

**Centre Name: ACE-THUDUC-1-FPT.**

**Address: 62 Street 36, Van Phuc Residential Area, Hiep Binh Phuoc Ward,Thu Duc City**

Easy Car

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | *Mr. Pham Cong Danh* | |
| **Batch.** | *T5.2303.E0* | |
| **Group** | *Group 1* | |
| **Serial No** | **Enrollment Number** | **Student Name** |
| *1.* | *Student1468208* | *Nguyễn Quốc Thiện* |
| *2.* | *Student1468189* | *Phan Đặng Hoàng Tiến* |
| *3.* | *Student1468213* | *Vũ Quốc Việt* |
| *4.* | *Student1237676* | *Trần Thanh Hải* |

***Month****: 01* ***Year****: 2024*



This is to certify that

|  |  |
| --- | --- |
| **Mr.** | **NGUYỄN QUỐC THIỆN** |
| **Mr.** | **PHAN ĐẶNG HOÀNG TIẾN** |
| **Mr.** | **VŨ QUỐC VIỆT** |
| **Mr.** | **TRẦN THANH HẢI** |

Have successfully Designed & Developed

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **EASYCAR** | | |

Submitted by:

|  |
| --- |
| **Mr. PHAM CONG DANH** |

Date Of Issue:

|  |  |
| --- | --- |
|  |  |

Authorized Signature:

|  |  |
| --- | --- |
|  |  |

|  |
| --- |
| **Content** |

Content

Acknowledge

Synposis

Review1 1

Review2 10

Review3 49

|  |
| --- |
| **Acknowledge** |

As we understand that the eProject is a step-by- step learning environment that closely simulates the class-room and Lab based learning environment into actual implementation. It is a projectimplementation at your fingertips!! An electronic, live juncture on the machine that allows you to.

* Practice step by step i.e. laddered approach.
* Build a larger more robust application.
* Usage of certain utilities in applications designed by user.
* Single program to unified code leading to a complete application.
* Learn implementation of concepts in a phased manner.
* Enhance skills and add value.
* Work on real life projects.
* Give a real life scenario and help to create applications more complicated and useful.
* Mentoring through email support.

We would like to send a great thank to our professor and others student for the addorable supports during the time in project.

Beside serveral mistakes we had made in the project, we hope to have more oppotunities to widen our knownledge in web developing world.

We also thank you so much for your infomation and the “demo” project from India APTECH to help us get more reality experience in this project. Since we have gained more experience from this project, we will not stop learning any thing as much as we can to be more successful in the future.

Your Sincerely,

Team Group 01.

|  |
| --- |
| **Synposis** |

The Objective of this program we aim is to give a sample project to work on real life projects. These applications help us build a larger more robust application.

The objective is not to teach us HTML/JavaScript but to provide us with a real life scenario and help us create basic applications using the tools.

Hence, we can revise the chapters before we start with the project.

This project is meant for students like us who have completed the module of HTML5 and PHP. These programs should be done in the Lab sessions with assistance of the faculty if required.

The EASYCAR website brings everything in the world to you such as historic monuments, information, natural knowledge or skill in technology...

It is very essential for us to have a clear understanding of the subject. We think we should go through the project and solve the assignments as per requirements given.

And get back eprojects@aptech.ac.in as the assigned schedule.

**REVIEW 1**

**Problem Defination**

**Easy Car** is an online website for rental of all types of cars. They have a collection of cars. Now they want to revamp their website with all the cars listed on the website. Consumers should find it easy to view various cars and decide upon booking the right cars.

A website needs to be designed to display various cars available with the store. Feedback must be included.

The website must allow website visitors to browse and search for car models using schedule or filter categories and attributes of Cars.

If customers want to book a car, they must register an account. A customer with an existing account can log in. They do not have the right to change any information other than their own.

Registered customers can leave reviews or questions about the product.

The website must allow administrators (business owners and their appointees) to view, update (when meeting requirements) existing products, categories, car attributes, registered customers , booked a car, customer information and feedback about the car booking.

If the store owner wants to change sensitive information (e.g. delete cars, ban customers, etc.), they must have the permission level “Admin”, otherwise they will not be able to access the functions This.Aside from managing product, categories, orders, etc. an admin can also view reports on product, orders and customers.

|  |
| --- |
| **Customer Requirements Specifications** |

1. **Customer’s Specific Requirements**

The website allows user register and login to become member, which can use all the function of the website like review car, rental car online, pay for order online.

1. **Functional Requirements**

There are 4 main actors:

* Visitor
* User
* Moderator
* Admin

1. Visitor
   1. Visitor can view website information

a.Description

Visitor can view the featured car in homepage, company history, car by category.

b.Functional requirement

Visitors can view website information like company history, contact, service… Visitors can view feature car in homepage,…

* 1. Visitor can register an account to be a user:

a.Description

Customers must be able to register a new account. Once they have their account, they can login and access more features. A guest, who is not logged in, can only browse products.

b.Functional requirement

User must provide the information below to create an account:

- Fullname

- Password

- Email (must be unique)

- Phone number

- Address

And some additional information.

* 1. Visitor can contact and send feedback to Easy Car

a.Description

Visitor can contact to Easy Car and send feedback to company

b.Functional requirement

Visitor can contact to Easy Car and send feedback to company, by phone, mail or feedback form in Contact Page

* 1. Visitor can search for car

a.Description

Visitor can search for car by their schedule, filter car by price, car type,..

b.Functional requirement

Visitor can search for car by their schedule, filter car by price, car type,..

* 1. Visitor can view car details

a.Description

Visitor can view the car details like name, brand, price,…

b.Functional requirement

Visitors can view car information such as title, description, price, review…

1. User

User can do all function like Visitor can, and other function:

* 1. User can login:

a. Description:

After registering an account, users can log in. Once logged in, they can modify their personal information, including their password. They can also access more features: service reviews, booking car rentals in the shopping cart.

b. Functional Requirement:

Users must have an existing account and not be a banned account. Users must provide their Email and Password to log in.

* 1. User can change profile and password:

a. Description:

User can change their profile info like name, date of birth,… and recover password

b. Functional Requirement:

User can change their profile info like name, date of birth,.. when they logged in. User also can change to new password

* 1. User can rent a car:

a. Description:

Users can find the car they want by searching or filtering results. Users can book a car appointment on the booking bar and proceed to checkout. They will be asked to provide billing and payment information.

b. Functional Requirement:

User must log in to use this function. User can only rent a car that are still available on their schedule. User needs to provide the information below to complete their order placement. User can checkout and pay for order by using e-wallet .

* 1. User can write review on car:

a. Description:

User can leave their comment and review on each car.

b. Functional Requirement:

User must login to use the review function. They must provide:

- Review content

* 1. User can view history order:

a. Description:

User view history order car.

b. Functional Requirement:

User can view history rental car, or status of their rental order in account information.

1. Moderator (Mod)

Mod must login to do these function:

* 1. Mod Can Manage Orders (Rental Car):

a. Description:

The site must allow Mod to interact with Orders. When a customer places an order, it will show up in the list of orders and these orders’ details..

b. Functional Requirement:

An Order must contain: ID , Car’ID, Account ID, pickup and return date. Mod can view a list of existing orders and see their info. Mod can change or cancel the order.

* 1. Mod can manage User

a. Description:

The site must allow Mod to interact with User’s Accounts. When a customer has registered an user account, they will show up in the User list. Mod can view a list of existing registered users. Admin can ban a user who violates site’s rules by delete user

b. Functional Requirement:

- Must login as Moderator to ban users

- Mod can search customers by Username and Emails

- Mod can ban(delete) user

* 1. Mod can manage Comment/Feedback

a. Description:

The site must allow Mod to interact with User’s Comment.As a Mod, they can hide violent comment by user, or reply

b. Functional Requirement:

- Must login as Moderator to hide comment, or reply user comment.

1. Admin

Admin can do all function of Moderator and these functions:

* 1. Admin Can Manage Website Content

a. Description:

Admin can manage website content such as advertising carousels,blogs. Admin can add or modify blogs, ads. Admin also can disable them out of website.

b. Functional Requirement:

- Must login as an admin to manage website content

- Admin provide content of blogs, ads, images,..

* 1. Admin can manage Car:

a. Description:

The site must allow admins to interact with car. An admin can view a list of existing cars. Admin can add a new car and modify car’s name, description, status, category,... Admin can delete car. Admin can view car’s details when click on the car.

b. Functional Requirement:

To add or modify a car, admin must provide: Car name , Type, Brand, Description, Color, Price , Image,...

Other requirements: Must login as an admin to add or modify car, Names are unique , Category must exist, Must be an Admin to delete product, Can only delete product which do not belong to user order, Admin can search product by titles, Admin can filter by categories.

* 1. Admin can manage Car Type:

a. Description:

The site must allow admins to interact with Car type. An admin can view a list of existing car type, insert a new type, and modify the type's name. When an Admin, they can delete car type not contain any existing car. Car type with no car is an empty type.

b. Functional Requirement:

To add or modify a car type, admin must provide: Type’s ID, Type’s Name

Other requirements: Must login as an admin to add or modify type, Names are unique, Type's ID must exist, Must be an Admin to delete type, Can only delete empty category.

## System Requirements:

**Hardware / Software requirement**

1. **Hardware**:

**Web Server**

* Processor Intel Core I3 or higher.
* Memory 6 GB RAM or greater.
* Modem/ADSL Internet access is required.

**Client**

* Processor Intel Core I3 or higher.
* Memory 6 GB RAM or greater.
* Monitor Super VGA (1024x768) or higher resolution.
* Modem/ADSL Internet access is required.

1. **Software**:

**Server:**

* Web server: XAMPP
* Operation System: Window 7 or later.
* Databases: MySQL.
* PHP - Laravel framework
* Browser: Google Chrome version 35.

**Client:**

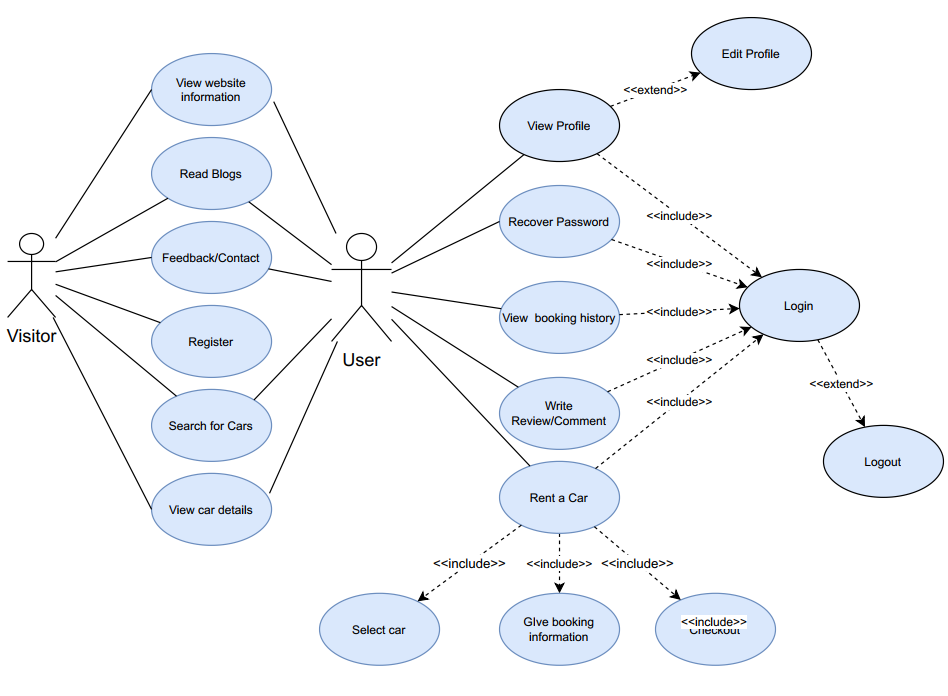
* Operation System Window 7 or higher.
* Browser Google Chrome , MS-Edge, Firefox

REVIEW 1 - TASK SHEET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. | Project Title:  Game Store | | Date of Preparation of Activity Plan:  25/12/2024 | | |
| No. | Task | Start Date | Actual Days | Member | Status |
| 01 | Acknowledgment | 25/12/23 | 01/01/24 | -Phan Đặng Hoàng Tiến | Completed |
| 02 | Synopsis | - Phan Đặng Hoàng Tiến | Completed |
| 03 | Customer Requirement Specification | - Nguyễn Quốc Thiện | Completed |
| 04 | Functional Requirement Specification | - Trần Thanh Hải | Completed |
| 05 | System Requirements | - Vũ Quốc Việt | Completed |
| 06 | Development Software | - Vũ Quốc Việt | Completed |

