

**Stationery Management**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | *Nguyen* Huu *Tri* | |
| **Semester:** | *III* | |
| **Batch No:** | *T1.2103.E1* | |
| **Group No:** | *03* | |
| **Order:** | **Full name** | **Roll No.** |
| *1.* | *Do Huy Hoang* | *Student1288223* |
| *2.* | *Nguyen Tran Hoang Phuc* | *Student1286085* |
| *3.* | *Van Khanh Phat* | *Student1275262* |
| *4.* | *Le Huu Thang* | *Student1302889* |

***Month****: 06* ***Year****: 2022*



This is to certify that

|  |  |
| --- | --- |
| **Mr.** | **Đỗ Huy Hoàng** |
| **Mr.** | **Nguyễn Trần Hoàng Phúc** |
| **Ms.** | **Văn Khánh Phát** |
| **Ms.** | **Lê Hữu Thắng** |

Have successfully Designed & Developed

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Stationery Management** | | |

Submitted by:

|  |
| --- |
| **Mr. NGUYEN HUU TRI** |

Date of Issue:

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

Authorized Signature:

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

Content

Acknowledge

Synopsis

Customer Requirement Specification

1. User use System…………………………………………………………………….…()
2. Functional Requirements……………………………………………………………...()

2.1. Employee

* 1. . Admin

System Requirements

1. Hardware Requirements
2. Software Requirements.

Review 1 ………………………………………………………………………..(1)

1. Problem Definition
2. Customer Requirements Specifications
3. System Requirements

Review 2 ………………………………………………………………………..(7)

1. Use Case
2. Data Flow Diagram
3. Site Map
4. Sequence

Review 3 ………………………………………………………………………..(26)

1. Screen Shots

Tasksheet ………………………………………………………………………..(39)

**Acknowledge**

As we understand that the Project 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, Mr.Le Huu Tri 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 03.

**Synopsis**

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 front-end or back-end technology 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 Semester III, as well as special subjects those are C#. These programs should be done in the Lab sessions with assistance of the faculty if required.

The website (in this project) is a world that brings everything about the potteries and crockeries into your experience.

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

Review 1

|  |
| --- |
| Problem Definition |

In recent years, all the organizations are asking for stationery via e-mail or sheet or the number of subscribers is outstanding and this leads to a waste of resources, difficult to control thoroughly. and labor intensive.

Because of these difficulties, an organization called HMT Technologies cooperated with our company to develop a website called "Stationery Management" to make it easier for the company to manage and store stock possible items. More than necessary and the supply of some items may be less than necessary, which reduces waste of resources and manpower.

With a friendly interface, simple, easy to use, online support team, enthusiastic, high-speed website to meet all access needs, helping employees in the company receive stationery quickly. No need to waste time commutied

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

1. **User use System**

* **Employee:**
* Employees log in to use the website to look up stationery specified for their department. Order stationery and track whether your stationery is approved or not. If the web doesn't have the stationery you need, you can chat and ask the supplier to view and browse.
* **Admin:**
* Web Manager: Can Create, update: employees, accounts, stationery and other web information and make reports.

1. **Functional Requirements**
   1. **Employee**

* **Search stationery:**
  + Employees choose the stationery in their department that they need.
  + The view should be a notification to the employee that the information they have entered is incorrect.
  + The site will return all the stationery that the employee searches for.
* **Request for providing:**
  + Employees can select stationery from the search list. The list will show pictures and information of those stationery.
  + Accept the company's terms and then claim.
  + Staff can edit stationery and quantity as required is provided.
  + Once the request is approved, an approval notice will be sent to the employee's email.
* **Login:**
* If you are an employee of the company, you can log in to the system by entering the employee Id and password provided by the company.
* Employees can view and update their profile information (Account name, date of birth, gender, email, phone, password).
* Employees can view their request history.
* View must show message if login failed or show employee name on site.
* **View News, Contact info of the web:**
* In the homepage shows the company's intro and there is no choice of stationery and shopping cart. When the employee logs in, it will appear.
* When employees need help, they can view contact information of the web: Email, phone to contact.
  1. **Admin**
* **Dashboard:**
* Admin can see an overview of approved requests: Total number of approved stationery, total of orders, total approved orders cost, rate of each product in the total approved orders.
* See the list of the all requests.
* View the list of new employees, total number of employees.
* **Employee management:**
* Create, update, view, delete employee information in the system by entering the account ID.
* Grant and revoke roles for employees.
* **Manage information about Web**:
* The information is displayed in a view that employees can directly select when searching for the stationery their department is issued with. Or use admin to filter when making reports, see dashboard. All value headers must be valid and no broken configuration
* Stationery: This is information about stationery. Admin can create stationery by id, office name, stationery information. Update information when there are changes.
* Account: This is information about the employee. Admin can create with employee id and employee information.
* Browsing list: This is the information generated when the Admin approves the request.
* Contact information: administrator can create or edit the web's information link system: Phone, email, working time

