**ASSIGNMENT FRONT SHEET**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Qualification** | **BTEC Level 5 HND Diploma in Computing** | | | |
| **Unit number and title** | **﻿WEBG301 – Project Web** | | | |
| **Submission date** |  | **Date Received 1st submission** | |  |
| **Re-submission Date** | 15/3/2023 | **Date Received 2nd submission** | | 15/3/2023 |
| **Group** | **Student ID & Name** | **Final Score** | | **Student’s signature** |
| 1. Luu Pham Anh Kiet – GCC200311 |  | | Luu Pham Anh Kiet |
| 2. Truong Quang Minh – GCC200321 |  | | Truong Quang Minh |
| 3. Huynh Quoc Tuan- GCC200083 |  | | Huynh Quoc Tuan |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | | |
| **Class** | GCCWEBHL | | **Assessor name** | Tran Thi Kim Khanh |

**OBSERVATION RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name:** | **Truong Quang Minh** | | |
| **Description of activity undertaken** | | | |
| * Create entity: product * CRUD of product * Create controller: AdminController, ProductController * Create cart page * Web interface design support * Participate in report writing | | | |
| **Assessment criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the assessment criteria** | | | |
|  | | | |
| **Student signature:** | **Truong Quang Minh** | **Date:** | **3/15/2023** |
| **Assessor name:** |  | | |
| **Assessor signature:** |  | **Date:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name:** | **Huynh Quoc Tuan** | | |
| **Description of activity undertaken** | | | |
| * Create entity: Category * CRUD of category * Create controller: AdminController, CategoryController * Web interface design support * Participate in report writing | | | |
| **Assessment criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the assessment criteria** | | | |
|  | | | |
| **Student signature:** | **Huynh Quoc Tuan** | **Date:** | **3/15/2023** |
| **Assessor name:** |  | | |
| **Assessor signature:** |  | **Date:** |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student name:** | **Luu Pham Anh Kiet** | | |
| **Description of activity undertaken** | | | |
| * Create entity: supplier * CRUD of supplier * Create controller: AdminController, SupplierController * Design about us page * Web interface design support * Participate in report writing | | | |
| **Assessment criteria** | | | |
|  | | | |
| **How the activity meets the requirements of the assessment criteria** | | | |
|  | | | |
| **Student signature:** | **Luu Pham Anh Kiet** | **Date:** | **3/15/2023** |
| **Assessor name:** |  | | |
| **Assessor signature:** |  | **Date:** |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **IV Signature:** | | |

**Assignment Brief 1 (RQF)**

**Higher National Certificate/Diploma in Computing**

|  |  |
| --- | --- |
| **Student Name/ID Number:** | 1. Huynh Quoc Tuan – GCC200311 2. Trương Quang Minh – GCC200321 3. Lưu Phạm Anh Kiệt - GCC200083 |
| **Unit Number and Title:** |  |
| **Academic Year:** |  |
| **Unit Assessor:** | **Hoang Nhu Vinh** |
| **Assignment Title:** |  |
| **Issue Date:** | **01 April 2021** |
| **Submission Date:** | **15/03/2023** |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Submission Format:** |
| *Format: The submission is in the form of documents*   * An individual evaluation document in PDF format. Writing must be professional.   *Submission*   * Students are compulsory to submit the assignment on the due date and in a way requested by the Tutor. * The form of submission will be a soft copy posted on [http://cms.greenwich.edu.vn/.](http://cms.greenwich.edu.vn/) * Remember to convert the word file into PDF file before the submission on CMS.   *Note:* |
| **Unit Learning Outcomes:** |

|  |  |  |
| --- | --- | --- |
| **LO1** Use MVC (Model – View – Controller) pattern with Symfony framework to develop dynamic Web Applications  **LO2** Use Doctrine ORM framework to query information from the database  LO3 Create Web API services to handle users’ request and return JSON responses via Postman REST client LO4 Apply best practices such as: Dependency Injection, Git and GitHub, Web security (CSRF, DdoS attack, | | |
|  | SQL Injection) in Web development |  |
| **Assignment Brief and Guidance:** | | |

**Scenario**: Design and implement a Student Management/Inventory Management/Shopping cart/CMS/RPG Game/Conference Scheduler using PHP (Symfony) and HTML / CSS / JavaScript. Your project must be approved by Lecturer and meet all the requirements listed below.

The project will be implemented by a team with 3 members maximum. Chapter 1 **General Requirements**

* **Use PHP** – the major part of your work should be PHP written
  + You **must use Symfony Framework**
    - The application must have at least **10 web pages (views**)
    - The application must have at least **4 independent entity models**
    - The application must have at least **4 controllers** o You have to additionally use

**HTML5, CSS3** to create the content and to stylize your web application

* + You may optionally use **JavaScript, jQuery, Bootstrap** o Use **PHP 7**

Chapter 2 **Forbidden Techniques and Tools**

* Using **CMS / blog systems** (like WordPress, Drupal and Joomla) is forbidden.
* Using **Shopping cart systems** (like OpenCart) is forbidden. Chapter 3 **Source Control**

Use a **source control system** by choice, e.g., **GitHub**, **BitBucket**

* Submit a link to your public source code repository
* Each student should have at least **10 meaningful commits**
* Each student should have **commits** in at least **3 DIFFERENT** days

**Tasks**

***Group Report***

Each group will be required to deliver a Group Report, which have following items:

* Produce a set of Users’ requirements by using User Story template
* Site map of the project
* Entity Relationship Diagram (ERD)
* Final Result of the project with evidences + GitHub repository evidences

+ Sample Source code with brief explanation Images of final application

* Conclusion and Lessons Learned
* The Document should be less than 20 pages

***Public Project Defense***

Each group will have to deliver a public defense of its work in front of the Lecturer.

Each group will have only 20 minutes for the following:

•

•

•

timing!

Each student demonstrates his/her part in the project

Each students shows the source code and explain how it works

Answer questions related to the project (and best practices in general) Please be strict in

Be well prepared for presenting maximum of your work for a minimum time. Bring your OWN LAPTOP. Test it

preliminarily with the multimedia projector. Open the project assets beforehand to save time.

Table of Contents

[Chapter 1 – Users’ requirements 15](#_Toc129790772)

[1.1. Project specification 15](#_Toc129790773)

[1.2. Users 15](#_Toc129790774)

[1.3. Input data is taken from the users 15](#_Toc129790775)

[1.4. Output data is displayed to the users 15](#_Toc129790776)

[1.5. The processes of the system 16](#_Toc129790777)

[1.6. Scope of work 17](#_Toc129790778)

[1.7. Customer acceptance criteria 17](#_Toc129790779)

[Chapter 2 – System Design 18](#_Toc129790780)

[2.1 About Symfony 18](#_Toc129790781)

[2.2. Git/GitHub 19](#_Toc129790782)

[2.3 System Architecture Diagram 21](#_Toc129790783)

[2.4 Use Case Diagram 22](#_Toc129790784)

[2.5 Entity Relationship Diagram 23](#_Toc129790785)

[2.6 Sitemap 24](#_Toc129790786)

[2.6.1 Home Page 24](#_Toc129790787)

[2.6.2 Administer 25](#_Toc129790788)

[2.7 Wireframes 25](#_Toc129790789)

[Chapter 3 – Implementation 32](#_Toc129790790)

[3.1 Sample Source Code 32](#_Toc129790791)

[3.2 Images of final Application 44](#_Toc129790792)

[3.3 GitHub Repository evidences 54](#_Toc129790793)

[Chapter 4 – Conclusion 56](#_Toc129790794)

[4.1 What went well 56](#_Toc129790795)

[4.2 What did no go well 56](#_Toc129790796)

[4.3 Lessons learned and further improvements 57](#_Toc129790797)

[References 58](#_Toc129790798)

Table of Figures

[Figure 1: What is symphony 18](#_Toc129800382)

[Figure 2: What is a git 20](#_Toc129800383)

[Figure 3: What is a Github 21](#_Toc129800384)

[Figure 4: System Architecture Diagram 21](#_Toc129800385)

[Figure 5: Use Case Diagram 22](#_Toc129800386)

[Figure 6: Entity Relationship Diagram 23](#_Toc129800387)

[Figure 7: Home site 24](#_Toc129800388)

[Figure 8: Admin site 25](#_Toc129800389)

[Figure 9: Home Page 25](#_Toc129800390)

[Figure 10: About us page 26](#_Toc129800391)

[Figure 11: Sign In page 26](#_Toc129800392)

[Figure 12; Register account 27](#_Toc129800393)

[Figure 13: View Product 27](#_Toc129800394)

[Figure 14: Add new product 28](#_Toc129800395)

[Figure 15: Update product 28](#_Toc129800396)

[Figure 16: View categories 29](#_Toc129800397)

[Figure 17: Add categories 29](#_Toc129800398)

[Figure 18: Update category 30](#_Toc129800399)

[Figure 19: View Supplier 30](#_Toc129800400)

[Figure 20: Add supplier 31](#_Toc129800401)

[Figure 21: Update supplier 31](#_Toc129800402)

[Figure 22: View account 32](#_Toc129800403)

[Figure 23 - Structure of system 33](#_Toc129800404)

[Figure 24 - Controller in system 33](#_Toc129800405)

[Figure 25 - Entity in system 34](#_Toc129800406)

[Figure 26 - Repository in system 34](#_Toc129800407)

[Figure 27 - Template in system 34](#_Toc129800408)

[Figure 28 - Show products function 35](#_Toc129800409)

[Figure 29 - Template of admin/product/products.html.twig part 1 35](#_Toc129800410)

[Figure 30 - Template of admin/product/products.html.twig part 2 36](#_Toc129800411)

[Figure 31 - Template of error\_admin.html.twig 36](#_Toc129800412)

[Figure 32 - Add new product function 37](#_Toc129800413)

[Figure 33 - Addnew product in ProductRepository 38](#_Toc129800414)

[Figure 34 - Template of admin/product/add.html.twig part 1 39](#_Toc129800415)

[Figure 35 - Template of admin/product/add.html.twig part 2 40](#_Toc129800416)

[Figure 36: Template of admin/product/add.html.twig part 3 40](#_Toc129800417)

[Figure 37 - Controller of show detail product 40](#_Toc129800418)

[Figure 38 – Controller of edit product 41](#_Toc129800419)

[Figure 39 - Function upload image in Admin Contoller 41](#_Toc129800420)

[Figure 40 - Template of edit product part 1 42](#_Toc129800421)

[Figure 41 - Template of edit product part 2 42](#_Toc129800422)

[Figure 42 - Template of edit product part 3 43](#_Toc129800423)

[Figure 43 - Function delete product 43](#_Toc129800424)

[Figure 44 - Function delete product in ProductRepository 44](#_Toc129800425)

[Figure 45 - Home page without login 45](#_Toc129800426)

[Figure 46 - Register page 46](#_Toc129800427)

[Figure 47 - Login page 46](#_Toc129800428)

[Figure 48 - Home page with login user 47](#_Toc129800429)

[Figure 49 - Home page with login admin 48](#_Toc129800430)

[Figure 50: Product Page 48](#_Toc129800431)

[Figure 51: Product Manager Page 49](#_Toc129800432)

[Figure 52: Add Product Page 49](#_Toc129800433)

[Figure 53: Edit Product Page 50](#_Toc129800434)

[Figure 54: Categories Manager Page 50](#_Toc129800435)

[Figure 55: Add Category Page 51](#_Toc129800436)

[Figure 56: Edit Category Page 51](#_Toc129800437)

[Figure 57: Supplier Manager Page 52](#_Toc129800438)

[Figure 58: Add Supplier Page 52](#_Toc129800439)

[Figure 59: Edit Supplier Page 53](#_Toc129800440)

[Figure 60: Account Manager Page 53](#_Toc129800441)

[Figure 61: Our repository on the GitHub 54](#_Toc129800442)

[Figure 62 - Some images about our commits on the GitHub (1) 55](#_Toc129800443)

[Figure 63: Some images about our commits on the GitHub (2) 56](#_Toc129800444)

Table of Tables

[Table 1: Customer acceptance criteria 18](#_Toc129790728)

# Chapter 1 – Users’ requirements

## 1.1. Project specification

High-end vehicles are sold on the website MinCar. You can locate a wide variety of luxury car brands on this website. The TMK team founded Min Car in 2023, and since then, it has come a long way from its initial in-store operations to engage more customers online. Team members Kiet, Tuan, and Minh are in charge of running the website. The website for the Windows Platform will be developed using HTML5, CSS, JavaScript, and PHP. All popular browsers, including Chrome, Microsoft, Opera, and others, must function properly with the website.

## 1.2. Users

In this project, 2 users use this site for their purposes:

* Admin: Min Car is the website of a small organization, so Minh, Tuan and Kiet are the founders and admins of the website. We use Min Ca to sell high-end automotive items to everyone who wants to buy or are interested in our products.
* Customers: Those who want to buy high-end cars sold on the web.

## 1.3. Input data is taken from the users

There are 2 main groups of data that the website takes:

Data taken from admin:

* Supplier information including the provider's name and address.
* Directory information including category name.
* Item information includes item name, image, description, price, item category name, item supplier name
* Account information including e-mail and password (encrypted) to login to the website.

Data taken from customers:

* Order information including full name, phone number, delivery address.

Data taken from the store:

* About me, contact and location.

## 1.4. Output data is displayed to the users

There are group data that the website takes:

Output data is displayed to admin:

* The list of providers consists of provider ID and provider name and address.
* The list of categories consists of category ID and category name.
* The list of items consists of product ID, item name, image, description, price, category name that product belongs to, and provider name that product belongs to.
* The list of users consists of user ID, email address, and password (encrypted).

Output data is displayed to the customer:

* List of products consists of product ID, product name, image, description, price, category name that product belongs to.
* Website information consists of about me, contact, and location.

## 1.5. The processes of the system

There are some processes of the system:

About Provider:

* Create a new provider.
* Update existing provider.
* Delete existing provider.
* Read all existing providers.

About Category:

* Create a new category.
* Update existing category.
* Delete existing category.
* Read all existing categories.

About Product:

* Create a new product.
* Update existing product.
* Delete existing product.
* Read all existing product.

About Account:

* Register a new account.
* Delete an existing account.
* Read all existing accounts.
* Login by using user name and password registered.
* Authorization for admins by using the account.

## 1.6. Scope of work

There are 2 parts of the scope of work in this project:

* Build a multi-page, online shopping website.

For both admin and customer to perform the following functions:

* View all items and view item details.
* View website introduction.
* See some ways to contact to store.
* Register new account or login existing account on provided link – Admin only.
* Add items to their cart

For administrator only to perform the following functions:

* Edit and view their account information.
* Before coming to the admin site, they must log in by using an account on the provided link.
* View, add, edit or delete all items.
* View, add, edit or delete all providers.
* View, add, edit or delete all categories.
* View, edit or delete all accounts.

## 1.7. Customer acceptance criteria

The table below shows some acceptance criteria:

|  |  |
| --- | --- |
| **No.** | **Customer’s Acceptance Criteria** |
| 1 | The homepage should be built with sections and can include a header section with pictures of different items. |
| 2 | The website should have links that allow users to navigate through a variety of other items, categories, and more. |
| 3 | Along with the photo, the item name and cost must be included. |
| 4 | Customers can see several ways to contact the store at the About us page. |
| 5 | Customers can add items to their cart. |
| 6 | Customers can place an order in their shopping cart. |
| 7 | Customers can read the introduction on About us page to understand the page |
| 8 | Customers can read the information of the items on the detail page. |

Table 1: Customer acceptance criteria

# Chapter 2 – System Design

## 2.1 About Symfony

Finally, what is Symfony? It is still a relatively new PHP framework and reusable PHP components vendor. It is both a business and a community of more than 600,000 developers and users from various nations. And from there, Symfony's utility derives. It has always followed the most recent trends because of its adaptability and vibrant community. Due to its open-source nature, Symfony may be customized to meet the unique requirements of each and every developer by adding any necessary modules by themselves. sponsoring sponsoring sponsor sponsors. Every user has access to a wide range of opportunities because to the scale and diversity of the Symfony community. After all, you can always add anything specifically suited to your framework if you can't find what you need here (accesto, 2021).

