UNIVERSITY OF ASIA PACIFIC

Department of Computer Science & Engineering

Course code: CSE 302

Course title: Object Oriented Programming II and Web Programming Lab

Hire Your Wheels

Motive: Hire the car & feel like your own.

Submitted to

Nuzhat Tabassum Progga

Lecturer, CSE, UAP

Team Members:

M Sakib Rahman : 22201240

H M Tahsin Sheikh : 22201243

Tasfia Tabassum : 22201248

1 Introduction

1.1 Objective

This project aims to develop a user-friendly web and mobile application platform that allows users to easily rent cars from a pool of available cars. The ”Hire Your Wheels” application will provide a seamless experience for finding, booking, and managing car rentals, empowering users with the flexibility of having a car without the commitments of ownership. Our main goal is to create an online platform where users can effortlessly access rental cars in a friendly and efficient environment. This application intends to streamline the car rental process, reducing the complexities associated with traditional methods.

1.2 Motivation

In today’s fast-paced world, the need for convenient and flexible transportation is ever-increasing. Owning a car comes with various responsibilities and costs, which may not be ideal for everyone. ”Hire Your Wheels” is motivated by the desire to provide an alternative solution that offers the freedom of having a car when needed, without the long-term financial and logistical burdens. We aim to create a platform where users can ”hire the car feel like their own,” emphasizing a sense of convenience, reliability, and personalized experience.

1.3 Critical Challenges

Developing a robust and trustworthy car-sharing platform comes with its challenges. Ensuring the safety and security of both users and cars is paramount. Building trust among users regarding car condition and availability will be crucial. Implementing an efficient system for car tracking, booking management, and payment processing will also be significant hurdles to overcome. Furthermore,

Addressing issues related to insurance, liability, and user verification will be critical for the success of ”Hire Your Wheels.”

1.4 Social and Environmental Impact

1. Convenience and Accessibility: Provides easy access to transportation for individuals who may not own a car or need an additional car temporarily.
2. Cost-Effectiveness: Can be more economical than owning a car, especially for occasional use.
3. Reduced Traffic Congestion and Emissions: Encouraging car sharing can potentially lead to fewer privately owned cars on the road, thus reducing traffic and environmental impact.
4. Efficient Resource Utilization: Maximizes the use of available cars, reducing the number of idle cars.
5. Community Building: Can foster a sense of community by enabling people to share resources.

2. Project Description

2.1 ”Hire Your Wheels” is a car-sharing application designed to connect users who need to rent a car with a fleet of available cars. The platform will offer a user-friendly interface for browsing available cars, checking their availability, making bookings, and managing rentals. Users will be able to search for cars based on location, type, and availability. The application will also handle payment processing, booking confirmations, and communication between users and the platform administrators. The core objective is to provide a seamless and trustworthy car-sharing experience, making users feel like they have their own car at their disposal.

2.2 Proposed Functionalities/Features

1. User Management:

- User registration and login.
- Profile creation and management.
- Secure payment integration.
- Booking history and management.

2. car Management:

- Listing available cars with details (model, make, location, features, pricing).
- Real-time availability status.

3. Booking and Rental Management:

- Easy car search based on various criteria.
- Booking and cancellation functionality.
- Rental agreement and terms.
- Check-in and check-out process.

4. Communication and Support:

- In-app messaging or contact options for support.
- Notifications for booking updates and reminders.

5. Reviews and Ratings:

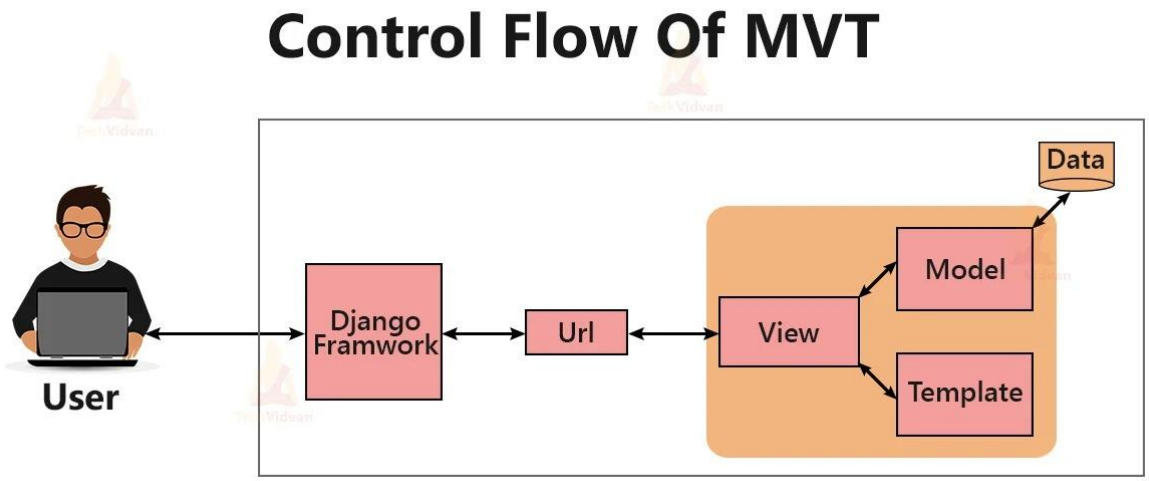
- System for users to review and rate cars and the service.

These features will collectively contribute to a comprehensive and user-centric car-sharing platform that aligns with the ”Hire the car feel like your own” motive.

3 Project Design

3.1 Architecture of the Proposed System

The ”Hire Your Wheels” application will likely follow a multi-tiered architecture, potentially involving a front-end (web and mobile applications), a back-end (API and application logic), and a database for storing data.



3.1.1 Models (Conceptual for Database):

The models.py file for the “Hire Your Wheels” based on the functionalities, we can envision the following key entities (models) for the database:

- User: Stores user information (ID, name, contact details, payment information, etc.).
- Cartype: Stores car typr information(ID, type, details)
- Car: Stores car details (ID, model, make, year, location, availability status, pricing, features, etc.).
- Reservation: Stores booking information (booking ID, user ID, car ID, start date/time, end
- date/time, booking status, total cost, etc.).
- Payment: Stores payment transaction details (payment ID, booking ID, user ID, amount, payment date, payment method, status, etc.).
- Notification: Stores user notification for cars reservation (notification ID, user ID, car ID, rating, comment, date, etc.).

3.1.2 **Views/Controllers (Conceptual for Application Logic):** The views.py file for the “Hire Your Wheels” based on the functionalities, we can envision the following key entities (models) for the database

The views.py file in the "Hire Your Wheels" project orchestrates the backend logic, linking user interactions with the application's core car rental functionalities. It encompasses a range of features, from user authentication and profile management to car browsing, booking management, and payment processing. The code likely integrates Django's authentication system for secure user access, employs form validation to ensure data integrity during user input and booking processes, and utilizes messages to provide feedback to the users.

User registration and login functionalities would involve form validation to ensure accurate user account creation and secure authentication. User profiles are managed through features allowing users to update their personal information, contact details, and payment methods.

car browsing and search are key aspects, with views enabling users to search for available cars based on criteria such as location, car type, make, model, and price. These views would handle filtering and displaying car listings to the user.

Booking management is a central functionality, allowing users to select cars, specify rental dates and times, and create bookings. The code would manage car availability, preventing double-bookings and

ensuring a smooth booking process. Views for viewing, modifying, and potentially cancelling bookings would also be included.

Payment processing would be handled through views that interact with payment gateways, securely processing user payments for rentals. These views would likely manage transaction status and provide confirmation to the user. Additionally, location services would be integrated, allowing users to view car locations on a map and potentially select pick-up and drop-off points. The views would handle the retrieval and display of location data.

The views in "Hire Your Wheels" are designed to interact effectively with the application's models (representing data like users, cars, bookings) and templates (for rendering the user interface), forming a crucial component of the Django MVT (Model-View-Template) architecture. This structure ensures a modular, maintainable, and scalable codebase, providing a comprehensive car rental management system for users.

The application logic will handle user interactions and data manipulation, including:

- User authentication and authorization.
- Handling user profile management.
- Displaying available cars based on search criteria.
- Processing booking requests and cancellations.
- Managing payment transactions.
- Handling car listings and updates (for administrators).
- Processing and displaying reviews and ratings.

3.1.3 Templates/User Interface (Conceptual for Presentation):

Templates:

In the "Hire Your Wheels" project, templates play a crucial role in defining the user interface and presentation of information to users interacting with the car rental platform. Django templates are used to structure the web pages, employing template inheritance to promote code reusability and maintainability. The base template, typically named "base.html," establishes the overall framework of the site, including common elements such as the header, footer, and navigation menus. This base template defines blocks that child templates can override to customize specific content areas for different pages.