|  |
| --- |
| SYstem requirements |

1. **Hardware Requirements**

**Server:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **CPU** |  |  | Processor type: |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | Core-compatible processor or faster | |  |
|  |  |  |  |  | Processor speed: |  |  |
|  |  |  |  |  | Recommended: 2.0 GHz or faster | |  |
|  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  |  |
|  | **OS** | |  |  | Microsoft Windows 7 with IIS 7 or higher | |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Memory (RAM)** |  |  | RAM: |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | Minimum: 3 GB | |  |
|  |  |  |  |  | Recommended: 4 GB or more |  |  |
|  |  |  |  |  | Recommended: 2.0 GHz or faster | |  |
|  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  |  |
|  |  | **Hard Drive** |  |  | Free space:  Minimum: 20 GB  Recommended: 50 GB or more  Maximum: Operating system maximum |  |  |

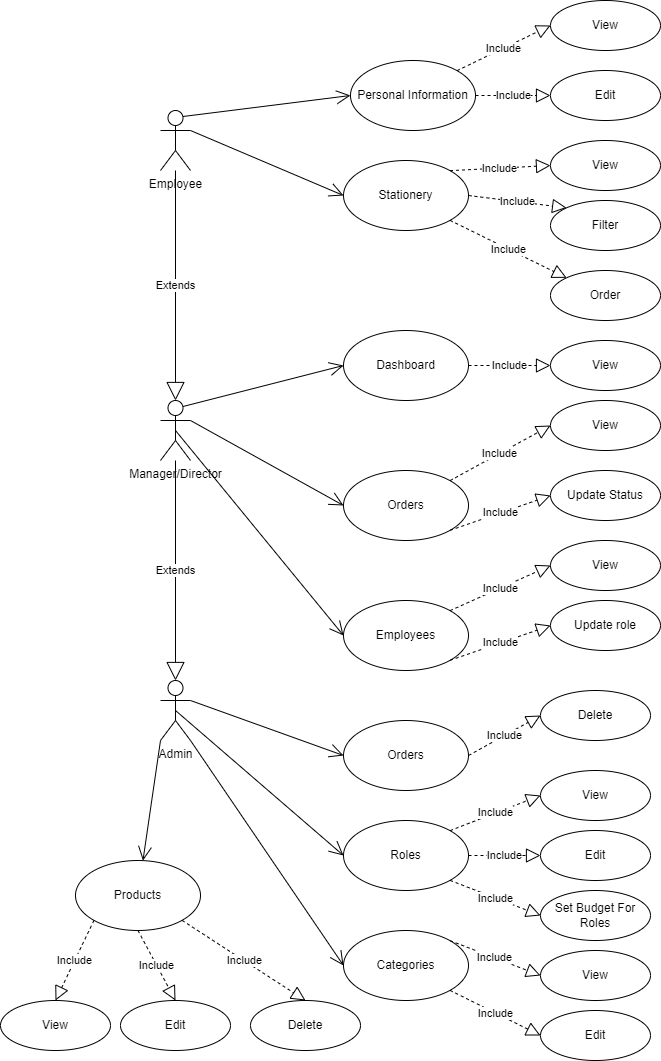
**Client:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Component** | | |  | **Requirement** | |
|  |  |  |  |  |  |
|  | **CPU** |  |  | Processor type: |  |
|  |  |  |  |
|  |  |  |  | Core-compatible processor or faster |  |
|  |  |  |  | Processor speed: |  |
|  |  |  |  | Recommended: 2.0 GHz or faster |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| **OS** | | |  | All OS(Window ,Linux ,Android ,Mac OS …) | |
|  | | |  |  | |
|  | **Memory (RAM)** |  |  | RAM: |  |
|  |  |  |  |
|  |  |  |  | Minimum: 3 GB |  |
|  |  |  |  | Recommended: 4 GB or more |  |
|  |  |  |  | Maximum: Operating system maximum |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Visual Studio 2013 or higher |  |  |