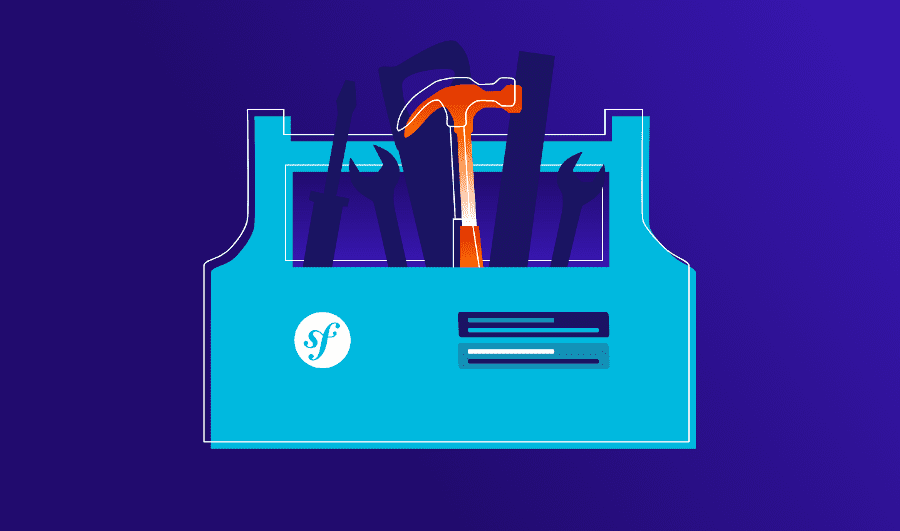


Figure 1: What is symphony

The Model-View-Controller Architecture:

To use PHP Framework Symfony in a more efficient way, it can be structured with MVC pattern. This stands for Model View Controller, and it is an architecture pattern, which helps you design a website or web app in a much more structured, layered, and logical way. It does so by dividing your project into three pieces:

* Model, i.e., everything related to **databases and maintaining the data** itself.
* View stands for components responsible for**presenting data to a user**.
* Controller, which is an intermediate between the other two. It connects them by sending data from Model to View, or, in other words**, processes data, updating it when needed, and explains how it should be presented to the user**

## 2.2. Git/GitHub

**Git:**

Distributed version control software is called Git. The use of version control allows you to save changes without erasing earlier iterations. When a Git repository is distributed, any developer using it has a copy of the whole repository, including every commit, every branch, and every file. This is a significant distinction if you're used to working with centralized version control systems. Regardless of your prior experience with version control, there are a few things you should be aware of before using Git (github, 2022):

* Branches are light and inexpensive, so it's OK to have a lot of them.
* Git uses SHA hashes, which compress text files, to save changes. Git is thus a very effective version control system (VCS) for software development, but it is less effective for binary files like photographs and videos.
* You can work on one locally on your computer and connect it to a shared repository because Git repositories may be connected. You can easily collaborate with others and push and pull changes to a repository in this way.

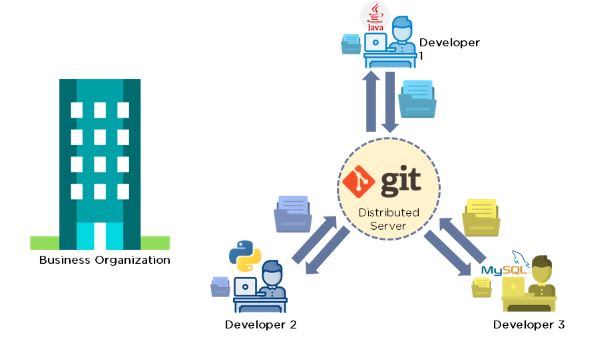


Figure 2: What is a git

**GitHub:**

GitHub is a for-profit company that offers a cloud-based Git repository hosting service. Essentially, it makes it a lot easier for individuals and teams to use Git for version control and collaboration. GitHub’s interface is user-friendly enough so even novice coders can take advantage of Git. Without GitHub, using Git generally requires a bit more technical savvy and use of the command line. GitHub is so user-friendly, though, that some people even use GitHub to manage other types of projects – like writing book. Additionally, anyone can sign up and host a public code repository for free, which makes GitHub especially popular with open-source projects. As a company, GitHub makes money by selling hosted private code repositories, as well as other business-focused plans that make it easier for organizations to manage team members and security. We utilize Github extensively at Kinsta to manage and develop internal projects (kinsta, 2022).



Figure 3: What is a Github

## 2.3 System Architecture Diagram

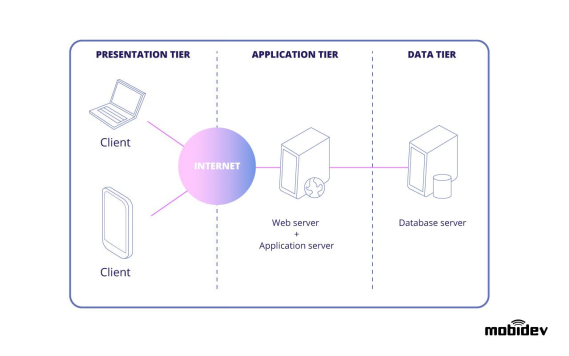


Figure 4: System Architecture Diagram

## 2.4 Use Case Diagram

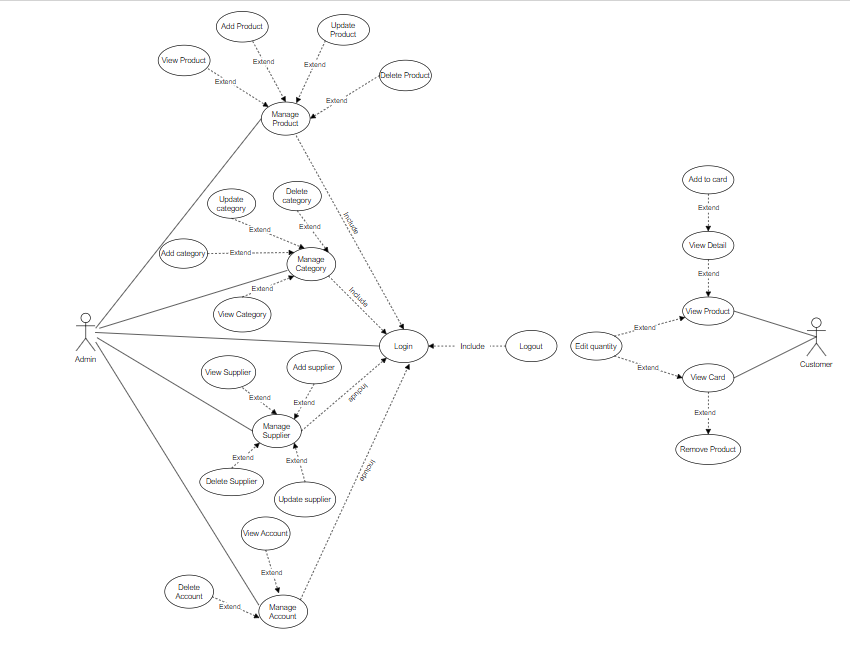


Figure 5: Use Case Diagram

As we can see in the use case diagram which consists of 2 parts:

* Admin: In this section, Admin has functions such as logging out, managing categories, suppliers, items, etc. Admin login is required to perform the above functions.
* Customers: In this section, customers can use functions such as view products, view details, add to cart... Customers no need to login to perform functions.

## 2.5 Entity Relationship Diagram



Figure 6: Entity Relationship Diagram

My diagram consists of 5 entities: product, category, supplier, user, cart.

* About product entity, it contains information about product such as id, name, price, created, image, quantity, cat\_ id, description, sup\_id.
* The category entity contains detailed information about the category such as id, cat\_name.
* The supplier entity contains information about the supplier such as id, sup\_name, address.
* Cart entity includes id, pro\_cart\_id, user\_cart\_id, quantity.
* User entity includes id, email, roles, password, name, address.

Relationships between entity sets:

* Category – product (1:N): A category can have many items, but each product is related to only one product type.
* Supplier – product (1:N): A supplier can have multiple entries, but each product is related to only one product supplier