REVIEW 2

## Use Case Diagram 1



**I. VISITOR and USER**

**1. Visitor**

**1.1 View website information**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | View website information | |
| **Actors** | Visitor/User |  |
| **Description** | View information about our company | |
| **Requirements** |  |  |
| **Status** | Show information page | |
| **Basic flow** | Actor action:  1. Actor click “About us” on the Navigation Bar | System Response:  2. The system will direct to the About Us Page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.2 Read blogs**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Read blogs | |
| **Actors** | Visitor/User |  |
| **Description** | Read blogs about car experiences, tips for trips,… | |
| **Requirements** |  |  |
| **Status** | Show blogs page | |
| **Basic flow** | Actor action:  1. Actor clicks “Blogs” on the Navigation Bar.  2. Actor clicks on the blog’s title to read. | System Response:  2. The system will redirect to the Blogs Page showing a list of blogs.  4. The system will redirect to the blog content page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.3 Register an account**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Register an account | |
| **Actors** | Visitor/User |  |
| **Description** | Visitor want to register to become an user of website | |
| **Requirements** |  |  |
| **Status** | Success: The account will be created  Fail: Refill information | |
| **Basic flow** | Actor action:  1. Actor click “Register” link.  3. Actor inputs register’s information and click ‘Register’ button. | System Response:  2. System redirects to Register page and displays Registration form with the following controls:  - ‘Full Name text field  - ‘Password’ text field  - ‘Confirm Password’ text field  - ‘Email’ text field  - ‘Phone’ text field  - ‘Address’ text field  - ‘Date of Birth' field  - ‘Photo’ field  - ‘Register’ button  - ‘Reset’ button  4. System validates the information.  5. System inserts the account into database  6. System |

**1.4 Search cars**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Search cars | |
| **Actors** | Visitor/User |  |
| **Description** | Visitor want to search for available cars on their schedule. | |
| **Requirements** |  |  |
| **Status** | Success: Show all cars that match with input schedule.  Fail: No cars is show. | |
| **Basic flow** | Actor action:  1. Actor input ‘pick-up date’, ‘return date’ and ‘car type’ (optional) then click ‘Search’ button. | System Response:  2. System redirects to Cars list Page, show available cars. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.5 View car details**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | View car details | |
| **Actors** | Visitor/User |  |
| **Description** | Actor want to view car details. | |
| **Requirements** |  |  |
| **Status** | Show car detail information. | |
| **Basic flow** | Actor action:  1. Actor click on ‘View Details’ button. | System Response:  2. System redirects to Cars Details Page, showing Car detail information. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2. User**

**2.1 Login**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Login | |
| **Actors** | User |  |
| **Description** | User who has registered an account can login. | |
| **Requirements** |  | |
| **Status** | Success: User is logged in to website  Fail: Refill information. | |
| **Basic flow** | Actor action:  1. Actor clicks on ‘Login’ button.  3. Actor inputs Email and Password, then click the ‘Login’ button. | System Response:  2. System redirects to Login Page with the following controls:  - 'Email' text field  - 'Password' text field  - 'Login' button  4. System check the email and password.  5. System redirects to Homepage. |
| **Alternative flow** | 1. Actor click “Forget Password” button.  3. Actor provide email and click “Send mail”.  5. Actor use new password to login. | 2. System redirect to Forget password page.  4. System check user’s email in database. If correct, send new password to user’s email. Otherwise, return message “Email not found”. |
| **Exception** | 1. Actor inputs invalid email and password. | System redirects to Login page with message: “Invalid email or password’ |

**2.2 Edit profile**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Edit profile | |
| **Actors** | User |  |
| **Description** | User who has logged in can edit profile | |
| **Requirements** |  |  |
| **Status** | Success: Actor can see and change their info  Fail: Refill information | |
| **Basic flow** | Actor action:  1. Actor click on “My Account” button.  3. Actor click on “Edit profile’ button.  5. Actor edit their profile, then click Save | System Response:  2. System redirects to Account page.  4. System show Edit profile form with following control:  - ‘Full Name’ text field  - ‘Email’ text field  - ‘Phone’ text field  - ‘Address’ text field  - ‘Date of Birth' field  - ‘Photo’ field  6. System validates the information.  7. System update account database and show update success message . |

**2.3 Log out**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Log out | |
| **Actors** | User |  |
| **Description** | User who has logged in can log out. | |
| **Requirements** |  | |
| **Status** | User can log out website | |
| **Basic flow** | Actor action:  1. Actor clicks on ‘Log out’ button. | System Response:  2. System delete user session data and redirect to Homepage. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.4 View rental history**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | View rental history | |
| **Actors** | User |  |
| **Description** | User want to view rental history | |
| **Requirements** | User who has logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. User click on “Rental History” button | System Response:  2. System redirect to Rental history page, show rental history information. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.5 Recover password**

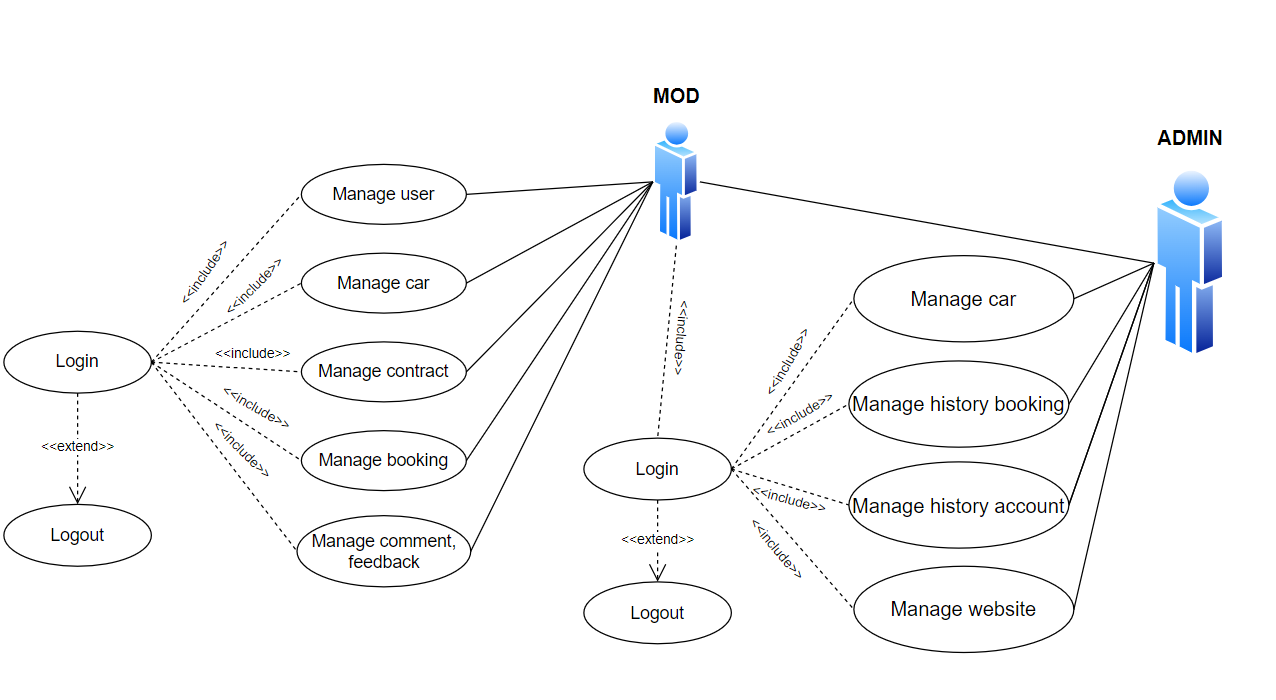
|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Edit profile | |
| **Actors** | User |  |
| **Description** | User who has logged in can recover password | |
| **Requirements** |  | |
| **Status** | Success: Actor can see and change their info  Fail: Refill information | |
| **Basic flow** | Actor action:  1. Actor click on “My Account” button.  3. Actor click on “Change Password” button.  5. Actor give current password and new password. | System Response:  2. System redirects to Account page.  4. System show ‘Change password’ form with following control:  - “Current password” field  -‘New password” field  6. System validates the password.  7. System update account database and show update success message. |
| **Alternative flow** |  |  |
| **Exception** | Actor action:  1. Input current password don’t match with the password of account.  2. ‘New password’ field left blank or incorrect format | System displays message to actor:  1.Current password is incorrect.  2. New password cannot be blank or incorrect format. |

|  |  |  |
| --- | --- | --- |
|  | **Prepared by (Student)** | **Approved by (Faculty)** |
|  | **Project Group No**: 01 | **PHAM CONG DANH** |
| **Signature** |  |  |
| **Date** |  |  |

**2.6 Rent a car**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use Case Name** | Rent a car | |
| **Actors** | User |  |
| **Description** | User want to rent a car from website | |
| **Requirements** | User who has logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. User click on “Rent car” button  3. User can change rental info, additional info like “Customer name”,“note”,… Then click “Checkout” button  5. User pay for rental. | System Response:  2. System redirect to Rent Car page with rental information (car, schedule,..) when search car.  4. System save rental data to session, redirect to Checkout page, show payment method, generate qrcode for pay.  6. System save rental data to database, show success message and redirect to Home page |
| **Alternative flow** |  |  |
| **Exception** |  |  |

## Use Case Diagram 2



**II. ADMIN and MODERATOR**

**1. Admin**

**1.1 Admin login**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Log in | |
| **Actors** | Admin | |
| **Description** | Admin can login to admin dashboard | |
| **Requirements** | Admin provide email and password | |
| **Status** | Success: show login message successful.  Fail: show message failed. | |
| **Basic flow** | Actor action:  1. Actor type email and password, then click “Login” button. | System response:  2. System validate information. If correct, redirect to system dashboard. Otherwise, show failed login message. |
| **Alternative flow** |  |  |
| **Exception** | - Actor type invalid email and password. | System show message:  - “Wrong email or password” |

**1.2 Admin logout**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Log out | |
| **Actors** | Admin | |
| **Description** | Admin can log out of the admin dashboard | |
| **Requirements** |  | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Actor click “Logout” link | System response:  2. System delete admin session data. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.3 Admin manage all functions of the moderator**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all functions | |
| **Actors** | Admin | |
| **Description** | Admin can reply comment and feedback | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Add, edit, delete information for moderators and users, vehicle contracts, manage the process of vehicle booking contracts, and comments and feedback.  3. Update or create new and delete information as requested by the admin. | System response:  2. Navigate to the management page according to each function for processing.  4. Save the added and edited data to the database, and return to the management page. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.4 Admin create category**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all category | |
| **Actors** | Admin | |
| **Description** | Admin can create category | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Click on the add category button.  3. Add information for the category , click on the add button. | System response:  2. Navigate to the page for adding category information.  4. System show add car form with following control:  - ‘Name’ text field  - ‘Description’ text field  - ‘Add’ button  - ‘Reset’ button  6. System validates the information.  7. System update account database and show update success message . |
| **Alternative flow** |  |  |
| **Exception** | Actor action:  1. ‘Name’ field left blank or incorrect format.  2. ‘Description’ field left blank or incorrect format. | System displays message to actor:  - ‘Name’ cannot be blank or incorrect format.  - ‘Description’ cannot be blank or incorrect format. |

**1.5 Admin delete category**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all category | |
| **Actors** | Admin | |
| **Description** | Admin can delete category | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click “delete” link. | System response:  2. System delete order in order database. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.6 Admin can edit category**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all category | |
| **Actors** | Admin | |
| **Description** | Admin can edit category | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Click on the edit category button.  3. Add information for the category , click on the edit button. | System response:  2. Navigate to the page for editing category information.  4. System show edit car form with following control:  - ‘Name’ text field  - ‘Description’ text field  - ‘Edit button  - ‘Reset’ button  6. System validates the information.  7. System update account database and show update success message . |
| **Alternative flow** |  |  |
| **Exception** | Actor action:  1. ‘Name’ field left blank or incorrect format.  2. ‘Description’ field left blank or incorrect format. | System displays message to actor:  - ‘Name’ cannot be blank or incorrect format.  - ‘Description’ cannot be blank or incorrect format. |