|  | **Visual Studio** | |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **Microsoft .NET Framework** | | | Version 4.5 or higher | |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  | Microsoft SQL Server 2012 or higher |  |  |
|  | **RDBMS** | |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **Front-End** | |  |  | HTML5CSS3(sass, Bootstrap 3), JavaScript(JQuery), | |  |
|  |  |  |  |  | Ajax |  |  |
|  |  |  |  |  |  | |  |
|  |  |  |  |  |  | |  |
|  | **Back-End** | |  |  | ASP.NET MVC 5, ASP.NET WEB API 2, Entity | |  |
|  |  |  |  |  | Framework |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **Client:** | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **Component** | |  |  | **Requirement** |  |  |
|  |  |  |  |  |  |  |  |
|  |  | **CPU** |  |  | Processor type: |  |  |
|  |  |  |  |  | Core -compatible processor or faster |  |  |
|  |  |  |  |  | Processor speed: |  |  |
|  |  |  |  |  | Recommended: 2.0 GHz or faster |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **OS** | |  |  | All OS(Window ,Linux) |  |  |
|  |  | |  |  |  |  |  |
|  |  | **Memory (RAM)** |  |  | RAM: |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  | Minimum: 2 GB |  |  |
|  |  |  |  |  | Recommended: 4 GB or more |  |  |
|  |  |  |  |  | Maximum: Operating system maximum |  |  |
|  |  | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | **Hard Drive** | |  |  | Free space: |  |  |
|  |  |  |  |  | Minimum: 20GB |  |  |
|  |  | |  |  | |  |  |
|  | **Web Browser** | |  | Firefox 60+,Chrome 60+ | |  |  |
|  |  |  |  |  |  |  |  |

