LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS6004NI Application Development

Assessment Weightage & Type

30% Group Coursework

Title (Where required): Hajur Ko Car Rental

Year and Semester 2022 Autumn

Word Count: 1800

Group Name:			
SN	**Student Name**	**College ID**	**University ID**
1	**Punam Thapa Magar**	**Np01cp4s210273**	**20048968**
2	**Poonam Singh Yadav**	**Np01cp4s210213**	**20049412**
3	**Aashutosh Sharma**	**Np01cp4s210087**	**20048803**
4	**Samita Sunuwar**	**Np01cp4s210059**	**20049016**
5	**Girish Shrestha**	**Np01cp4s210082**	**20049993**

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgement

This Group Coursework on the Application Development Module was prepared in compliance with the bachelor's third-year, second-semester regulations of London Metropolitan University. It is the result of the cooperation and support of module tutor **Mr. Abhishek Humagain.**

We have taken endeavors in this venture. However, it would not have been possible without our module tutors' generous support and assistance. First of all, we'd like to express our gratitude to him. We owe a lot of gratitude to our highly regarded tutor, **Mr. Abhishek Humagain,** for his direction and constant supervision, as well as for providing the necessary project information and supporting the project's preparation. Your precious time has absolutely assisted us in giving significance to our undertaking with working. Second of all, we want to express our gratitude to Islington College for providing us with a wealth of possibilities, resources, and support, as well as a fantastic learning environment. Finally, we would like to express our sincere appreciation to the following individuals: Mr. Aashutosh Sharma, Mr. Girish Shrestha, Miss Punam Thapa Magar, Miss Poonam Singh Yadav, and Miss Samita Koinch Sunuwar, for their alliances, benevolence, commitment, and devotion to leading this Groupwork completion at the scheduled deadline.

In addition, our thanks and admiration also go towards our friends, group members, tutors and everyone who have willingly helped us out with their abilities.

Table of Contents

Table of Tables

Table of Figures

1. Introduction

We have been given a group coursework assignment that accounts for 30% of the module grade total and requires us to develop a management system for Hajur Ko Car Rental that will streamline the rental process and offer a seamless user experience for both operators and clients. The main objective of the course is to develop and build a C# web application that will help a car rental business maintain track of its inventory, stock, and rented cars. The system that we need to develop for Hajur Ko Car Rental is supposed to be written in the C# programming language on the ASP.NET platform using the integrated Visual Studio Environment. This system's main objective is to facilitate the management of leased and returned vehicles for Hajur Ko Car Rental.

Continue Reading…

Users will only have restricted access to some system features in the created system, but administrators will have access to more features than users, and staff members would have complete system access. This implies that neither the customer nor the employees are able to accomplish all of the tasks that the admin and staff are capable of. The activities that both a user and an admin can carry out are all accessible to staff, on the other hand.

2. Detail Instruction to Run the Program

2.1. Home Page

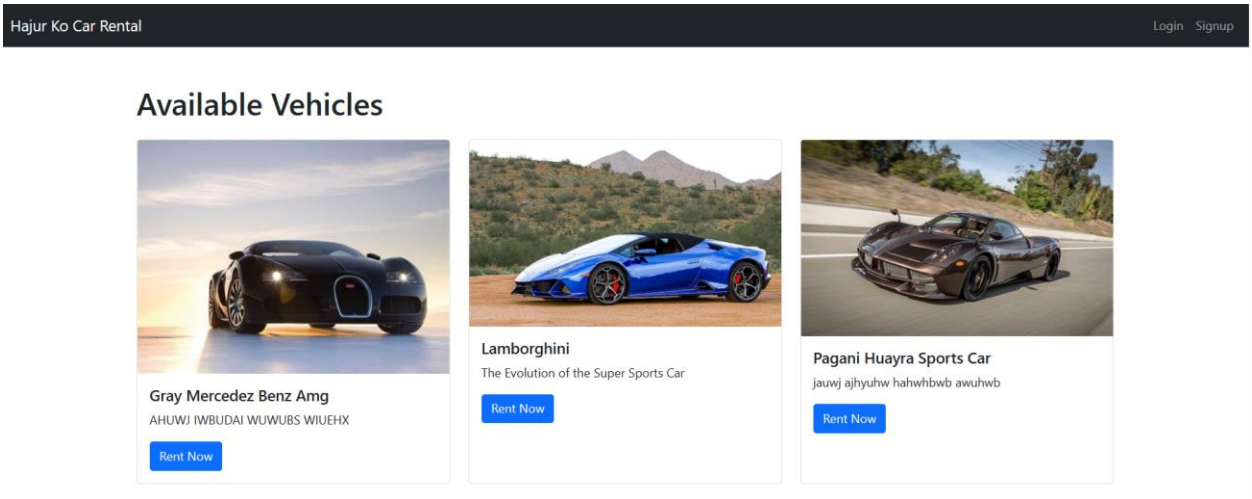

Figure 1 Home Page

2.2. Login

After the user presses the Login button as shown above, they are directed to User Login page as follows:

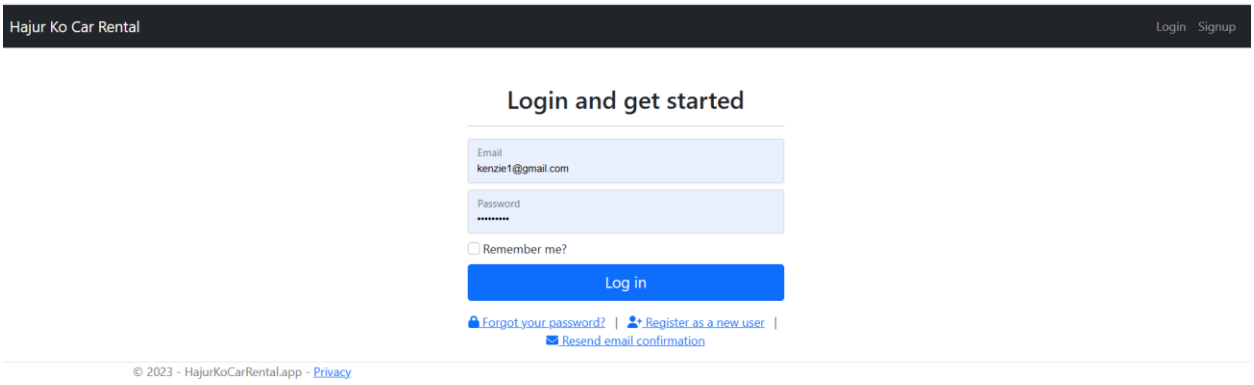

Figure 2 Login Page

2.3. Admin Dashboard

If the user logs in successfully, an admin dashboard is displayed as follows:

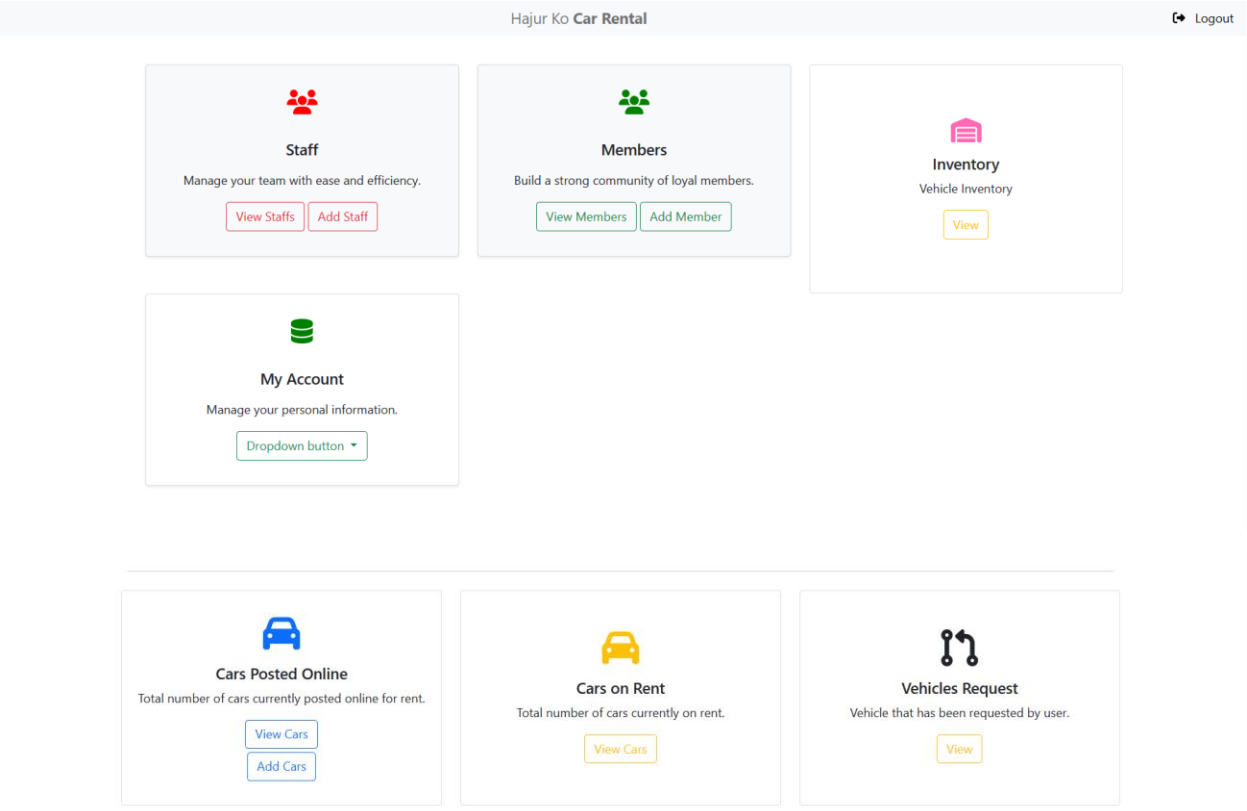

Figure 3 Admin Dashboard

2.3.1. Add Staff

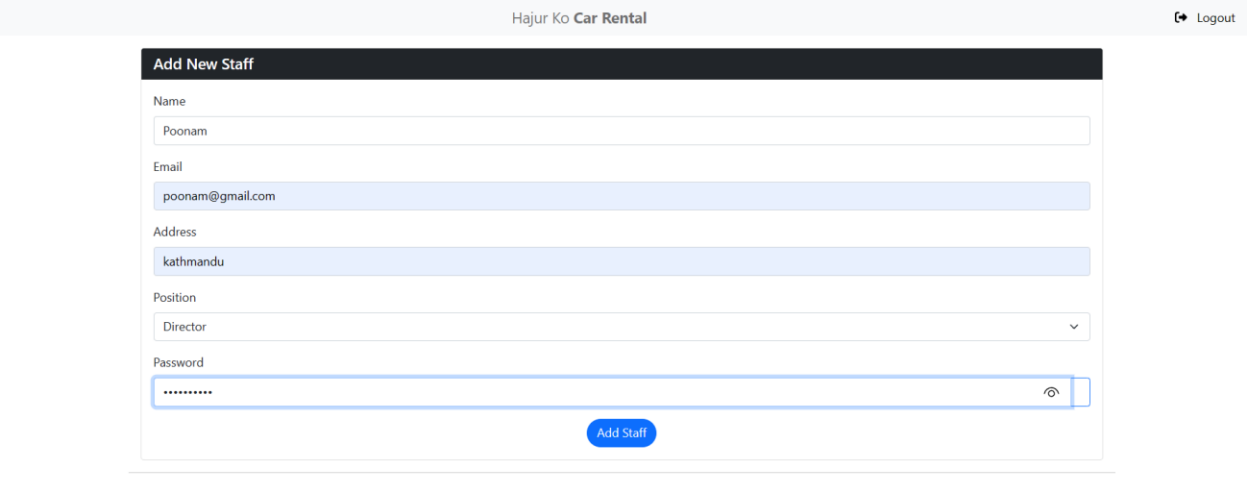

Figure 4 Adding Staff By Admin

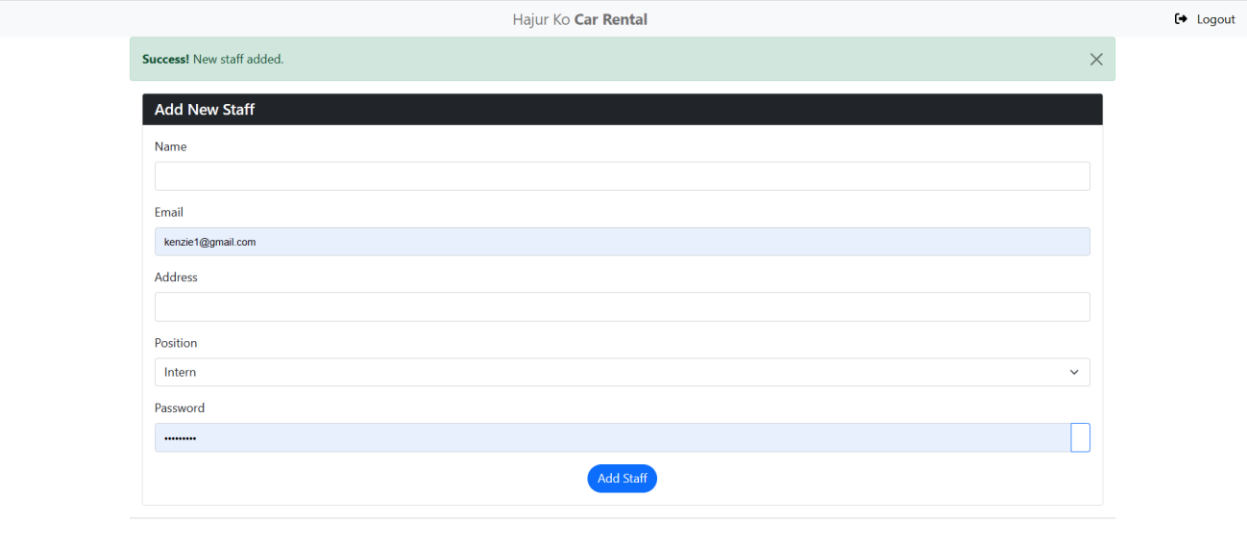

Figure 5 Successfully New Staff Added

2.3.2. View Staff

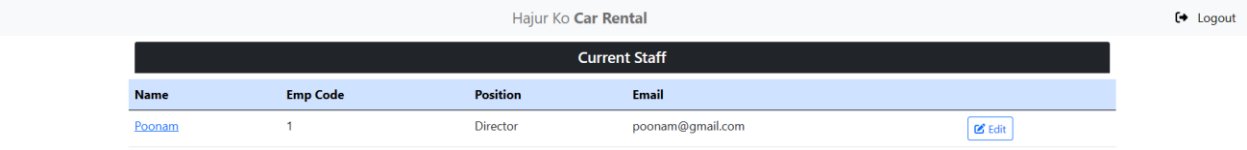

Figure 6 List of Staff

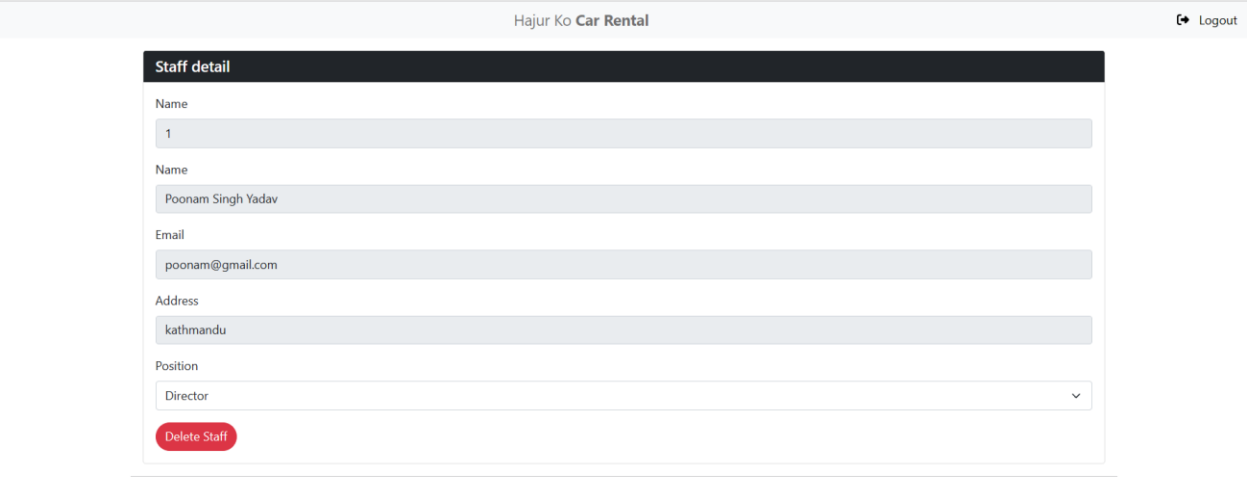

Figure 7 Edited Staff Details

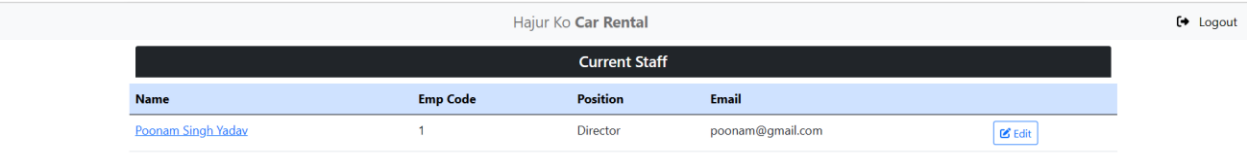

Figure 8 Updated Staff Details

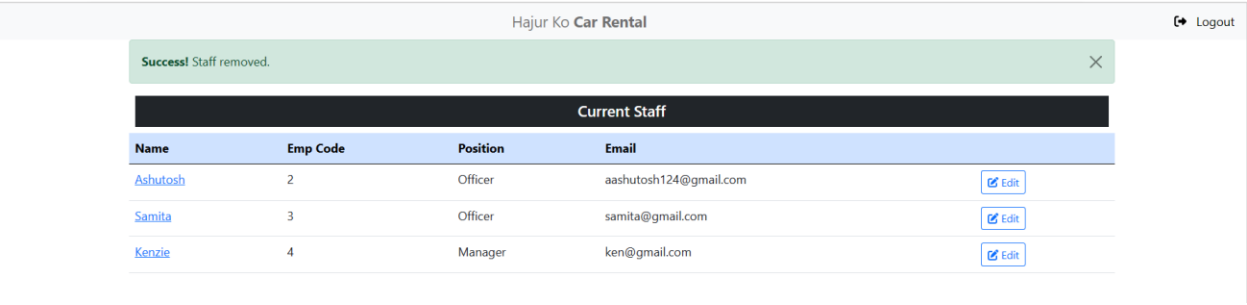

Figure 9 Employee Code 1 Deleted Successfully

2.3.3. Add Vehicle for Rent

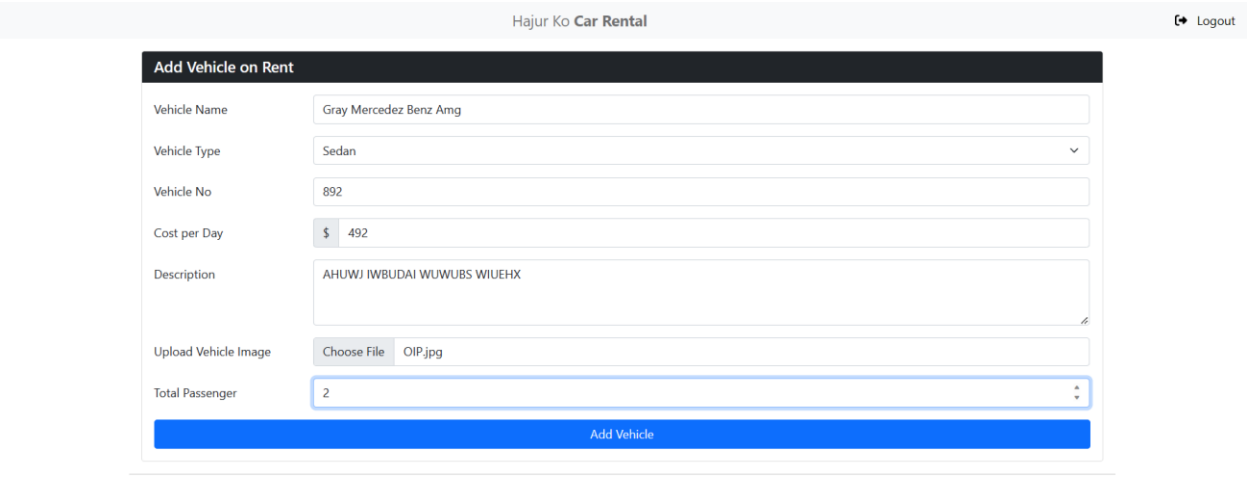

Figure 10 Adding Vehicle for Rent

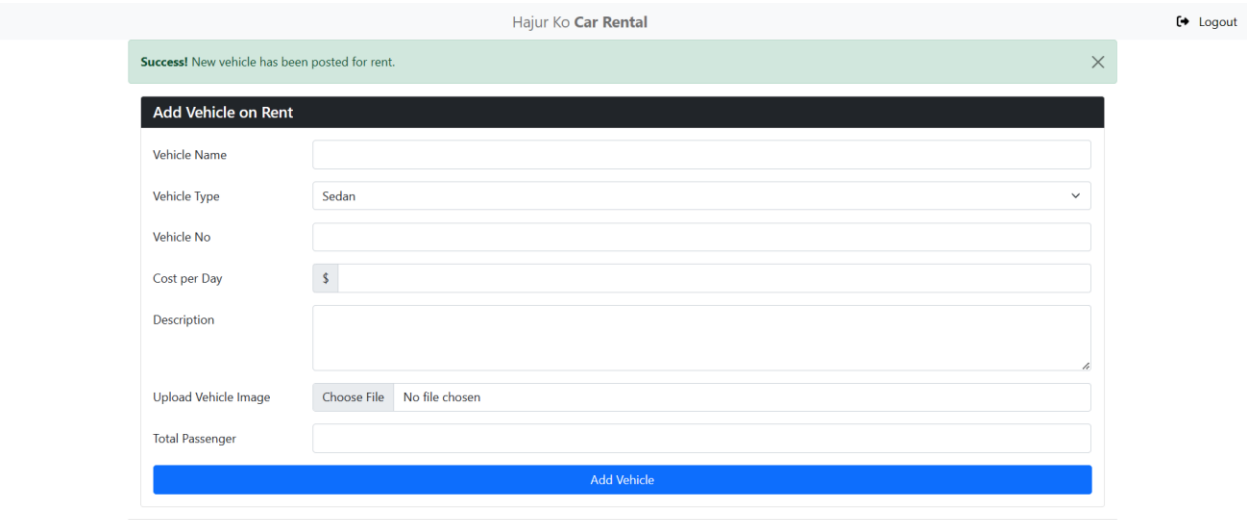

Figure 11 New Vehicle added successfully

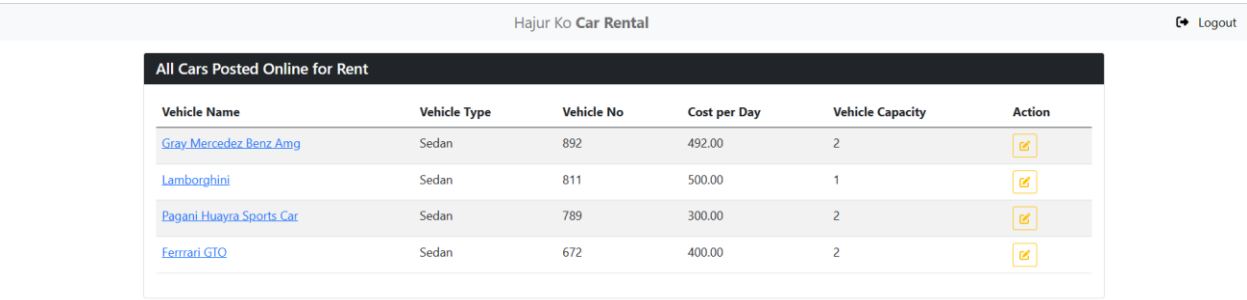

Figure 12 List of Cars for Rent

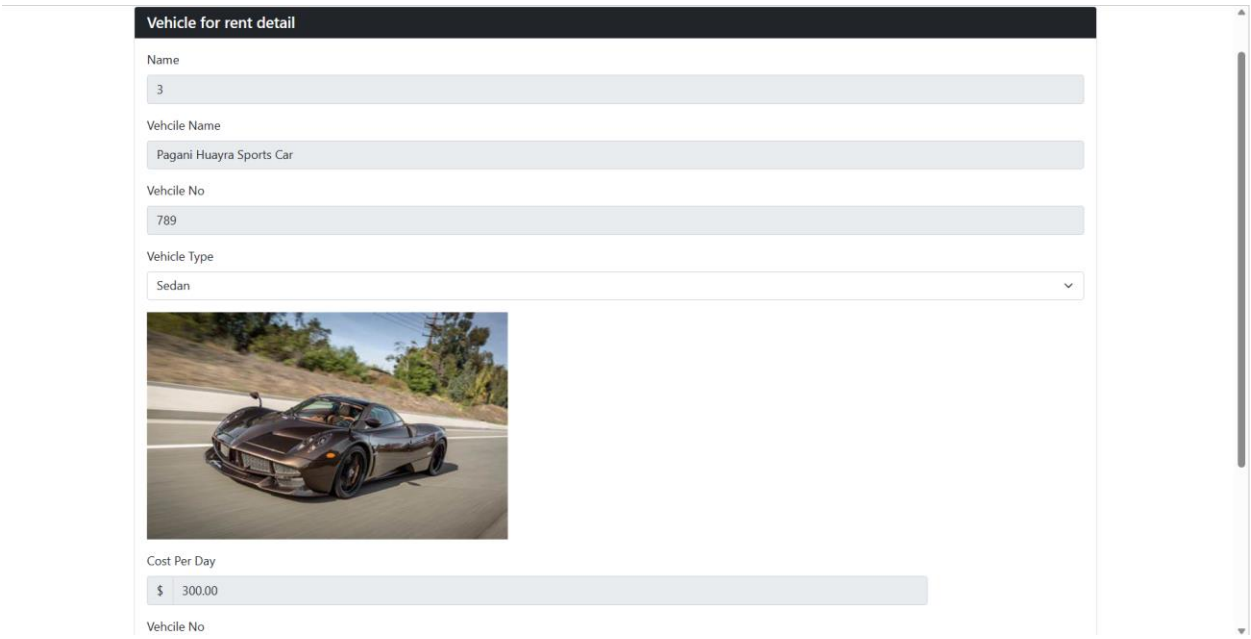

Figure 13 Vehicle for Rent Detail

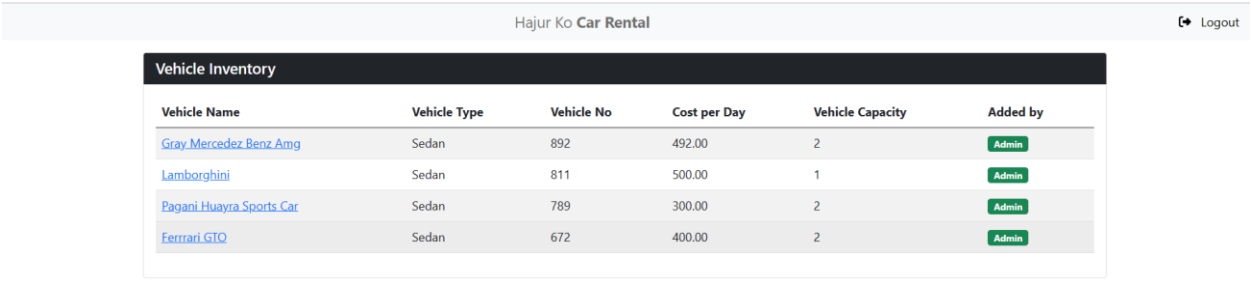

Figure 14 Vehicle Added by Admin For Rent

2.4. Staff Dashboard

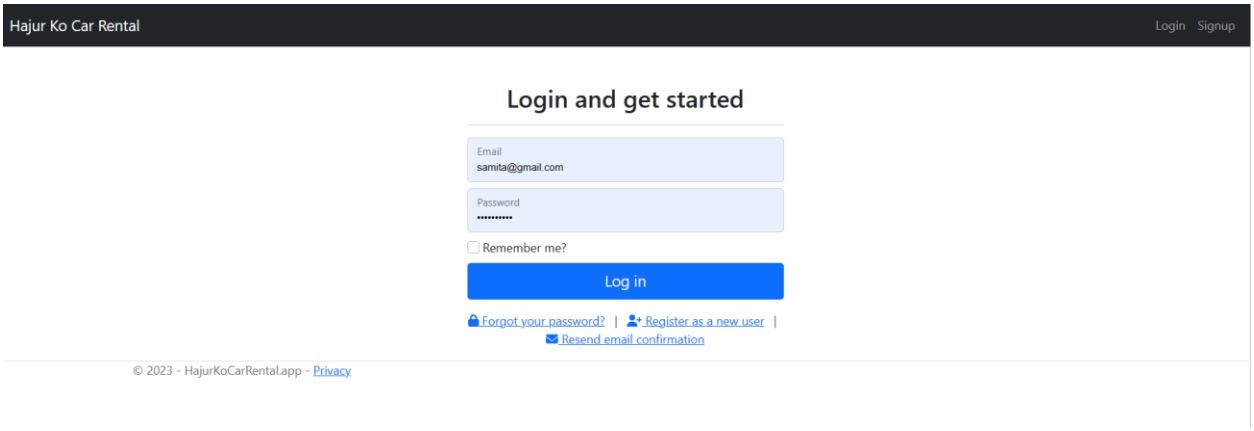

Figure 15 Staff Login

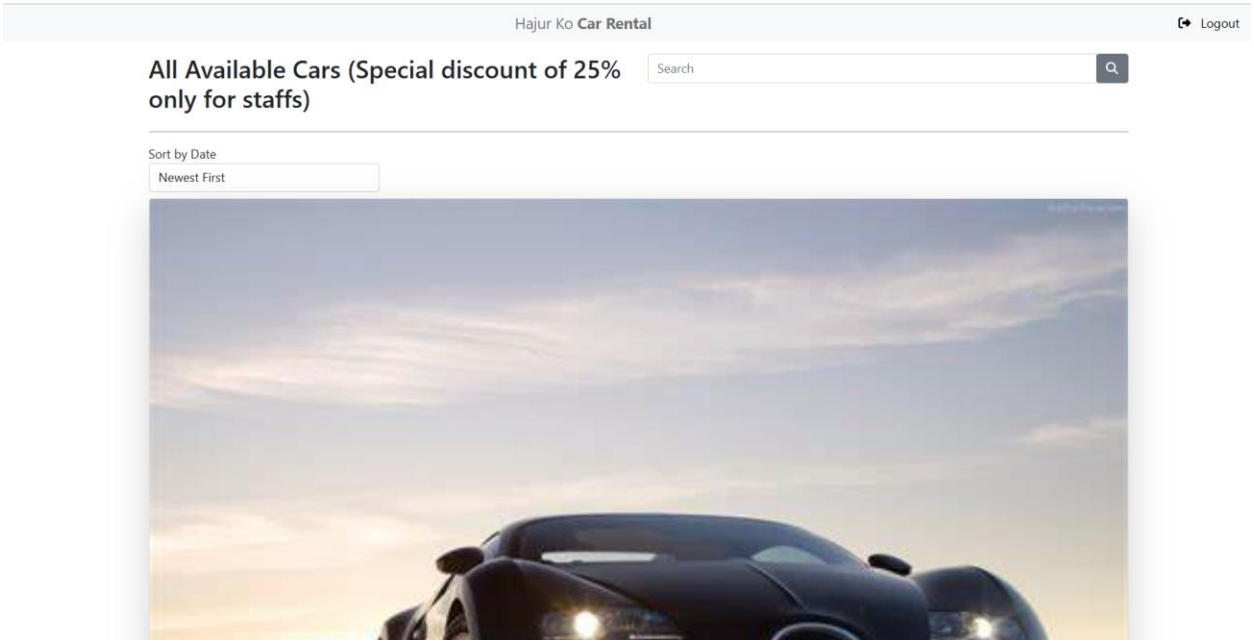

Figure 16 Discount of 25% only for staffs for renting car for themselves

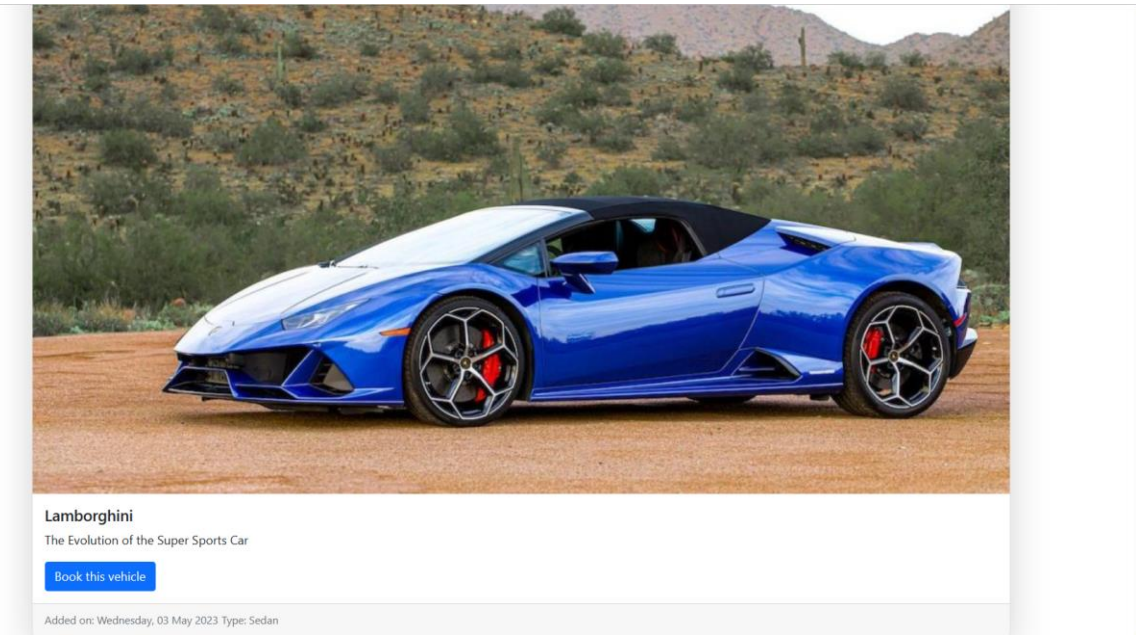

Figure 17 Booking the vehicle for rent