## 2.6 Sitemap

### 2.6.1 Home Page

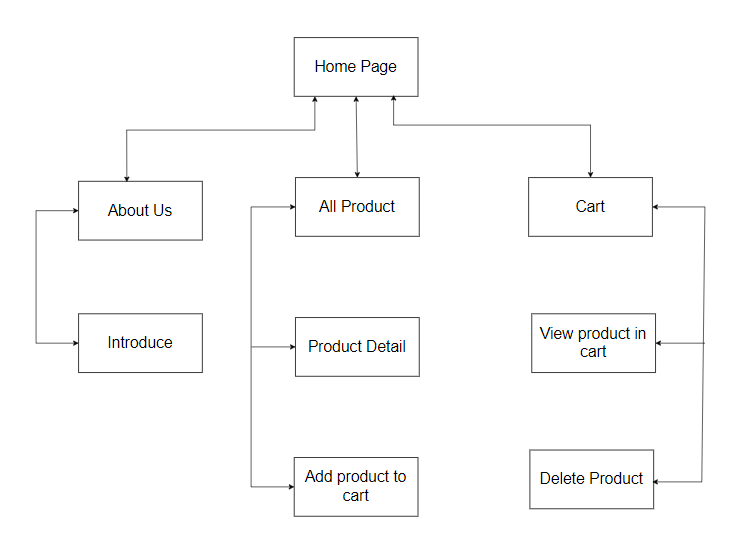


Figure 7: Home site

### 2.6.2 Administer

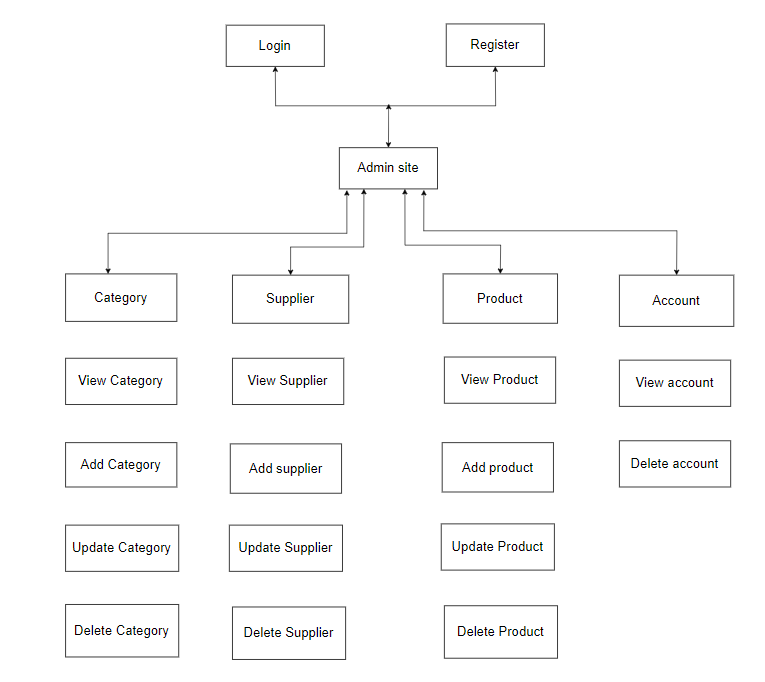
****

Figure 8: Admin site

## 2.7 Wireframes

* **Customer:**

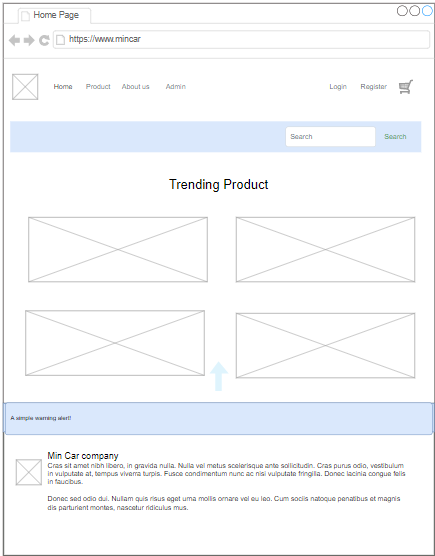


Figure 9: Home Page

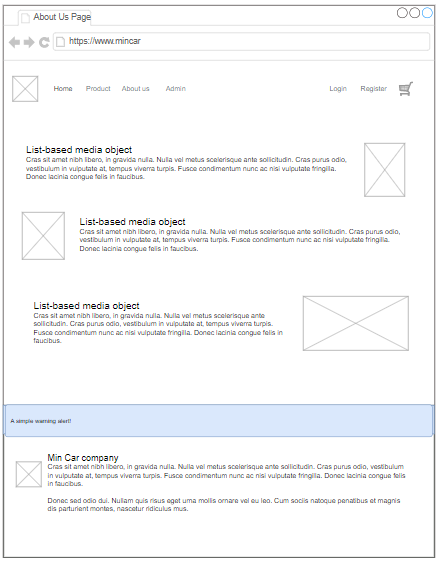


Figure 10: About us page

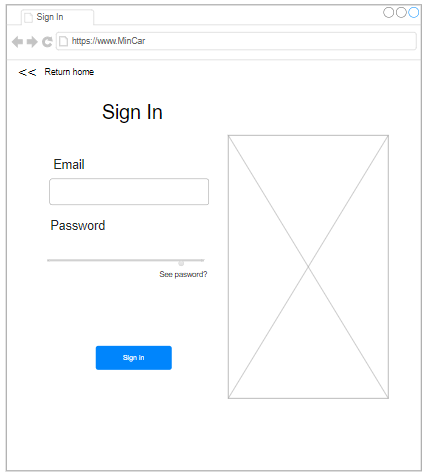


Figure 11: Sign In page

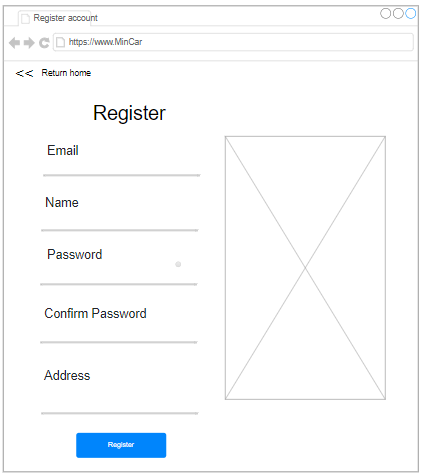


Figure 12; Register account

* **Admin – Product:**



Figure 13: View Product

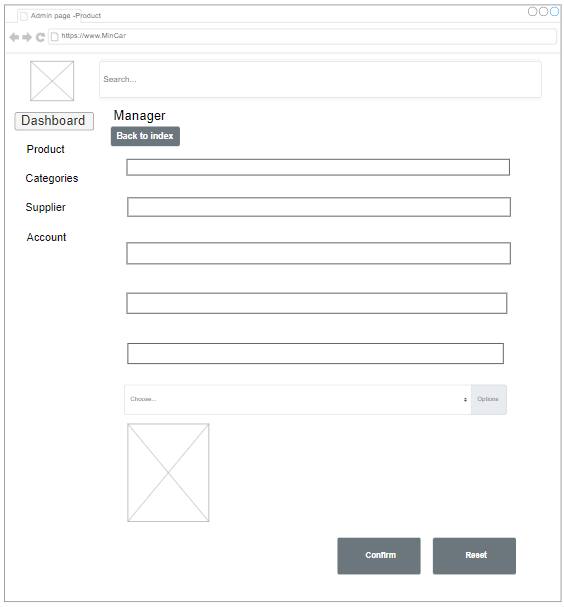


Figure 14: Add new product

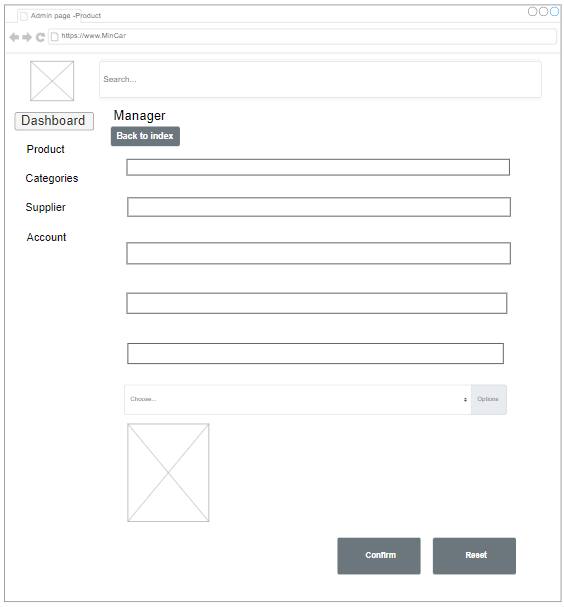


Figure 15: Update product

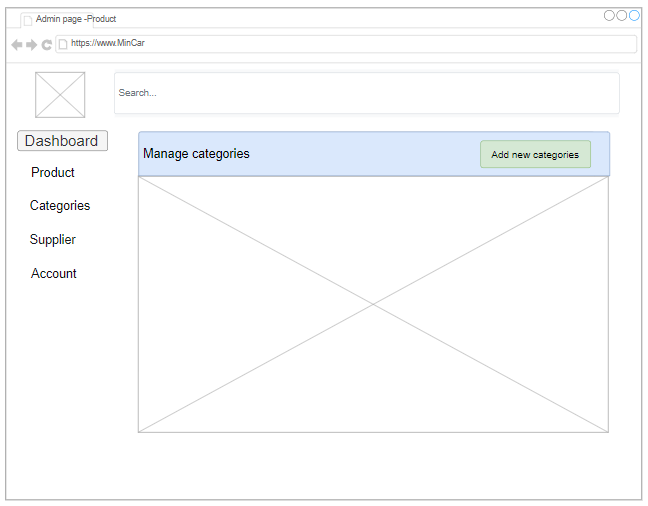


Figure 16: View categories

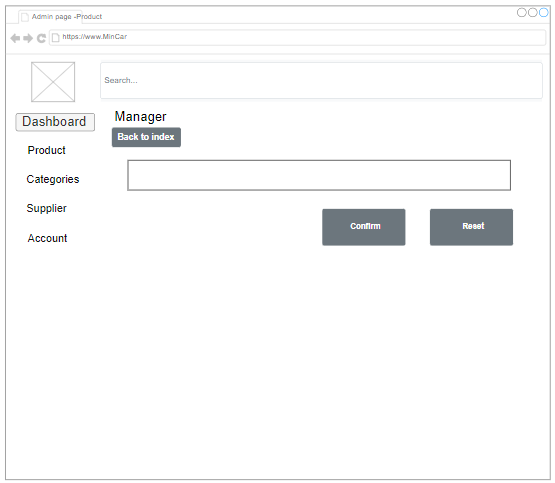


Figure 17: Add categories

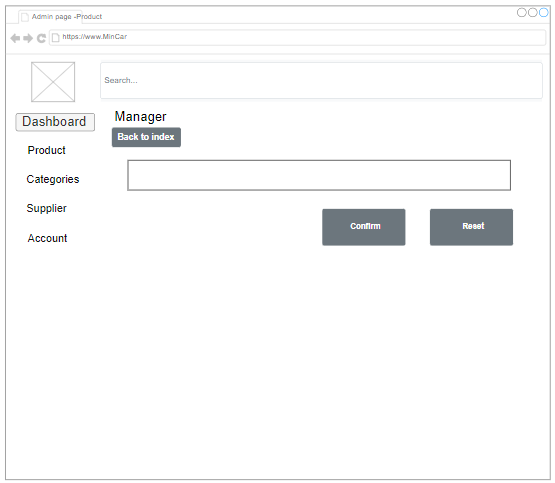


Figure 18: Update category

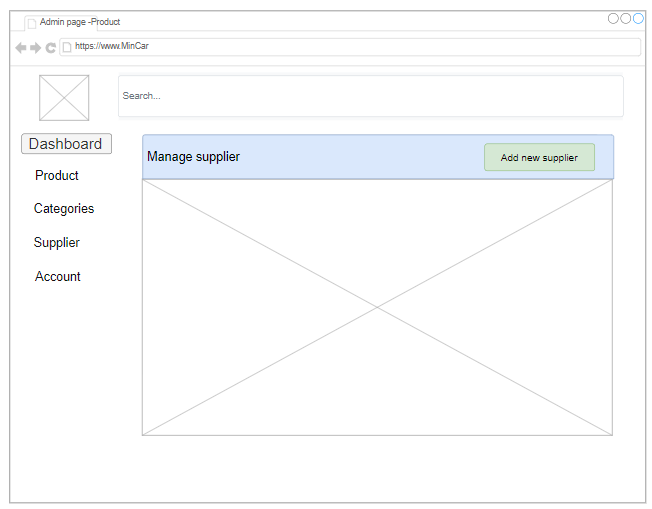


Figure 19: View Supplier

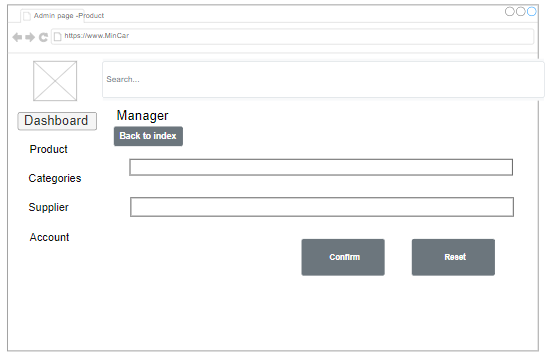


Figure 20: Add supplier

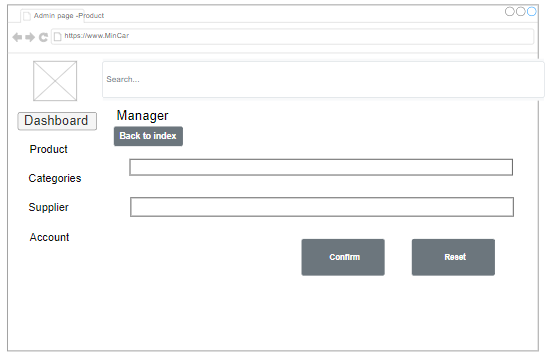


Figure 21: Update supplier

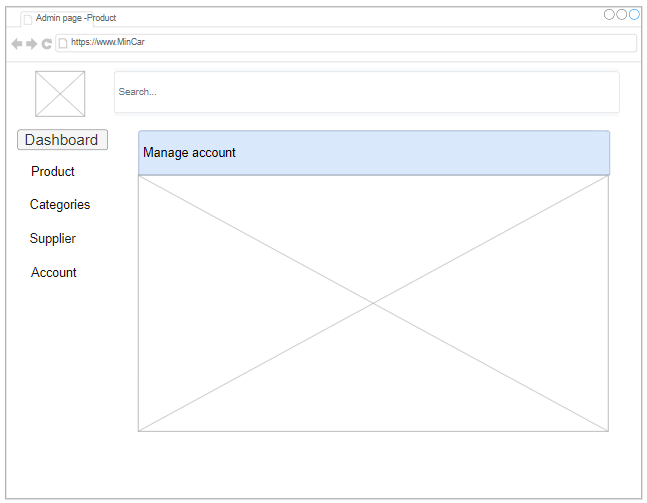


Figure 22: View account

# Chapter 3 – Implementation

## 3.1 Sample Source Code

* Structure of system

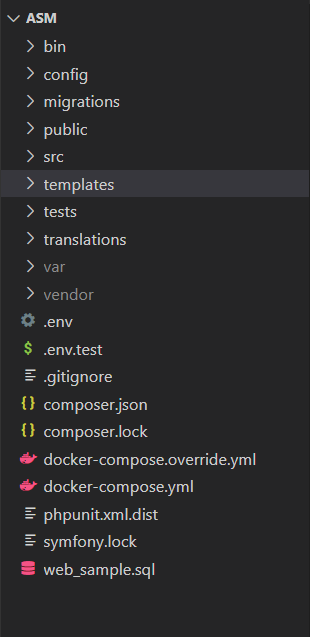


Figure 23 - Structure of system

* Overview Cotroller in system

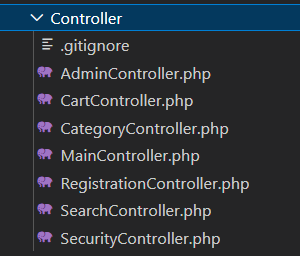


Figure 24 - Controller in system

* Overview Entity in system

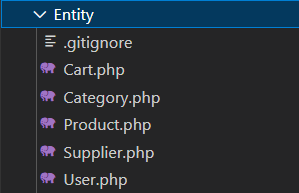


Figure 25 - Entity in system

* Overview Repository in system

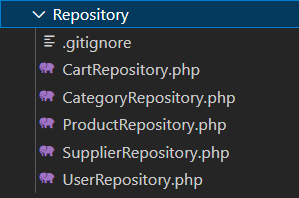


Figure 26 - Repository in system

* Overview Template in system

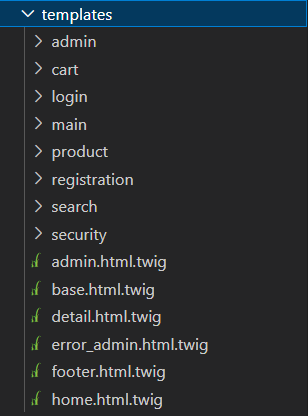


Figure 27 - Template in system

* Show products function

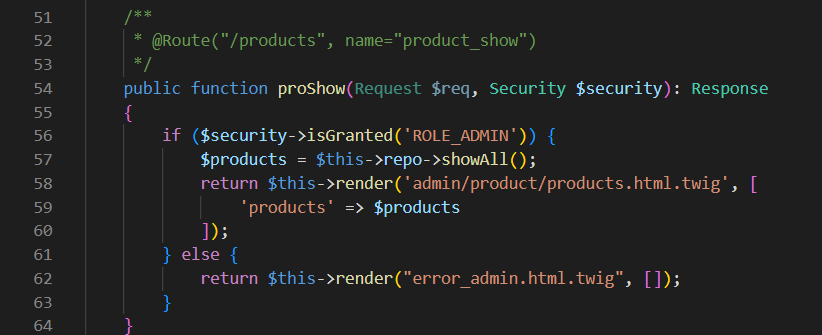


Figure 28 - Show products function

This is the function that displays the products contained in the database. First, this function checks the user's role. If admin, the site returns admin/product/products.html.twig. Otherwise, if you are not an administrator, you will not be able to access it and will return error\_admin.html.twig

* Template of admin/product/products.html.twig

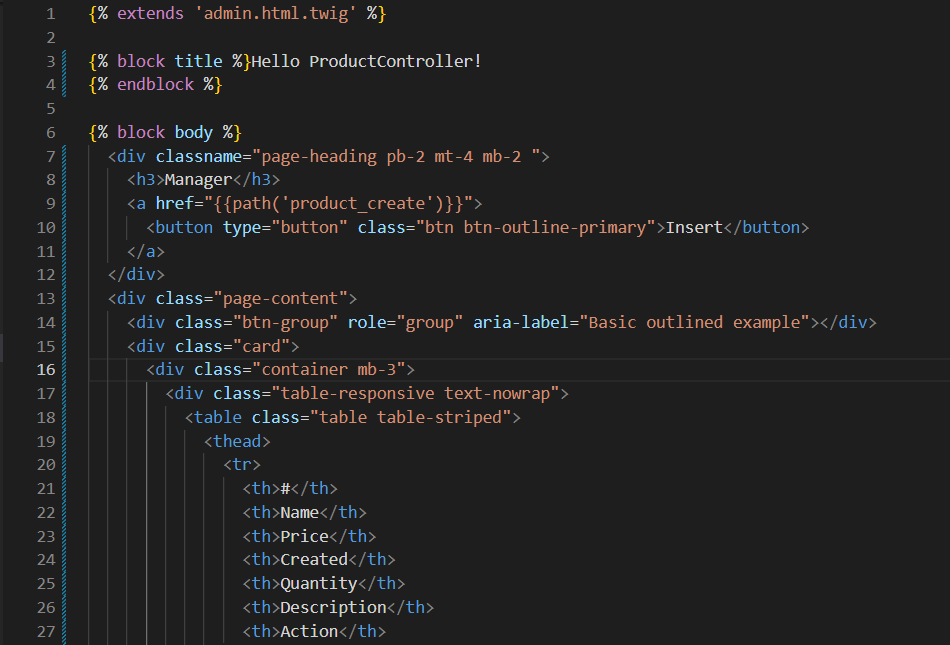


Figure 29 - Template of admin/product/products.html.twig part 1



Figure 30 - Template of admin/product/products.html.twig part 2

Here the website will display the products included in the store's database and the functions that the administrator can edit, add and remove products

* Template of Error\_admin.html.twig

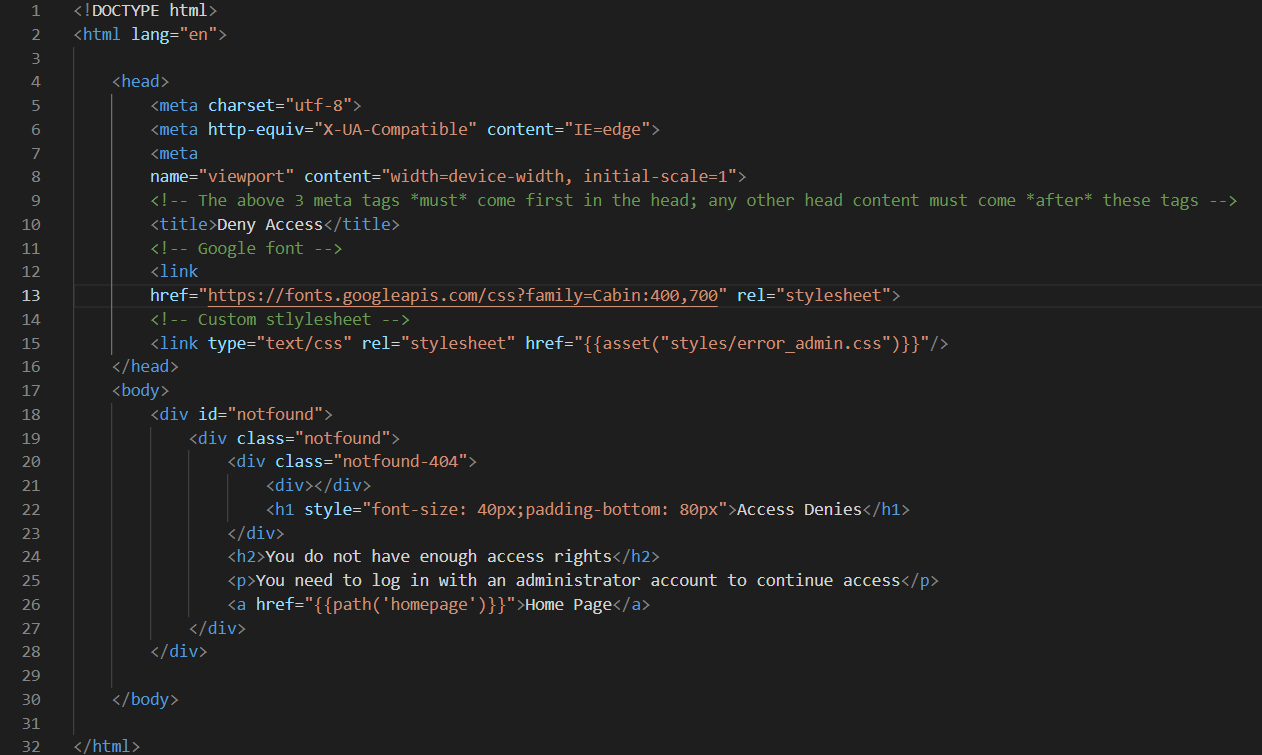


Figure 31 - Template of error\_admin.html.twig

This template is responsible for displaying on the screen with the message that the user does not have access to the page without sufficient permissions. And the user can only return to the home page to continue doing other actions

* Add new product function



Figure 32 - Add new product function

This function will result in the creation of a new product. When a user uses this function to create a new product, a form with information about the new product, including its name, price, description, created, quantity, image, supplier ID, and category ID, is created for them to fill out. When a user enters legitimate data and clicks the submit icon, the function validates the information and adds it to the database. The function will hash the picture's name using the MD5 algorithm and determine its file extension before adding it to the database (jpg or png). The picture will then be transferred to the images section, which has the path '../public/uploads/images,' and added to the database.