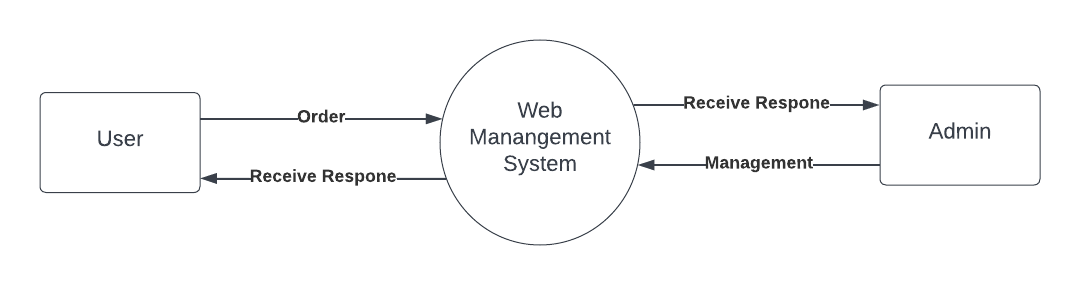
REVIEW 2

|  |
| --- |
| USE CASE |

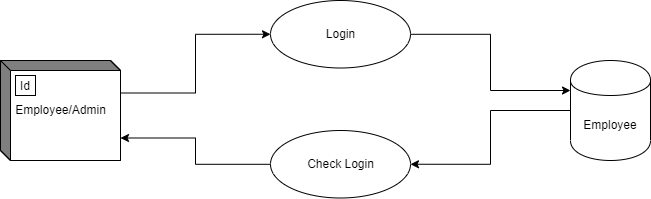


|  |
| --- |
| Data flow diagram |

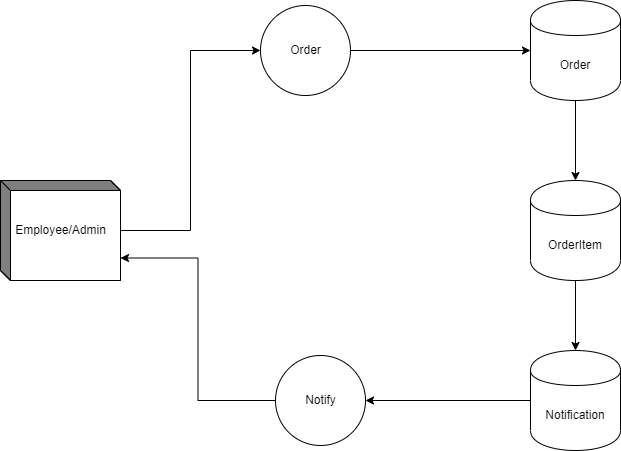
DFD.1



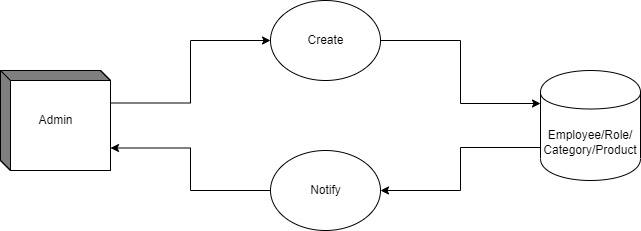
1.1. Login



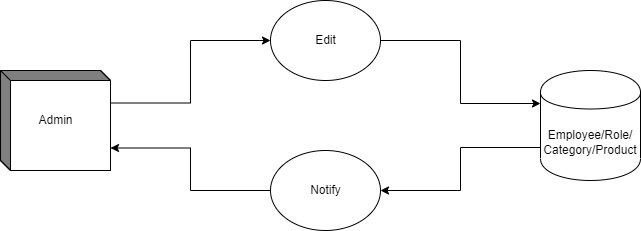
1.2. Order



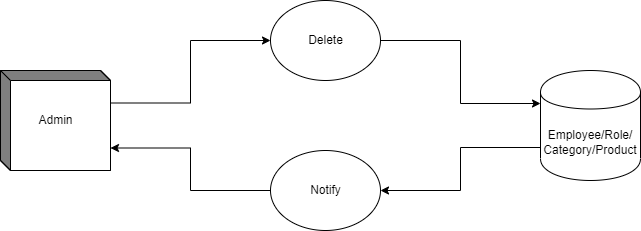
1.3. Create



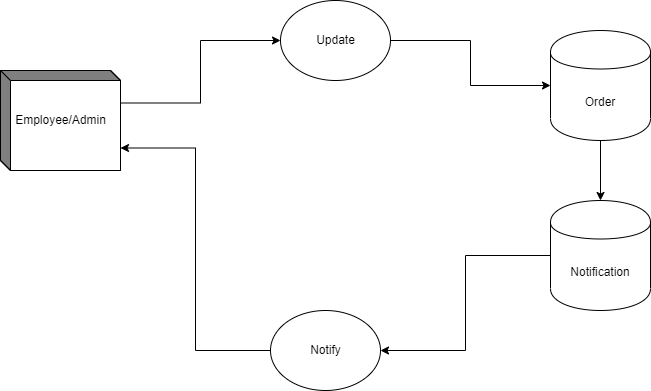
1.4. Edit



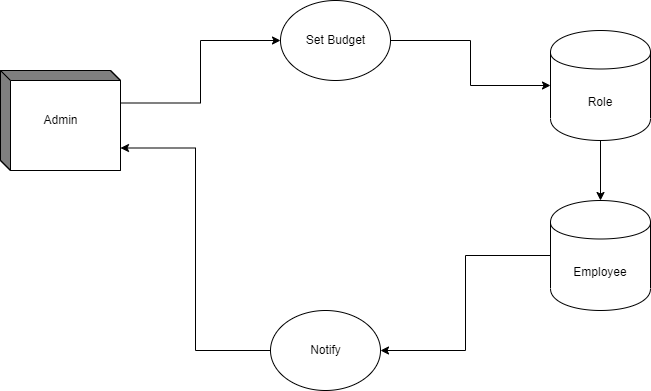
1.5. Delete



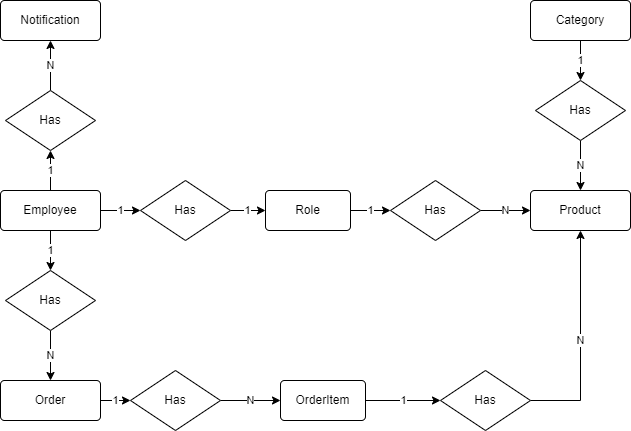
1.6. Update Order



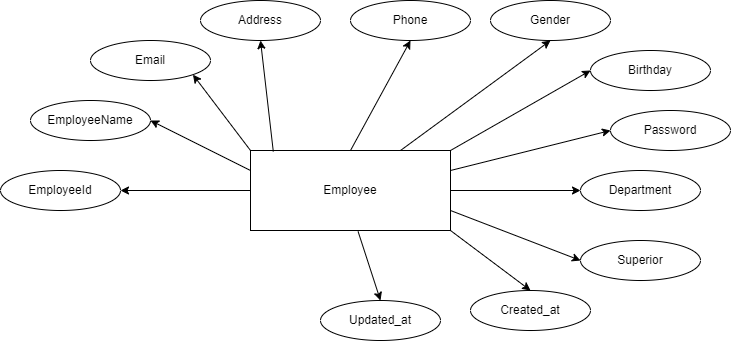
1.7. Set Budget

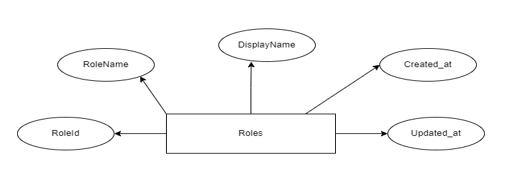


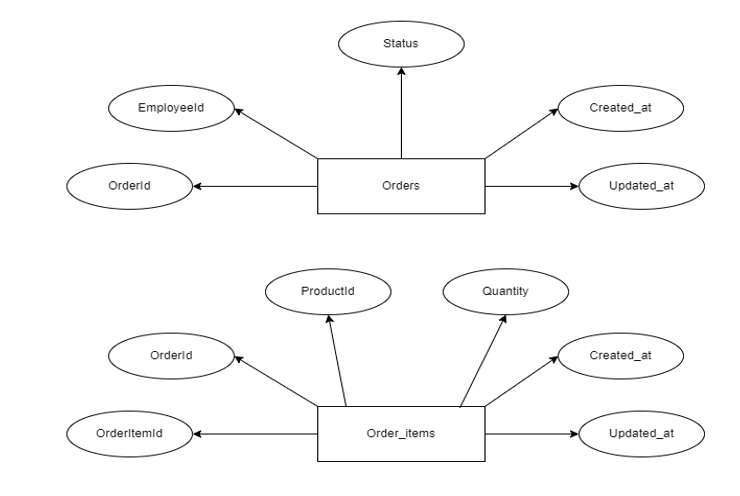
1. Entity Relationship Diagram

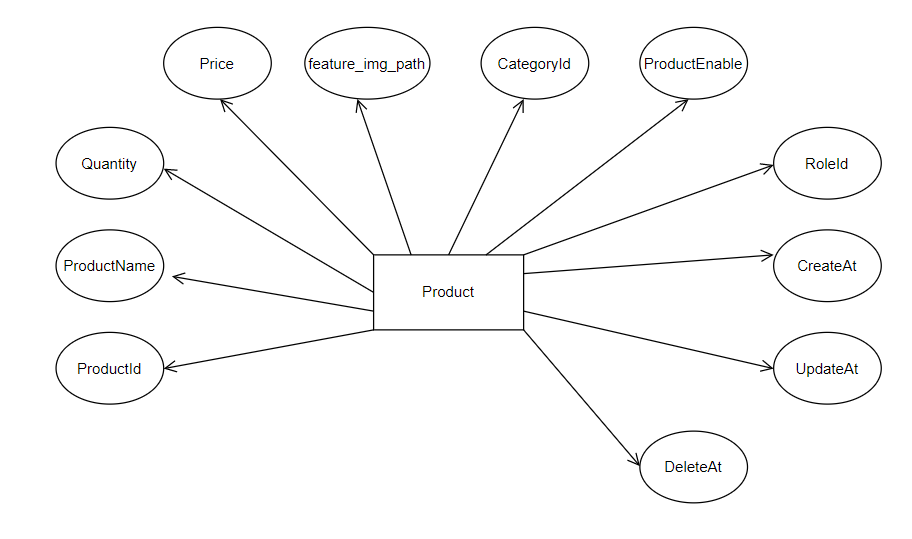