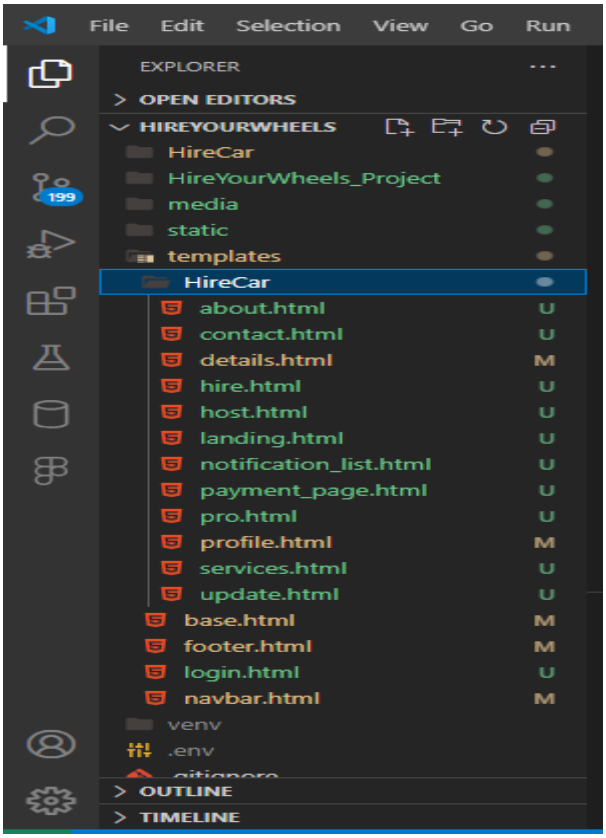
1. {% extends %}:

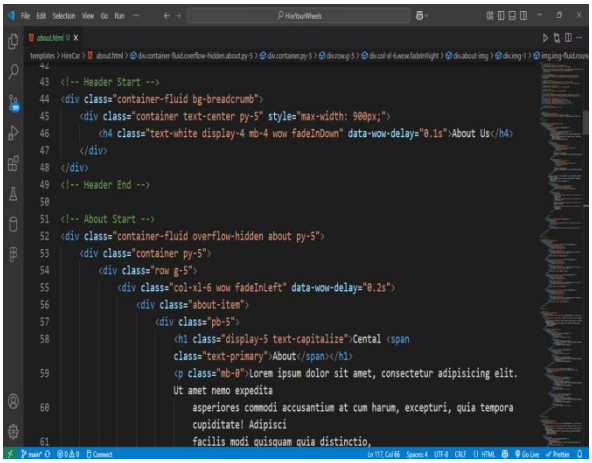
The {% extends %} tag is used by a child template to inherit the structure and content of the base template. This mechanism establishes a hierarchy, where the child template builds upon the layout defined in the base template, ensuring a consistent design and reducing code duplication across the website.

2. {% block %}:

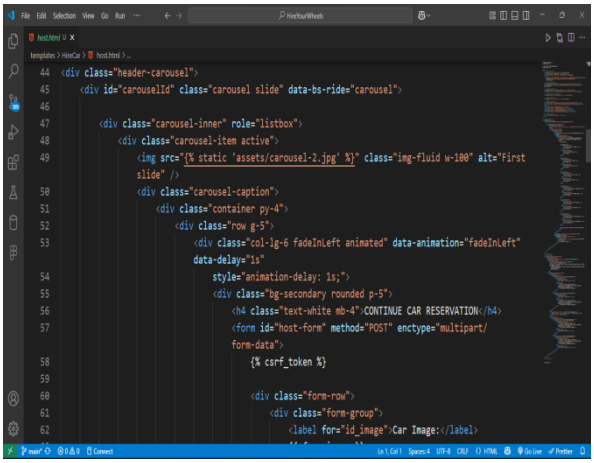
The {% block %} tag defines a content block within a template. In the base template, these {% block %} tags act as placeholders for content that can be customized or overridden by child templates. Child templates use the same block names to insert specific content into those predefined areas, allowing for page-specific customization while maintaining the overall site structure and design.

Some HTML page are shown here.

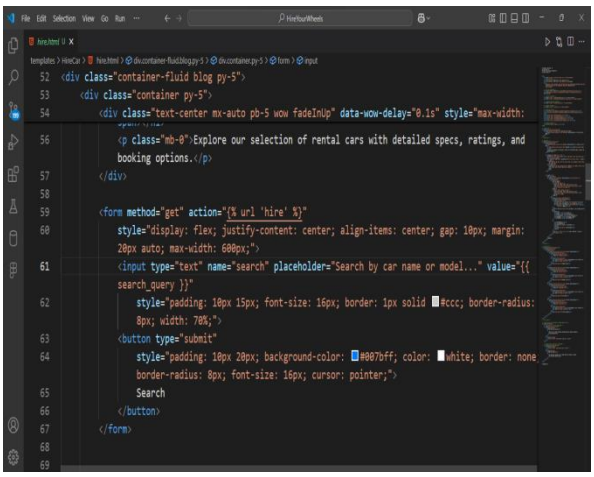




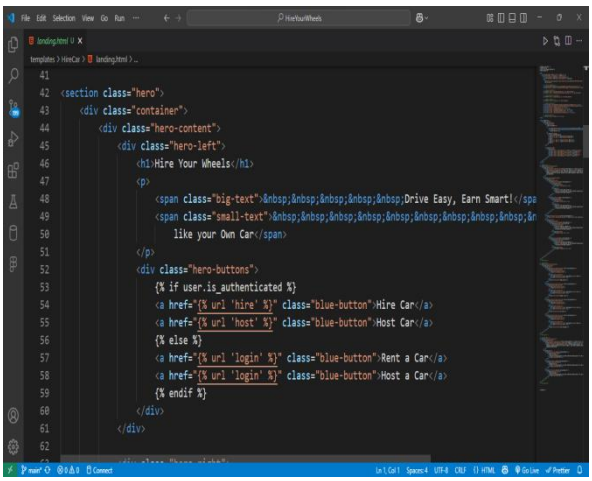
```
<!-- Header Start -->
<div class="container-fluid bg-breadcrumb">
    <div class="container text-center py-5" style="max-width: 900px;">
        <h4 class="text-white display-4 mb-4 wow fadeInDown" data-wow-delay="0.1s">About Us</h4>
    </div>
</div>
<!-- Header End -->

<!-- About Start -->
<div class="container-fluid overflow-hidden about py-5">
    <div class="container py-5">
        <div class="row g-5">
            <div class="col-xl-6 wow fadeInLeft" data-wow-delay="0.2s">
                <div class="about-item">
                    <div class="pb-5">
                        <h1 class="display-5 text-capitalize">Cental <span
                        class="text-primary">About</span></h1>
                        <p class="mb-0">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
                        Ut amet nemo expedita
                            asperiores commodi accusantium at cum harum, excepturi, quia tempora
                            cupiditate! Adipisci
                            facilis modi quisquam quia distinctio,
```



```
<div class="header-carousel">
    <div id="carouselId" class="carousel slide" data-bs-ride="carousel">

        <div class="carousel-inner" role="listbox">
            <div class="carousel-item active">
                <img src="{% static 'assets/carousel-2.jpg' %}" class="img-fluid w-100" alt="First
                slide" />
                <div class="carousel-caption">
                    <div class="container py-4">
                        <div class="row g-5">
                            <div class="col-lg-6 fadeInLeft animated" data-animation="fadeInLeft"
                            data-delay="1s"
                                style="animation-delay: 1s;">
                                <div class="bg-secondary rounded p-5">
                                    <h4 class="text-white mb-4">CONTINUE CAR RESERVATION</h4>
                                    <form id="host-form" method="POST" enctype="multipart/
                                    form-data">
                                        {% csrf_token %}

                                        <div class="form-row">
                                            <div class="form-group">
                                                <label for="id_image">Car Image:</label>
```



```
<div class="container-fluid blog py-5">
    <div class="container py-5">
        <div class="text-center mx-auto pb-5 wow fadeInUp" data-wow-delay="0.1s" style="max-width:
        <p class="mb-0">Explore our selection of rental cars with detailed specs, ratings, and
        booking options.</p>
    </div>

    <form method="get" action="{% url 'hire' %}"
        style="display: flex; justify-content: center; align-items: center; gap: 10px; margin:
        20px auto; max-width: 600px;">
        <input type="text" name="search" placeholder="Search by car name or model..." value="{{
        search_query }}"
            style="padding: 10px 15px; font-size: 16px; border: 1px solid #ccc; border-radius:
            8px; width: 70%;">
        <button type="submit"
            style="padding: 10px 20px; background-color: #007bff; color: white; border: none
            border-radius: 8px; font-size: 16px; cursor: pointer;">
            Search
        </button>
    </form>
```



```
<section class="hero">
    <div class="container">
        <div class="hero-content">
            <div class="hero-left">
                <h1>Hire Your Wheels</h1>
                <p>
                    <span class="big-text">     Drive Easy, Earn Smart!</spa
                    <span class="small-text">         &r
                        like your Own Car</span>
                </p>
                <div class="hero-buttons">
                    {% if user.is_authenticated %}
                    <a href="{% url 'hire' %}" class="blue-button">Hire Car</a>
                    <a href="{% url 'host' %}" class="blue-button">Host Car</a>
                    {% else %}
                    <a href="{% url 'login' %}" class="blue-button">Rent a Car</a>
                    <a href="{% url 'login' %}" class="blue-button">Host a Car</a>
                    {% endif %}
                </div>
            </div>
```

Here we show host.html ; hire.html; profile.html and home.html

manage.py:

The manage.py file is a command-line utility that is fundamental to every Django project, including "Hire Your Wheels." [cite: Demo_doc.pdf - 1, 2] It provides a set of commands that developers can use to perform various administrative tasks related to the project. These tasks include running the development server for testing the application locally, creating and applying database migrations to manage the application's data models, and creating new Django apps within the project structure. [cite: Demo_doc.pdf - 1, 2] The manage.py file serves as the central entry point for executing many Django-specific management commands. Common commands that would be used in the "Hire Your Wheels" project include run server to start the development server, make migrations to generate database migration files based on changes in the models, and migrate to apply these migrations to the database.

settings.py:

The settings.py file is a core configuration file within the "Hire Your Wheels" Django project. [cite: Demo_doc.pdf - 1, 2] It contains crucial settings that govern the behaviour of the application. Key configurations within this file would include database settings, specifying how the application connects to and interacts with the database storing user information, car details, booking records, and payment data. Middleware configurations, which define the sequence of request and response processing, would also be set here. Template settings, indicating how Django finds and renders the HTML templates for the user interface, are another important aspect configured in settings.py. Additionally, time zone settings, security configurations such as the secret key, and potentially settings for integration with third-party services like payment gateways or mapping APIs would be found in this file. The INSTALLED_APPS setting would list all the applications that are part of the "Hire Your Wheels" project. This file holds sensitive information and is critical for the proper functioning and security of the application.

admin.py:

The admin.py file is used to register the application's models with Django's built-in admin interface. In the **"Hire Your Wheels"** project, models like CarType, Car, CustomUser, Reservation, Profile, and Notification are registered using admin.site.register([...]). This allows project administrators to efficiently manage car types, car listings, user accounts, reservations, user profiles, and system notifications through a web-based admin dashboard. The Django admin site simplifies the process of adding, updating, and deleting records in these models.

forms.py:

The forms.py file defines Django forms that handle user input, validation, and form rendering for various features in the **"Hire Your Wheels"** platform. It includes both ModelForm classes (linked to models) and standard Form classes for specific use cases.