**1.7 Admin add products**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all products | |
| **Actors** | Admin | |
| **Description** | Admin can create products | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Click on the add product button.  3. Add information for the car , click on the add button. | System response:  2. Navigate to the page for adding product information.  4. System show add car form with following control:  - ‘Name car’ text field  - ‘Brand’ text field  - ‘Color’ field  - ‘Type’ field  - ‘Year’ field  - ‘Capacity’ text field  - ‘Price’ text field  - ‘Overview’ text field  - ‘Images’ field  - ‘Thumbnail’ field  - ‘Add’ button  - ‘Reset’ button  6. System validates the information.  7. System update account database and show update success message . |
| **Alternative flow** |  |  |
| **Exception** | Actor action:  1. ‘Name car’ field left blank or incorrect format.  2. ‘Brand’ field left blank or incorrect format.  3. ‘Color’ field left blank or incorrect format.  4. ‘Capacity’ field left blank or incorrect format.  5. ‘Price’ field left blank or incorrect format.  6. ‘Overview’ field left blank or incorrect format.  7. ‘Images’ incorrect format.  8. ‘Thumbnail’ incorrect format. | System displays message to actor:  - ‘Name car’ cannot be blank or incorrect format.  - ‘Brand’ cannot be blank or incorrect format.  - ‘Color’ must choose.  - ‘Capacity’ cannot be blank or incorrect format.  - ‘Price’ cannot be blank or incorrect format.  - ‘Overview’ cannot be blank or incorrect format.  - ‘Images’ must not be empty and must be in the correct format.  - ‘Thumbnail’ must not be empty and must be in the correct format. |

**1.8 Admin edit products creation**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all products | |
| **Actors** | Admin | |
| **Description** | Admin can edit products | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Click on the edit product button.  3. Edit information for the car , click on the save button. | System response:  2. Navigate to the page for page for editing product information.  4. System show edit car form with following control:  - ‘Name car’ text field  - ‘Brand’ text field  - ‘Color’ field  - ‘Type’ field  - ‘Year’ field  - ‘Capacity’ text field  - ‘Price’ text field  - ‘Overview’ text field  - ‘Images’ field  - ‘Thumbnail’ field  - ‘Save’ button  - ‘Reset’ button  6. System validates the information.  7. System update account database and show update success message . |
| **Alternative flow** |  |  |
| **Exception** | Actor action:  1. ‘Name car’ field left blank or incorrect format.  2. ‘Brand’ field left blank or incorrect format.  3. ‘Color’ field left blank or incorrect format.  4. ‘Capacity’ field left blank or incorrect format.  5. ‘Price’ field left blank or incorrect format.  6. ‘Overview’ field left blank or incorrect format.  7. ‘Images’ incorrect format.  8. ‘Thumbnail’ incorrect format. | System displays message to actor:  - ‘Name car’ cannot be blank or incorrect format.  - ‘Brand’ cannot be blank or incorrect format.  - ‘Color’ must choose.  - ‘Capacity’ cannot be blank or incorrect format.  - ‘Price’ cannot be blank or incorrect format.  - ‘Overview’ cannot be blank or incorrect format.  - ‘Images’ must not be empty and must be in the correct format.  - ‘Thumbnail’ must not be empty and must be in the correct format. |

**1.9 Admin delete products**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage all products | |
| **Actors** | Admin | |
| **Description** | Admin can delete products | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click “delete” link. | System response:  2. System delete order in order database. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**1.10. Admin delete banners and blogs**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage banner and blogs | |
| **Actors** | Admin | |
| **Description** | Admin can delete products | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click “delete” link. | System response:  2. System delete order in order database. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan**: EASYCAR | **Document Name**: REVIEW II | **SWD/Form** No.01/RV2/Ver1.0 |
| **Effective Date**:25/12/2023 | **Version** 1.0 | **Page No**: **29** of **1999** |

**1.11 Admin create banners and blogs**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage banner and blogs | |
| **Actors** | Admin | |
| **Description** | Admin can create banner and blogs | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Actor click on the add Banner/Blogs button.  3. Add information for Banner/Blogs, click on the add button. | System response:  2. Navigate to the page for adding information of Banner/Blogs.  4. Save the added data to the database, and return to the management page. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of user account information |

**1.12. Admin edit banners and blogs**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage banner and blogs | |
| **Actors** | Admin | |
| **Description** | Admin can edit banner and blogs | |
| **Requirements** | Admin is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Actor click on the edit Banner/Blogs button.  3. Add information for Banner/Blogs, click on the edit button. | System response:  2. Navigate to the page for editing information of Banner/Blogs.  4. Save the added data to the database, and return to the management page. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of user account information |

**2. Moderator**

**2.1 Moderator login**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Log in | |
| **Actors** | Moderator | |
| **Description** | Moderator can login to admin dashboard | |
| **Requirements** | Moderator provide email and password | |
| **Status** | Success: show login message successful.  Fail: show message failed. | |
| **Basic flow** | Actor action:  1. Actor type email and password, then click “Login” button. | System response:  2. System validate information. If correct, redirect to system dashboard. Otherwise, show failed login message. |
| **Alternative flow** |  |  |
| **Exception** | - Actor type invalid email and password. | System show message:  - “Wrong email or password” |

**2.2 Moderator logout**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Log out | |
| **Actors** | Moderator | |
| **Description** | Moderator can log out of the admin dashboard | |
| **Requirements** |  | |
| **Status** |  | |
| **Basic flow** | Actor action:  1. Actor click “Logout” link | System response:  2. System delete admin session data. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.3 Moderator create car rental contract management**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Car rental contract management | |
| **Actors** | Moderator | |
| **Description** | Moderator can create car rental contracts | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click on the button to “create” a car rental contract.  3. Add contract information to button “create”. | System response:  2. Navigate to the car rental contracts page.  4. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of order information |

**2.4 Moderator edit car rental contract management**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Car rental contract management | |
| **Actors** | Moderator | |
| **Description** | Moderator can edit car rental contracts | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click on the button to “edit” a car rental contract.  3. Edit the information on the contract, click on the “Save” button . | System response:  2. Navigate to the car rental contracts page.  4. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of order information |

**2.5 Moderator delete car rental contract management**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Car rental contract management | |
| **Actors** | Moderator | |
| **Description** | Moderator can delete car rental contracts | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click “delete” link. | System response:  2. System delete order in order database. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.6 Moderator contract process management**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Confirm the contract with the customer | |
| **Actors** | Moderator | |
| **Description** | Moderator can confirm | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. After confirming with the customer (calling), click on the ‘Details’ button  3. Click the ‘Confirmed’ button. | System response:  2. Navigate to the contract details page.  4. Change the status of the car rental contract to Confirmed.  5. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.7 Moderator confirm handing over the vehicle**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Confirm handing over the vehicle | |
| **Actors** | Moderator | |
| **Description** | Moderator can confirm | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. At the details page, click on the Delivered button. | System response:  2. Change the status of the car rental contract and the vehicle status to Delivered. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.8 Moderator confirm that the customer has returned the vehicle**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Confirm that the customer has returned the vehicle | |
| **Actors** | Moderator | |
| **Description** | Moderator can confirm | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. At the details page, click on the Received the vehicle button. | System response:  2. Change the status of the car rental contract and the vehicle status to Received the vehicle back. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.9 Moderator create manage user accounts**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage user accounts | |
| **Actors** | Moderator | |
| **Description** | Moderator can create user account | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click on the add customer button on the system management page.  3. Add contract information to button “create”. | System response:  2. Navigate to the customer's additional information page.  4. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of user account information |

**2.10. Moderator edit manage user accounts**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage user accounts | |
| **Actors** | Moderator | |
| **Description** | Moderator can edit user account | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click on the edit button for customer information.  3. Edit the necessary information of the customer, and press the “save” button. | System response:  2. Navigate to the customer information editing page.  4. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** | - Actor enters incorrect information into the input fields. | System show message:  - error requesting re-entry of user account information |

**2.11. Moderator delete manage user accounts**

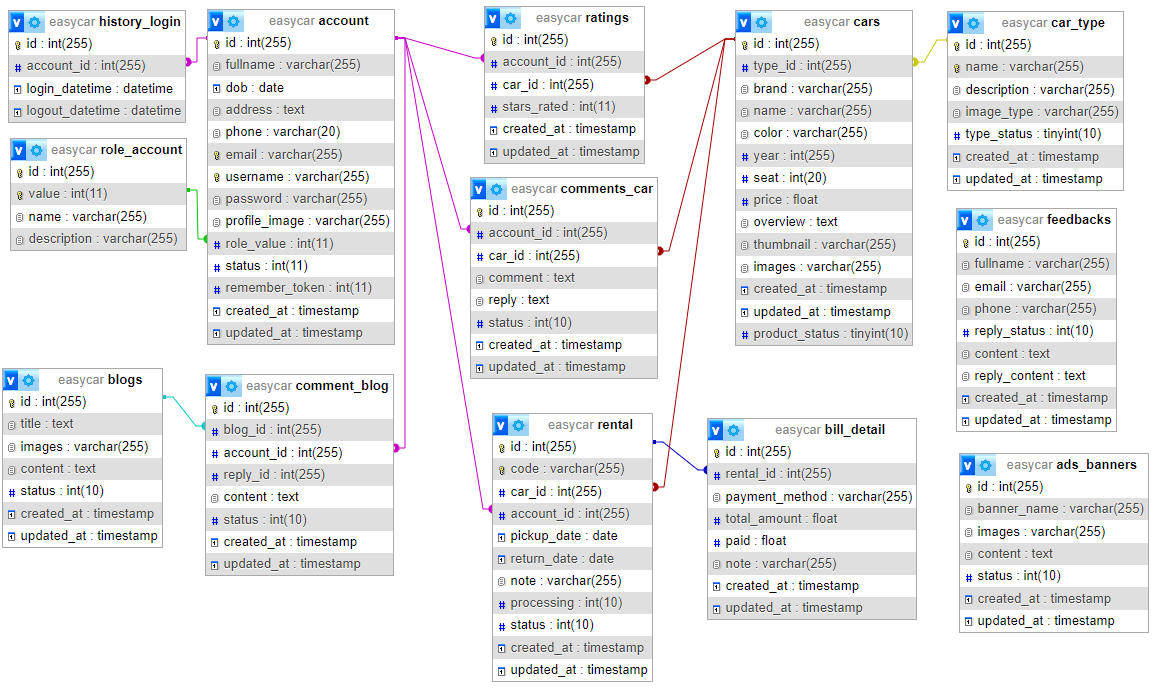
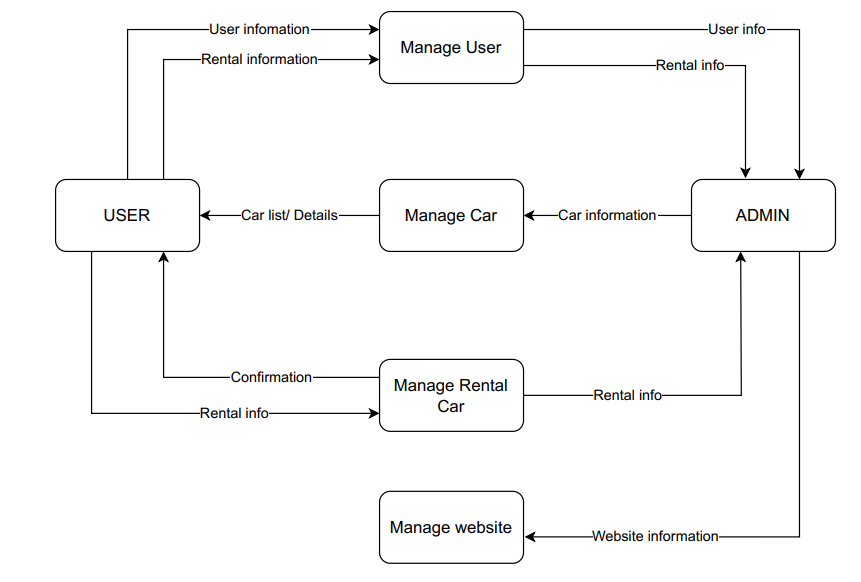
|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage user accounts | |
| **Actors** | Moderator | |
| **Description** | Moderator can delete user account | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click “delete” link. | System response:  2. System delete user account in account database. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

**2.12. Moderator manage customer feedback and comments**

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Manage comment, feedback | |
| **Actors** | Moderator | |
| **Description** | Moderator can reply comment and feedback | |
| **Requirements** | Moderator is logged in | |
| **Status** |  | |
| **Basic flow** | Actor action create :  1. Actor click the reply button for comments and feedback.  3. Click to select hide or show status. | System response:  2. Navigate to the comments, feedback page.  4. Save the added data to the database and navigate to the page management. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