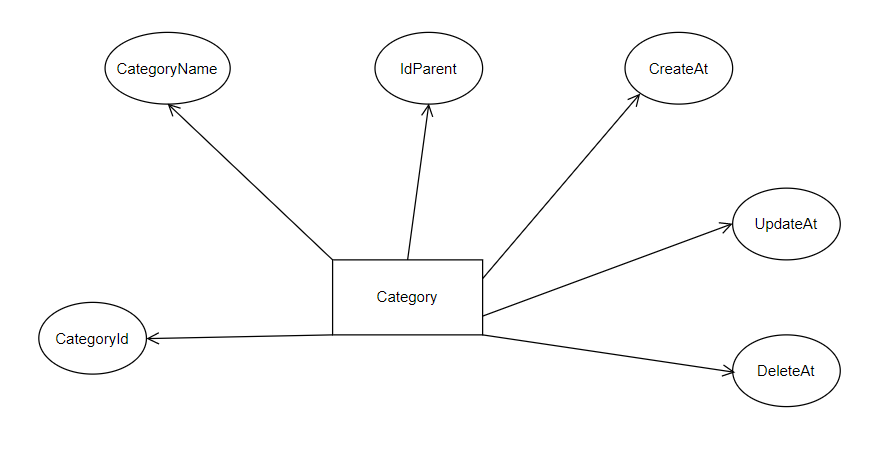
## ENTITY REATION DIAGRAM PROPERTIES











1. Table design

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Employees** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | EmployeeId | Varchar(255) | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | EmployeeName | Varchar(255) |  | Not null | User name |
| 3 | Email | Varchar(255) |  | Not null | User email |
| 4 | Address | Varchar(255) |  | Not null | User Address |
| 5 | Phone | Varchar(255) |  | Null | User Phone |
| 6 | Gender | Varchar(255) |  | Not Null,  default(0) | User Gender |
| 7 | Birthday | date |  | Not null | User Birthday |
| 8 | Password | Varchar(255) |  | Null | User password |
| 9 | Department | Varchar(255) |  | Null | User delete time |
| 10 | Superiors | Varchar(255) |  | Not Null | Superiors ID |
| 11 | Role\_Id | int | FK | Not null | Role ID |
| 12 | IsAdmin | bit |  | Null |  |
| 13 | Budget | int |  | Null | Budget |
| 14 | Created\_at | datetime |  | Null | User create time |
| 15 | Updated\_at | datetime |  | null | User update time |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Role** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | RoleId | Int | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | RoleName | varchar(255) |  | Not null | Roles identification |
| 3 | Budget | varchar(255) |  | Null | Role description |
| 4 | created\_at | datetime |  | Null | Roles creation time |
| 5 | updated\_at | datetime |  | Null | Roles update time |