Key Forms:

- CarForm
  A model form based on the Car model, allowing hosts to add or edit car listings. It includes fields

such as image, make, model, year, car type, features, daily rate, and availability. A Textarea widget is used for the features field.

- ReservationForm
  A model form used to create a booking/reservation. It allows users to select pickup and dropoff dates using DateInput widgets styled with custom classes.
- LoginForm
  A custom login form extending Django's built-in AuthenticationForm, using placeholder-styled fields for username and password.
- SignupForm
  A custom form for user registration. It includes fields for full name, username, phone, email, address, and role (host or renter). The form validates that the password and confirm password match and securely hashes the password before saving the user.
- ProfileUpdateForm
  A model form tied to the Profile model, allowing users to update their profile picture and bio.

urls.py:

Admin Section

path('admin/', admin.site.urls): Links to the Django admin site for managing models such as Car, Reservation, and User.

Landing Page and Information Pages

path('', Hire_views.landing, name='landing'): The homepage of the website.

path('contact', Hire_views.contact, name='contact'): A page to show contact information.

path('about', Hire_views.about, name='about'): A page with details about the website and company.

path('services', Hire_views.services, name='services'): A page that outlines the services provided by the platform.

User Authentication

path('login/', Hire_views.login_signup_view, name='login'): A view that handles both login and signup functionality.

path('logout/', LogoutView.as_view(next_page='landing'), name='logout'): Log the user out and redirect to the landing page.

Host and Car Rental Features

path('host/', Hire_views.host, name='host'): A page where users can host their cars for rent.

path('hire/', Hire_views.hire, name='hire'): A page for users to browse available cars for hire.

path('profile/', Hire_views.profile, name='profile'): The profile page where users can view and edit their profile.

path('profile/upload', Hire_views.profile_upload, name='profile_upload'): A page for uploading a profile picture or other profile-related media.

path('details/<uuid:id>/', Hire_views.details, name='details'): A page displaying details of a specific car, identified by a UUID.

Car Management

path('update/<uuid:id>/', Hire_views.update_car, name='update_car'): A view to update the details of a car using its unique UUID.

path('delete/<uuid:id>/', Hire_views.delete_car, name='delete_car'): A view to delete a car using its unique UUID.

Reservation Management

path('update_reservation/<uuid:id>/', Hire_views.update_reservation, name='update_reservation'): A view to update reservation details.

path('delete_reservation/<uuid:id>/', Hire_views.delete_reservation, name='delete_reservation'): A view to delete a reservation.

Notification Management

path('notifications/', Hire_views.notification_list, name='notification_list'): A page showing the list of notifications for the user.

path('notifications/mark_as_read/<uuid:notification_id>/', Hire_views.mark_as_read, name='mark_as_read'): A view to mark a specific notification as read.

path('notifications/mark_all_as_read/', Hire_views.mark_all_as_read, name='mark_all_as_read'): A view to mark all notifications as read.

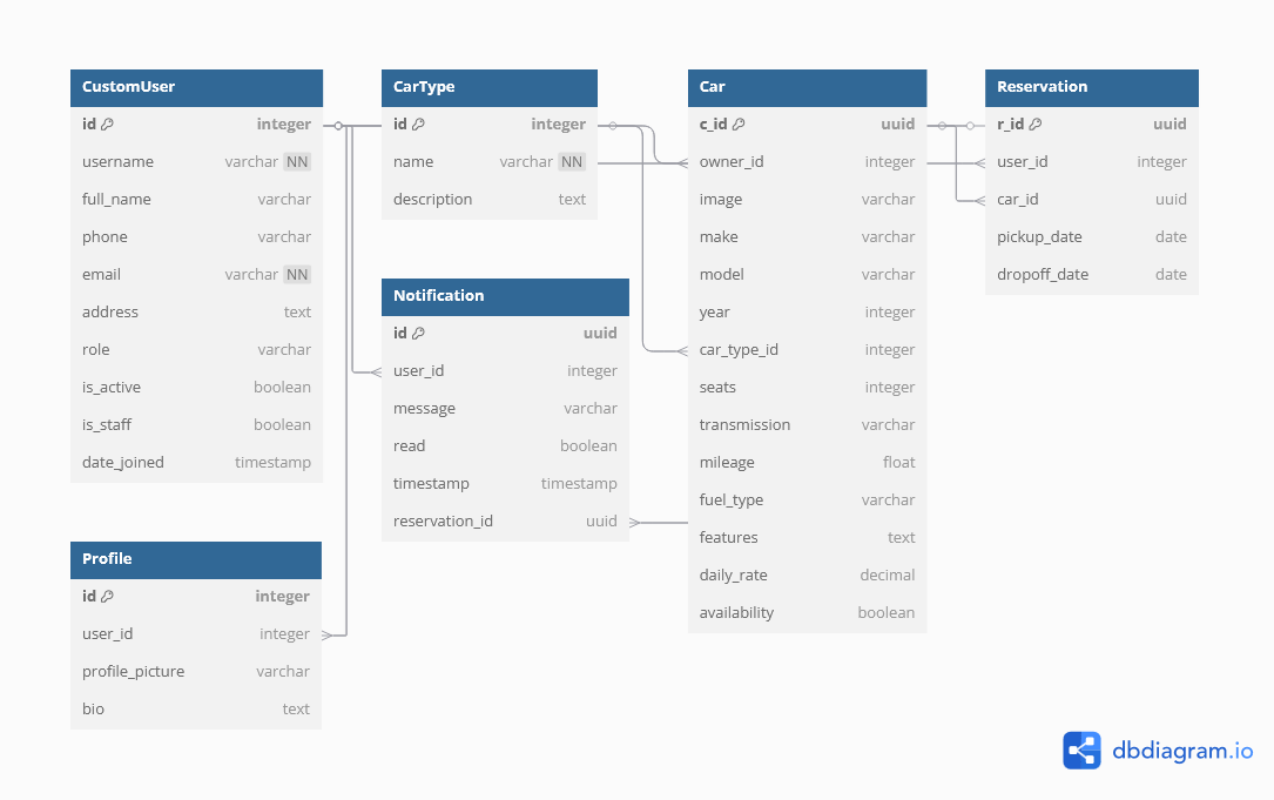
Payment Process

path('payment/', Hire_views.payment_page, name='payment_page'): A page where users can complete their payment for booking a car.

Static and Media Files:

+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT): This ensures that media files (e.g., car images, profile pictures) are served properly during development.

3.2 ER Diagram



Description:

The Entity-Relationship (ER) diagram for this Python project models the relationships and structure of the database for a car-sharing application. It includes entities such as User, car, Booking, Payment, Location, car Type, and Rating, each representing a distinct aspect of the application. Relationships are established through various types of associations, such as one-to-one, one-to-many, and many-to-many, linking users, cars, bookings, payments, and locations. This ER diagram captures the essential components and their connections, providing a blueprint for the underlying database design in the Python project. The entities and their relationships are given below:

1. **User:**
    - Attributes: role_choice, user_id, name, email, phone, , address_info
    - Dependencies: None
2. **Car:**
    - Attributes: car_id, model, make, year, reg_no, availability, price_per_day, location_id, type_id
    - Dependencies: None
3. **Reservation:**
    - Attributes: user, message, read, timestamp, reservation
    - Dependencies:
      User (ForeignKey)
      reservation(ForeignKey)
4. **Notification:**

- Attributes: user_id, reservation_id, amount, payment_date, payment_method, status
- Dependencies:
  - reservation(Foreign Key)

5 . **CarType:**

- Attributes: type_id, type_name, description
- Dependencies: None

6 . **Profile:**

- Attributes: user, car, profile_picture, bio
- Dependencies:
  - User (ForeignKey)
  - Car (ForeignKey)

4. Project Implementation

4.1 Tools

We utilized a range of tools, languages, frameworks, and other resources to develop this project and enhance our experience while also maintaining efficiency. The tools are:

VS Code:

In the context of our project, "Hire Your Wheels," an online platform facilitating car rentals and related services, Visual Studio Code (VS Code) emerged as our preferred development environment. Leveraging VS Code's lightweight design and powerful capabilities enhances the efficiency of our Django-based project. We utilized its extensions like:

1. **Python Extension by Microsoft:**

Provides robust support for Python development, the core language of our project.

2. **Django Extension:**

Offers specialized features like template highlighting, autocompletion, and quick navigation, tailored for Django projects.