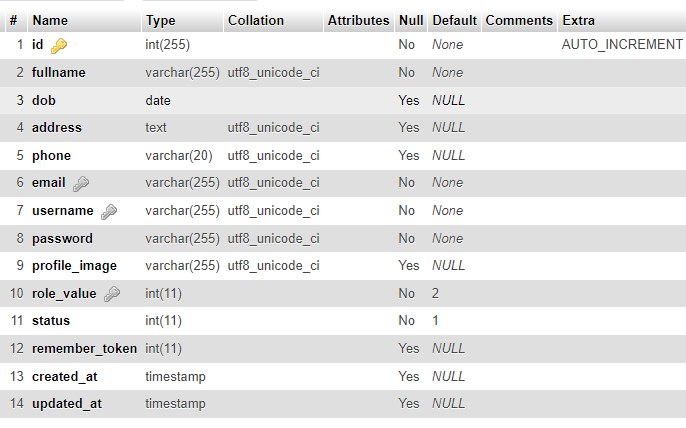
## DATA FLOW DIAGRAM (DFD)

## ENTITY RELATIONSHIP DIAGRAM (ERD)

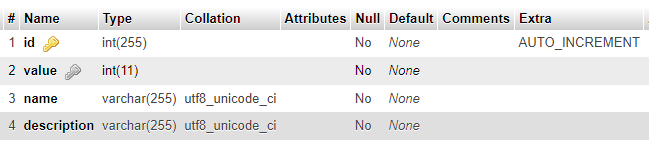


## Table Design:

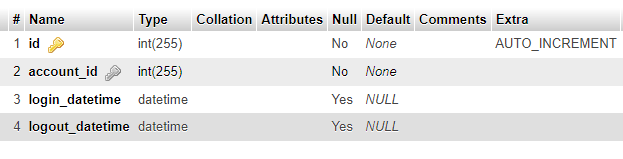
1. **Table accounts:**



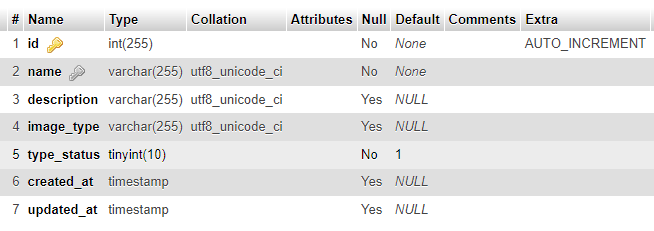
1. **Table role\_account:**



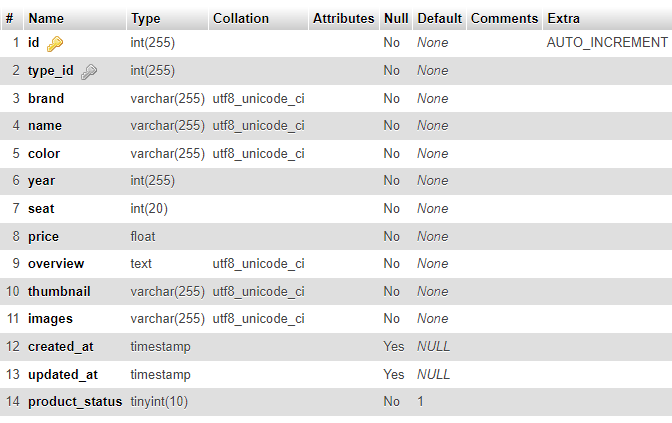
1. **Table history\_login:**



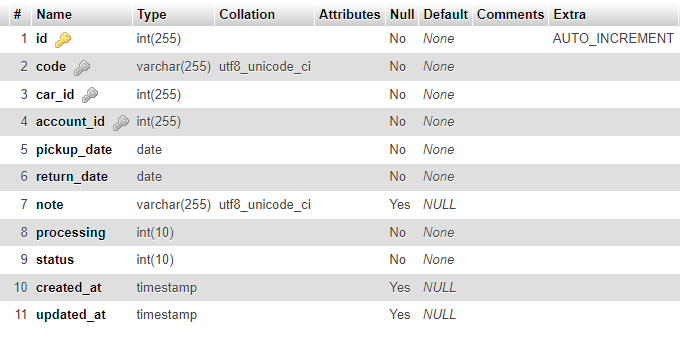
1. **Table car\_type:**



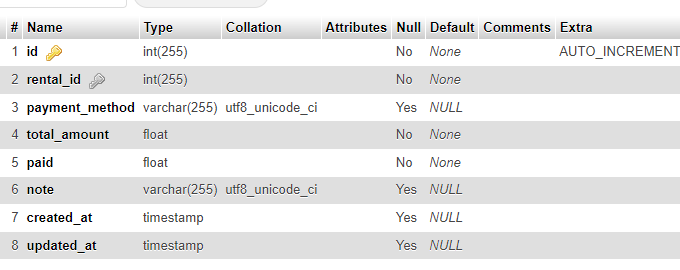
1. **Table cars:**



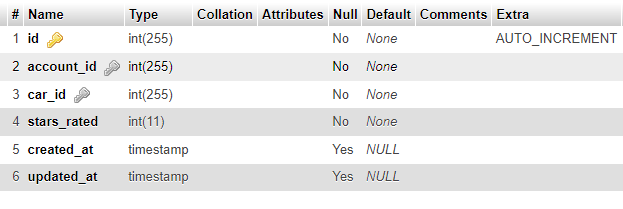
1. **Table rental:**



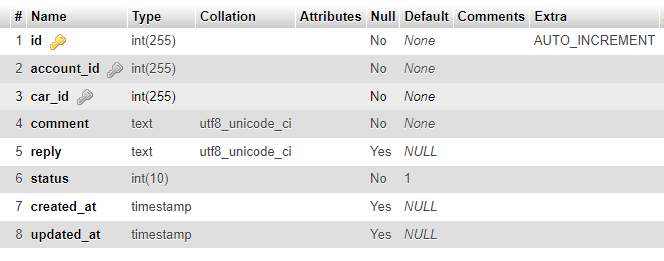
1. **Table bill\_detail:**



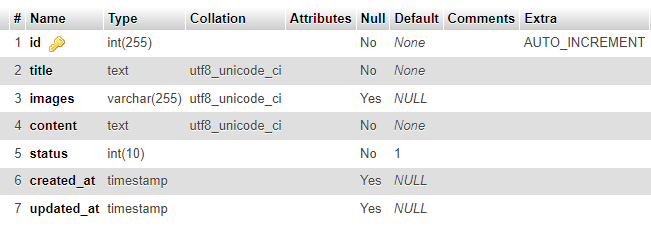
1. **Tables ratings**



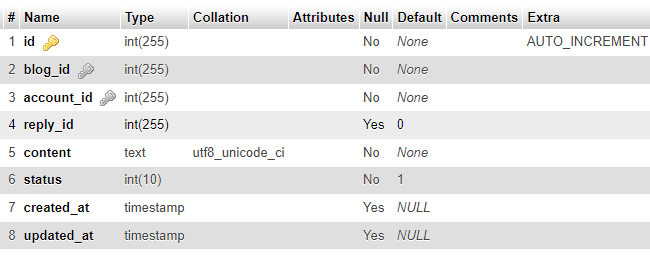
1. **Table comments\_car:**



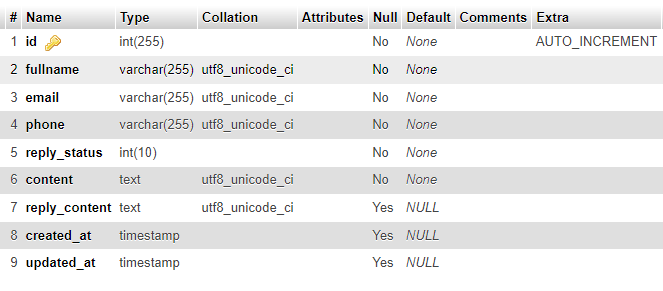
1. **Table blogs**



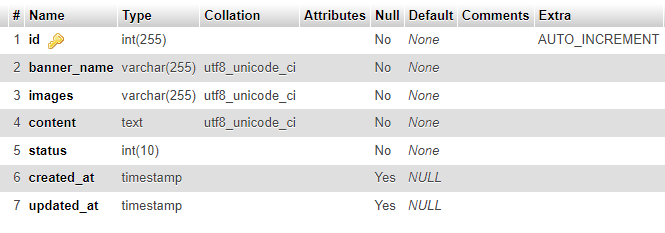
1. **Table comment\_blog:**



1. **Table feedbacks**



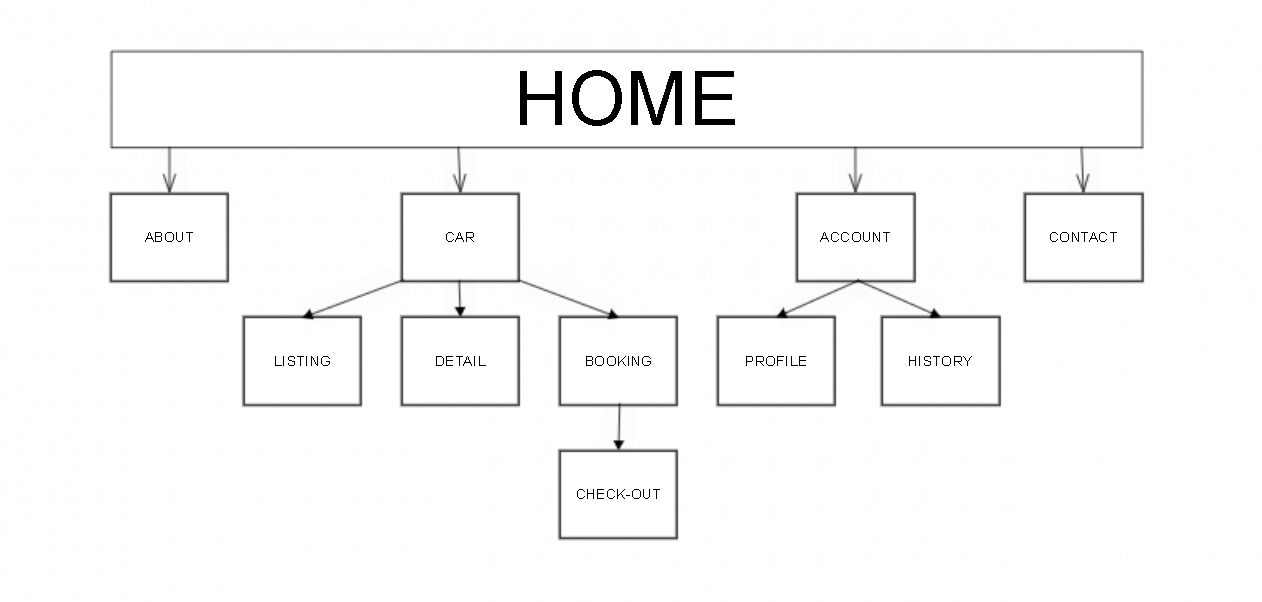
1. **Table ads\_banners**



REVIEW 2 TASKSHEET

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Ref. | Project Title:  Game Store | | Date of Preparation of Activity Plan:  25/12/2024 | | |
| No. | Task | Start Date | Actual Days | Member | Status |
| 01 | Use Case Visitor | 25/12/23 | 10/01/24 | -Phan Đặng Hoàng Tiến | Completed |
| 02 | Use Case User | - Vũ Quốc Việt | Completed |
| 03 | Use Case Moderation | - Nguyễn Quốc Thiện | Completed |
| 04 | Use Case Admin | - Trần Thanh Hải | Completed |
| 05 | Data Flow Diagram | - Phan Đặng Hoàng Tiến | Completed |
| 06 | ERD | - Vũ Quốc Việt | Completed |
| 07 | Table Design | - Trần Thanh Hải | Completed |