* Add new product in ProductRepository

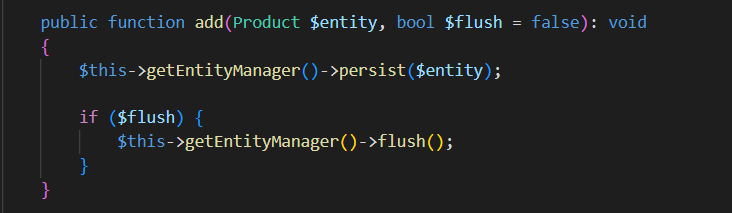


Figure 33 - Addnew product in ProductRepository

In order for the add function to do its job, the function will first run because the flush variable has been set to "true", which means it will flush the database all the changes to the objects that have been changed. queue up to this point based on the information the user entered.

* Template of Admin/product/add.html.twig



Figure 34 - Template of admin/product/add.html.twig part 1



Figure 35 - Template of admin/product/add.html.twig part 2



Figure 36: Template of admin/product/add.html.twig part 3

This is the form of the add product page, including the product attributes and the Confirm button

* Show detail product function

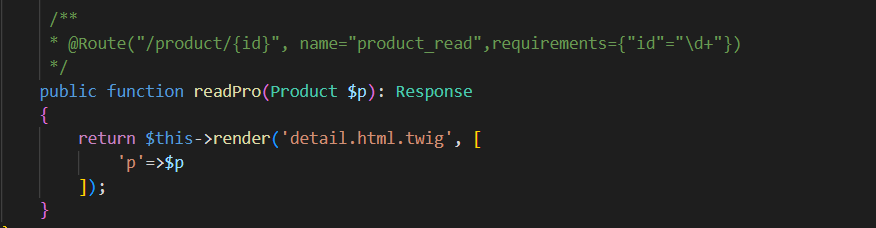


Figure 37 - Controller of show detail product

When user clicks on a product on the home page or view details, the site performs an action called product\_read to display the details of the product based on that product's ID and return details.html.twig

* Edit product function



Figure 38 – Controller of edit product

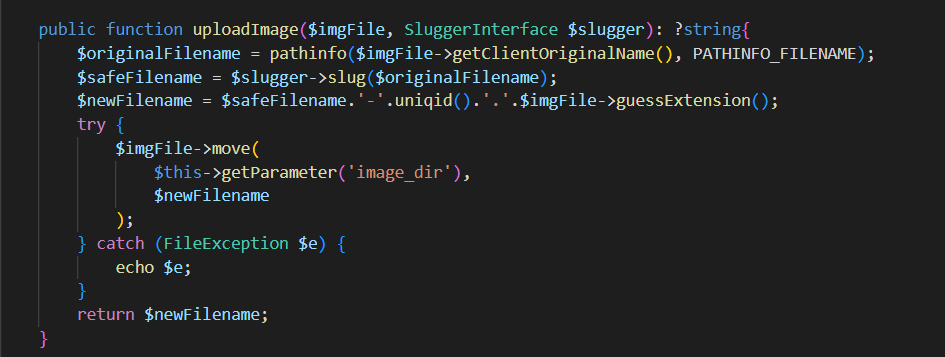


Figure 39 - Function upload image in Admin Contoller

This is the page used to update and change product information based on product ID. After the user presses the Update button, the information that the user changes will be updated in the database.

* Template of Edit product function

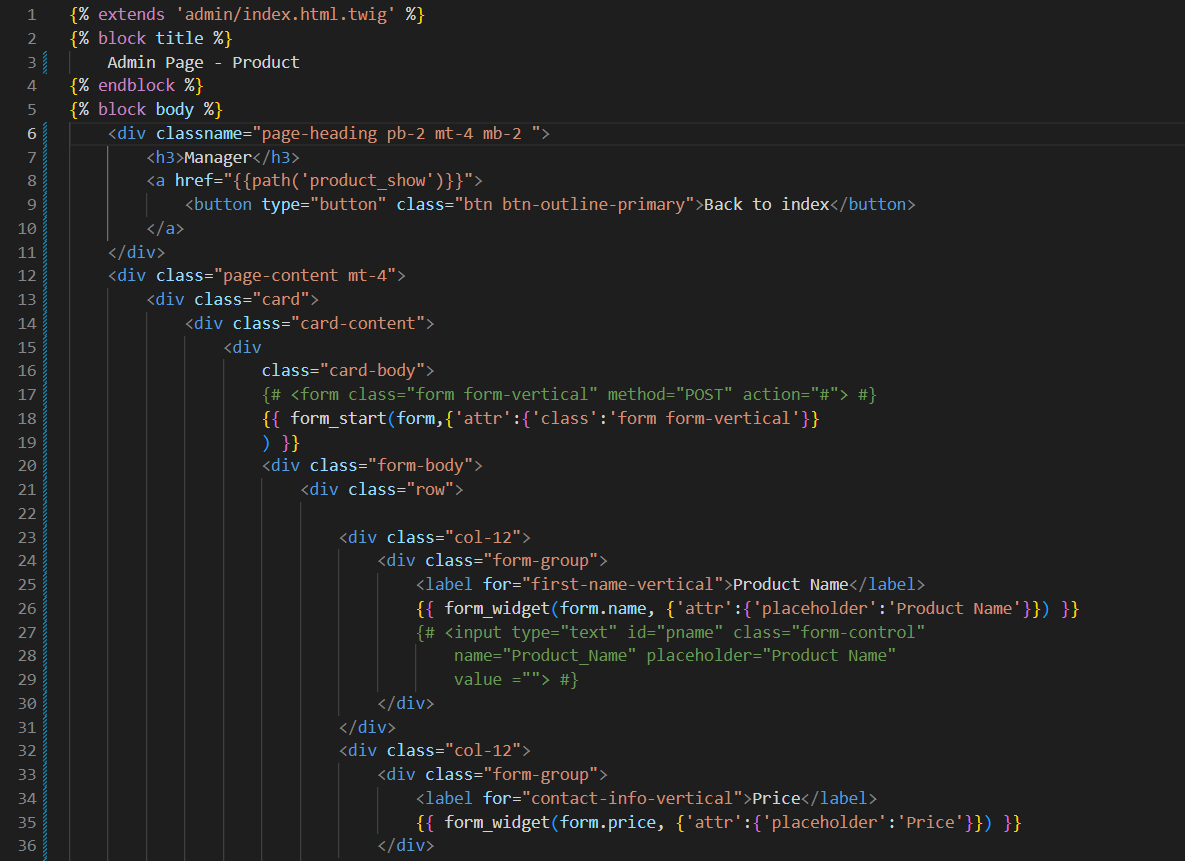


Figure 40 - Template of edit product part 1



Figure 41 - Template of edit product part 2

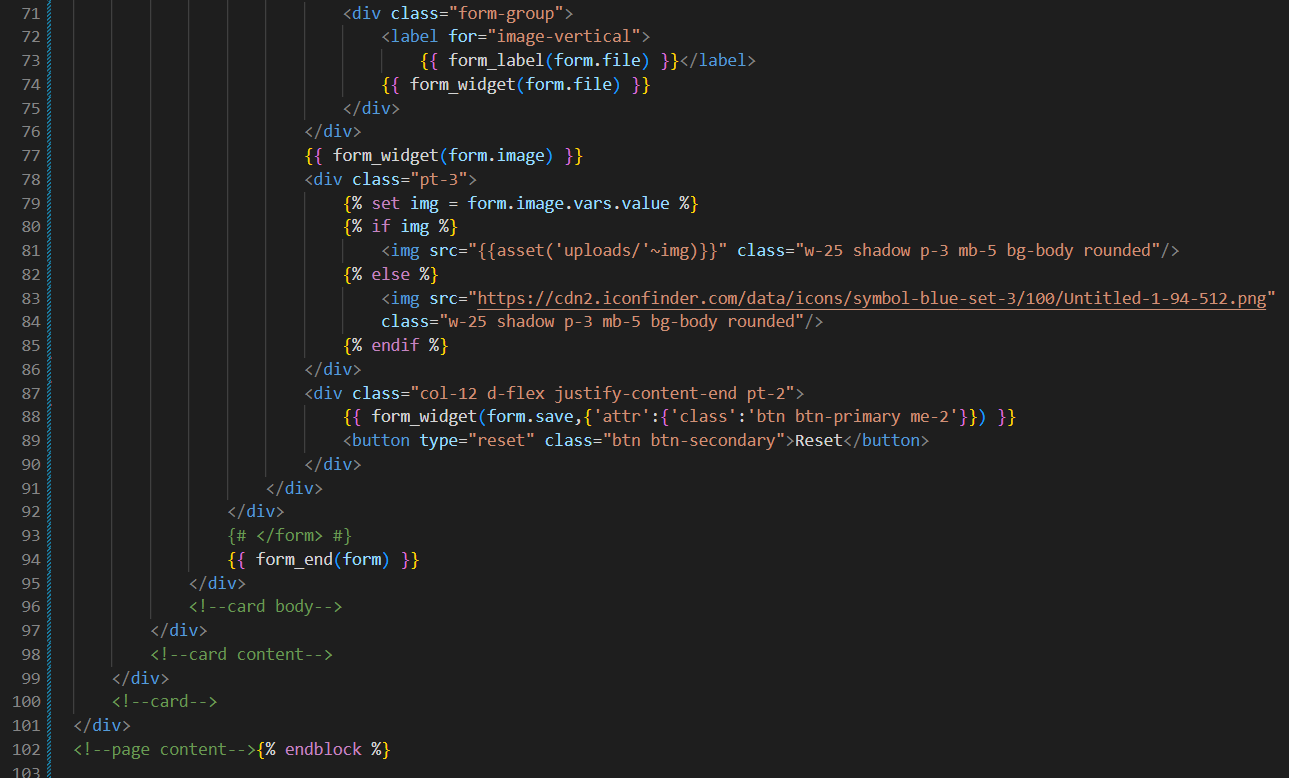


Figure 42 - Template of edit product part 3

* **Delete product function**



Figure 43 - Function delete product

The website uses the product's ID to delete the product from the shop when a user selects the garbage can symbol on the product administration page. This method obtains the product id and crsf token when the user hits the icon. This method compares these values to the values received when the "clear" button is pressed. If it does, the function deletes all product details and redirects the user to the product viewer function.

* Delete product in ProductRepository

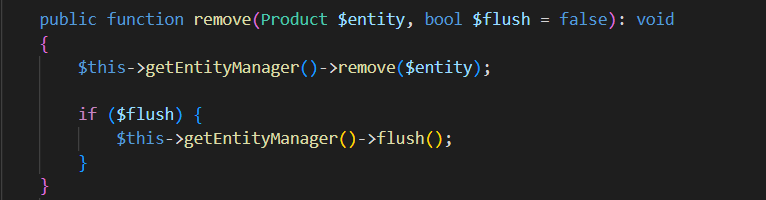


Figure 44 - Function delete product in ProductRepository

Because the variable flush has already changed to "true," when the user selects the "Delete" option to erase a product, the function remove, which appears like "$productRepository->remove($product, true)," will be performed. This implies that all modifications to items queued up to this moment will be flushed to the database.