3. **HTML, CSS, JavaScript:**

For the frontend development of our "Hire Your Wheels" project, which focuses on providing a comprehensive and user-friendly car rental experience, HTML, CSS, and JavaScript play pivotal roles in creating a seamless and intuitive interface.

- **HTML** serves as the structural foundation, defining the layout and organization of various elements on our web pages, such as car listings, booking forms, and user dashboards.
- **CSS** comes into play for styling, ensuring a visually appealing design and consistent branding across the site. We also utilize **Bootstrap**, a CSS framework, to work with many buttons and fonts to style elements responsively.

- **JavaScript** adds interactivity and dynamic behaviour, enabling features like real-time availability updates, interactive maps for location selection, form validation, and responsive user interfaces. Together, these technologies empower our frontend development, allowing us to deliver an engaging and efficient user experience on the "Hire Your Wheels" platform, enhancing accessibility and usability for our users.

Python and Django:

For the backend development of our "Hire Your Wheels" project, Python is the cornerstone, chosen for its suitability in creating a robust and scalable environment. Python, being a versatile and readable language, is instrumental in crafting the intricate logic that powers the backend functionalities of our online car rental platform.

The **Django framework**, built on Python, provides a structured foundation that streamlines the development process, allowing us to efficiently manage various aspects of our project, such as user authentication, car management, booking processing, and payment integration. Additionally, its extensive ecosystem and Django's built-in features simplify tasks such as database management, ensuring that our backend operations are not only powerful but also maintainable.

Database (SQLite3):

This is the default database of Django, and we utilized it to store all of the application's data. This includes user information, car details, booking records, payment information, location data, and car type specifications. We can also update and delete the data from the database, and we can use filters and queries to retrieve the desired data efficiently.

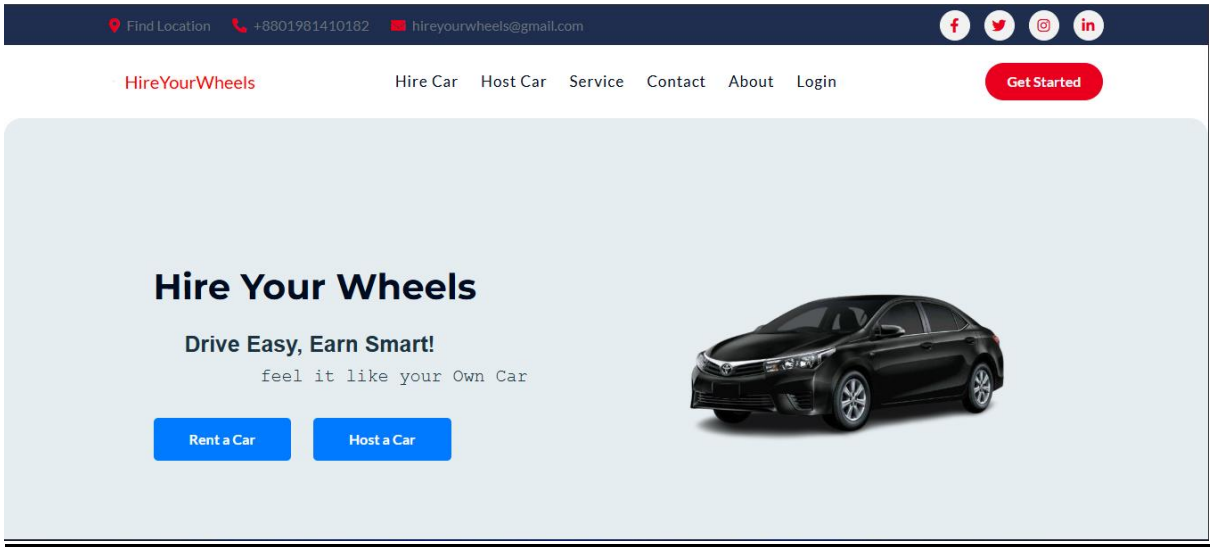
Google reCAPTCHA (v3):

Google reCAPTCHA v3 operates in the background, employing advanced risk analysis algorithms to evaluate user interactions without disrupting the user experience. By assigning a risk score to each action (such as user registration or booking attempts), it enables our system to make informed decisions about whether a specific interaction is likely to be legitimate or potentially malicious (e.g., bot activity). This proactive approach strengthens the integrity of our platform by mitigating the risks associated with fraudulent activities and ensuring that sensitive information (like user credentials and payment details) remains protected.

4.2 Example

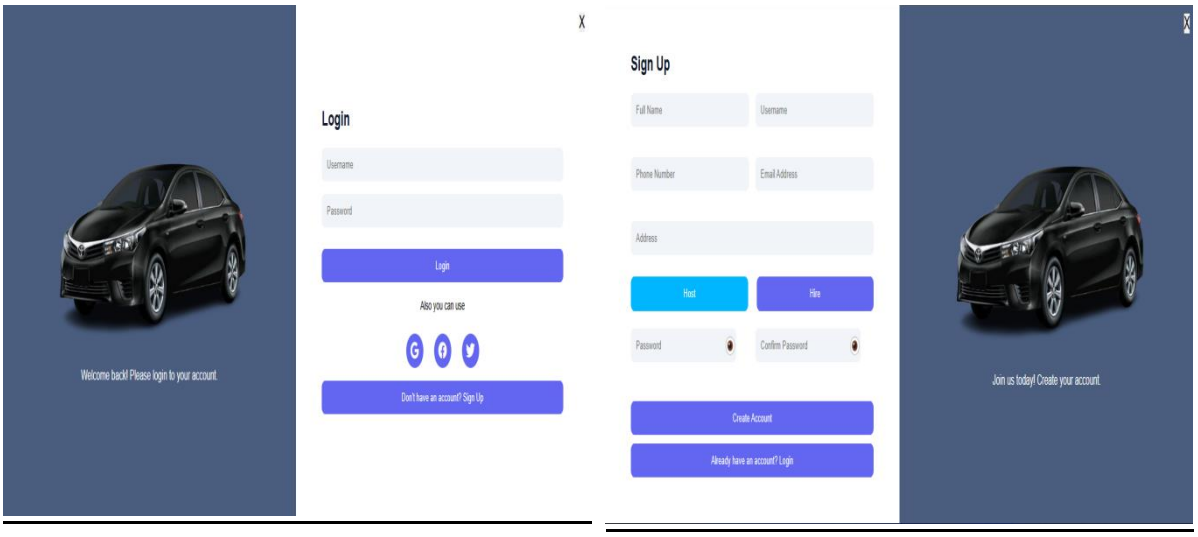
Homepage before Login:

This Is our homepage, There are many other things, but this is just the basic homepage,If the user is not logged in, the navbar will show a 'Login' option



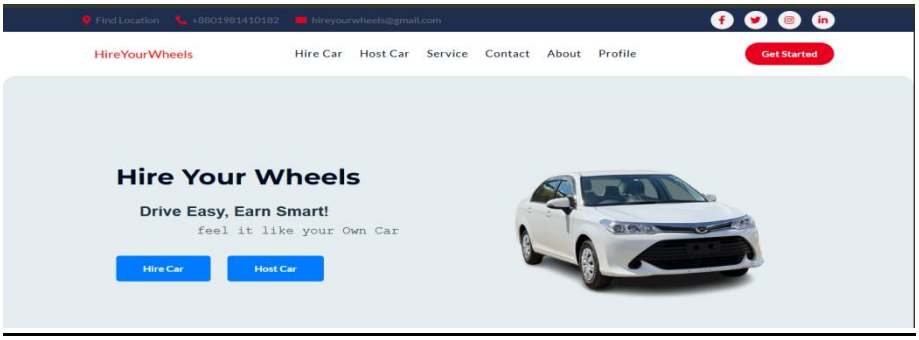
Login and Signup:

Everyone can easily Login and Signup



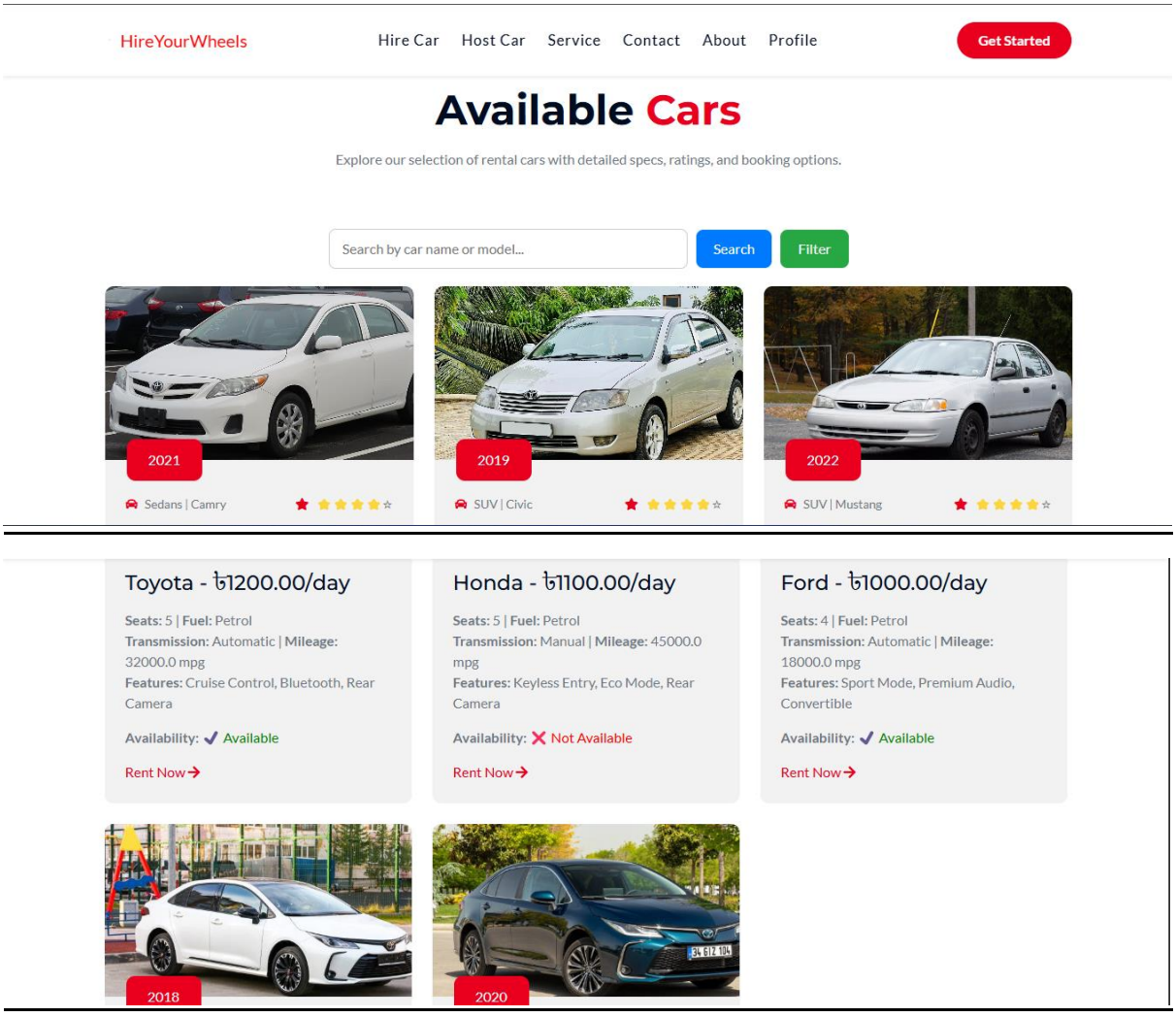
Homepage before Login:

Once logged in, it will change to a 'Profile' option.



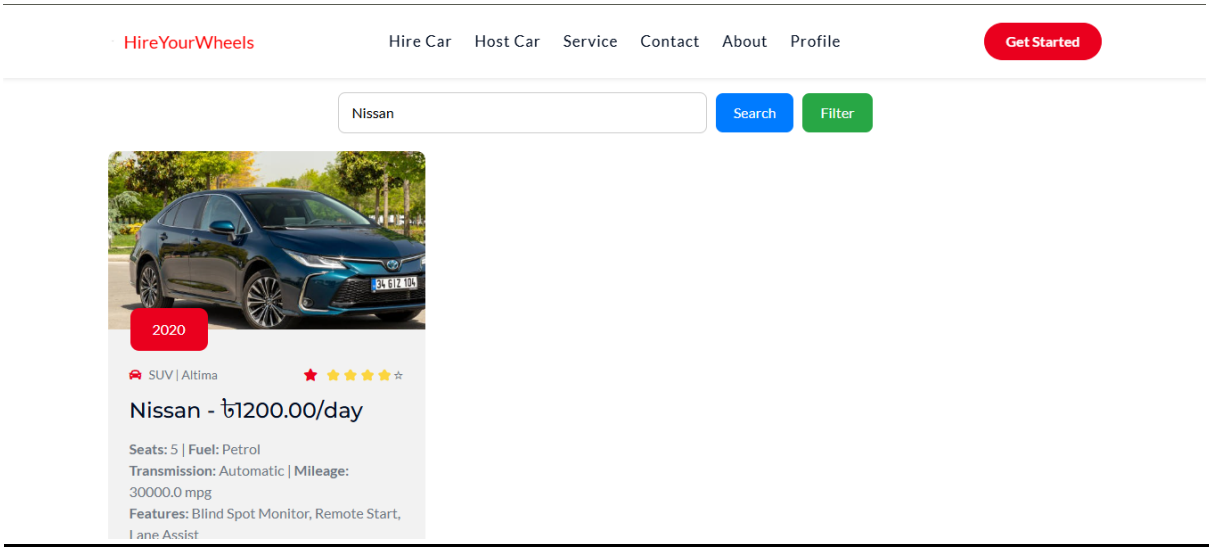
Hire:

When the user selects the 'Hire' option, all available cars will be displayed, and they can choose the one they like.



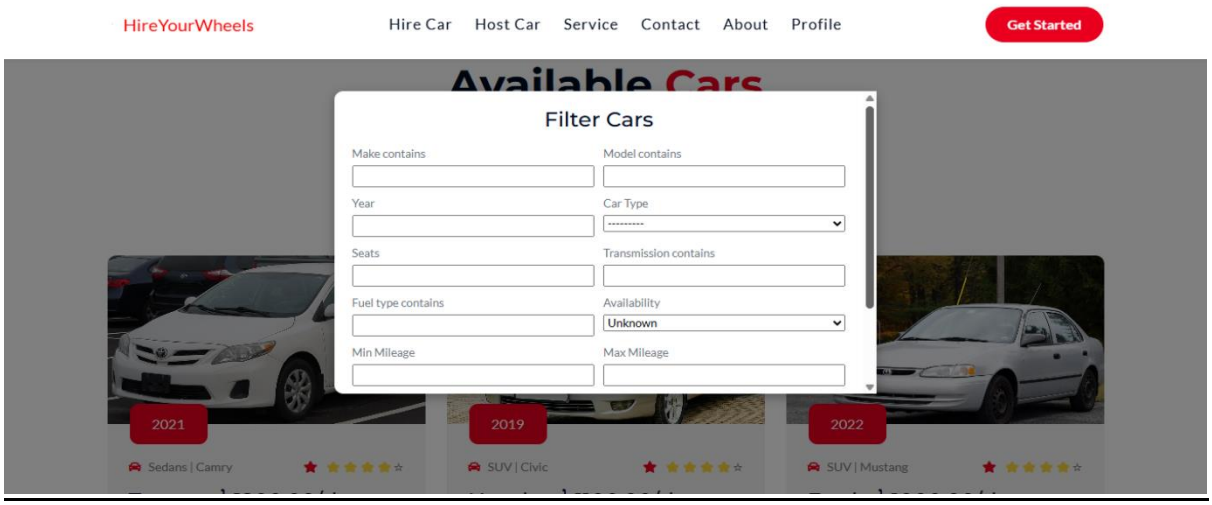
Search:

If the user searches by name or model, they will find the car they prefer.

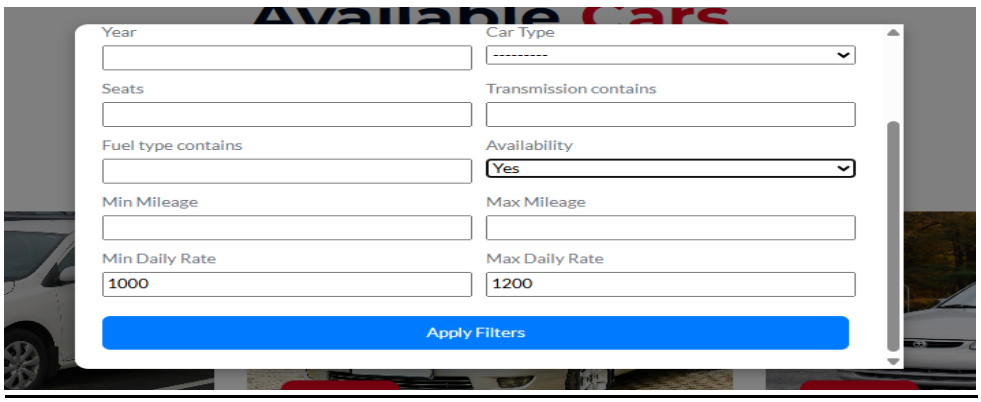


Filter:

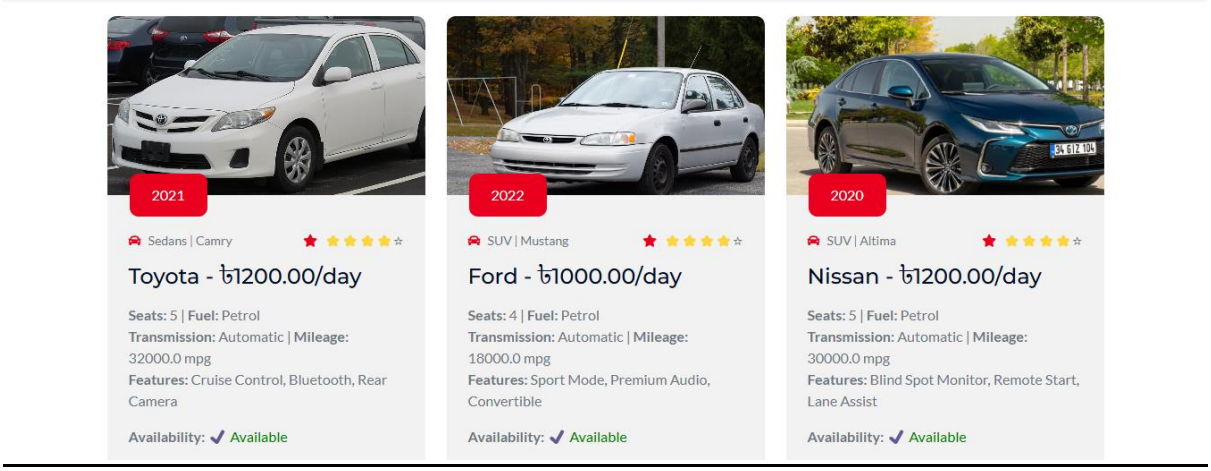
User can also use filter to find his choice easily