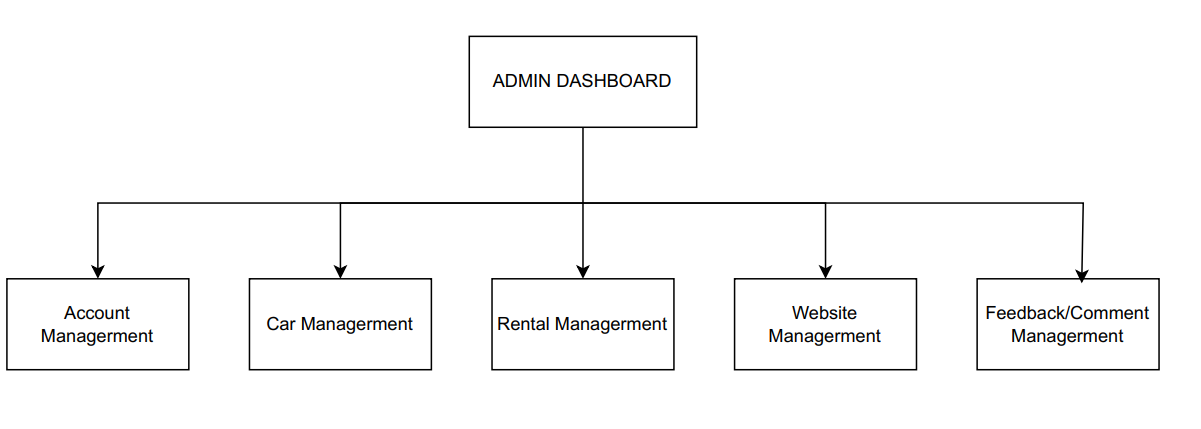
REVIEW 3



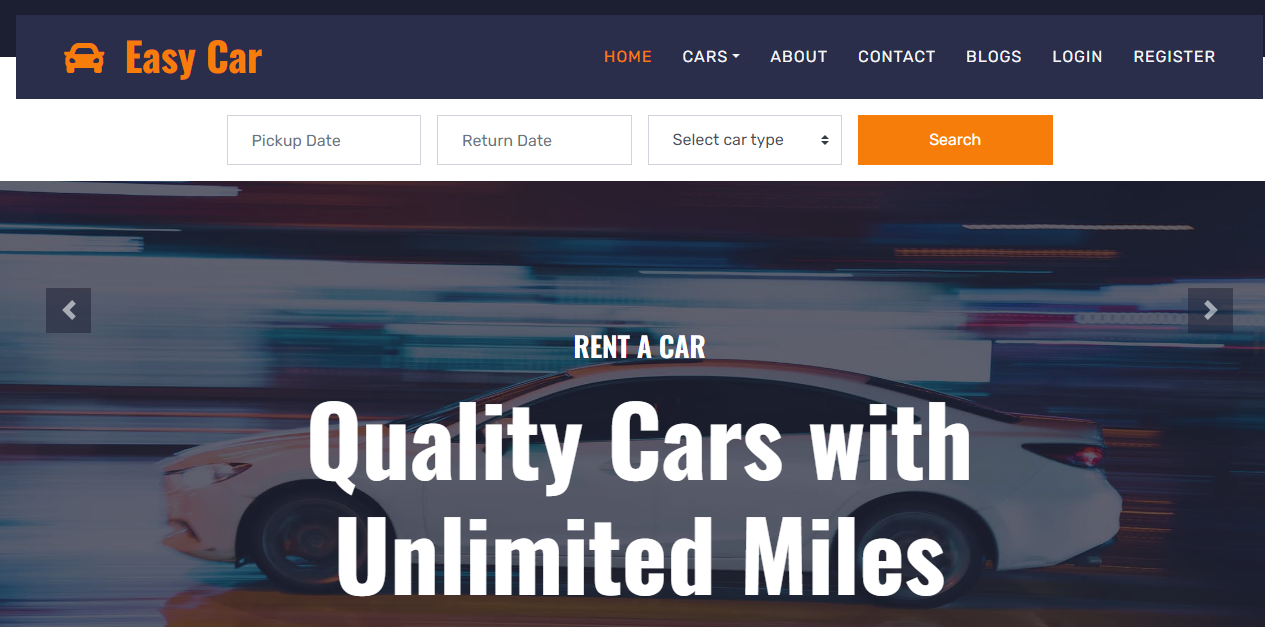
1. **Sitemap:**

ád

**II. Screenshot:**

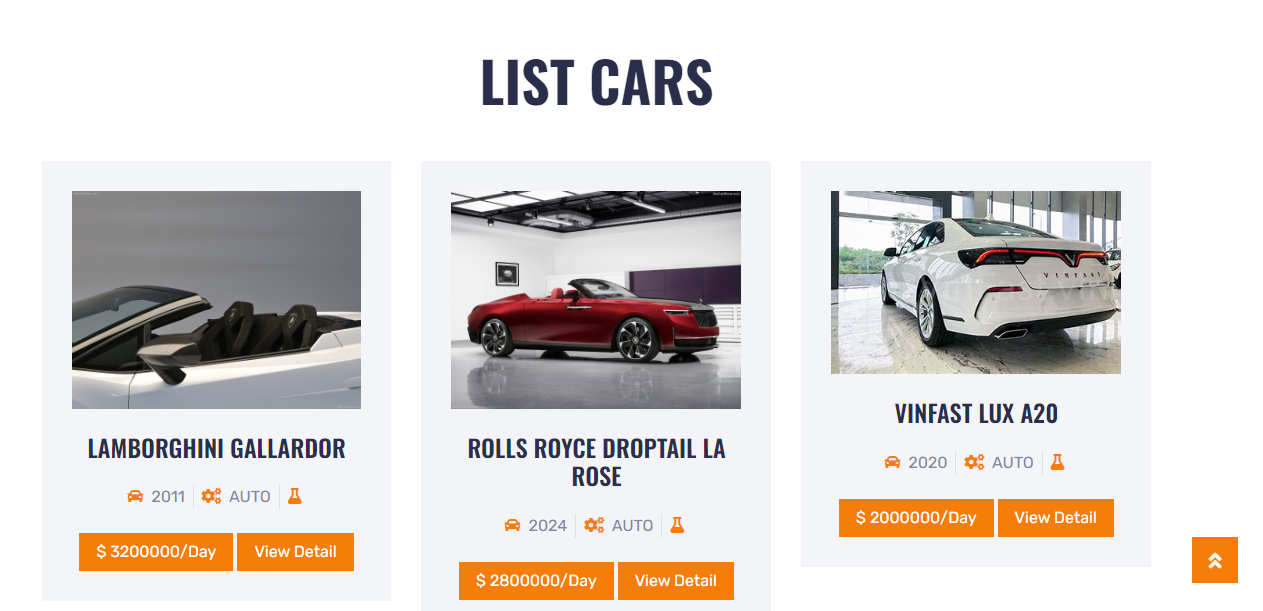


* + - 1. **Homepage:**



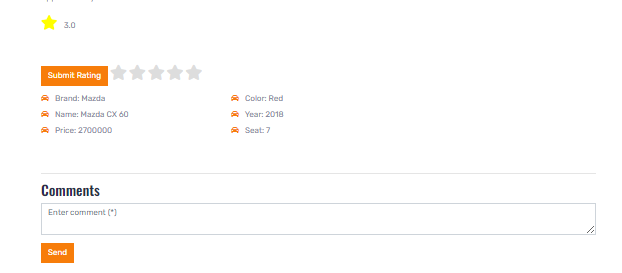
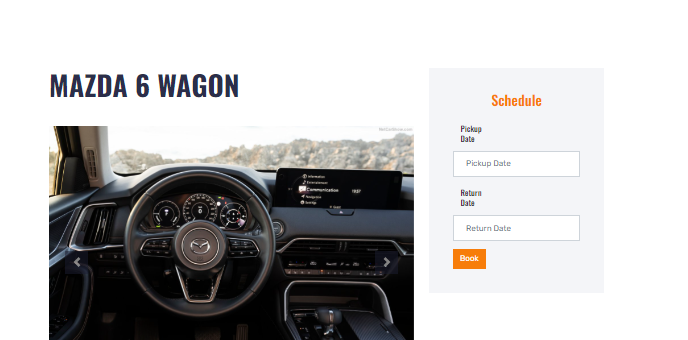
User or visitor can accesc to other page by click on button in NavBar. Visitor can search car by schedule input and click search button.

* + - 1. **Car List**



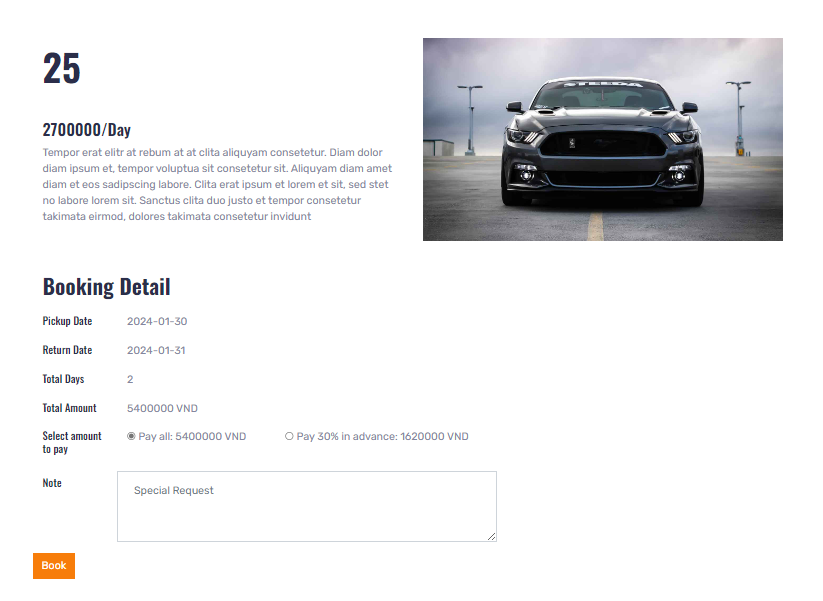
This page will show all car of each category (car type). Visitor click on View Detail button to redirect to Car Detail Page

* + - 1. **Car Details**



User must log in to book car and comment, rate. User enter schedule and click Book button to book this car.

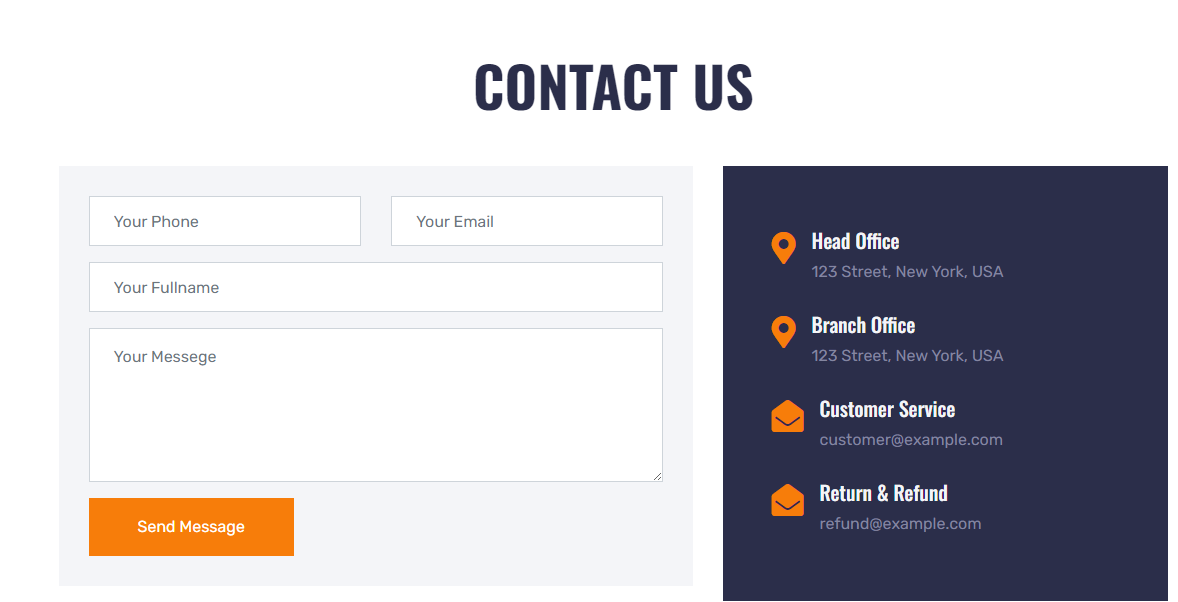
* + - 1. **Rental Page**



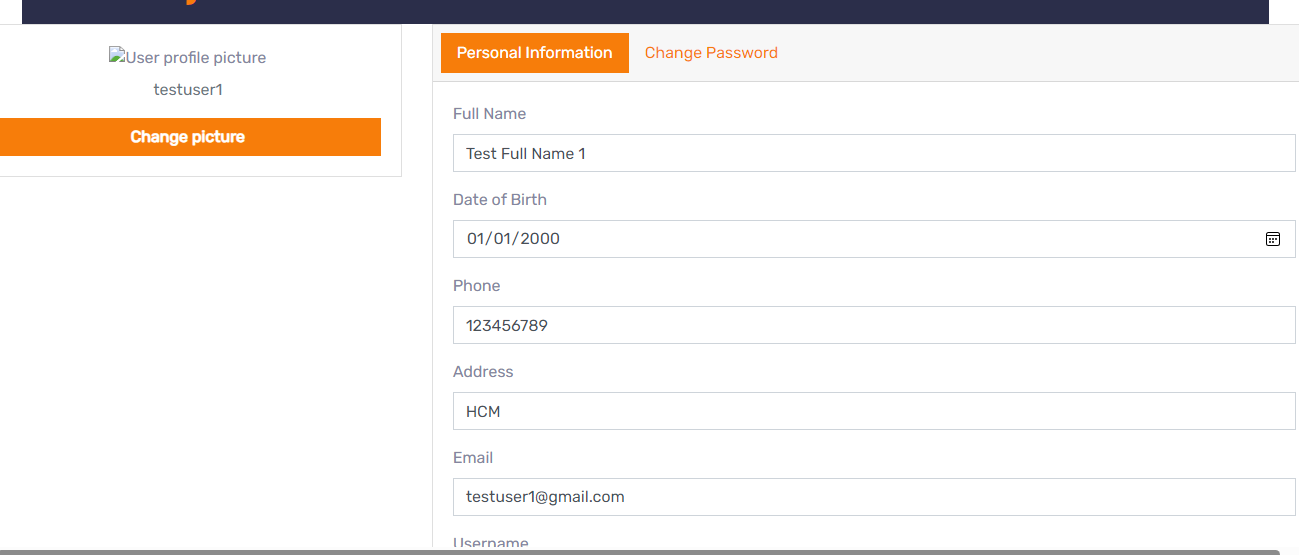
User choose the amount to pay in advance, and can leave note, then click Book. System will redirect to payment VNPay, after payment, website will notice to user success or failed booking.