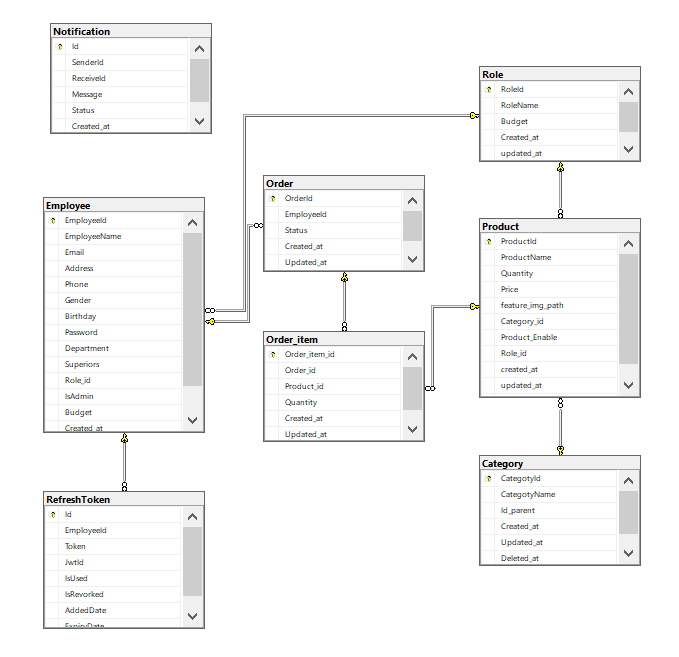
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Category** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | CategoryId | Int | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | CategoryName | varchar(100) |  | Not null | Category name |
| 3 | Id\_parent | Int |  | Null |  |
| 4 | created\_at | datetime |  | Null | Category create time |
| 5 | updated\_at | datetime |  | Null | Category update time |
| 6 | Deleted\_at | datetime |  | null | Category delete time |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Product** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | ProductId | int | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | ProductName | varchar(255) |  | Not null | Product name |
| 3 | Price | int |  | Not null | Product price |
| 4 | Quantity | int |  | Not null | Product quantity |
| 5 | Image | varchar(255) |  | Not null | Image path |
| 6 | CategoryId | int | Foregin key | UNSIGNED, Not null | Id category |
| 7 | ProductEnable | bit |  |  | Product enbale |
| 8 | Role\_id | int | Foregin Key | Not null | Id Role |
| 9 | created\_at | datetime |  | Null | Product create time |
| 10 | updated\_at | datetime |  | Null | Product update time |
| 11 | Deleted\_at | datetime |  | null | Product delete time |

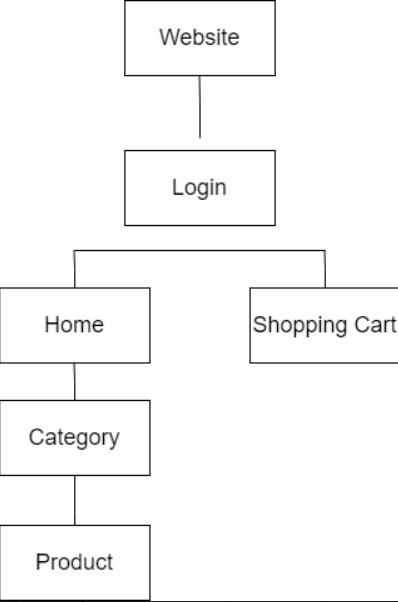
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | OrderId | int | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | EmployeeId | varchar(255) | Foreign key | Not null | Id User |
| 3 | Status | varchar(255) |  | Not null | Status Order |
| 4 | created\_at | datetime |  | Null | Order create time |
| 5 | updated\_at | datetime |  | Null | Order update time |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Order\_item** | | | | | |
| **Serial** | **Fields** | **Data type** | **Keys** | **Constrains** | **Description** |
| 1 | OrderItemId | int | Primary key | UNSIGNED, Not null | Sequence number |
| 2 | OrderId | int | Foreign key | UNSIGNED, Not null | Id order |
| 3 | ProductId | int | Foreign key | UNSIGNED, Not null | Id product |
| 4 | Quantity | int |  | Not null | Quantity |
| 5 | created\_at | datetime |  | Null | Order item create time |
| 6 | updated\_at | datetime |  | Null | Order item update time |