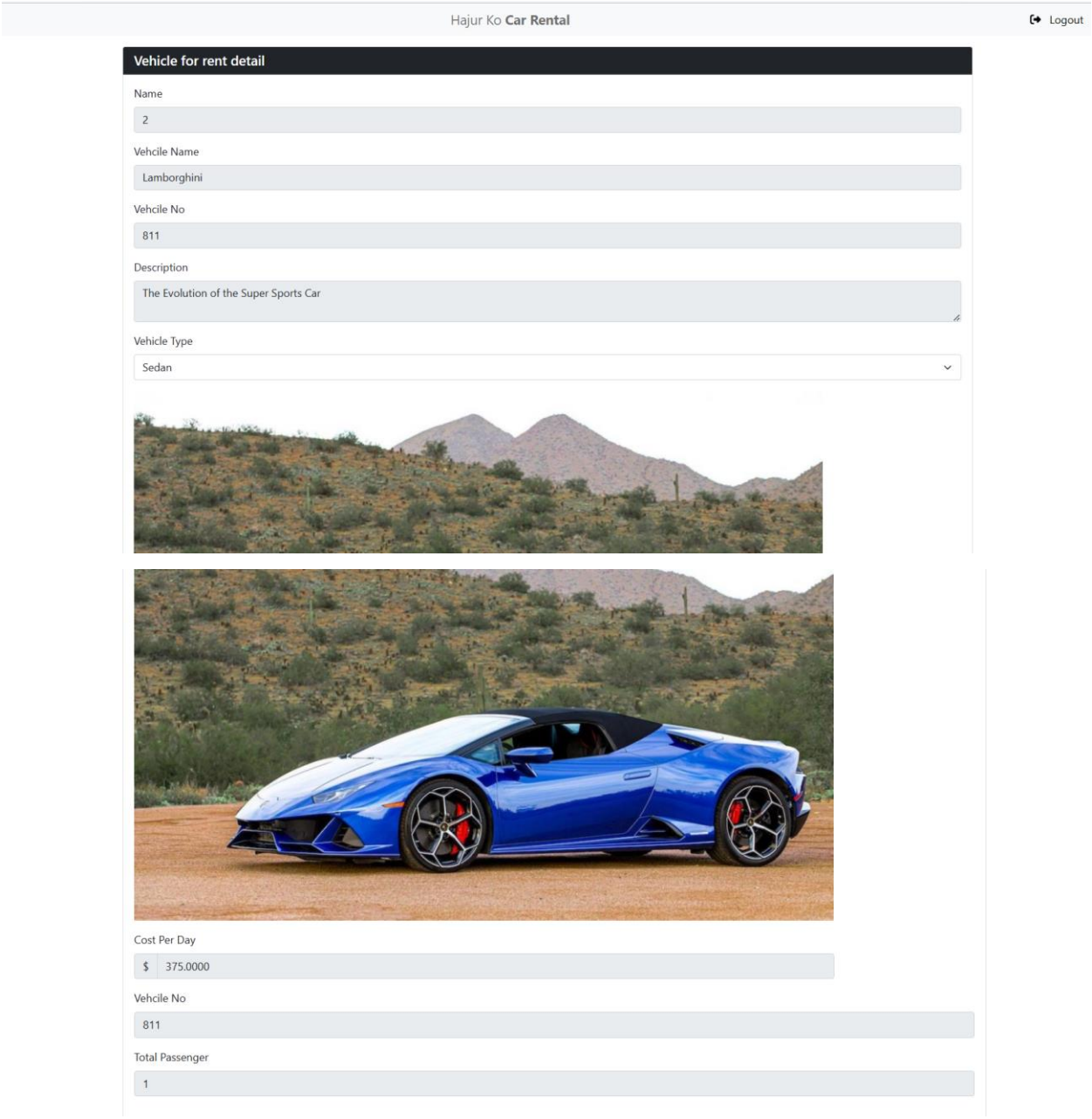

Figure 18 Vehicle Details To be Rented

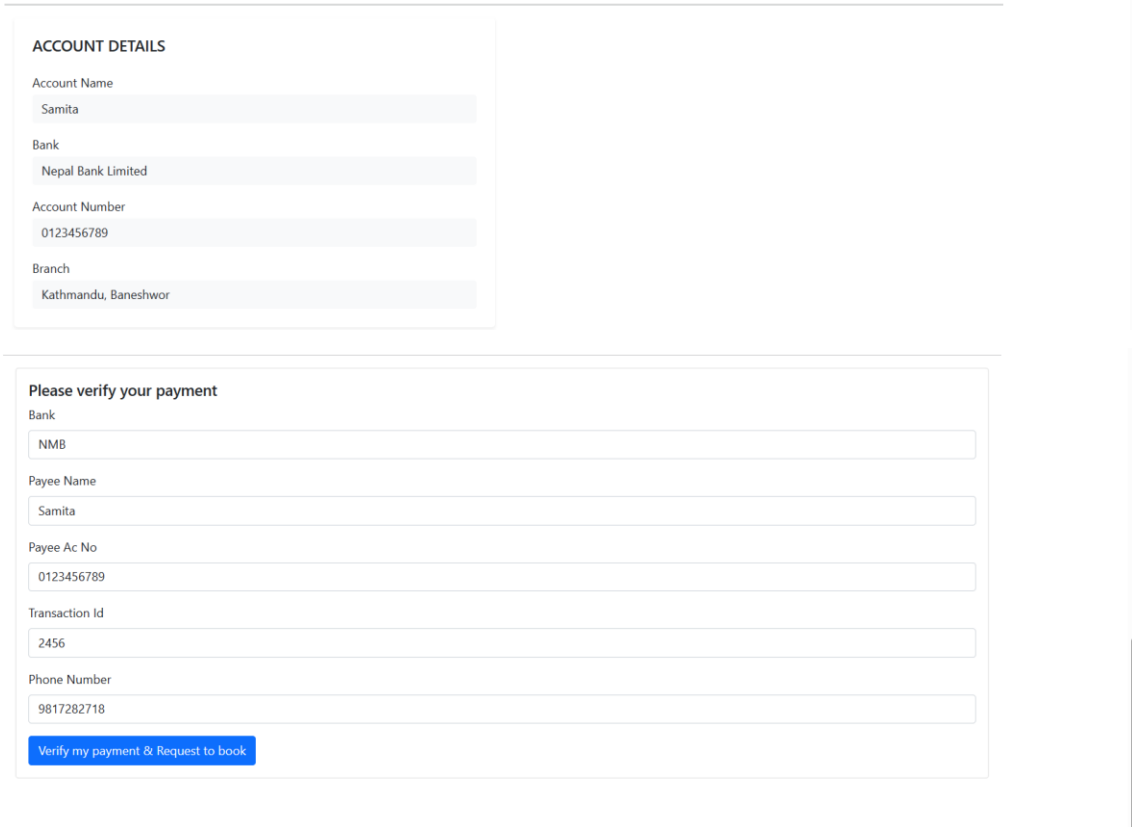

Figure 19 Account Details and Payment Verification

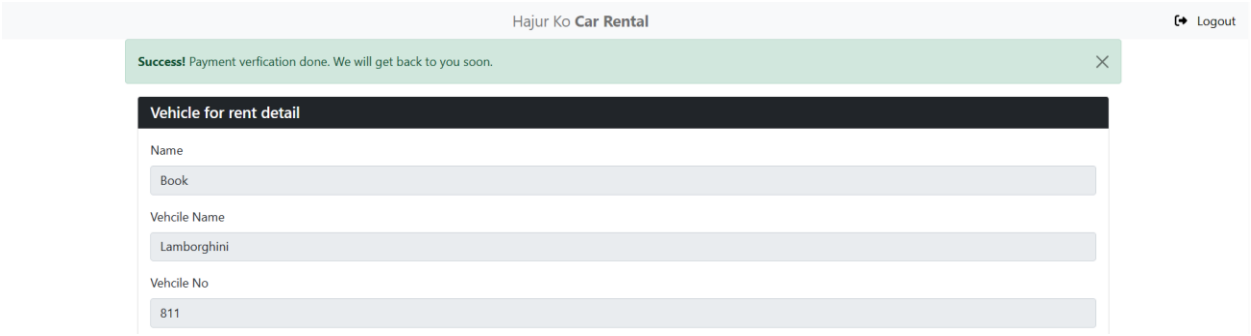

Figure 20 Payment Verification Done Successfully

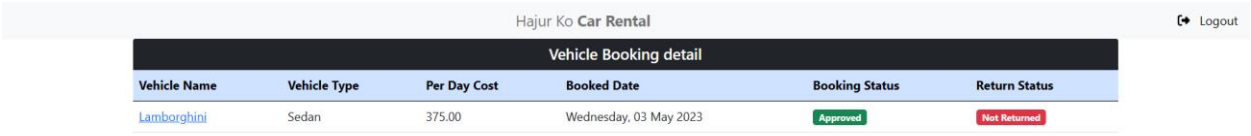

Figure 21 Rental Request Approved

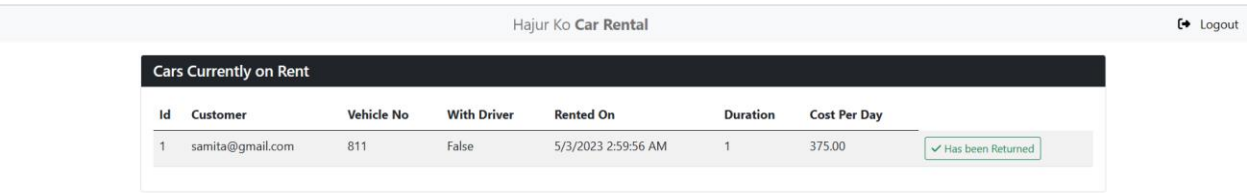

Figure 22 List of Cars on Rent

2.5. User Dashboard

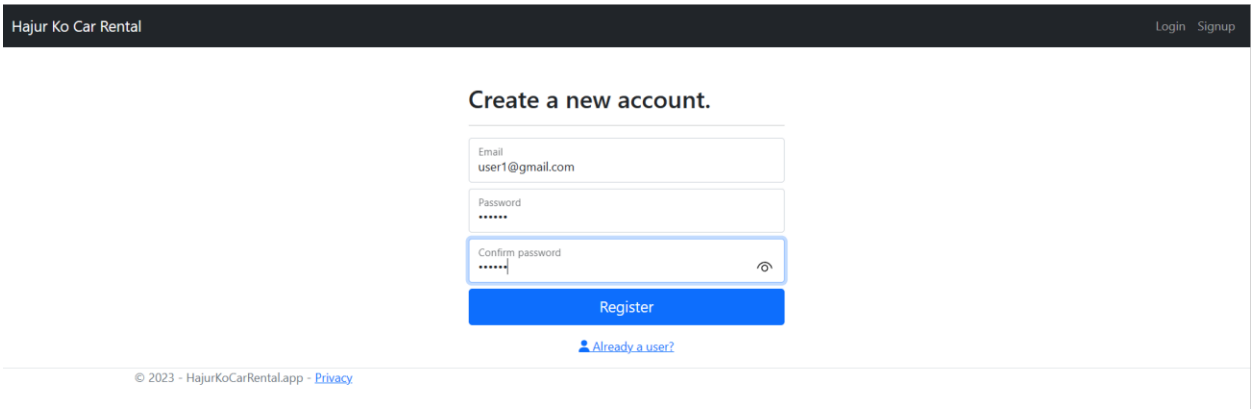

Figure 23 Register as a new user

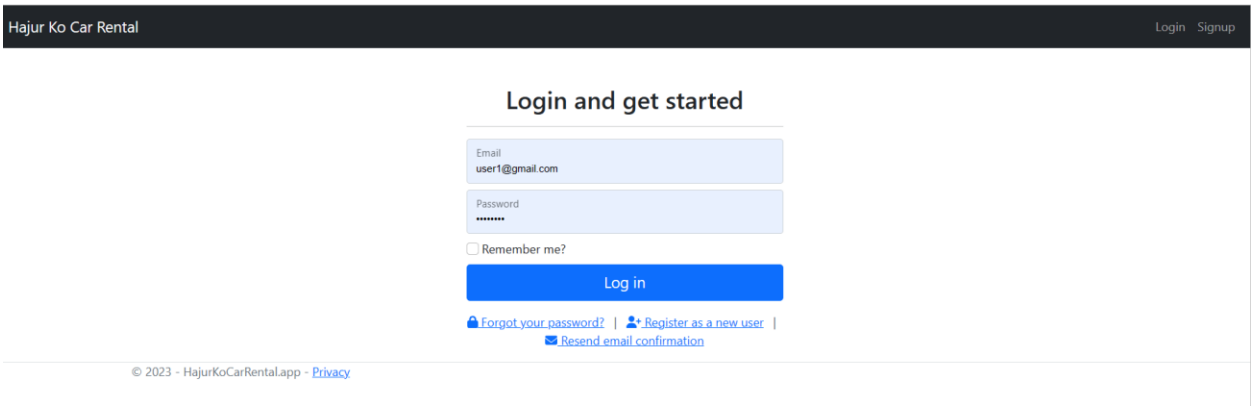

Figure 24 Login as a new user

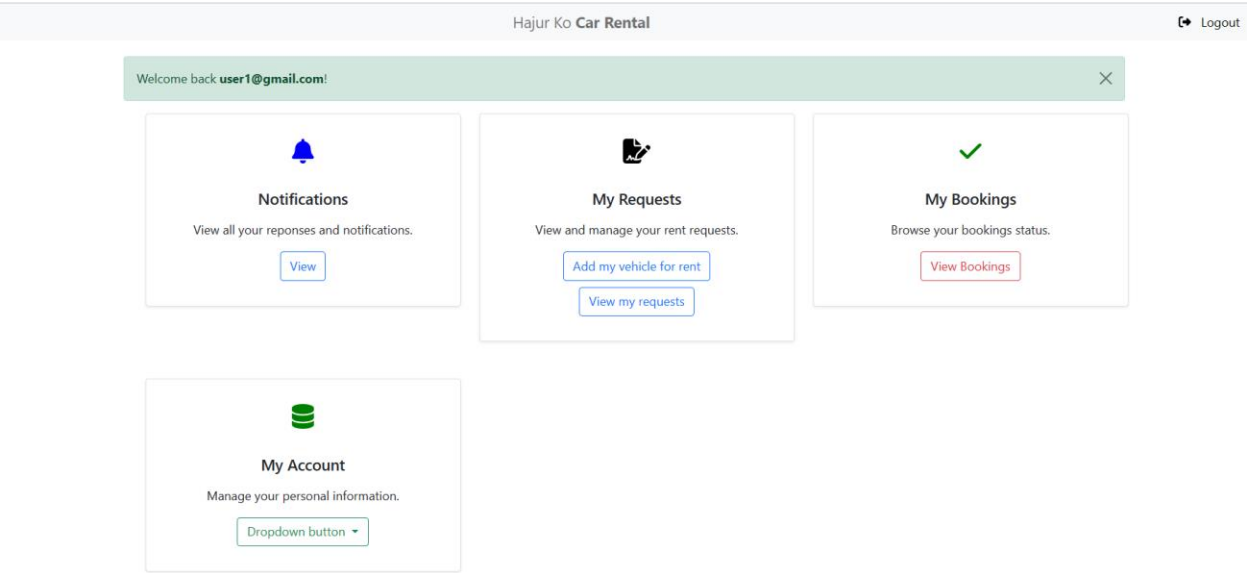

Figure 25 User Dashboard

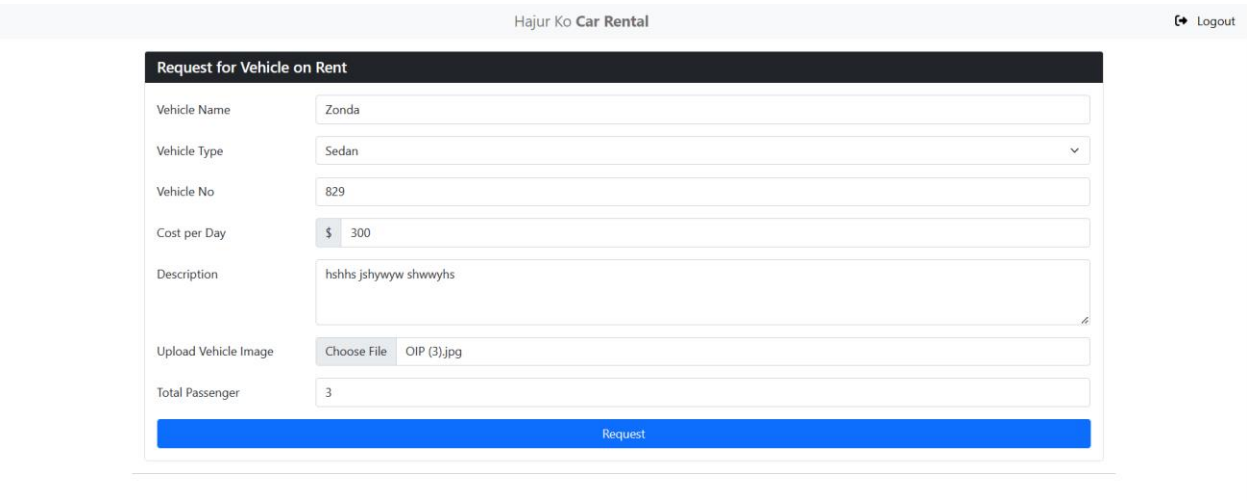

Figure 26 Requesting a vehicle for rent

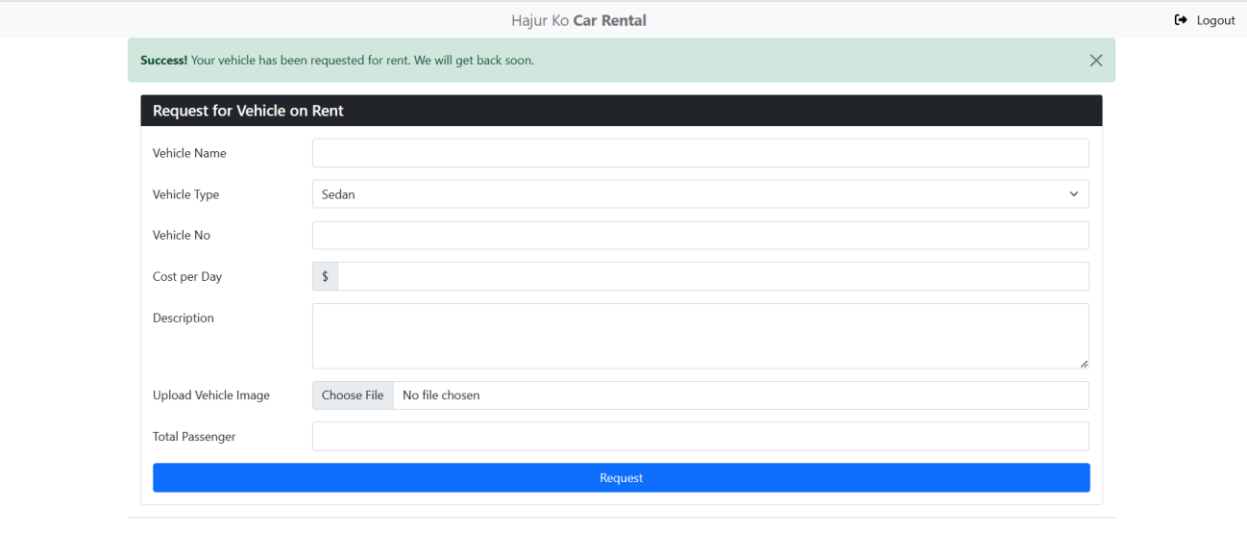

Figure 27 Vehicle Requested Successfully for Rent

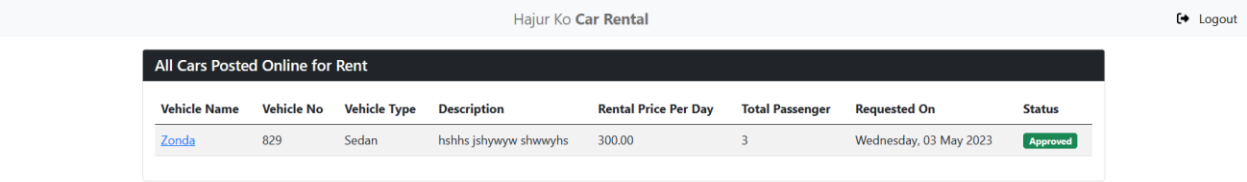

Figure 28 List of approved requested Vehicle

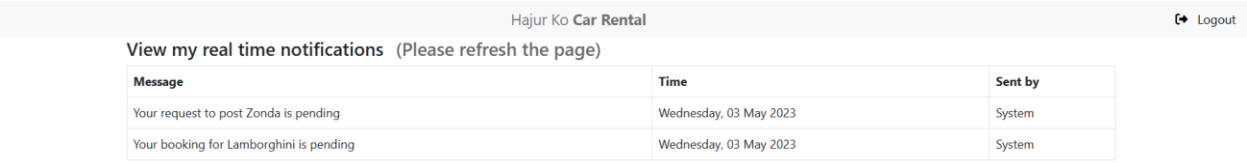

Figure 29 Pending request for renting cars

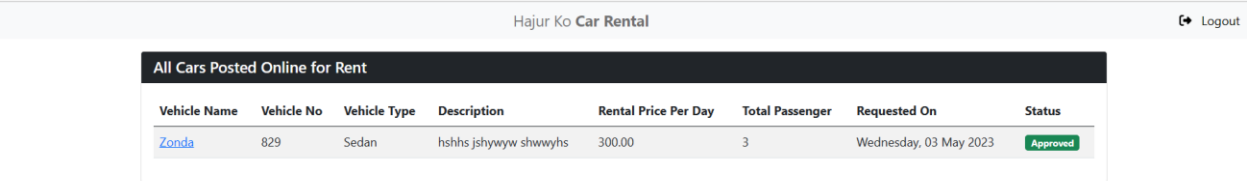

Figure 30 Rental request approved by the staff

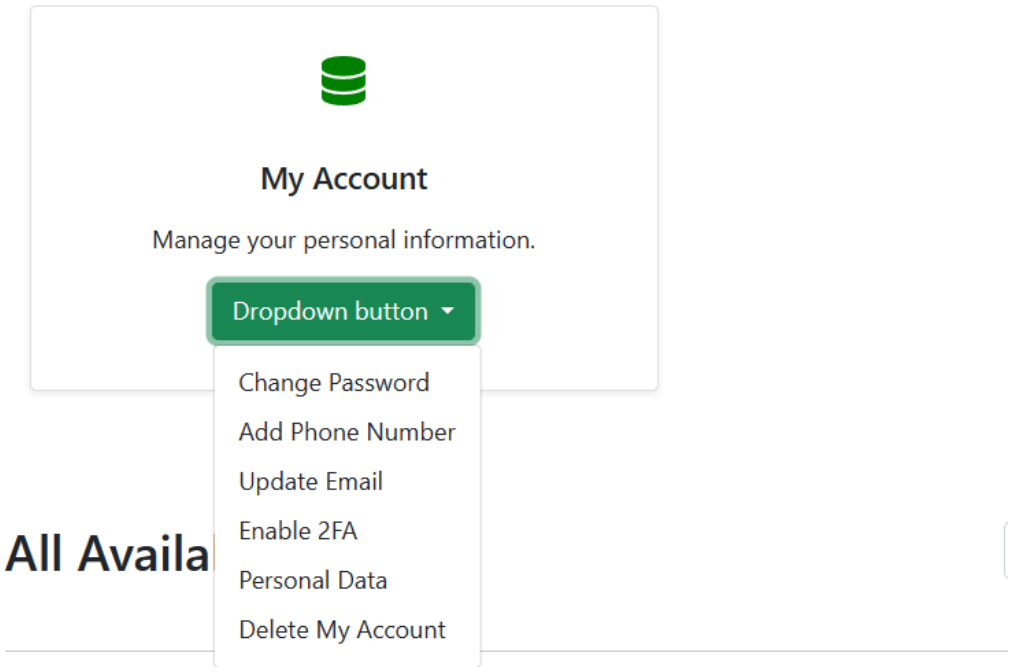

Figure 31 Account Settings

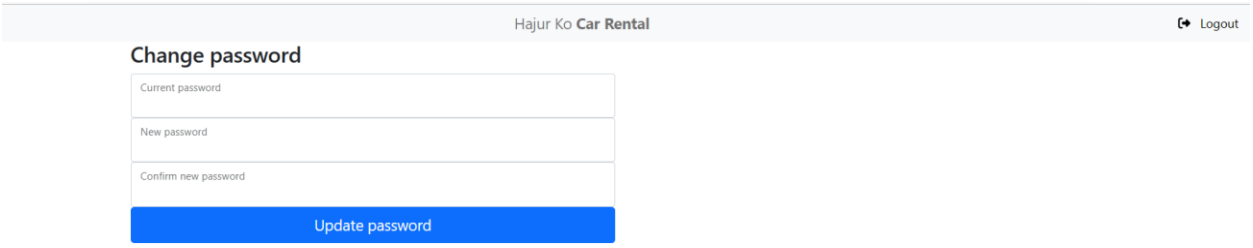

Figure 32 Change Password

3. Description of Solution Design