* + - 1. **Contact**

Visitor can contact to website, input all inforation and message to website. EasyCar will reply to your message by email or phone.

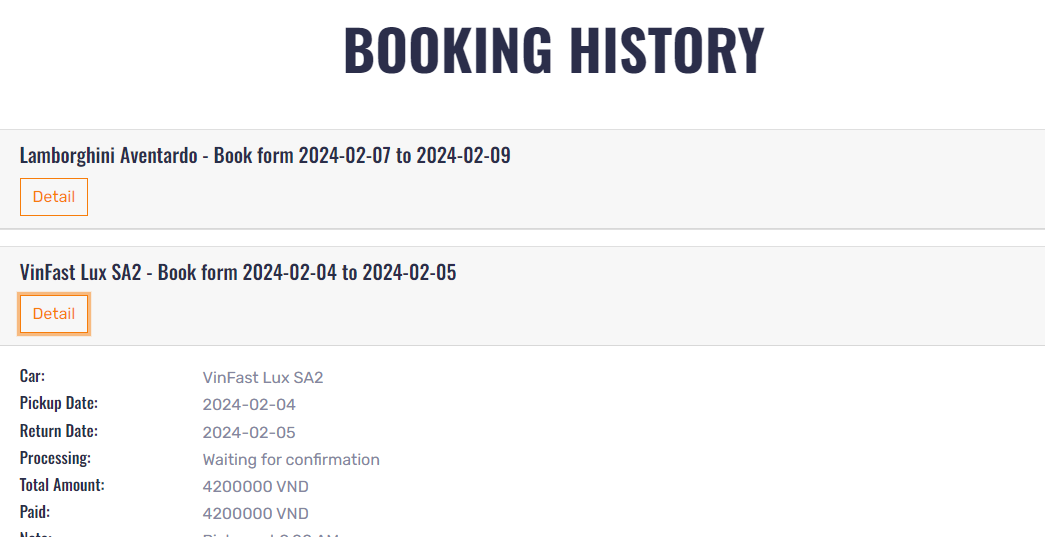


* + - 1. **Profile User**

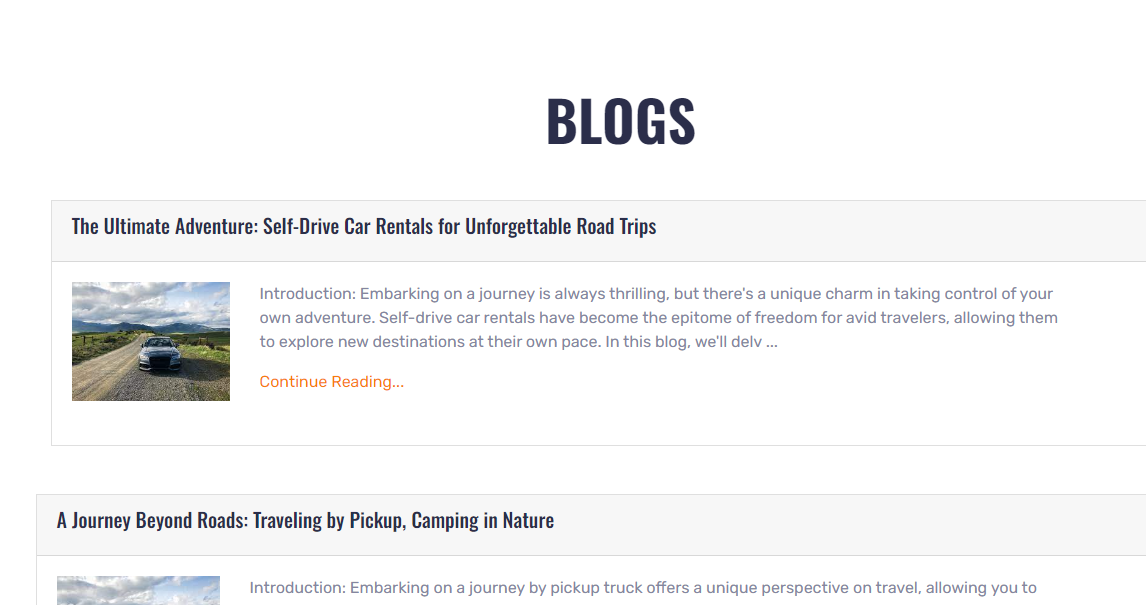


User can view personal info and edit profile, change password.

* + - 1. **Booking History**

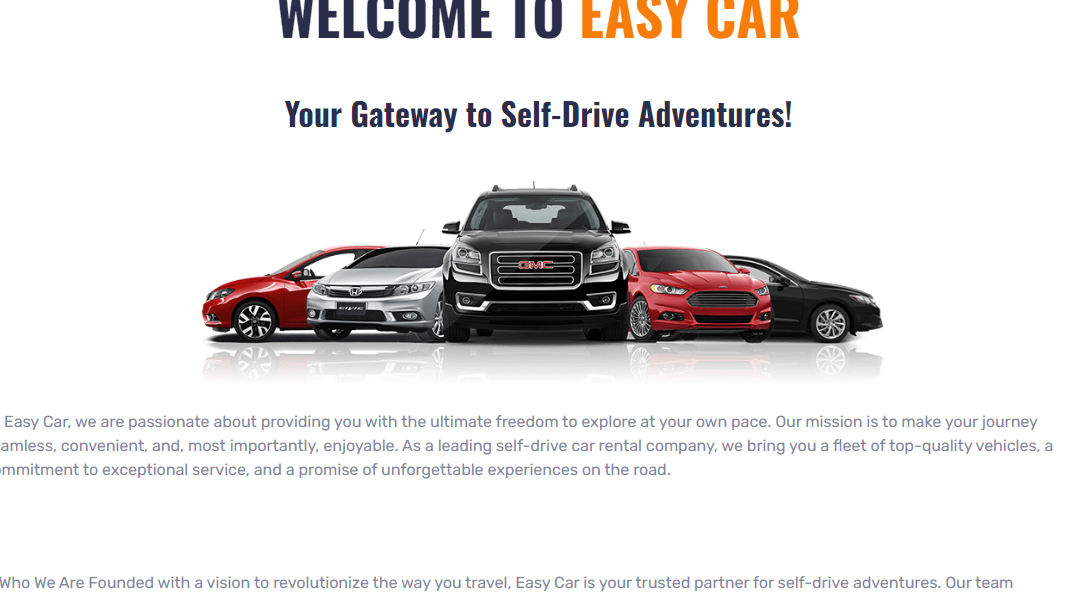


* + - 1. **Blogs**



Blog Page will display all blog. Click on Continue Reading to read full content of that blog.

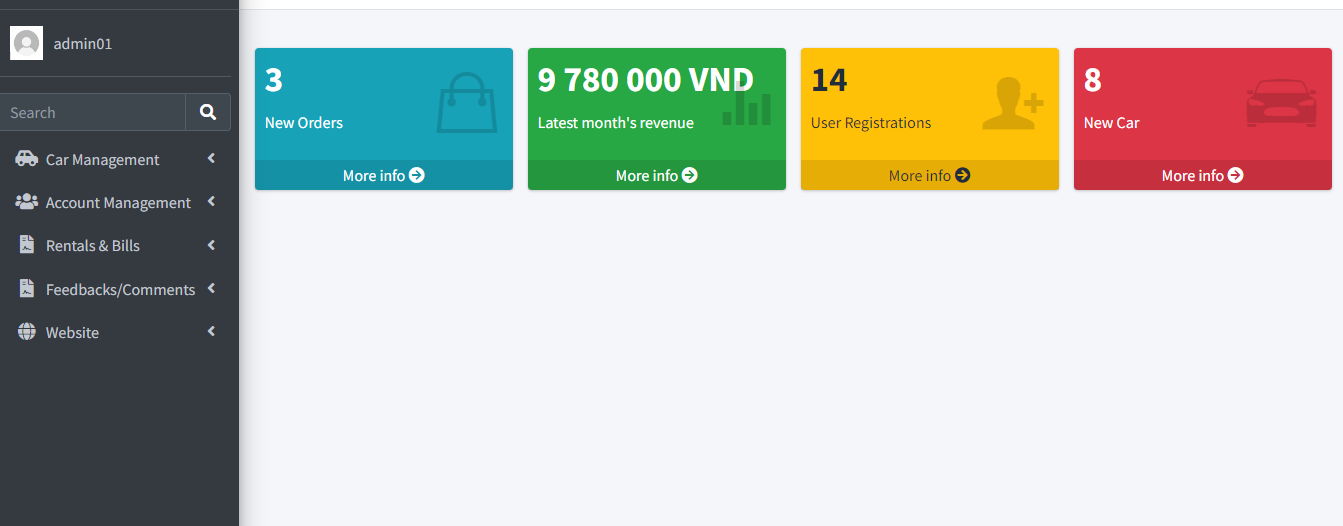
* + - 1. **About us**



This page will show information of Easycar

**ADMIN DASHBOARD:**

**Main dashboard page**

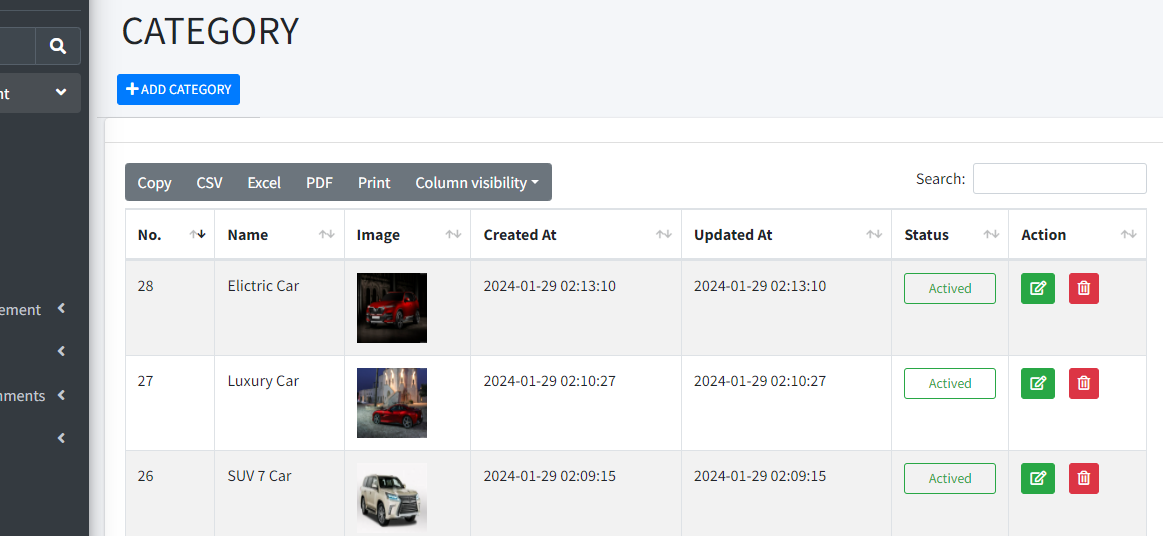


Display overall information of website in nearest 30days, click to More info in each card to view more detail.

