panels is a 'Success' message indicating that the user was successfully saved.

## Backend Development

The Backend implements the following business logic for the Quicktrips website:

1. User Management:
  - a. Users can register, log in, and log out securely.
  - b. Users have the ability to manage their profiles, including updating personal and contact information.
  - c. Password management is integrated to allow users to set and update their passwords securely.
  - d. Session management controls user sessions with features like session expiration and prevention against session hijacking.
2. Car Management:
  - a. The admin has the capability to add new cars, update existing car details, and remove cars from the platform.
  - b. Cars are searchable by brand or other attributes, and availability is updated in real-time based on current bookings.
3. Reservation System:
  - a. Users can search for available cars based on their preferences and reserve cars for rental.
  - b. The reservation process includes authentication steps, terms and conditions agreement, and filling out necessary personal and car details.

- c. Reservation management allows admins to cancel reservations and track rented cars.
- 4. Transaction and Payment Management:
  - a. Transactions can be managed through various payment options like credit/debit card or Bkash.
  - b. Users fill in required payment information, and the system verifies transaction details for security.
  - c. Payment is only required after the rental service is concluded.
- 5. Feedback and Review System:
  - a. After completing a rental, users are prompted to leave feedback and rate their experience.
  - b. Both positive and negative reviews are displayed to maintain transparency and help other customers make informed decisions.
- 6. Emergency and Support Services:
  - a. A dedicated emergency contact option is available for users to reach out in case of urgent issues.
  - b. A user-friendly AI chatbot ("Bhorosha") assists users by answering queries and helping with the booking process.
- 7. Rental History and Documentation:
  - a. Users can view their complete rental history, including past bookings and payments.
  - b. Documentation like receipts can be downloaded for record-keeping.
- 8. Technology and Security:
  - a. The system uses Django framework, integrating front-end technologies like HTML, CSS, JavaScript, and back-end technologies including Python.
  - b. Email verification and secure password reset functionalities enhance the security of the platform.

```

10     urlpatterns = [
11         path('', views.index, name = 'home'),
12         path('login', views.login, name = 'login'),
13         path('logout', views.handlelogout, name = 'logout'),
14         path('signup', views.signup, name = 'signup'),
15         path('loggedin', views.loggedin, name = 'loggedin'),
16         path('profile', views.profile, name='profile'),
17         path('edit_profile', views.edit_profile, name='edit_profile'),
18         path('editted', views.editted, name='editted'),
19
20         path('contact/',views.contact,name = 'contact'),
21
22         path('vehicles.html', views.vehicles, name='vehicles'),
23
24         path('car_history<int:car_id>', views.car_history, name='car_history'),
25
26         path("bill",views.bill,name = "bill"),
27         path("order",views.order,name = "order"),
28         path("about.html",views.about_us, name="about_us"),
29         path("rent_history.html",views.rent_history, name='rent_history'),
30         path('payment/<int:order_id>/<float:total_rent>/', views.payment, name='payment'),
31         path('review.html',views.review,name = 'review'),
32         path('cancel_order/<int:order_id>', views.cancel_order, name='cancel_order'),
33
34         path("message_request.html",views.message_request, name='message_request'),
35     ]

```

Figure : All urls

```

class Signup(models.Model):
    email = models.CharField(max_length=50)
    f_name = models.CharField(max_length=30)
    l_name = models.CharField(max_length=30)
    mobile = models.CharField(max_length=20)

class Contact(models.Model):
    time = models.CharField(max_length=50,default="")
    email = models.CharField(max_length=150, default="")
    name = models.CharField(max_length=150, default="")
    message = models.TextField(default="")
    reply = models.TextField(default="")

    def __str__(self):
        return f"{self.name} - {self.reply}"

class Car(models.Model):
    car_id = models.AutoField(primary_key=True)
    car_name = models.CharField(max_length=30, default="")
    car_color = models.CharField(max_length=30, choices=[('red', 'Red'), ('blue', 'Blue'), ('green', 'Green'), ('black', 'Black'), ('white', 'White')])
    car_desc = models.CharField(max_length=300, default="")
    price = models.IntegerField(default=0)
    image = models.ImageField(upload_to="car/images", default="")
    manufacturing_date = models.DateField() # New attribute for manufacturing date
    damages = models.TextField(blank=True, null=True) # New attribute for damages, optional

    CAR_STATUS_CHOICES = [
        ('available', 'Available'),

```

Figure : Classes defining models

```

def rent_history(request):
    # Assuming 'email' is the key used to store the user's email in the session
    user_email = request.session.get('email', None)
    if user_email:
        # Fetching orders for the logged-in user
        user_orders = Order.objects.filter(user_email=user_email)
    else:
        user_orders = []

    # Passing orders to the rent_history template
    return render(request, 'rent_history.html', {'user_orders': user_orders})

from django.views.decorators.csrf import csrf_exempt
from django.http import JsonResponse

@csrf_exempt
def payment(request, order_id, total_rent):
    order = get_object_or_404(Order, order_id=order_id)
    if request.method == 'POST':
        order.payment_status = 'paid'
        order.save()
        car = get_object_or_404(Car, car_id=order.selected_car_id)
        car.car_status = 'available'
        car.save()
        # Optionally, you can return a JSON response or redirect to a success page
        return JsonResponse({'success': True})
    return render(request, 'payment.html', {'order': order, 'total_rent': total_rent})