3.1. For Admin

- The admin can login using username and password as credential.
- The admin can add new staff members.
- The admin can assign the roles of the staff members.
- Admin can add rates for the rental cars.
- Admin can add or remove cars for renting in the car rental system.
- Admin can publish offers.
- Admin can validate client request.
- Admin can change their login password.
- Admin can view cars status if it is on rent or currently available.
- Admin should be able to provide a charge for repair bill to the customer.
- Admin should be able to provide 10% discount for regular customers.

3.2. For Staff

- Staff can add or remove cars for renting in the car rental system.
- Staff can publish offers.
- Staff can validate client request.
- Staff can change their login password.
- Staff can view cars status if it is on rent or currently available.
- Staff should be able to provide a charge for repair bill to the customer.
- Staff should be able to provide 10% discount for regular customers.

3.3. For User

- Customer can register using credentials.
- Customer can login using registered credentials.
- Customer can change their password.
- Customers can view the cars for rental.
- Customers can initiate rental request.

- Customers can be notified after request approval and about special offers.
- Customer can pay for rental and damage charge

4. System Architecture

This system was developed using the ASP.NET core MVC framework, often known as Model View Controller. The Model-View-Controller (MVC) framework is an architectural/design pattern that separates an application into three main logical components Model, View, and Controller. Each architectural component was built to handle specific development aspects of the application. The essential component of an application is the controller for the reason that it allows communication between the views and the model, serving as a middleman. It processes all the business logic and incoming requests, manipulate data using model component and interact with the view to render the final output. Similarly, the view component was used for all the UI logic of the application. The Bootstrap portion of an application are referred to as the View. Lastly, the model component of an application was responsible for managing the application's data and business logic. It interacts with a database or other data storage system to retrieve and store data (geeksforgeeks, 2023).

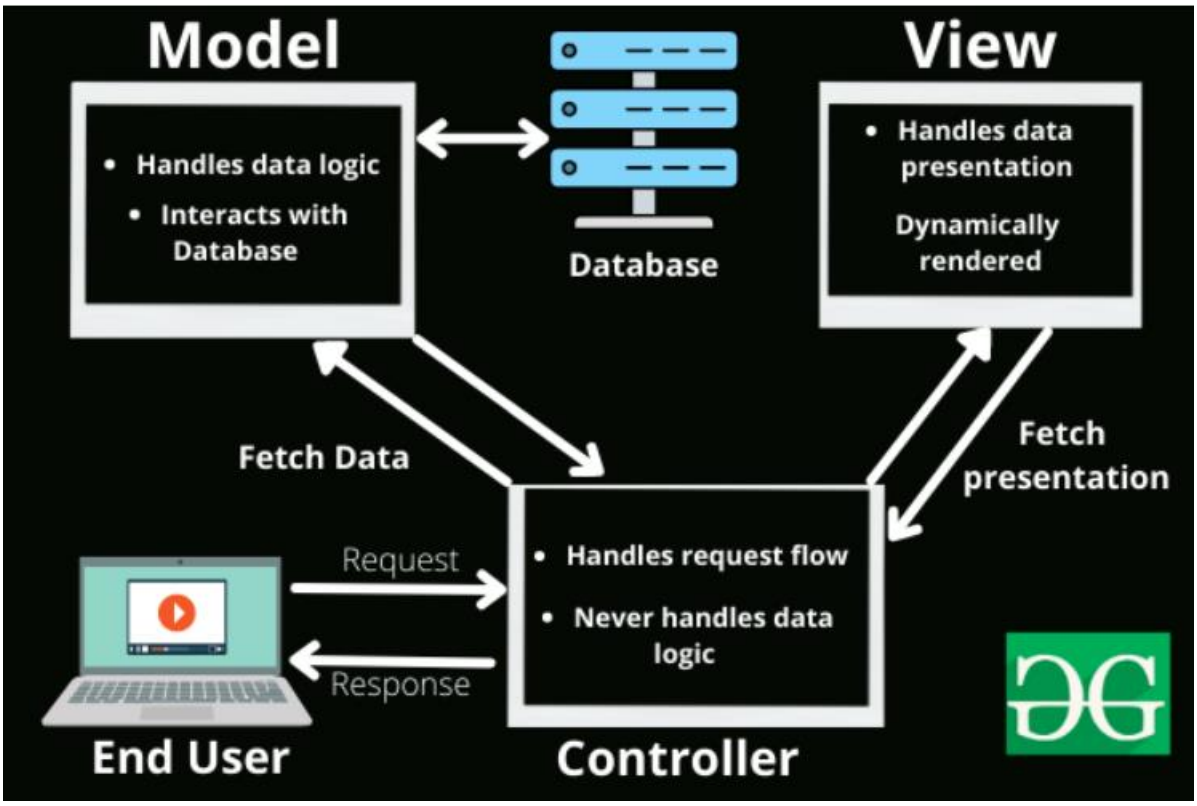

Figure 33 MVC Architecture Design

Class	Purpose
AddCarsOnlineController	This class represents a controller for adding and managing cars online. It contains methods for adding, updating, and deleting cars, as well as viewing a list of all available cars.
AddStaffController	This is a controller class that handles requests related to adding, viewing, updating, and deleting staff members in the Hamro Car Rental application
CarsOnRentController	This class is a controller class that handles the data of the rented cars.
DashboardController	This is a controller class that handles requests related to the Dashboard and provides methods for displaying the dashboard page and retrieving all available vehicles.
HomeController	This class is a controller class that handles the view of the home page with a list of available cars and it retrieves the list of available cars from the database
NotificationController	This class is a controller class that handles the notification for rental request and approval request retrieving data from the database.
VehicleRequestController	This class is a controller class that handles the Vehicle Request feature of the application. It retrieves and updates the data related to vehicle requests and updates the notifications accordingly.