Look a demo, Availability - yes, rate 1000 to 1200

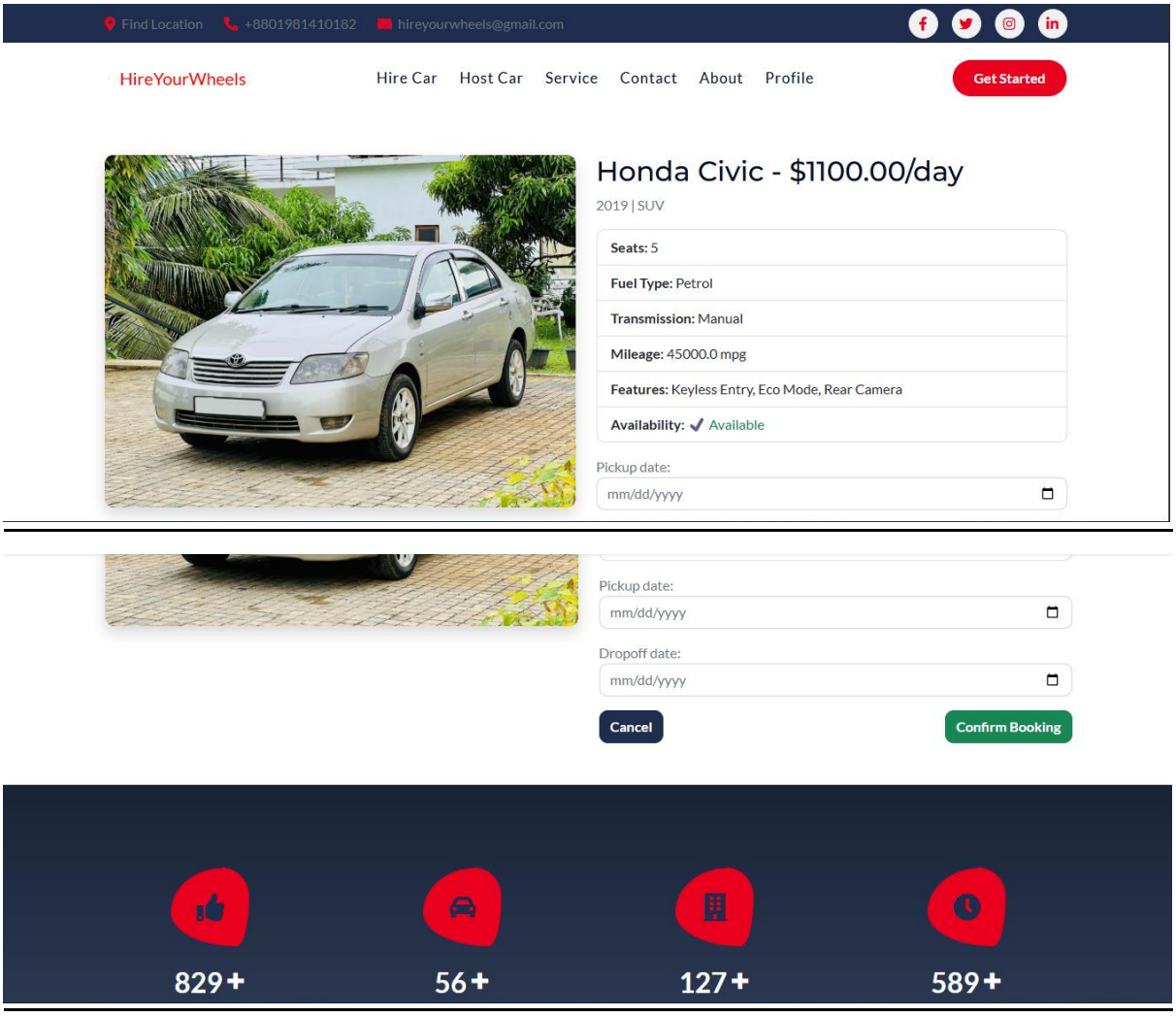


Filter will help the user to find his choice more efficiently



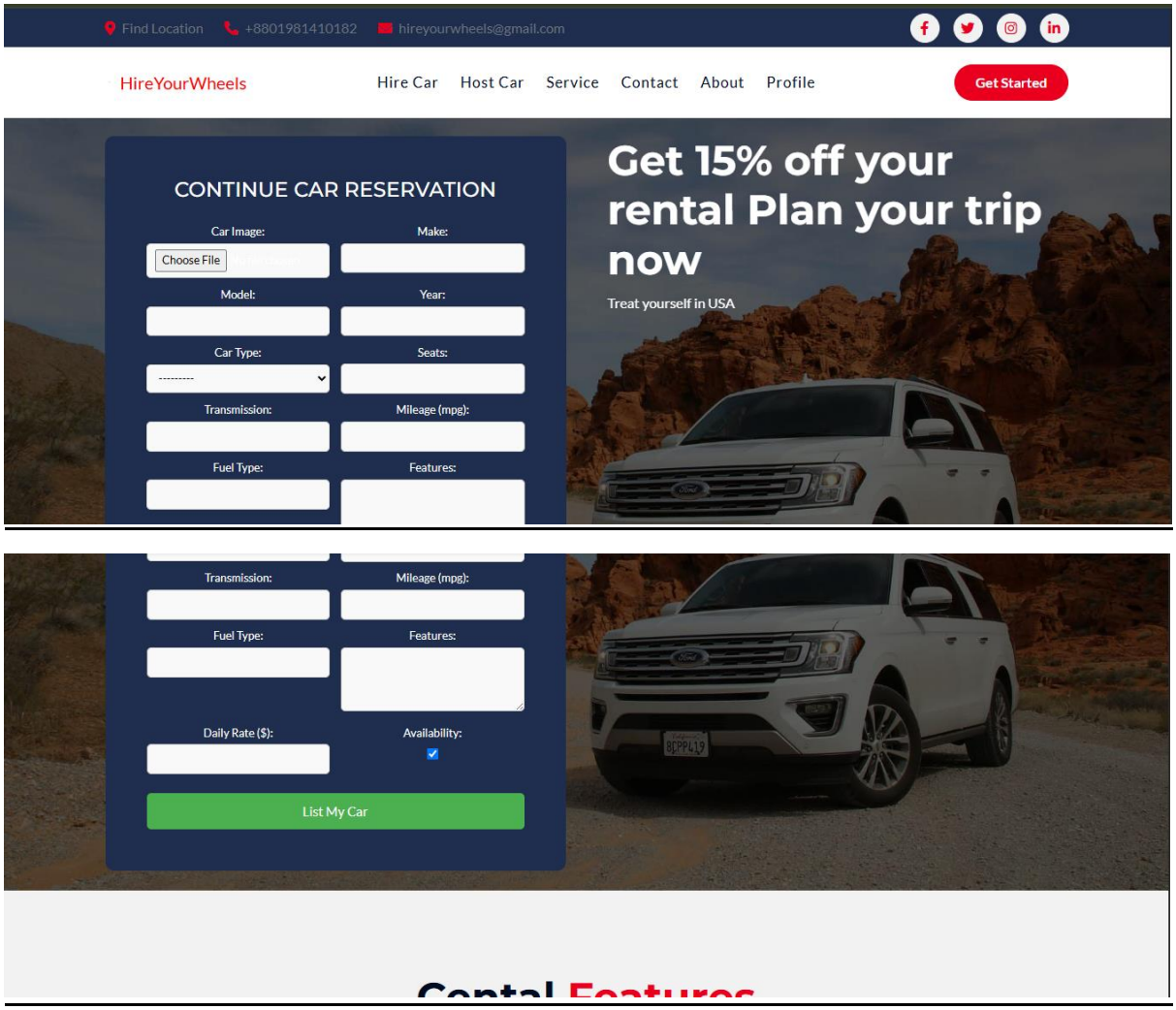
Details:

When the user clicks the 'Rent' button, they will be taken to the car's details page. If they like the car, they can select the pickup and dropoff dates and then confirm the booking.