## 3.2 Images of final Application

* Home page without login

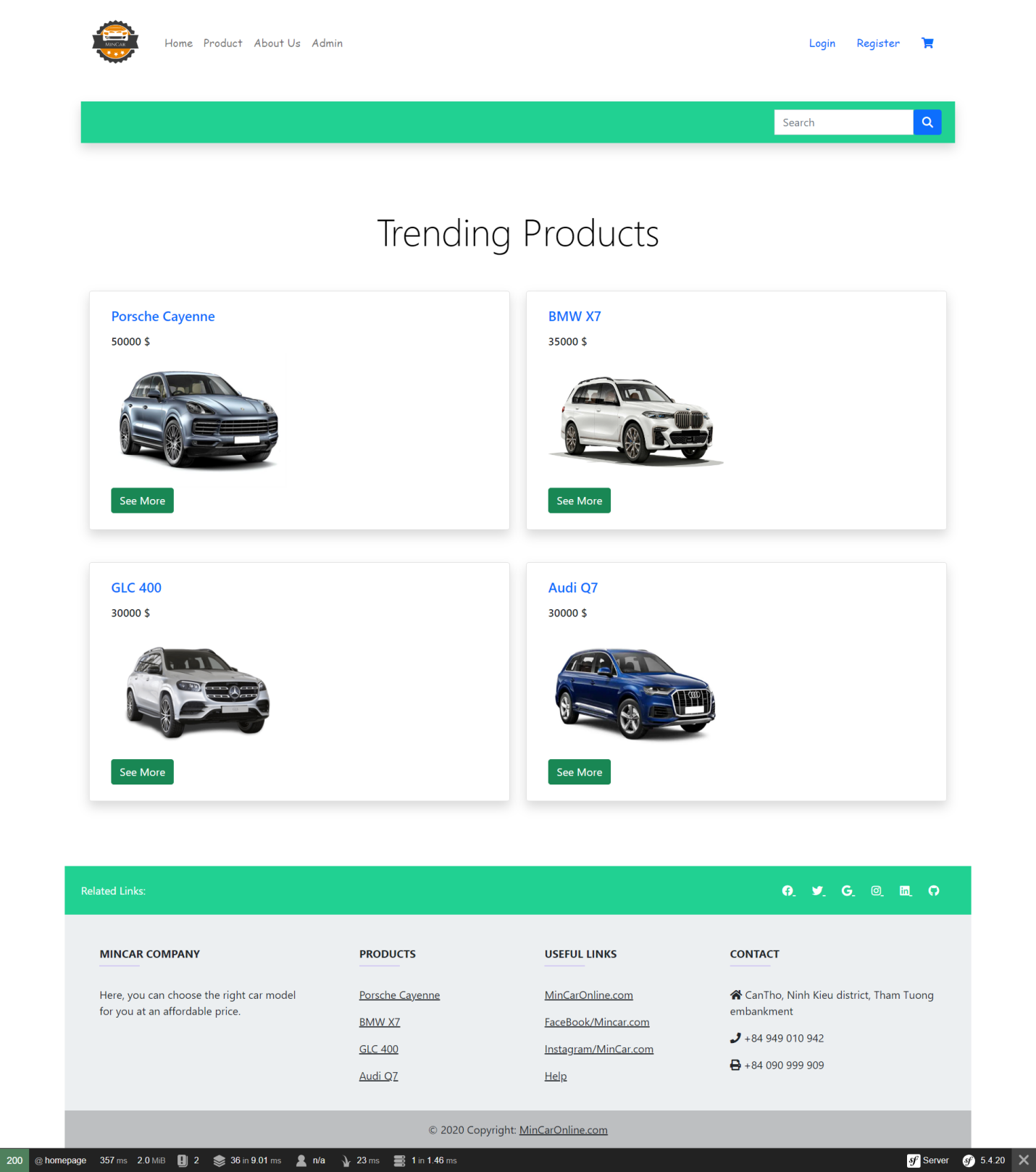


Figure 45 - Home page without login

* Register page

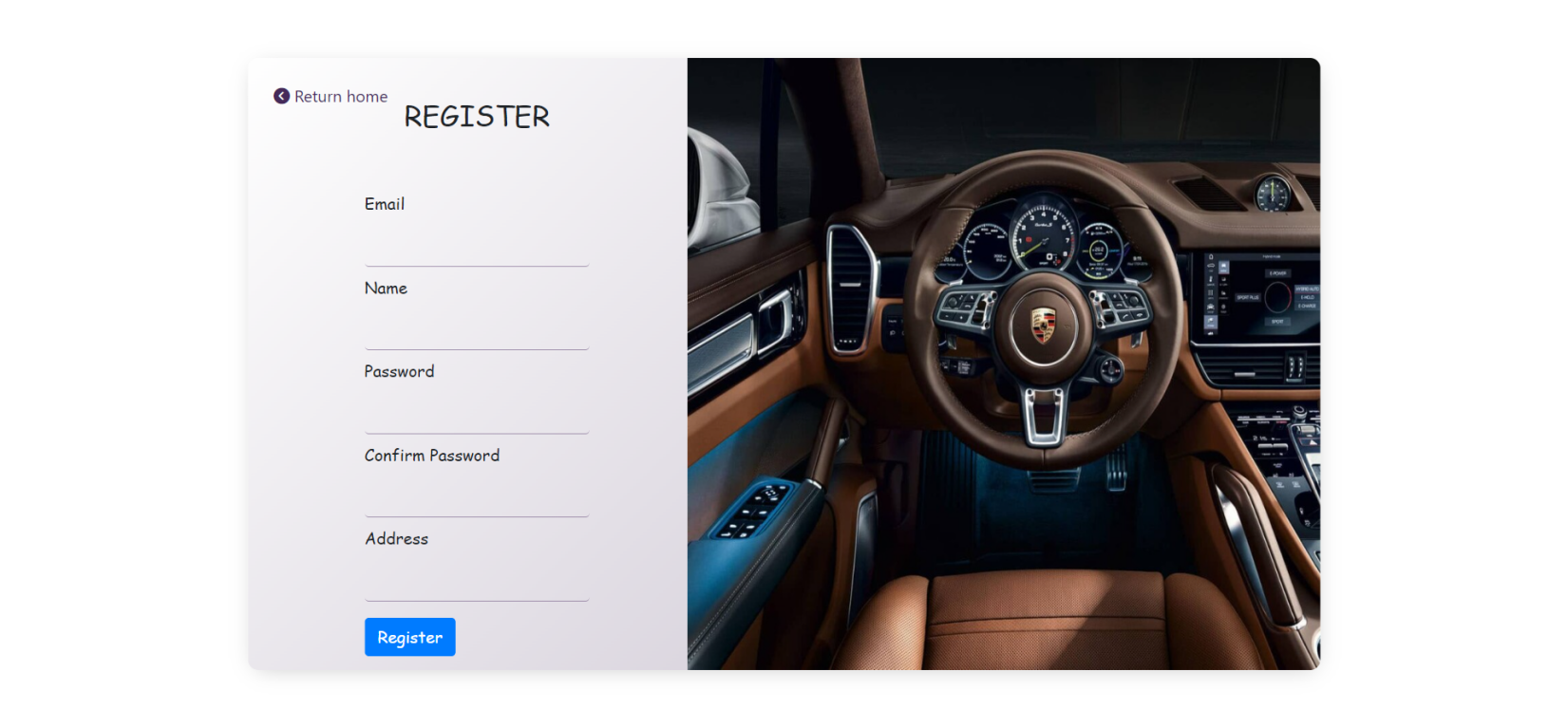


Figure 46 - Register page

* Login page

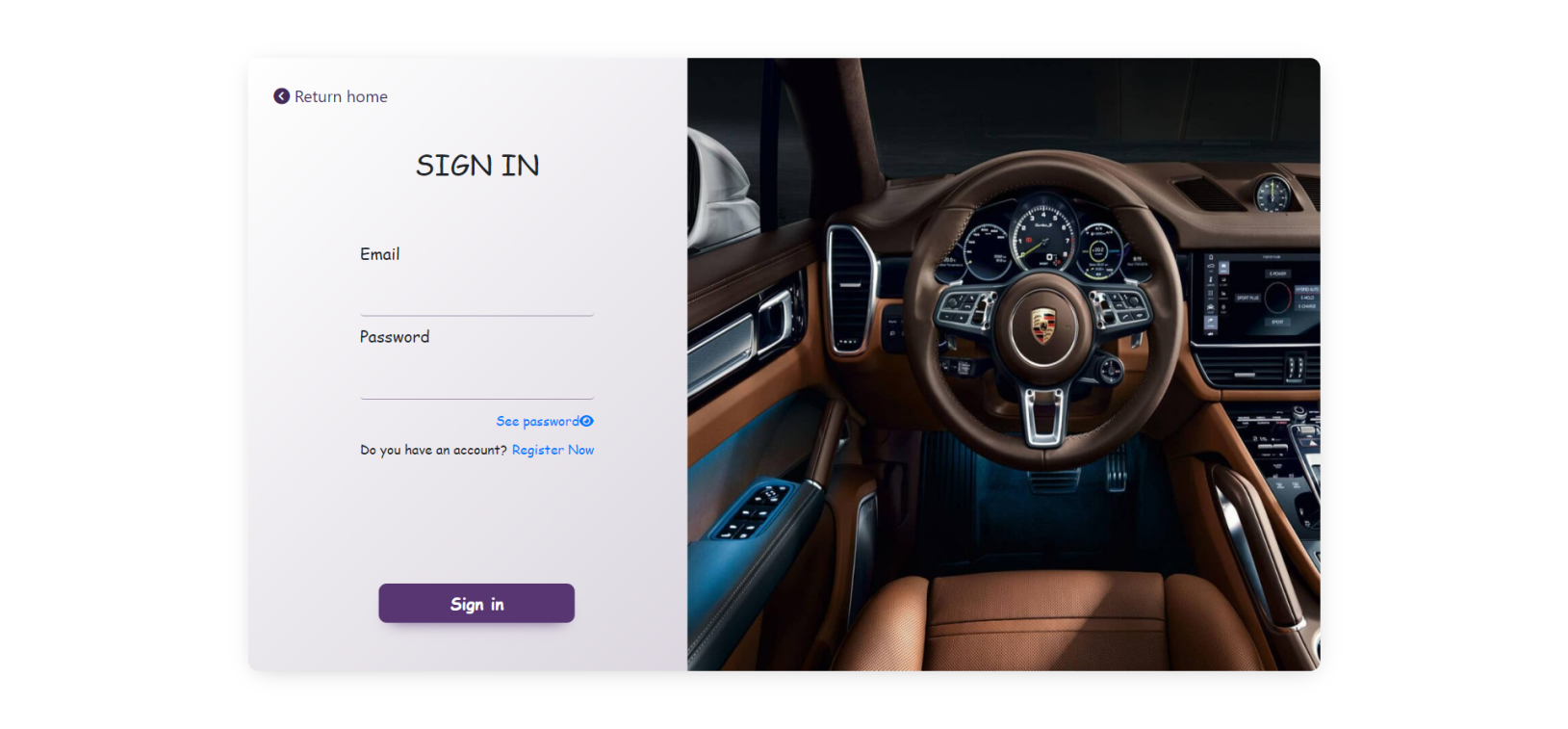


Figure 47 - Login page

* Home page with User

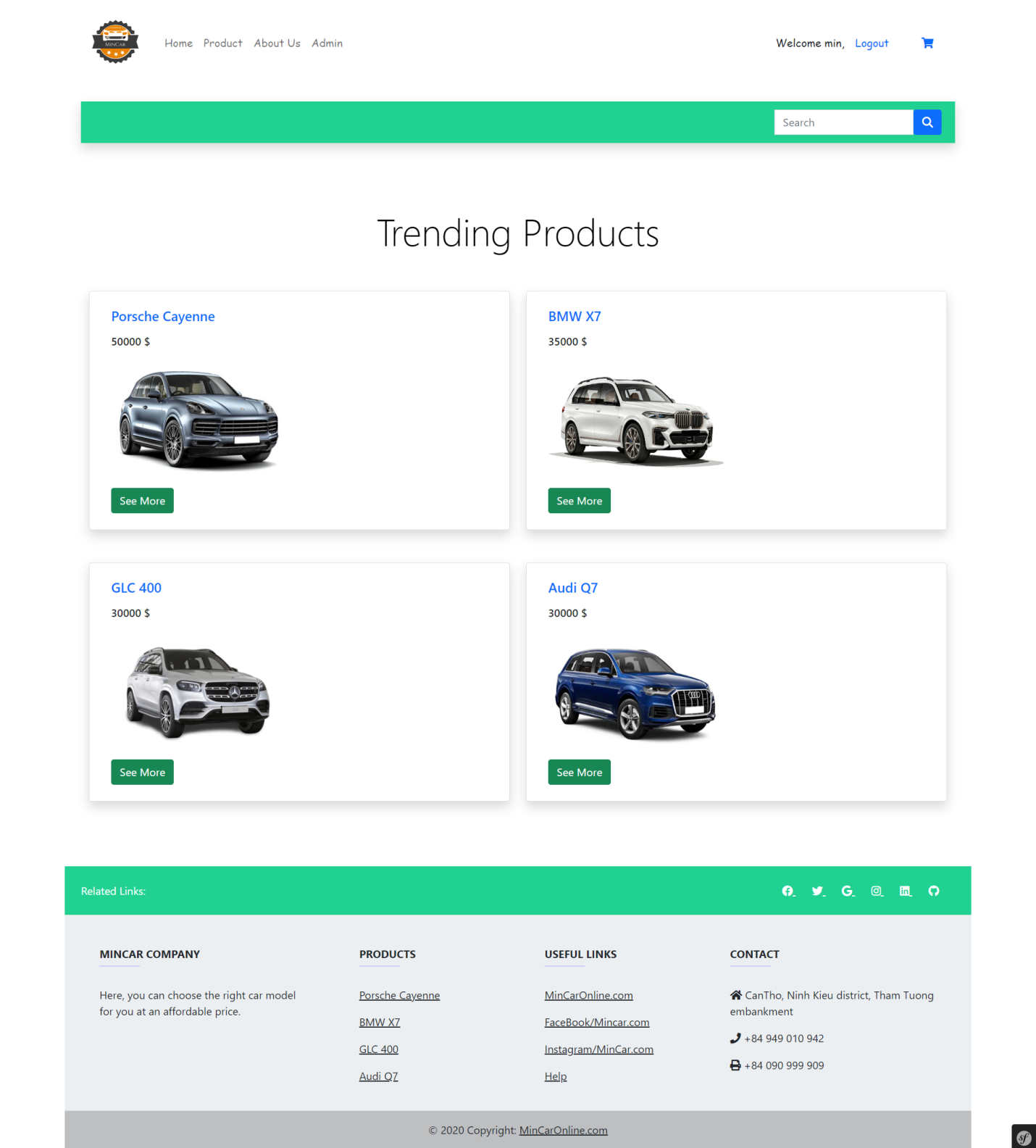


Figure 48 - Home page with login user

* Home page with Admin

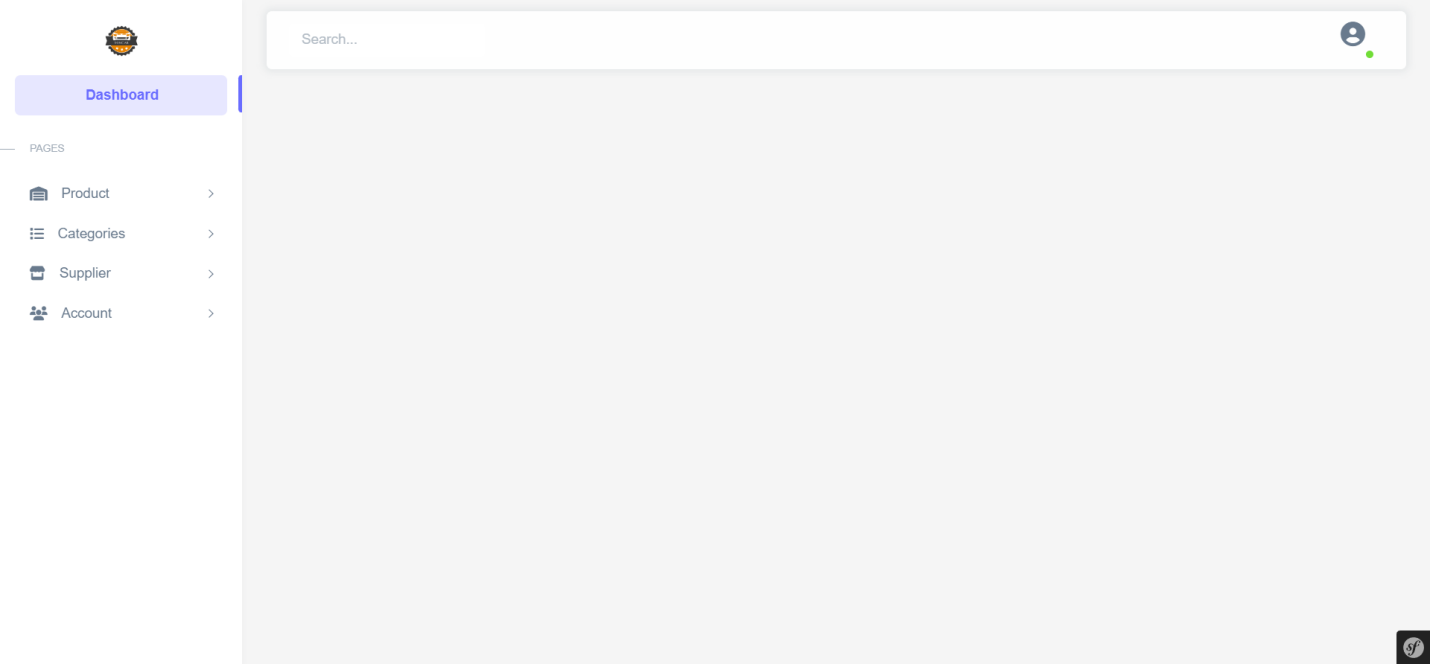


Figure 49 - Home page with login admin

* Product page

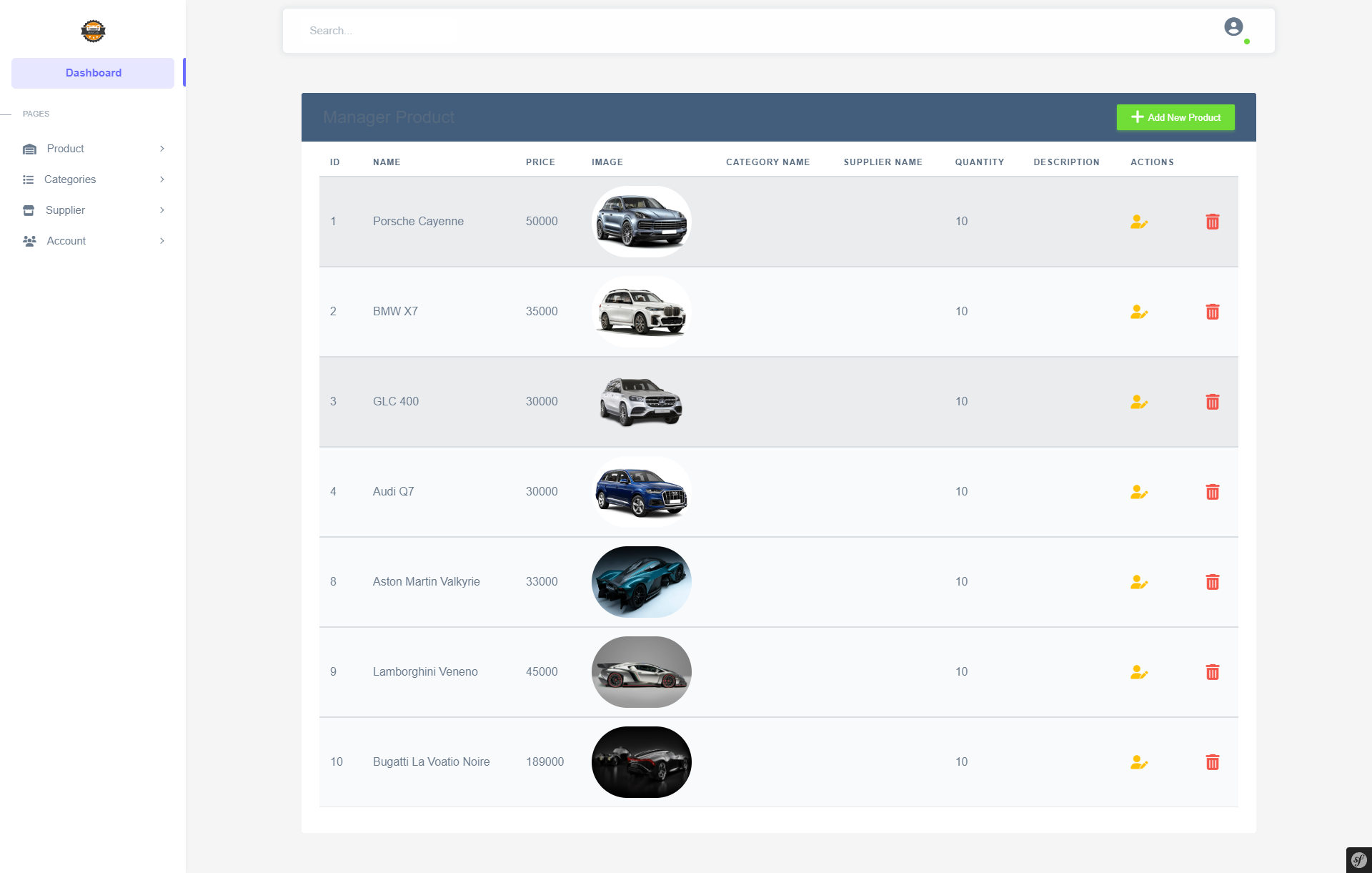


Figure 50: Product Page

* Product Manager Page

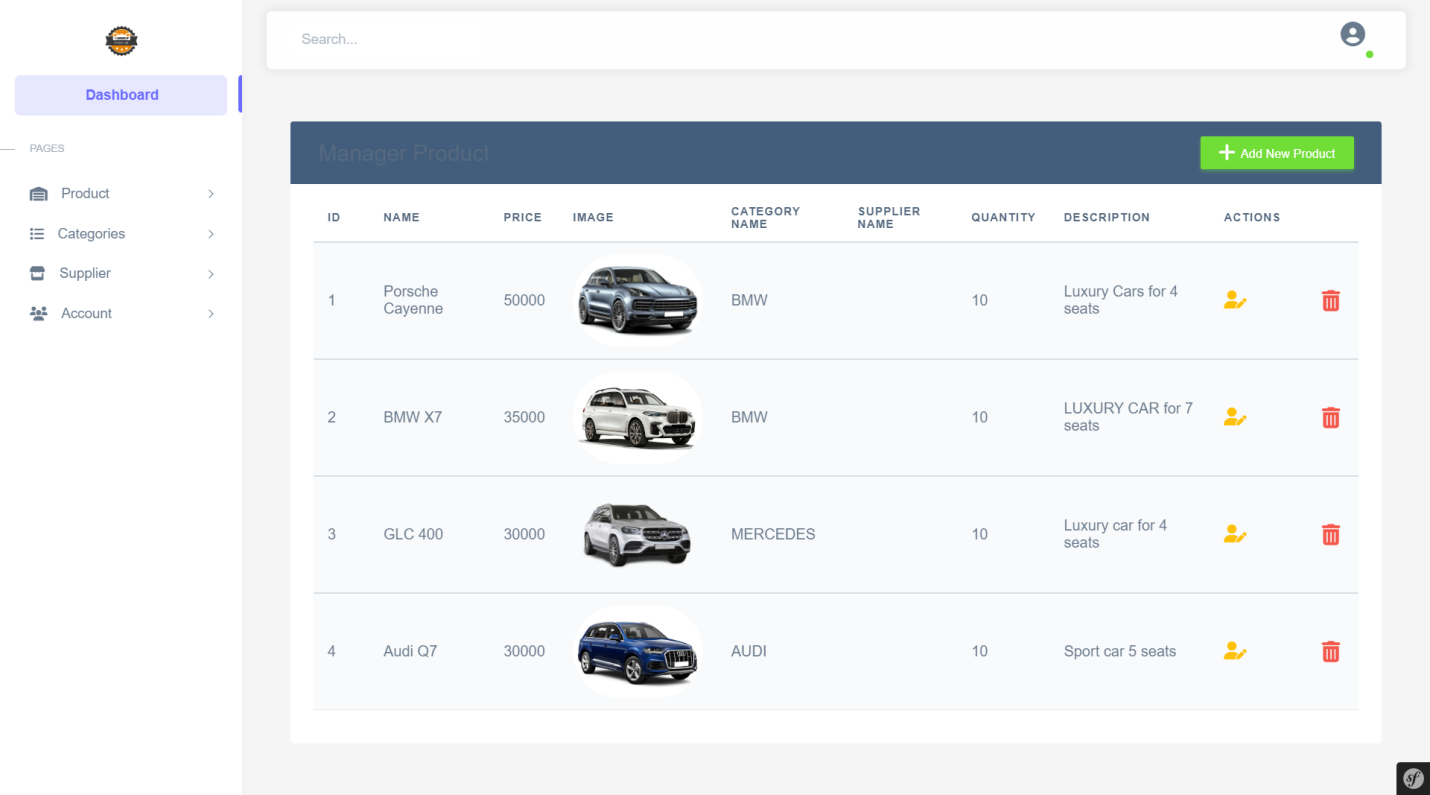


Figure 51: Product Manager Page

* Add product page

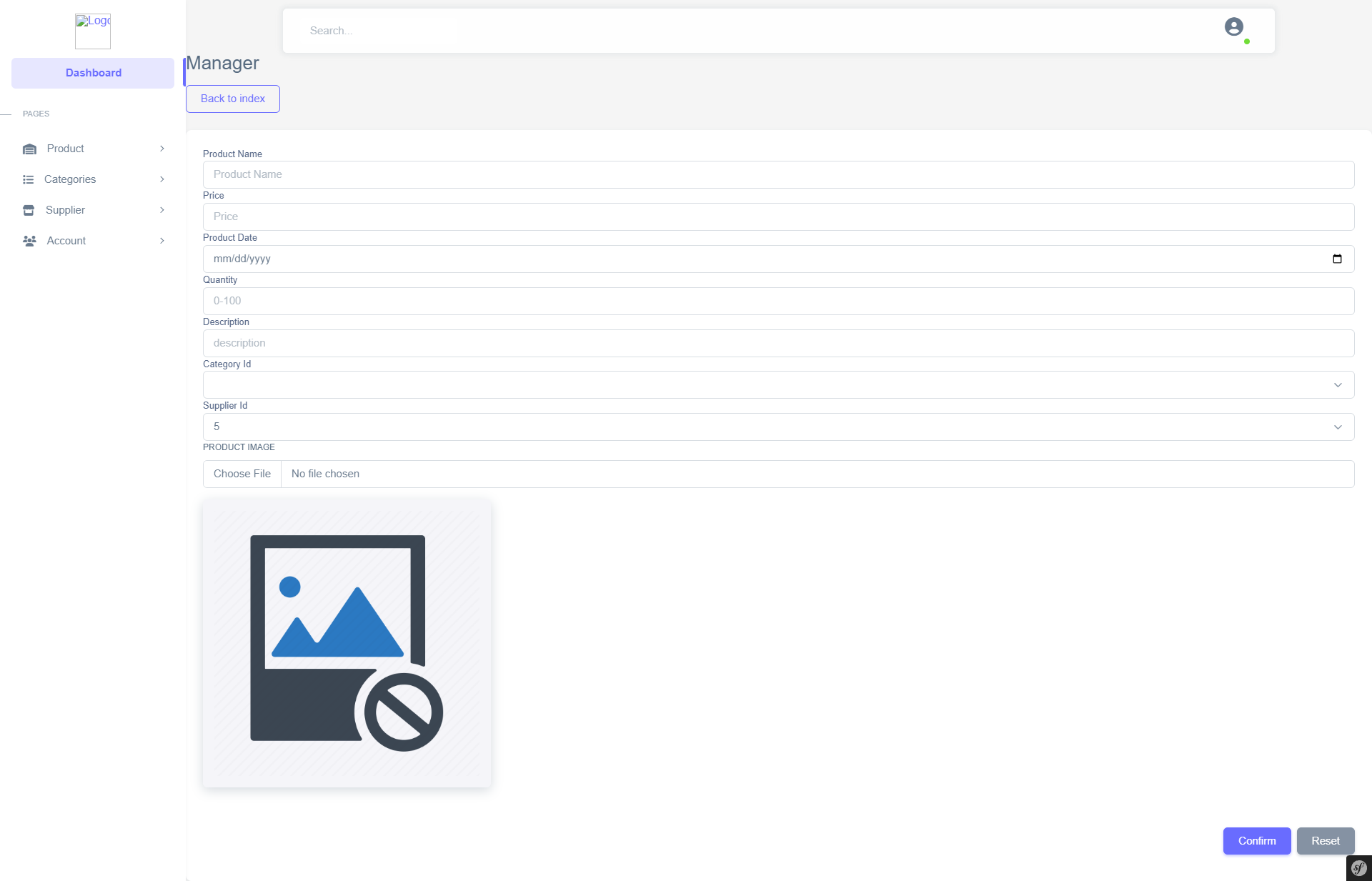


Figure 52: Add Product Page

* Edit product page

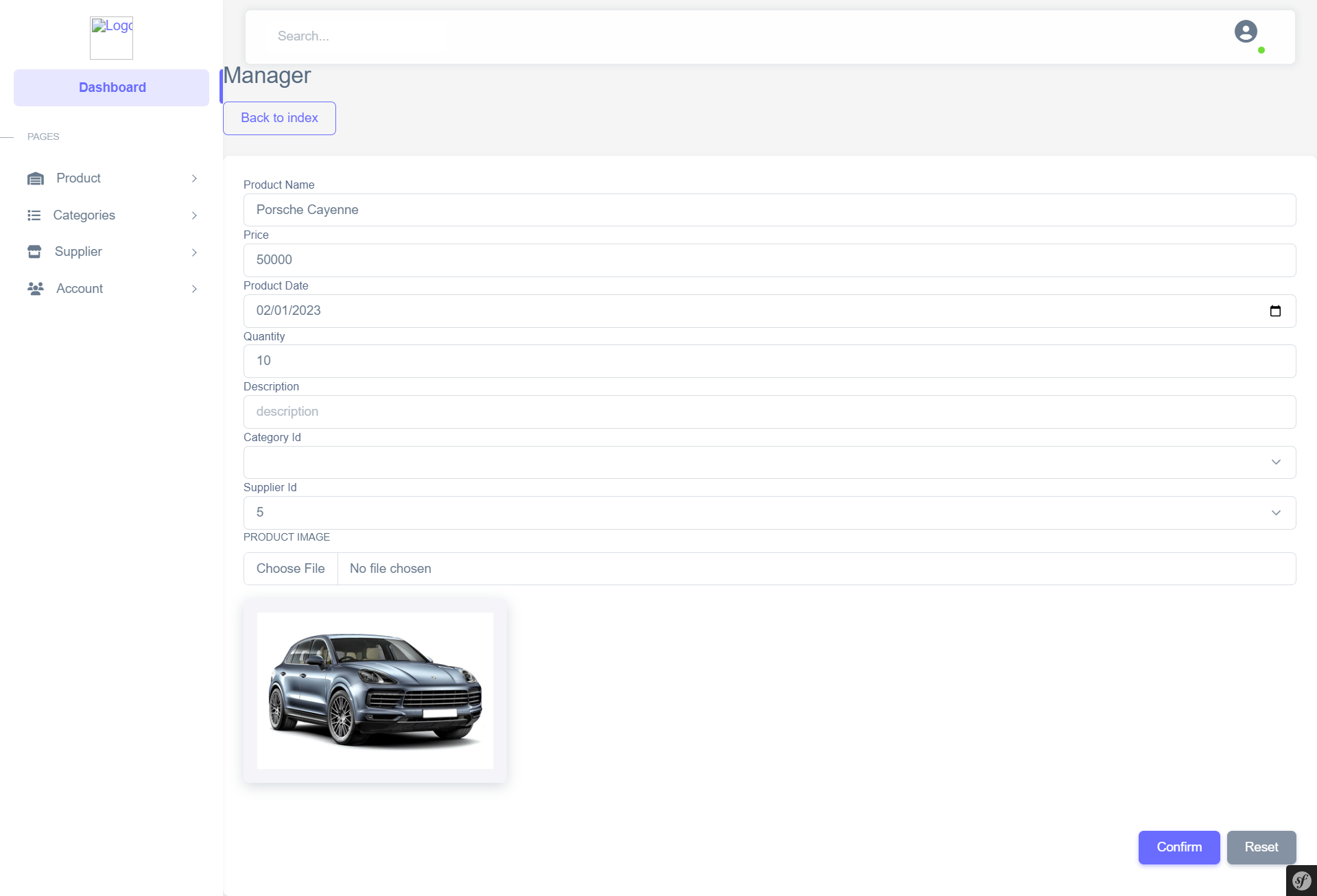


Figure 53: Edit Product Page

* Categories manager page

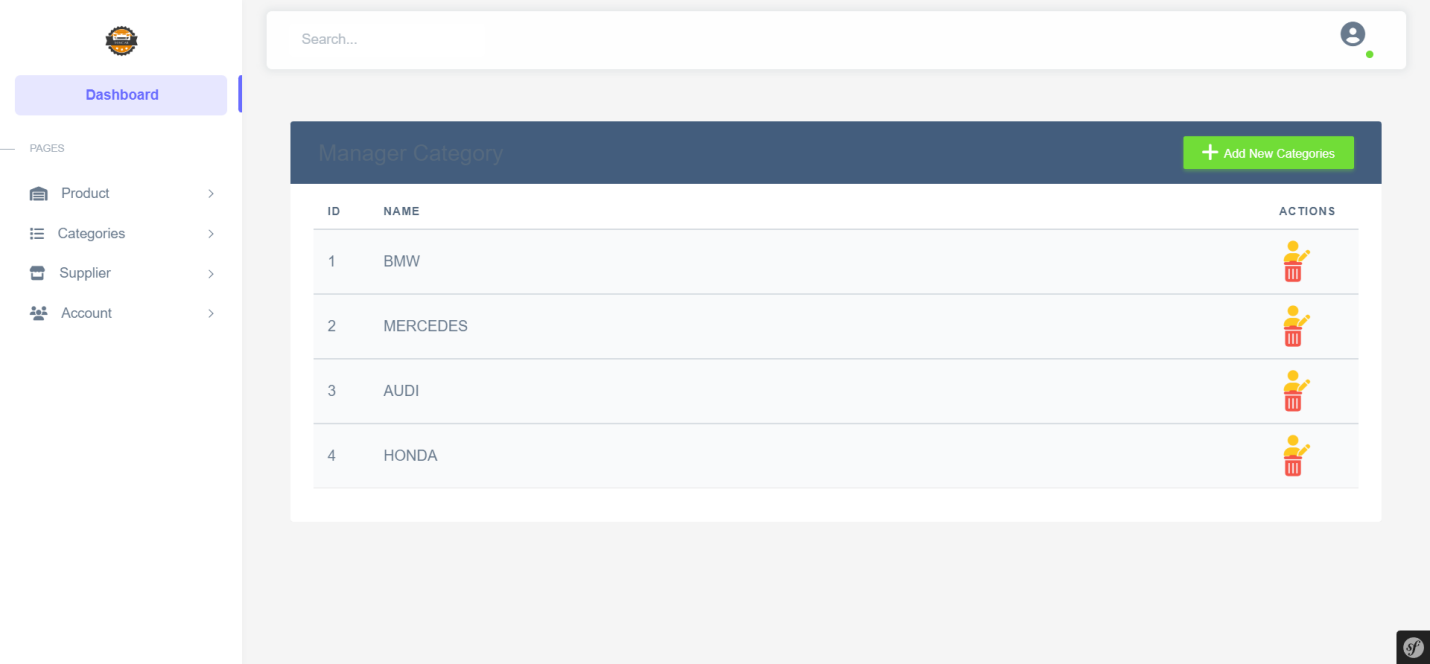


Figure 54: Categories Manager Page

* Add category page

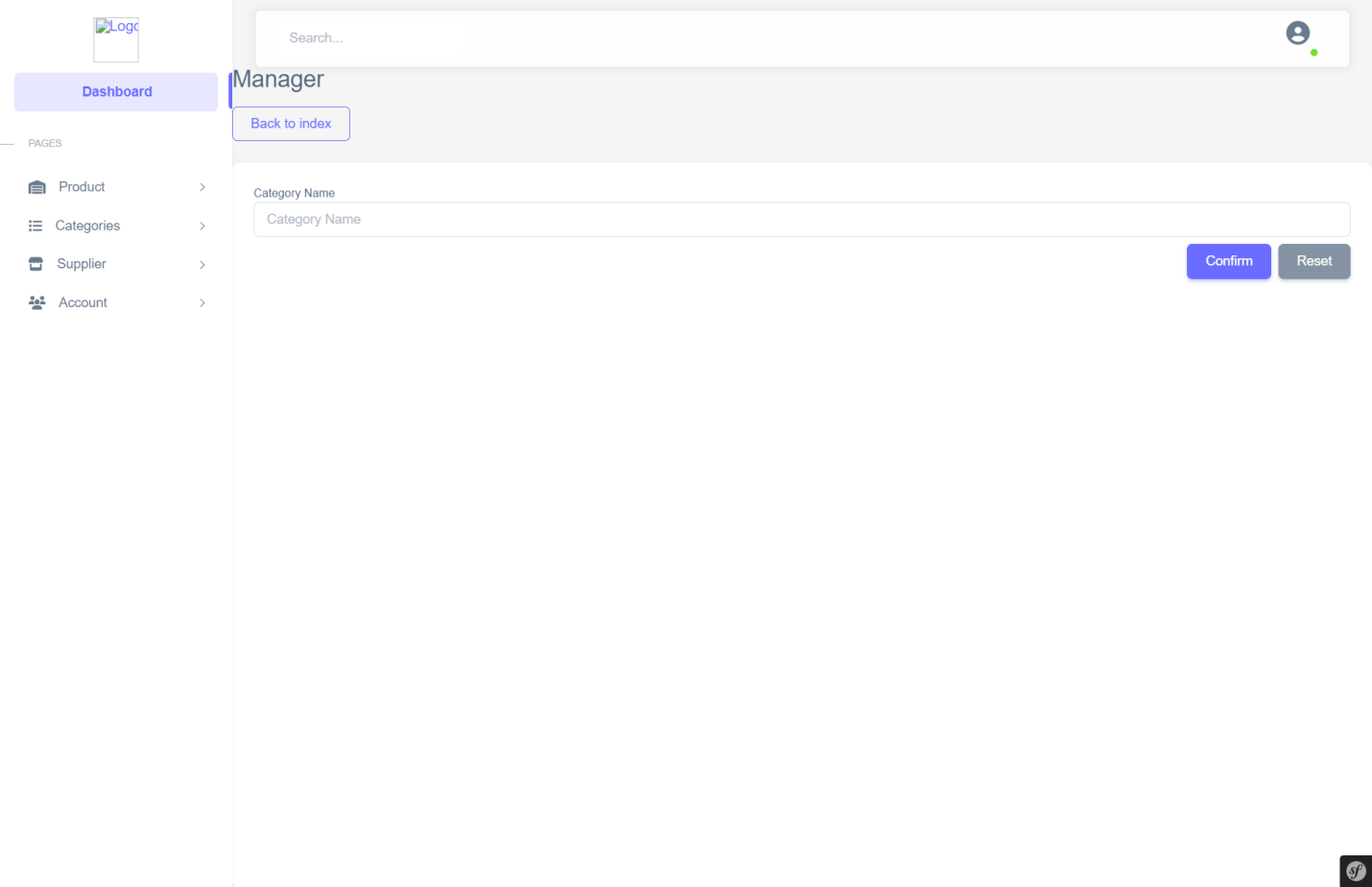


Figure 55: Add Category Page

* Edit category page

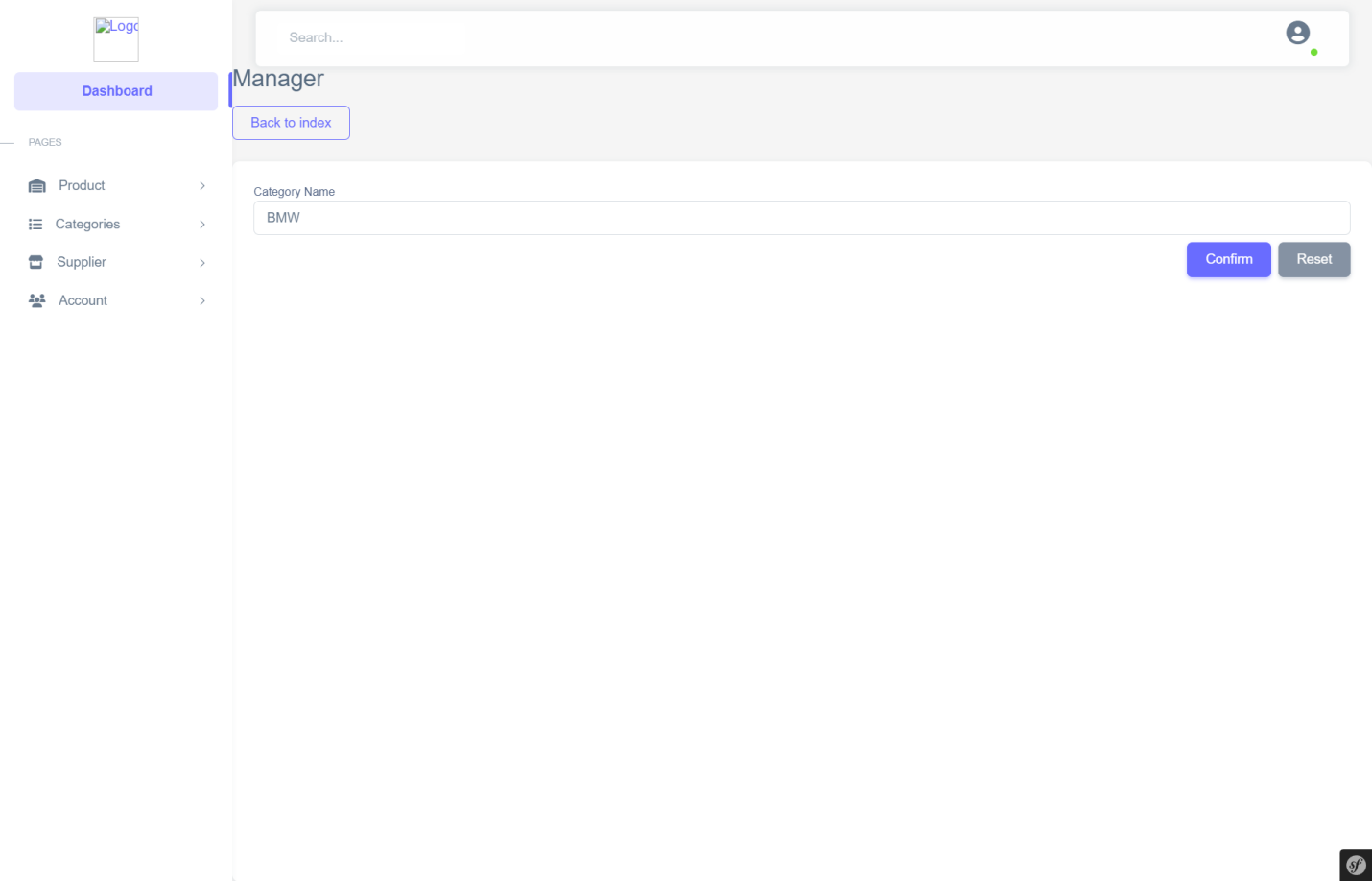


Figure 56: Edit Category Page

* Supplier manager page

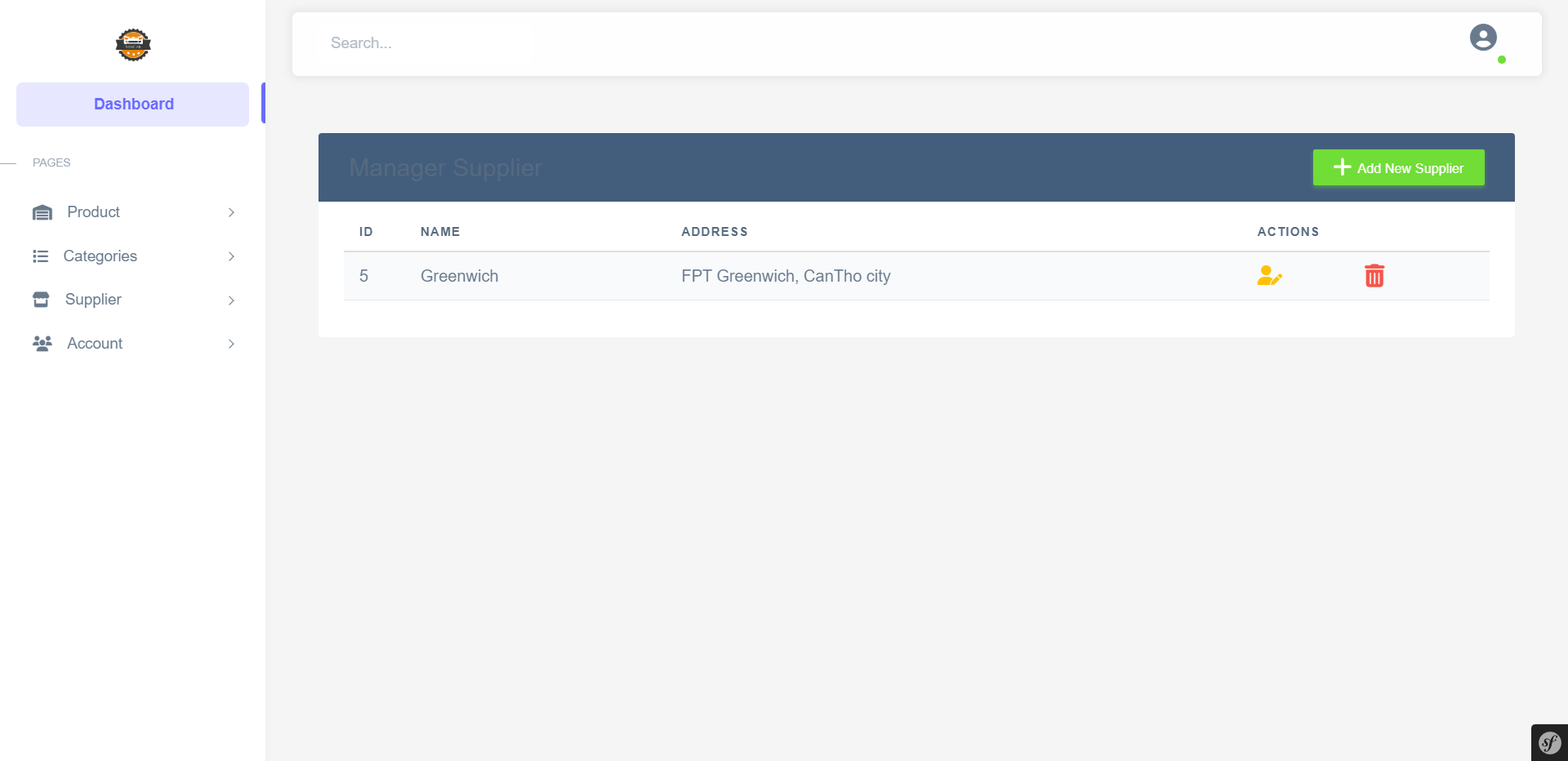