Table 1 Class Table

5. Architecture Diagram of the System

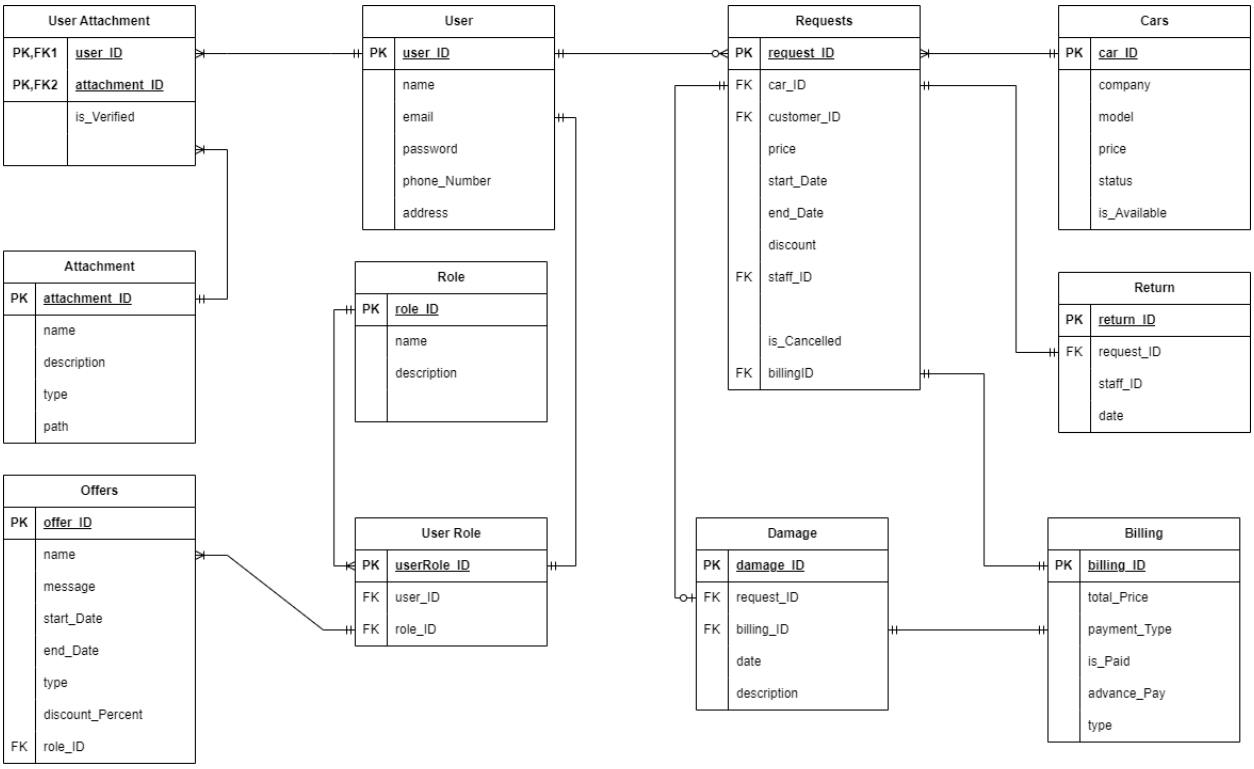

Figure 34 Entity Relation Diagram

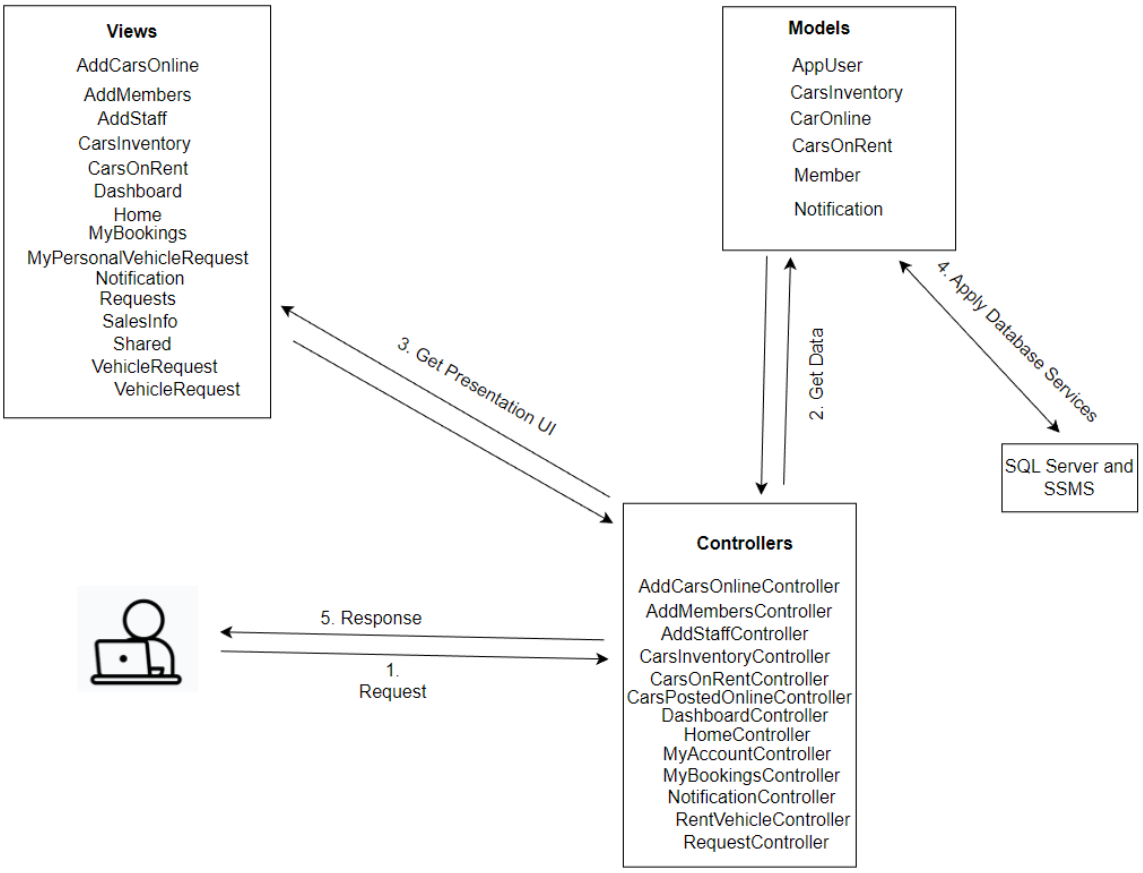

Figure 35 Architecture of the system

6. Detailed Description of the Classes' Properties and Method

The different classes of the system are described below with their respective methods and properties:

6.1. AddCarsOnlineController

Properties:

_applicationDbContext: An instance of the ApplicationDbContext class, which is used to access the database and perform CRUD operations on the CarsOnline and CarsOnlineInventory entities.

Methods:

Index(): A GET method that retrieves a list of all cars online from the database using the _applicationDbContext instance and returns the corresponding view.

Add(): A GET method that returns the view for adding a new car online.

Add(data: CarsOnlineVM): A POST method that receives data for a new car from the view and adds it to the database using the _applicationDbContext instance. It also adds the new car to the CarsOnlineInventory table.

View(id: int): A GET method that retrieves a car with a specific ID from the database using the _applicationDbContext instance and returns the corresponding view.

Update(id: int): A GET method that retrieves a car with a specific ID from the database using the _applicationDbContext instance and returns the view for updating the car's information.

Update(id: int, data: CarsOnline): A POST method that receives updated data for a car from the view, updates the car in the database using the _applicationDbContext instance, and returns the corresponding view.

Delete(id: int): A POST method that receives the ID of a car to be deleted from the view, deletes the car from the database using the _applicationDbContext instance, and returns a redirect to the index page.

6.2. AddStaffController

Properties

- **private readonly ApplicationDbContext _applicationDbContext:** A reference to an instance of the ApplicationDbContext class that is used to interact with the database.

- **private readonly UserManager<IdentityUser> _userManager:** A reference to instance of the UserManager class that is used for user authentication and authorization.

Methods

public async Task<IActionResult> Index(): This method handles the GET request to view the list of all staff members. It retrieves a list of staff members from the database and returns a view that displays the list.

public IActionResult Add(): This method handles the GET request to add a new staff member. It simply returns a view with a form to add a new staff member.

public async Task<IActionResult> Add(StaffVM data): This method handles the POST request to add a new staff member. It creates a new Staff object with the data provided by the user, adds it to the database, creates a new IdentityUser object with the user's email and password, and adds it to the user database. It also adds a new UserRole object to the database that associates the new user with the "Hamro Car Rental's Staff" role.

public async Task<IActionResult> View(int id): This method handles the GET request to view the details of a particular staff member. It retrieves the staff member with the given ID from the database and returns a view that displays the staff member's details.

public async Task<IActionResult> Update(int id): This method handles the GET request to update the details of a particular staff member. It retrieves the staff member with the given ID from the database and returns a view with a form to update the staff member's details.

public async Task<IActionResult> Update(int id, Staff data): This method handles the POST request to update the details of a particular staff member. It retrieves the staff member with the given ID from the database, updates its properties with the new data provided by the user, and saves the changes to the database. It then returns a view that displays the updated staff member's details.

public async Task<IActionResult> Delete(int id): This method handles the POST request to delete a particular staff member. It retrieves the staff member with the given ID from the database, removes it from the database, and saves the changes. It then redirects the user to the list of all staff members.

Continue Reading…

7. Individual Reflection

A total of five individuals were part of our group for this group project, and each participant had a distinct task to do, such as the front end, back end, report of the program, etc.

The Individual Reflection of each team members has been kept in the appendix section i.e., Individual Reflection.

8. Conclusion

This group coursework was relegated to us for creating a management system for Hajur Ko Car Rental to manage their business utilizing the C# programming language on the ASP.NET platform using the built-in Visual Studio Environment. First and foremost, this assignment was given to us, and in order to finish it on time, we were required to work in groups.