```

```

from django.contrib import admin
from home.models import Signup

```

```

# Register your models here.
admin.site.register(Signup)
admin.site.register(Contact)
admin.site.register(Car)
admin.site.register(Order)
admin.site.register(Review)

```

```

# Email configuration
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_USE_TLS = True
EMAIL_HOST_USER = 'quicktrips178@gmail.com'
EMAIL_HOST_PASSWORD = 'tmfdfjjiugypkecf'

```

Figure : Some functions for backend operations

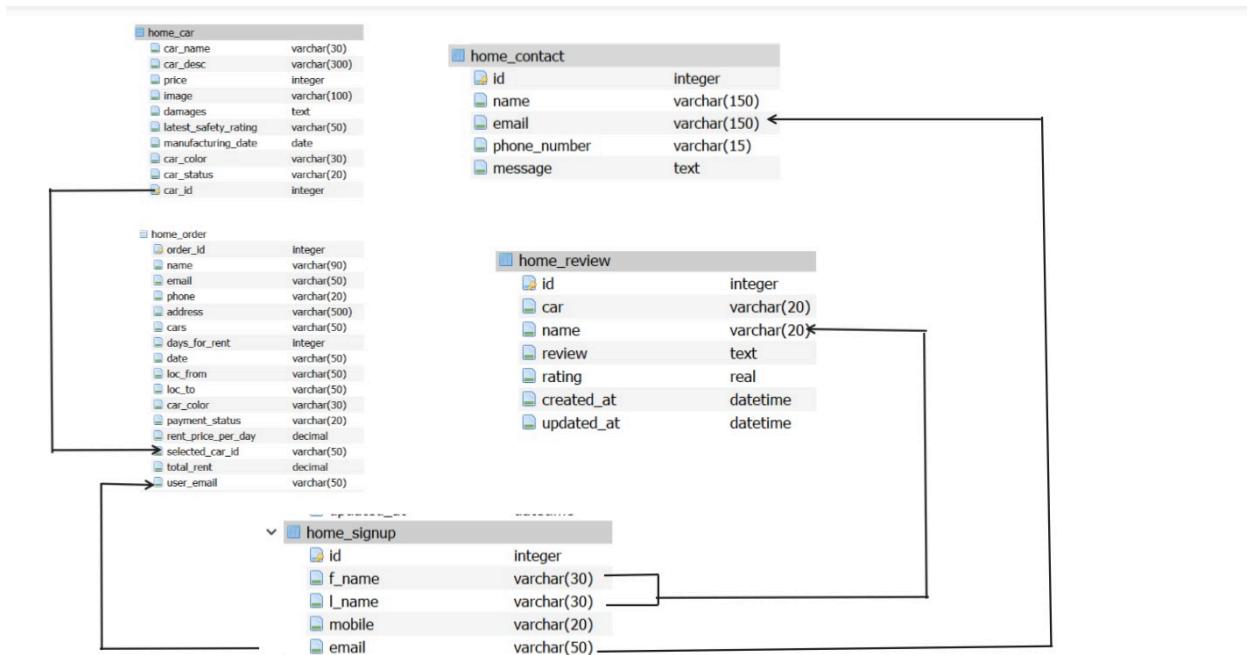


Figure:Database Schema

## **Technology (Framework, Languages)**

Framework : Django

Frontend: Html,CSS, JavaScript

Backend: Python,Javascript

Database:sqlite3

## **Github Repository**

Link: <https://github.com/Yasir2521/Car-Rental-System.git>

## Individual Contribution

ID	Name	Contribution
23341112	Munawar Mahtab Ansary	<ul style="list-style-type: none"> <li>● Vehicle management</li> <li>● Vehicle search and filter</li> <li>● Email verification</li> <li>● Forget Password</li> <li>● Homepage</li> <li>● Display Available Vehicles</li> </ul>
23341117	Yasir Arafat Saimon	<ul style="list-style-type: none"> <li>● Login system</li> <li>● Signup system</li> <li>● Billing system</li> <li>● User management system</li> <li>● Edit user profile</li> <li>● Rent History page</li> </ul>
21201124	Mim Raihan	<ul style="list-style-type: none"> <li>● About Us page</li> <li>● Add Review system</li> <li>● Display Review system</li> <li>● Navigation bar</li> <li>● Emergency Contact system</li> <li>● Dashboard</li> </ul>
21241012	Nibir Neelim	<ul style="list-style-type: none"> <li>● Download bill</li> <li>● View profile system</li> <li>● Transaction management system</li> <li>● 3D car model</li> <li>● Car history page</li> <li>● AI chatbot</li> </ul>