Figure 57: Supplier Manager Page

* Add supplier page

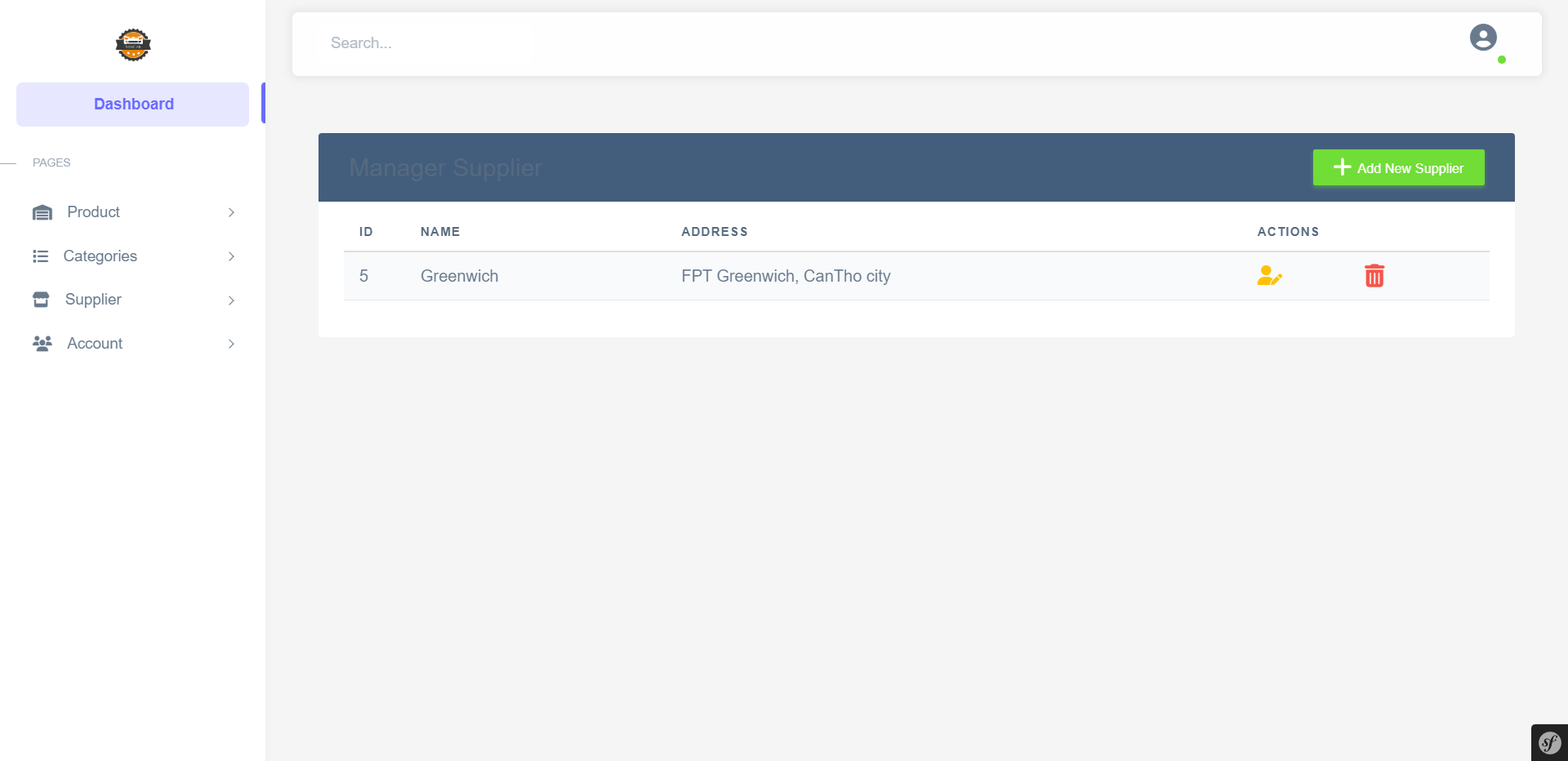


Figure 58: Add Supplier Page

* Edit supplier page

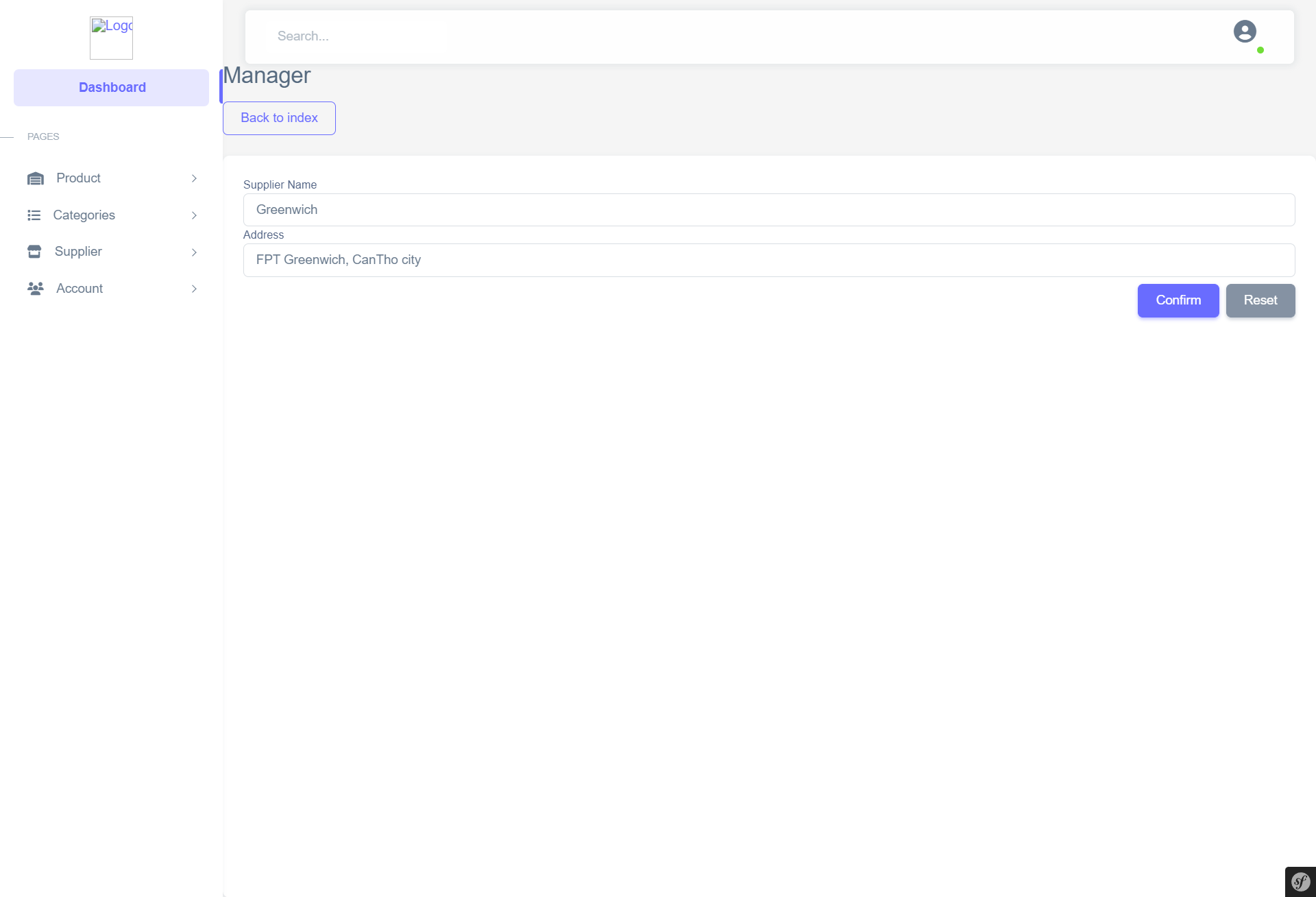


Figure 59: Edit Supplier Page

* Account manager page



Figure 60: Account Manager Page

## 3.3 GitHub Repository evidences

GitHub is a platform for developers to exchange knowledge and source code. That's why we decided to use

GitHub in this project to easily exchange the source code with each other if there is any change in the project.

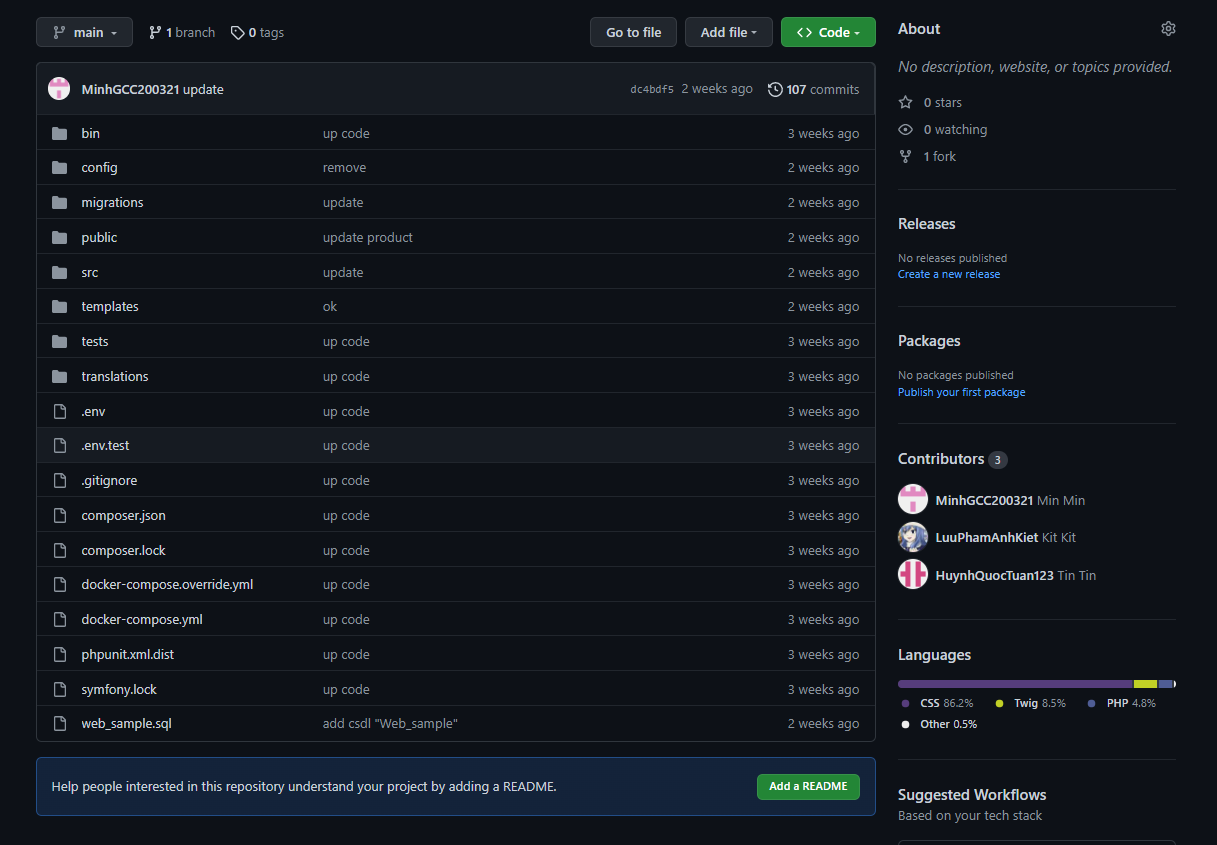


Figure 61: Our repository on the GitHub

Here are a few proof images of the source code repository in our project throughout the process that we code on the GitHub platform.

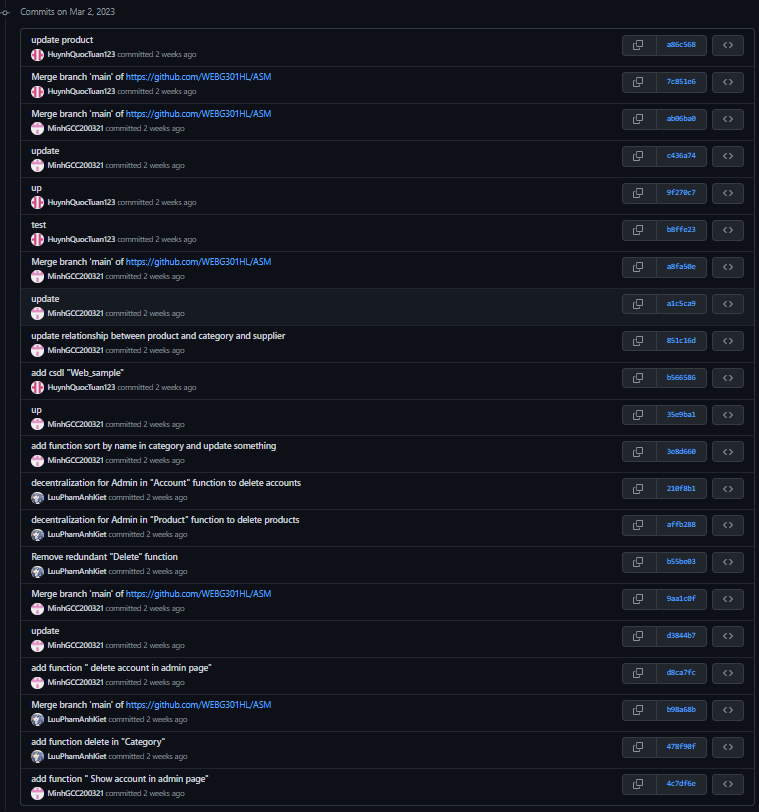


Figure 62 - Some images about our commits on the GitHub (1)

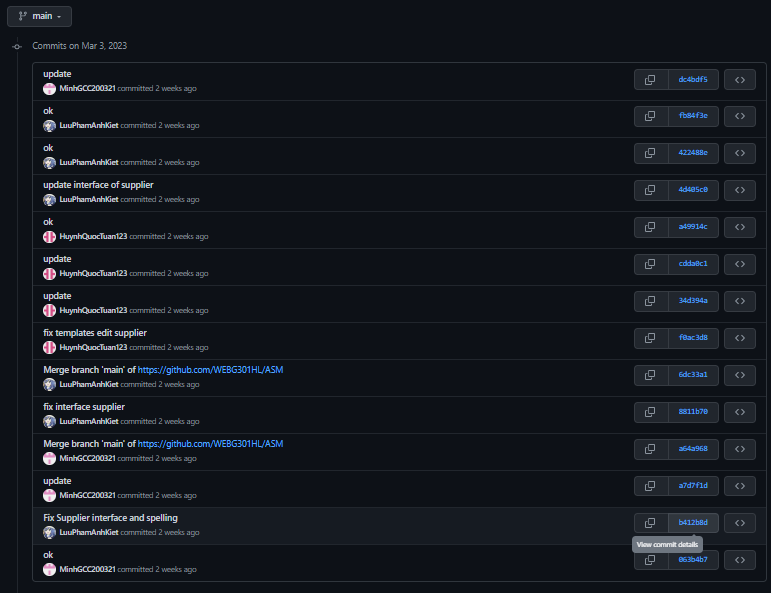


Figure 63: Some images about our commits on the GitHub (2)

GitHub Repository: <https://github.com/WEBG301HL/ASM>

# Chapter 4 – Conclusion

## 4.1 What went well

We effectively built a sales website with all of the required features. Basic functions needed of any website used to handle a specific product include