Continue Reading…

Despite the fact that it was challenging and troublesome from the outset, the coursework was finished on schedule and submitted on time as well. We had the chance to learn numerous new things and it assisted us in gaining a deeper understanding of the module and how back-end development functions, which we knew about. It was a decent encounter to foster a program utilizing Visual Studio 2022 and it was enjoyable to chip away at this project as well.

9. References

geeksforgeeks, 2023. *MVC Framework Introduction.* [Online] Available at: https://www.geeksforgeeks.org/mvc-framework-introduction/ [Accessed 30 April 2023].

10. Appendix

10.1. Introduction

The created system will be a web application with capabilities like login and registration for different users, such as Admin, Customer, and Staff. The created system will give users the opportunity to carry out a number of tasks, including adding and removing automobiles from the system's list of available rentals, publishing offers, authenticating clients' requests, changing passwords, renting a car to a member and keeping account of when that car is returned as well as keeping tabs on any damages to the vehicle. The system additionally enables the display of sales with a data filtering option based on customer names, viewing client rental history, and authorizing by a specific staff member's name. Additionally, it enables you to see the cars that are currently available and those that are being rented. The system will allow you to see the automobiles that have regularly been hired and the ones that haven't. It also keeps track of clients who commonly rent cars and those who haven't rented anything in a while (3 months). The technology also allows customers to register for accounts via which they may submit rental requests and update their passwords as needed. Similar to this, Hajur Ko Car Rental can benefit from a variety of operations and features to help run their business successfully.

10.2. Description of the classes, properties and methods

10.2.1. CarsOnRentController

Properties:

- **_applicationDbContext**: an instance of the ApplicationDbContext class for accessing the data source.
- **_userManager:** an instance of the UserManager class for managing user information.

Methods:

- **CarsOnRentController():** a constructor method that initializes the _applicationDbContext and _userManager properties.
- **Index():** a HTTP GET method that returns the view for displaying a list of cars available for rent.
- **Staff():** a HTTP GET method that returns the view for displaying a list of cars available for staff to manage.
- **Update(int id, [FromForm] MyBookedVehicle data):** a HTTP POST method that updates the information of the car that has been rented by a user with the provided id.
- **Returned(int id):** a HTTP POST method that handles the return of a rented car with the provided id. It removes the car from the list of rented cars, creates a notification, and returns the view for displaying a list of available cars.

10.2.2. DashboardController

Properties:

- **_applicationDbContext:** A private readonly ApplicationDbContext object that represents the database context.
- **_userManager:** A private readonly UserManager<IdentityUser> object that manages user-related operations.

Methods:

- **Index():** A method that returns the view of the dashboard page to the system.
- **AllAvailableVehicle():** A method that retrieves all available vehicles from the database and returns them as a list to the view. This method is decorated with the [HttpGet] attribute, indicating that it handles GET requests. It uses the _applicationDbContext object to retrieve data from the database and the ToListAsync() method to asynchronously convert the data into a list. The method then returns the list to the view using the View() method.

10.2.3. HomeController

Properties:

- **_logger:** an instance of the ILogger interface that is used for logging messages.
- **_applicationDbContext:** an instance of the ApplicationDbContext class that is used for accessing the database.
- **_userManager:** an instance of the UserManager<IdentityUser> class that is used for managing user authentication and authorization.

Methods:

- **Index():** a method that returns the view of the home page with a list of available cars. It retrieves the list of available cars from the database using the _applicationDbContext instance.
- **Privacy():** a method that returns the view of the privacy policy page.
- **Error():** a method that returns the view of the error page with an ErrorViewModel object containing information about the error. It uses the _logger instance to log the error message.
-

10.2.4.NotificationController

Properties:

- **_applicationDbContext:** an instance of the ApplicationDbContext class representing the database context.
- **_userManager:** an instance of the UserManager<IdentityUser> class representing the user manager for handling user-related operations.

Methods:

- **NotificationController:** Constructor method that initializes the _applicationDbContext and _userManager properties with the corresponding parameter values passed to it.
- **Index:** A HttpGet method that retrieves all the notifications from the database in descending order based on the Id and returns a view containing the notifications.

10.2.5. VehicleRequestController

Properties:

- **private readonly ApplicationDbContext _applicationDbContext:** an instance of the ApplicationDbContext class which is used to interact with the database.
- **private readonly UserManager<IdentityUser> _userManager:** an instance of the UserManager class which is used to manage users.

Methods:

- **public async Task<IActionResult> Index():** a method that handles HTTP GET requests for the Vehicle Request page. It retrieves the data for all the booked vehicles and returns the corresponding view.
- **public async Task<IActionResult> Update(int id**): a method that handles HTTP GET requests for the Update page for a specific booking. It retrieves the data for the booking with the specified id, and returns the corresponding view.
- **public async Task<IActionResult> Update(int id, [FromForm] MyBookedVehicle data):** a method that handles HTTP POST requests for the Update page for a specific booking. It updates the booking data with the submitted data and saves it to the database. If the vehicle booking status is approved, it creates a new entry in the CarsOnRent table and removes the vehicle from the CarsOnline table. It also creates a new notification for the customer and updates it to the database. The method then returns the corresponding view.

10.3. Individual Reflection

10.3.1. Punam Thapa Magar

As a member of the group project for the “Hajur Ko Car Rental”, my primary responsibility was to work on the backend and reporting parts of the system. Looking back on my work, I believe that I was able to make significant contributions of the project, and I learned a lot in the process. Working on the backend development part of the project was a challenging but rewarding experience. I spend a considerable amount of time designing the data models, setting up the database, and creating the APIs that would allow the frontend team to interact with the backend. One of the key challenges I faced was ensuring that the data models were efficient and well-designed, which required a deep understanding of the project requirements and careful consideration of the various relationships between different entities. In addition to working on the backend, I was also responsible for the reporting part of the system. This involved creating reports which includes all the detailed information about the system such as the detailed instruction to run the program, system architecture, testing and so on.

Despite these difficulties, I believe I was able to develop a reliable and effective backend that complied with the project's specifications. The ability to directly observe how my coding and design choices could affect the system's overall functionality was one of the aspects of this task that I most appreciated. I also valued the chance to collaborate closely with group members because it gave me the chance to pick up new abilities and methods from my peers.

Overall, I believe that my contributions to the backend development and documentation parts of the project played a critical role in the success of the car rental system. By working closely with the other members of the development team, I was able to create a robust and efficient backend that was well-documented and easy to use, ultimately resulting in a successful project outcome.

10.3.2. Poonam Singh Yadav

As a team member, I begin by discussing our role in the project and our responsibilities for the backend and report part of the car rental system. My tasks included designing and implementing the backend architecture, as well as creating the report for the project. Working on the backend and report parts was a challenging yet rewarding experience for me. I encountered several obstacles, such as debugging errors and integrating different technologies and libraries, but I was able to overcome them with the help of my team members. To complete the project, I used various tools and techniques, including C# programming language, SSMS, Microsoft SQL server, visual studio 2022.

Throughout this project, I was to improve my programming skills, gain a deeper understanding of the backend architecture and report development process, and learn how to collaborate effectively with other team members. Overall, my contribution to the project was valuable in helping the team achieve its objectives. Through working on the backend and report parts, I feel more prepared for future projects and career opportunities, such as working as a full-stack developer. Looking back, I could have improved by being more proactive in seeking feedback and asking questions. For future projects, I plan to be more open to constructive criticism and communicate more effectively with my team members.

10.3.3. Aashutosh Sharma

To begin, I started by comprehending the project requirements and the expected features of the system. This involved engaging in discussions with the project leader and my team members to ensure that I had a clear understanding of the project scope. After understanding the requirements, I commenced working on the frontend and backend aspect of the system. This entailed designing the user interface, generating the essential forms, and integrating the mandatory functionality utilizing C#.

During the project, I encountered several challenges, such as arranging the various frontend elements, addressing errors and misunderstandings, and collaborating with team members. Nevertheless, I managed to overcome these challenges by relying on my experience and communicating effectively with the team. Throughout the project, I acquired valuable knowledge and skills in C# programming and frontend development. Furthermore, my collaborative and communication abilities improved, which are essential in team settings.

Overall, successfully completing the frontend and backend section of the car rental system project demonstrated my C# programming expertise and my capability to work collaboratively with others towards a shared objective.

10.3.4. Samita Sunuwar

To start with, I first observed the project, its specifications, and its features, which include discussions with the team leader and in groups, and I came away with a clear grasp of the project. My work on the front-end of the system, which consists of design, forms, and integrating functionality in C# with the aid of HTML, CSS, Javascript, and Bootstrap, began as soon as I had a firm grasp of the needs and features.

The highly typed nature of C#, which can be difficult for developers accustomed to dynamically typed languages to learn, was one of several difficulties I encountered while working on this project. Furthermore, I found it frustrating that the setup for the development environment could be difficult. However, I was able to overcome all of these difficulties in the end. There were some miscommunications and collaborations with group members.

As a result, creating the front end of a web application for the vehicle rental system in C# was satisfying and had numerous benefits, such as the abundance of libraries and frameworks that were readily accessible, a robust type system, and built-in support for data access, security, and authentication. Although C# and ASP.NET may have a learning curve, the benefits of creating reliable and scalable web applications make the effort worthwhile.

10.3.5. Girish Shrestha

Work had to be done for the application development project using a variety of tools. It wasn't an easy task, but overall, it was a rewarding and fruitful experience. The abilities and knowledge gained during the procedure will undoubtedly be useful for upcoming initiatives. To identify the roles, responsibilities, and tasks assigned to each member, numerous meetings were held. Our group leader did the finest job of guiding us through the project.

In the same way, I developed experience in the C# programming language and.Net technologies. The front-end's usage and element organization presented the most challenges. But with the help of the members, getting around mistakes and misunderstandings became simpler. It was possible to learn a lot from each person. My capacity for collaboration and communication with other team members was put to the test. Significant gains were observed in terms of knowledge, experience, confidence, and collaborative abilities. All tasks were completed successfully. Despite all the challenges, the outcome was satisfying.

10.4. Conclusion

The requirements needed to finish this project were initially finalized by the group members. The task was then divided into multiple components, with an equal amount of each component going to each team member. For our team to be able to leverage the .Net technology and the C# programming language by integrating the numerous stated features in to the web application, it took some time for our team members to become familiar with both of these technologies. While doing this coursework, we needed to confront the challenges, particularly in the backend section which was a big deal as we had to built the whole features of the system ourselves as numerous mistakes occurred which was also a lot of work. Yet in order conquer the confusion and troubles, a lot of exploration was finished with respect to the applicable sections. Cent percent exertion was given to finish the undertaking allocated in this topic. Regular interaction with instructors, consistent exertion, and a lot of exploration, going through the lecture slides and surfing the web-assisted with sound information was done to solve the problems. Because this was a group project, our team members worked together to solve any issues that arose during the system's development. We likewise, became more acquainted with various terms which were utilized in the program, we knew about various terms alongside their capabilities and it was extremely beneficial for us and greatly improved our understanding of .Net technology and C# programming language.