The side bar include route to all managegement page of each class.

**Categories**

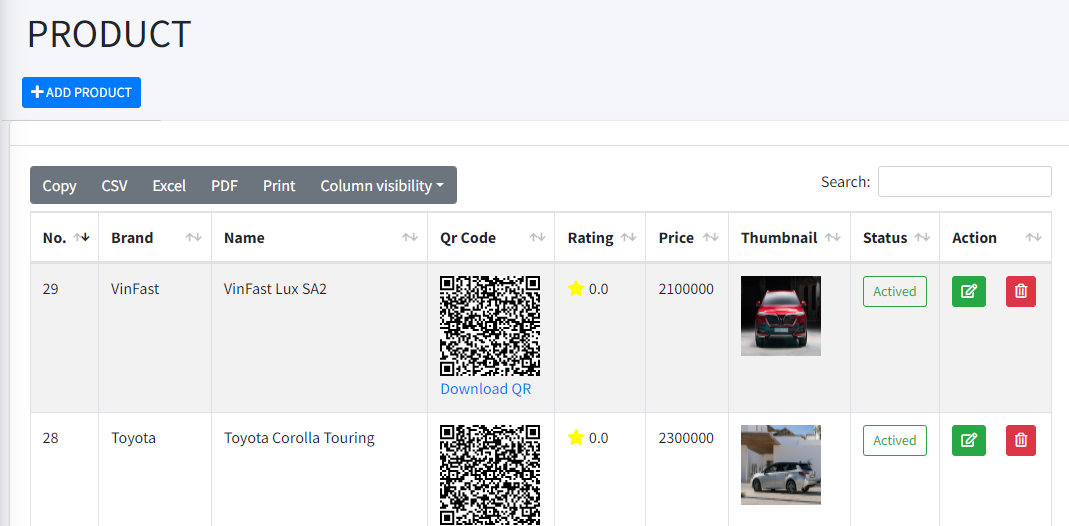
Click on Categories button on sidebar to redirect categories page:



Admin can manage Cartegories (Car types) on this page: add new categories, edit or delete exist categories

**Cars**

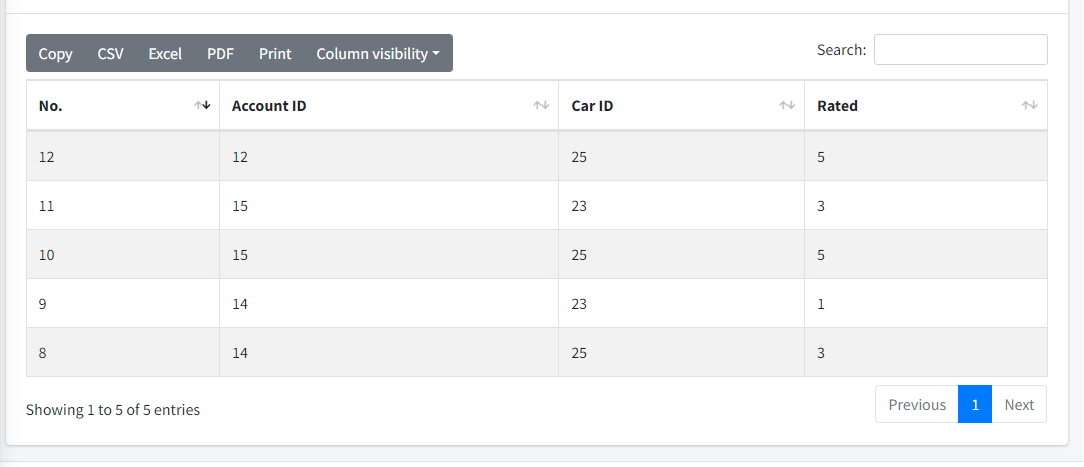
Click on Cars button on sidebar to redirect Car management page:



Admin can manage Car infoon this page: add new car, edit or delete existed car by click the button.

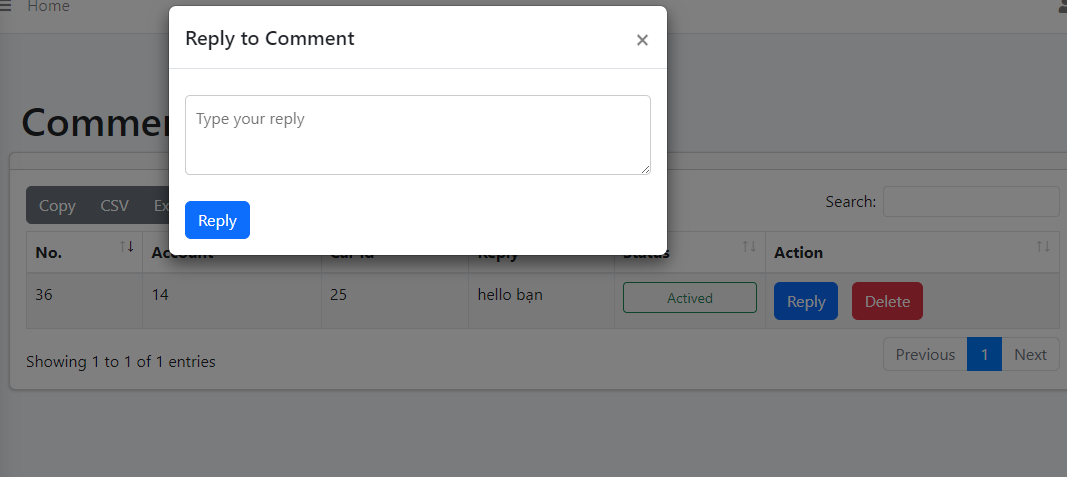
**Car ratings**

This page show list of rating that user have rated to the car.

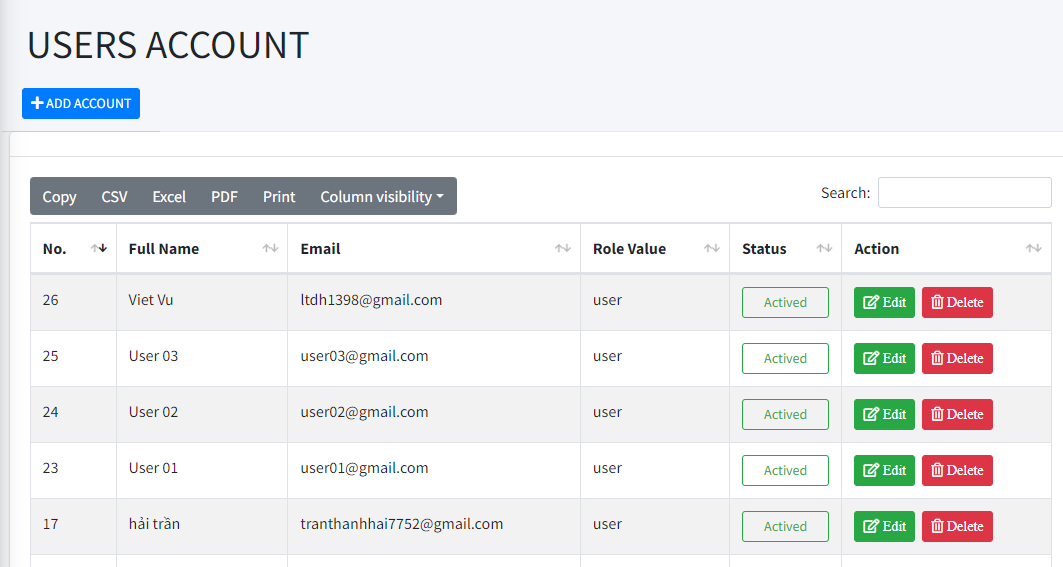


**Comments car**

This page show list of comment that user leave on each car detail page. Admin can reply to the comment here

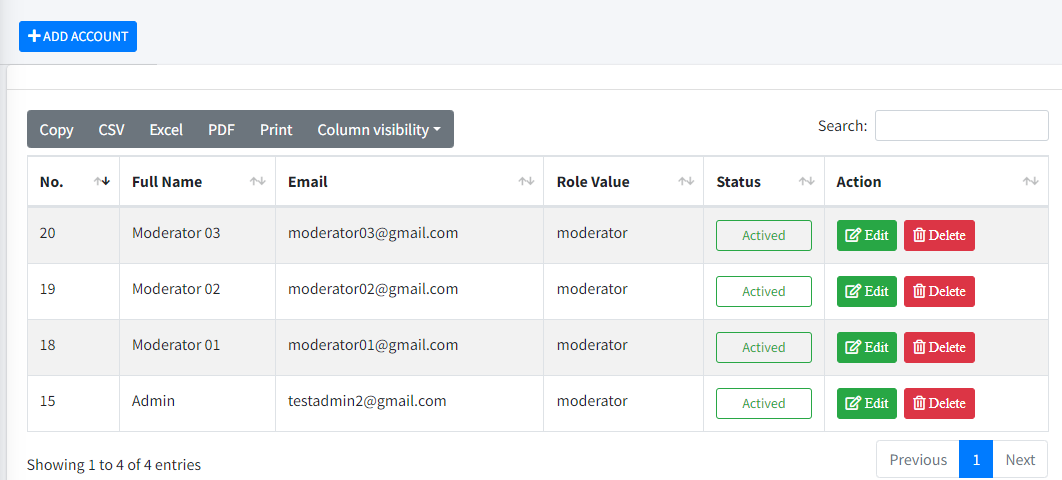


**Account user**



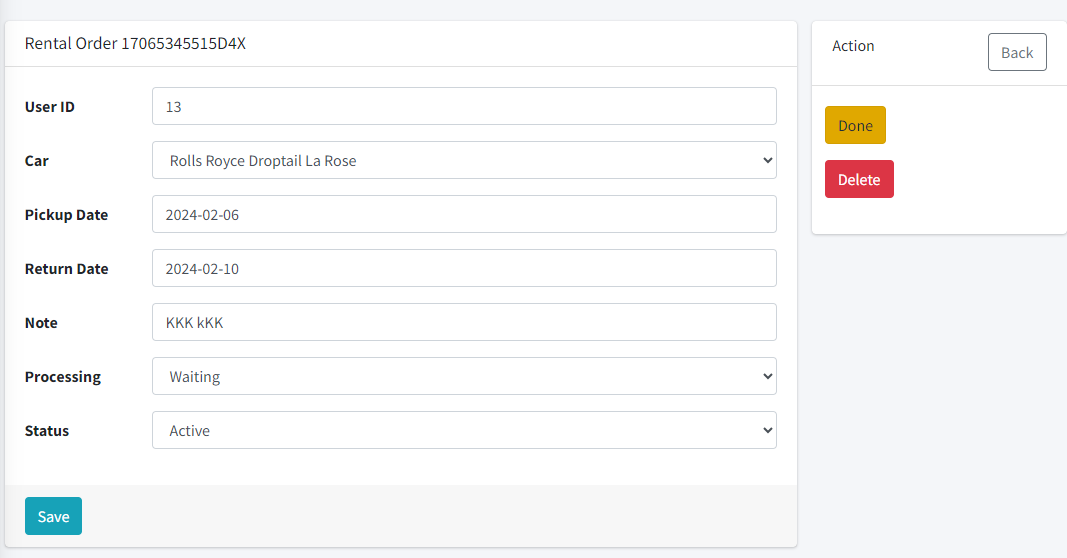
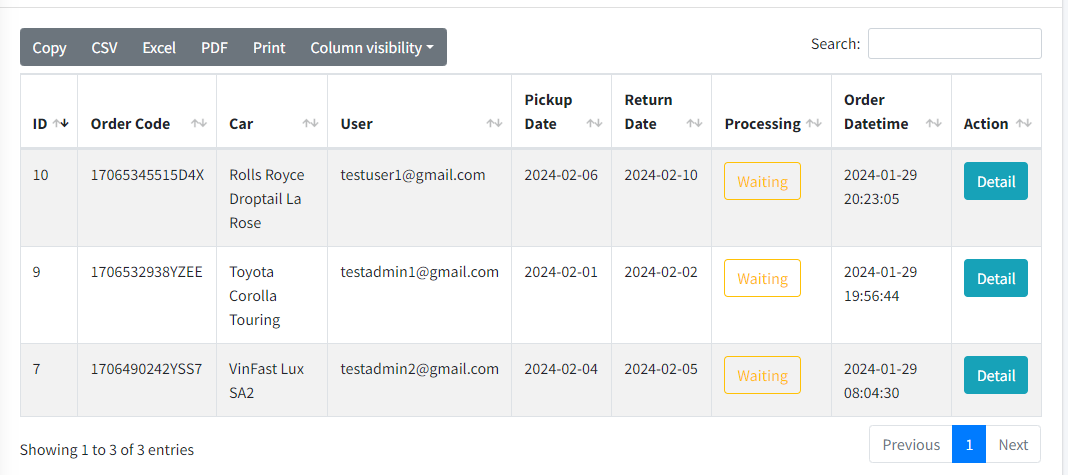
This page show list of all user have registered an account.Admin can edit user profile (if user request), or ban.delete account. Admin also make an user account to moderator of website.