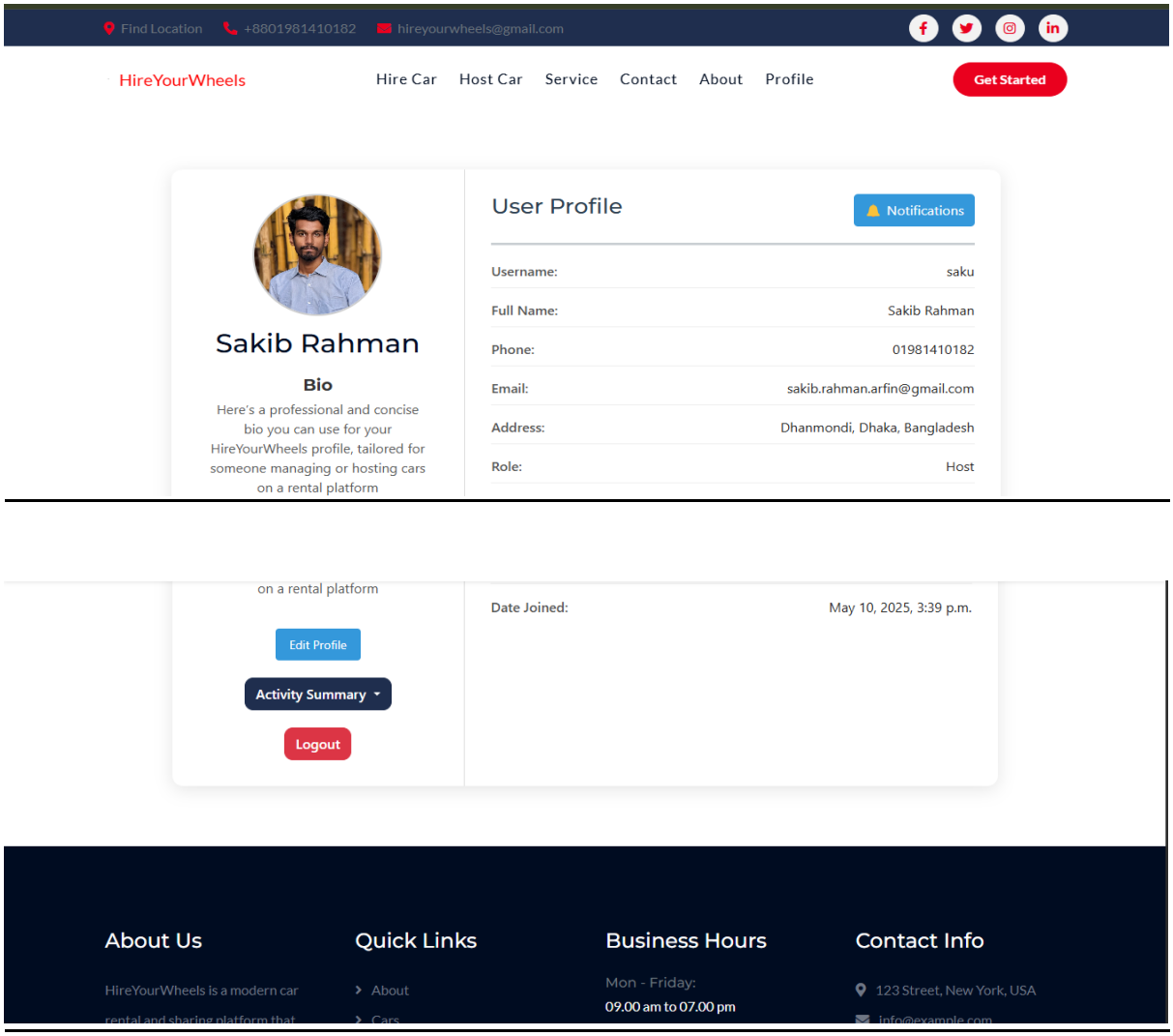
Host:

From the 'Host' option, a user can easily host their car.



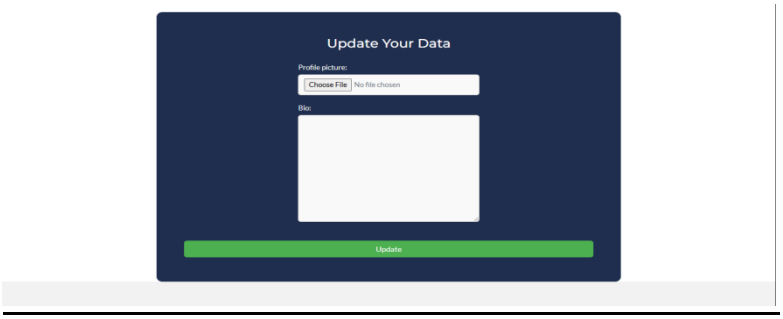
Profile:

Each account will have its own profile section—whether the user is a host or a renter. From the profile, they can view the cars they've hosted or rented. Additionally, users will be able to log out, edit their profile, and view notifications from the same section.



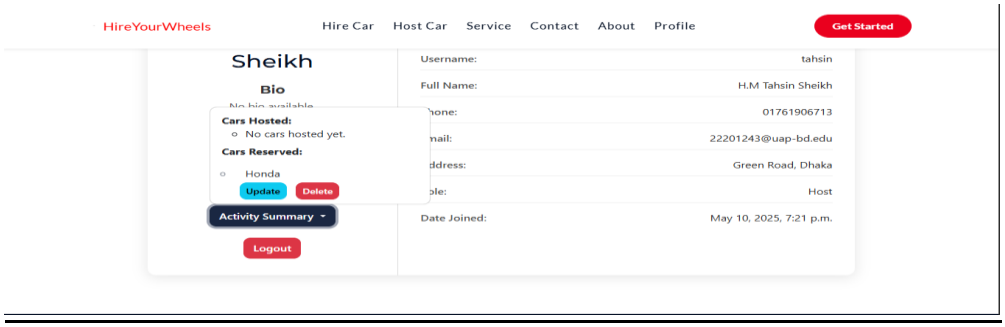
Add or Update Imgae or Bio :

Everyone can edit or upload his bio and picture



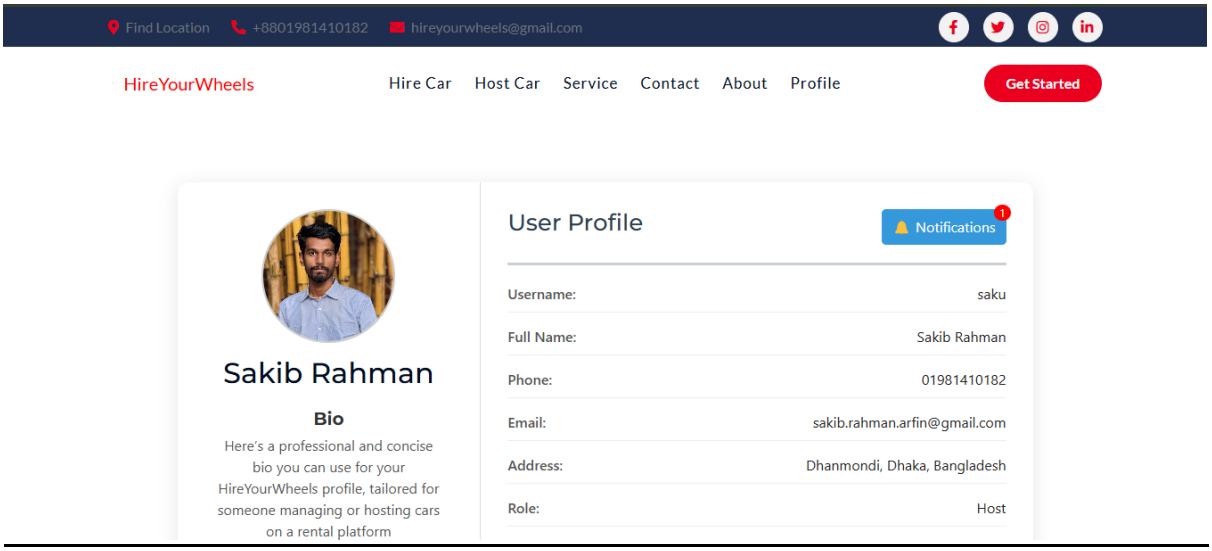
Activity Summary:

If he press Activity summary button then can see how many car he host and how many car he get rent, he can also edit and delete

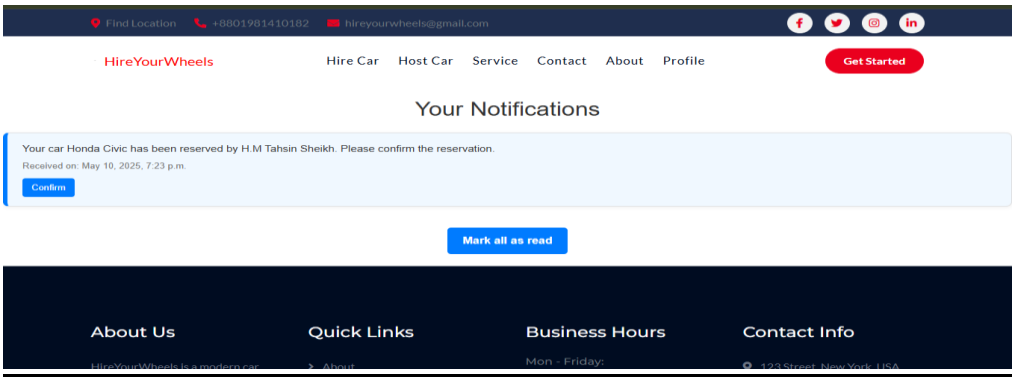


Notification:

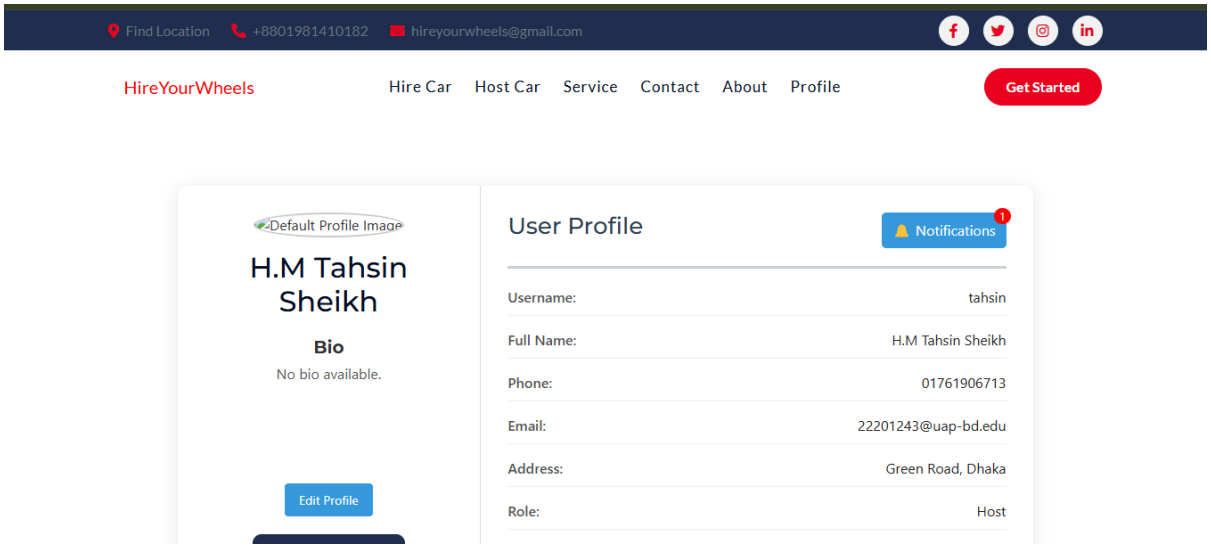
When a booking is confirmed, a notification will be sent to the person who hosted the car."Would you like to include how the notification will be sent.