the ability to make, display, update, and delete data. The website is also used to add login and registration methods in order to establish a decentralized function between users and administrators in order to prevent risks associated with data safety and security. Users will be prevented from viewing the path that only the admin account has access to because of the website's access control feature. Furthermore, product information or product specifics are frequently open to the public and can be viewed without requiring membership or login. The website is up and operating, and all features are functioning properly. The user interface is also well designed and clearly displayed to the user.

## 4.2 What did no go well

Beside the positive aspects of the website, there are a few flaws that must be addressed. The website has yet to notify users when they have successfully logged in, when the user deletes the basket, and when the user adds goods to the shop. Furthermore, even if the user or administrator enters negative numbers for the amount of products and the price, the system will still add them to the database.

## 4.3 Lessons learned and further improvements

Website overview in terms of user interface is not good, need to edit and update the user interface. Make the website as user-friendly as possible and show off the products on sale in the store. For user accounts, there are no verification methods through Gmail or phone number yet, which affects the capacity of accounts when oversubscribed. In addition, the website's functions are not perfect, it is only temporary for users and needs to be updated regularly.

Further improvement:

* Develop more functions to edit personal information.
* Improve the user interface for all pages of the website.
* Add confirmation messages when the customer increases the number of products in the cart and deletes them.
* Add a function that allows customers to update the number of products they order.
* Limit the number of products when the customer adds to the cart, must be greater than 0 and less than or equal to the number of products in the store.
* Add function payment.

# References

accesto, 2021. *accesto.* [Online]   
Available at: https://accesto.com/blog/what-is-php-framework-symfony-explained-for-executives/

github, 2022. *github.* [Online]   
Available at: https://github.com/git-guides

kinsta, 2022. *kinsta.* [Online]   
Available at: https://kinsta.com/knowledgebase/what-is-github/