**Account Moderator**



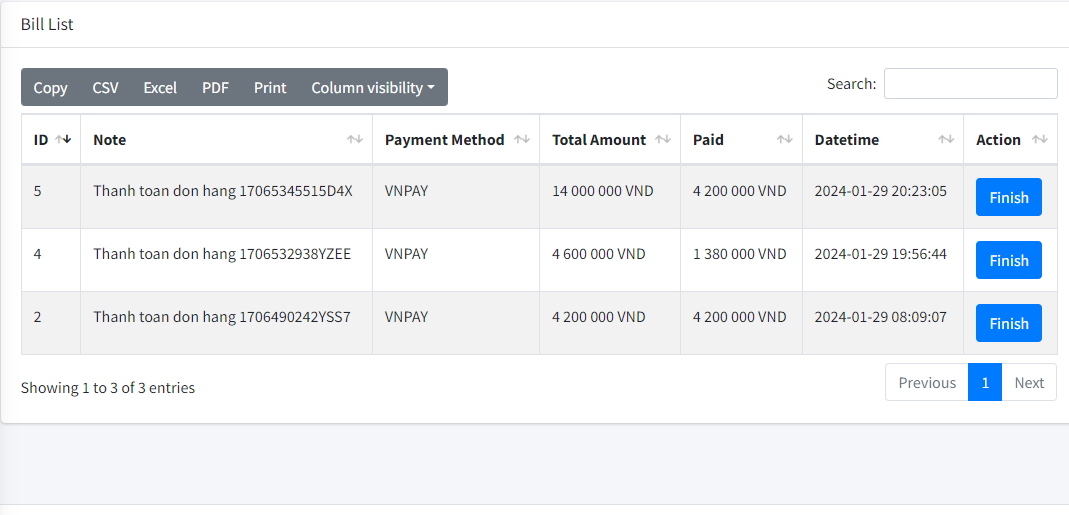
This page show list of all moderator of website.Admin can edit mod profile, or ban.delete account. Admin also can make mod account to user account.

**Rental Order**



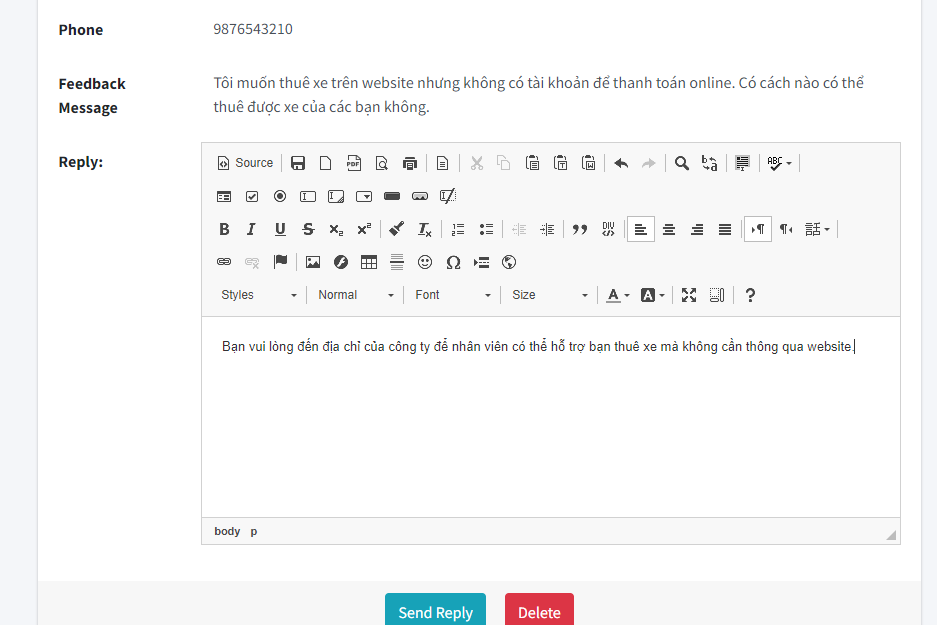
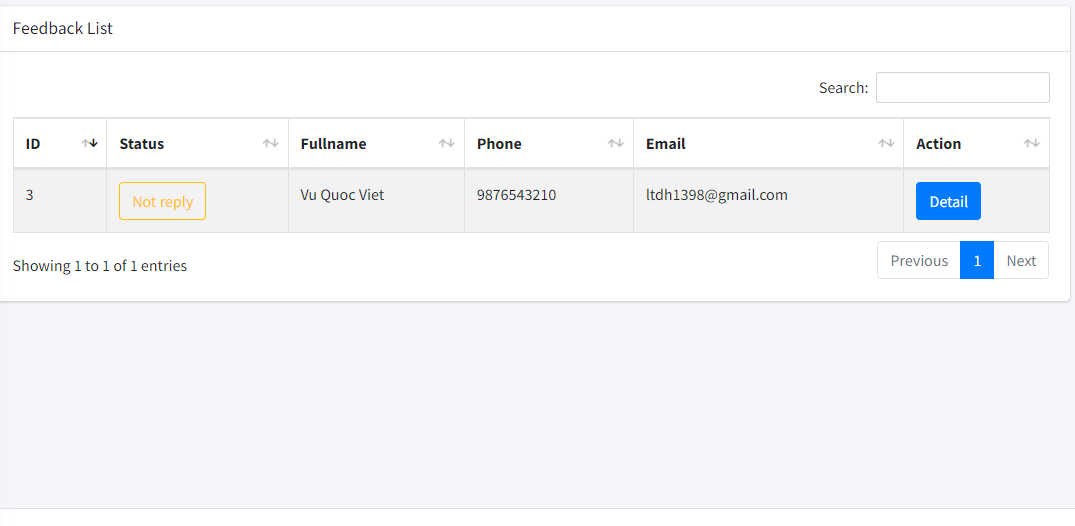
This page show all rental order. Admin can edit each order, delete order and bill of that order. Admin change processing status of each booking after confirm with user, or after user pickup/ return car.

**Bill Detail**



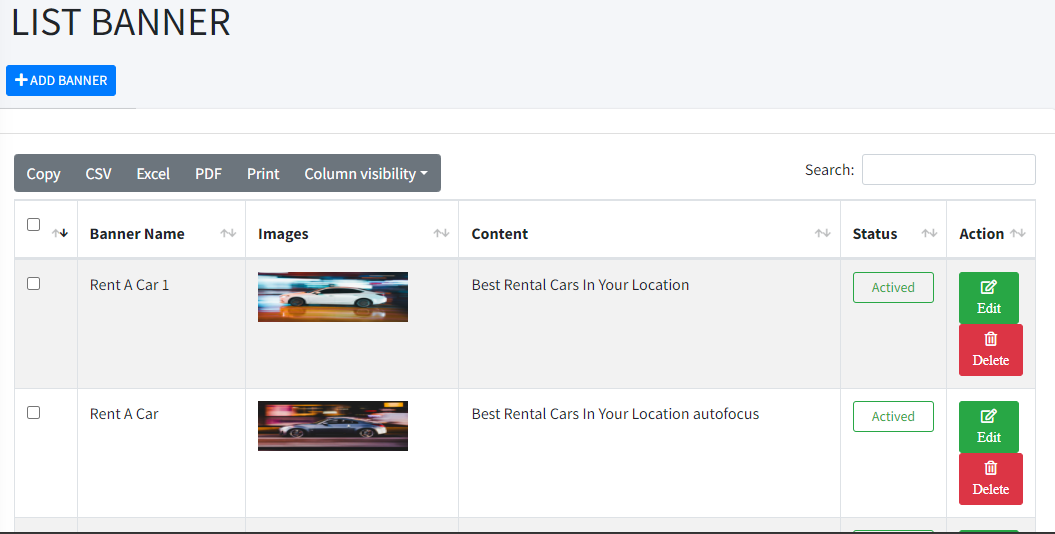
This page show all bill detail. Admin can confirm finish payment of user.

**Feedback**



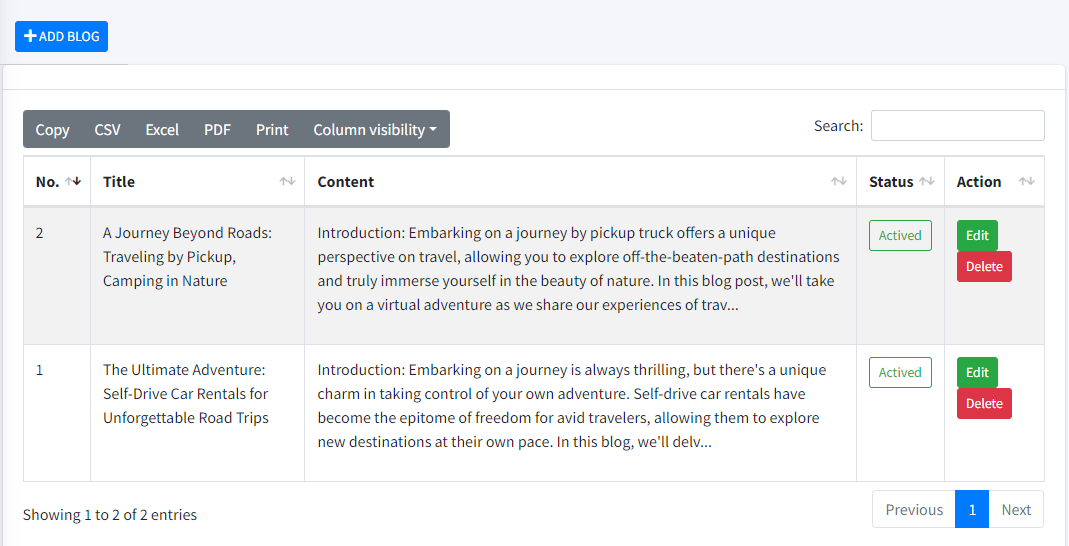
This page show all feedbacks receive from user/visitor. Admin can see all information of visitor, and can reply to them by email, or by phone

**Ads banner/Carousel**



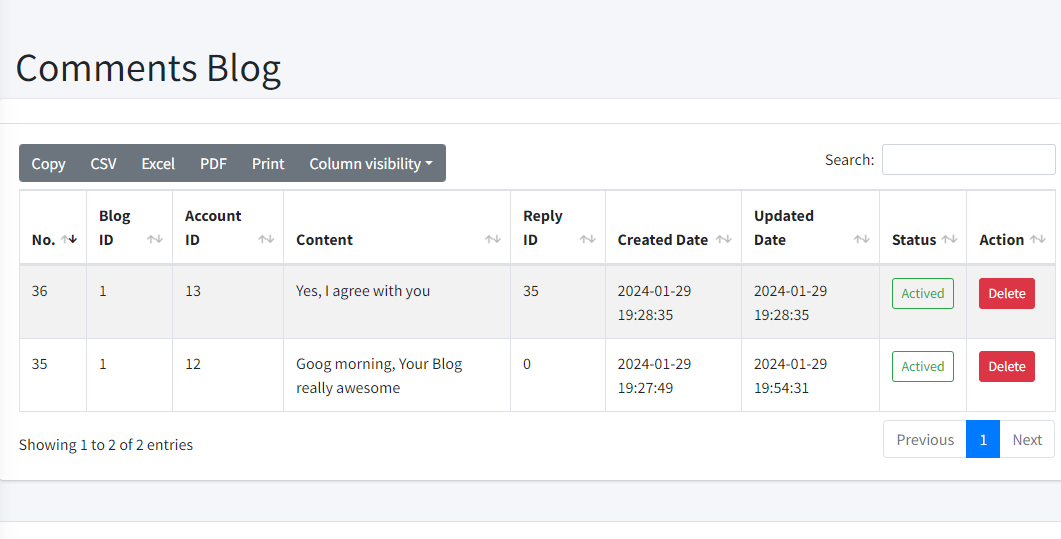
Admin can create, change or edit carousel for ads that display in website.

**Blogs**



Admin can create a blogs, edit or delete blog in website

**Blogs comment**



This page show all comment in each blog. Admin can hide or delete violent comment