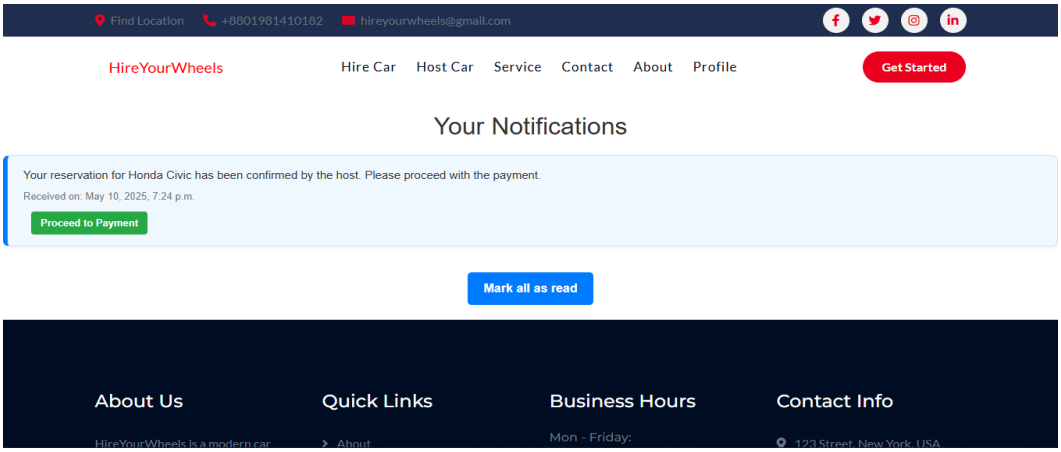
A notification will be sent like this, and then the host will confirm the booking.



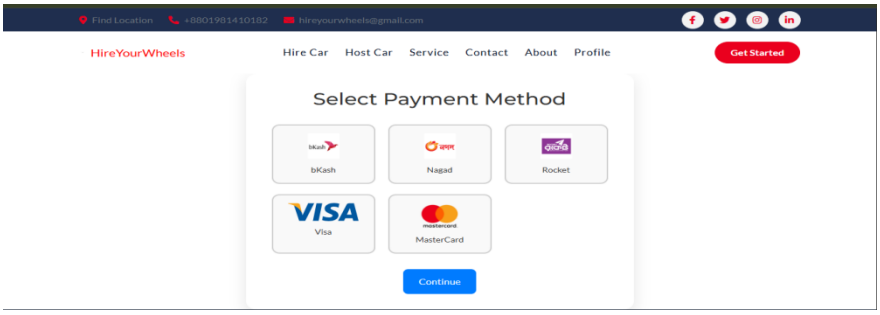
When the host confirms the booking, a notification will be sent to the renter.



And the payment option will be unlocked.



When the user clicks 'Proceed to Pay', the payment page will be displayed.



5. Team Contribution

Tasks	22201240 - Sakib Rahman Arfin	22201243 - H M Tahsin Sheikh	22201248 - Tasfia Tabassum
HTML page creation	These are the HTML file names: • base.html • car_list.html • car_detail.html • booking_create.html, • booking_detail.html, • booking_update.html	These are the HTML file names: • payment_process.html, • payment_confirmation.html • Database design and implementation details • Payment integration details • API integration details (if any)	These are the HTML file names: • home.html • user_login.html • user_register.html • user_profile.html • about_us.html
HTML page linking and URL mapping	• Worked on urls.py and project structure.	• Worked on urls.py	None
Design with CSS	These are the CSS file names: • base.css • car_list.css • car_detail.css • booking.css (for booking-related pages)	These are the CSS file names: • user_auth.css (for login/register)	These are the CSS file names: • home.css • user_profile.css • about_us.css • payment.css
Database creation in (models.py) and registration in (admin.py)	These are the tables/classes: • car • Booking • Location • Handled migrations and registration.	These are the tables/classes: • User (Built-in) • Payment Completed forms.py (User forms, Booking form). • Helped with bug fixes related to user authentication and payments.	These are the tables/classes: • Rating
Backend Logic handling (views.py, forms.py) and displaying data	• Completed views.py for car listings and details. • Worked on HTML files to show car data.	None	None
Features and bug fixing	• Implemented car search/filtering. • Created booking forms. • Implemented car detail view. • Fixed bugs related to car display.	• Helped with Bug fixing	None

	• **Implemented viewing bookings in user profile.**		
Version Control	• **Managed the project's version control via GitHub.**	**None**	**None**
Report Writing	• **Managed the project's version control via GitHub.** • **Introduction** • **Project Description** • **Architecture and details** • **Team contributions** • **ER Diagram**	• **ER Diagram and details**	• **Conclusion and Future work** • **Requirements Analysis** • **Testing and Evaluation**

6. Conclusion:

In conclusion, the "Hire Your Wheels" project presents a comprehensive and innovative web and mobile application platform, addressing critical challenges in the transportation sector. With a primary focus on providing convenient and flexible car rental services, "Hire Your Wheels" encompasses features such as user management, car browsing and search, booking management, payment processing, location services, car types, and a rating and review system. The project aims to streamline the car rental process, reduce complexities associated with traditional methods, and empower users with the freedom of having a car when needed.

The motivation behind "Hire Your Wheels" is rooted in the increasing need for accessible and flexible transportation solutions in today's fast-paced world. The project tackles critical challenges, including building trust among users regarding car condition and availability, ensuring the security of transactions and user data, and implementing an efficient system for managing bookings and logistics. The ambition to provide a seamless and user-friendly experience reflects the commitment to offering a reliable and convenient alternative to car ownership.

Future Work:

The future trajectory of the "Hire Your Wheels" project holds tremendous potential for growth and impact. To address the challenges of online car rental adoption, focused efforts on user education and trust-building initiatives are essential. Engaging with users through feedback mechanisms, providing transparent information about car conditions and insurance options, and offering excellent customer support will be crucial in refining and enhancing the platform's acceptance.

The implementation of advanced features such as real-time car tracking, dynamic pricing, and integration with navigation services remains a pivotal aspect of future development. Overcoming challenges associated with ensuring accuracy, reliability, and scalability will require collaboration with experts in geolocation technologies, data analytics, and software engineering.

Moreover, as the project evolves, expanding the range of services (e.g., adding insurance options, roadside assistance), collaborating with car rental companies, and incorporating emerging technologies (e.g., AI-powered customer support, electric car integration) will contribute to "Hire Your Wheels'" continual relevance and impact.

In the broader scope, potential partnerships with local businesses, tourism agencies, and transportation authorities could amplify the project's societal benefits and contribute to the improvement of transportation accessibility and efficiency.

CEP Mapping:

How K's are addressed through the project:

Ks	Attribute	How Ks are addressed through project
K3	**Engineering Fundamentals**	To create a viable model and construct a functional website, our project requires knowledge of foundational concepts like database administration and web development.
K4	**Specialist Knowledge**	We made advantage of our administrative specialist knowledge, data field expertise, and Django framework expertise. To store data, we also require an appropriate database administration system.
K5	**Engineering Design**	To construct everything in an orderly fashion, we used Django's MVT structure and produced system designs such as ER diagrams.
K6	**Engineering Practice**	VSCode will be used to construct this project, and languages including HTML, CSS, JavaScript, and Python will be used. Frameworks like Django and Bootstrap will be used. We stored the necessary data in a SQLite3 database.
K7	**Comprehension**	The concept makes a significant impact in emergency situations by providing quick access to physicians, blood banks, and medications.

How P's are addressed through the project:

Ps	Attribute	How Ps are addressed through project
P1	**Depth of Knowledge Requirement**	To create practical features, we had to pool our many talents, including web programming, databases, Django, and considering how it benefits society.
P2	**Depth of analysis requirement**	Further research on the tracking system is necessary for this job. In order to implement our rental automobile

		monitoring system and the optimal route, we should put in place an efficient tracking system. system. It will assist our project in achieving its ultimate objective.
P6	**Extent of stack holder**	able to register, maintain their account, and view their activity Customers can view a variety of vehicles and make a choice.
P7	**Interdependence**	The system functions as a whole thanks to the cooperation of all its components, including user accounts, product orders, emergency requests, and appointments.

How A's are addressed through the project:

As	**Attribute**	**How As are addressed through project**
A1	**Range of Resources**	We employed a variety of resources, including people (physicians, developers), tools (Django, Bootstrap), data (hospitals, users), and financial planning.
A4	**Consequences for society and the environment**	People benefit from quicker medical assistance, fewer paperwork (which is environmentally friendly), and time and effort savings through online services.