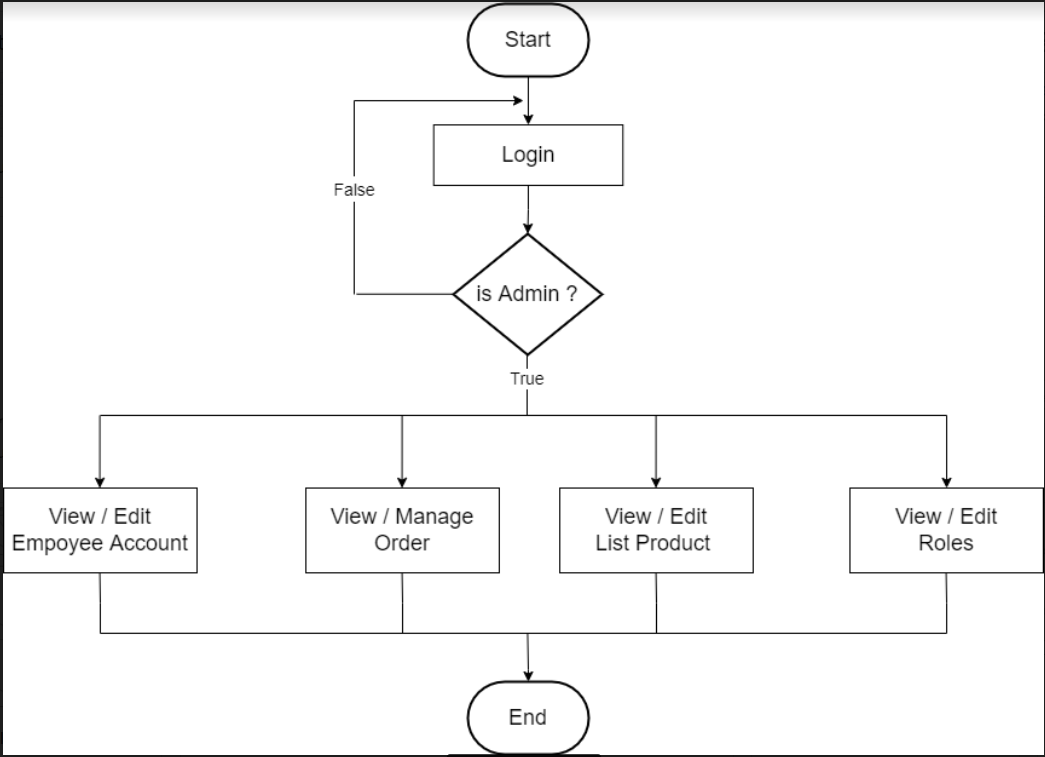
1. Relationship Diagram



|  |
| --- |
| StiteMap |



1. **Admin**

****

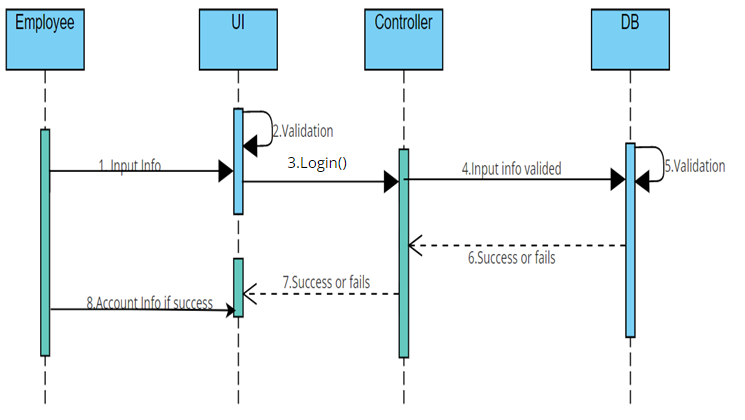
1. **User**

**Diagram

Description automatically generated**

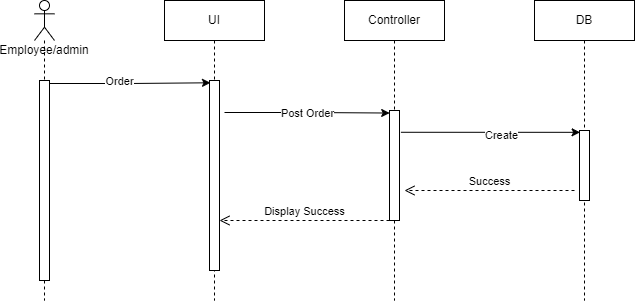
|  |
| --- |
| SEQUENCE |

**A. Login (ALL)**



|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | |  | |
| **Use case name** | | Login | |
| **Actor** | | Employee/Admin | |
| **Description** | | Employee/Admin wants to logon to website | |
| **Requirements** | | Employee/Admin provides account ID and password. | |
| **Pre-conditions** | | Employee/Admin created. | |
| **Post -Condition** | | Success: employee/admin login to website | |
| **Basic flow** | | **Actor action** | **System Responses** |
| 1. Input Account info and submit.  8. If success passenger logon to website | 1. UI Valadate 2. System call Login() to controller 3. Controller get match info from DB. 4. Validate info in DB 6 . DB return result.   7. Controller return view  login success or fail |
|  | Prepared by (Student) | | Approved by (Teacher) |
| Date: 07 July 2022 | All Member | | Nguyen Hữu Trí |

**B. Order (ALL)**

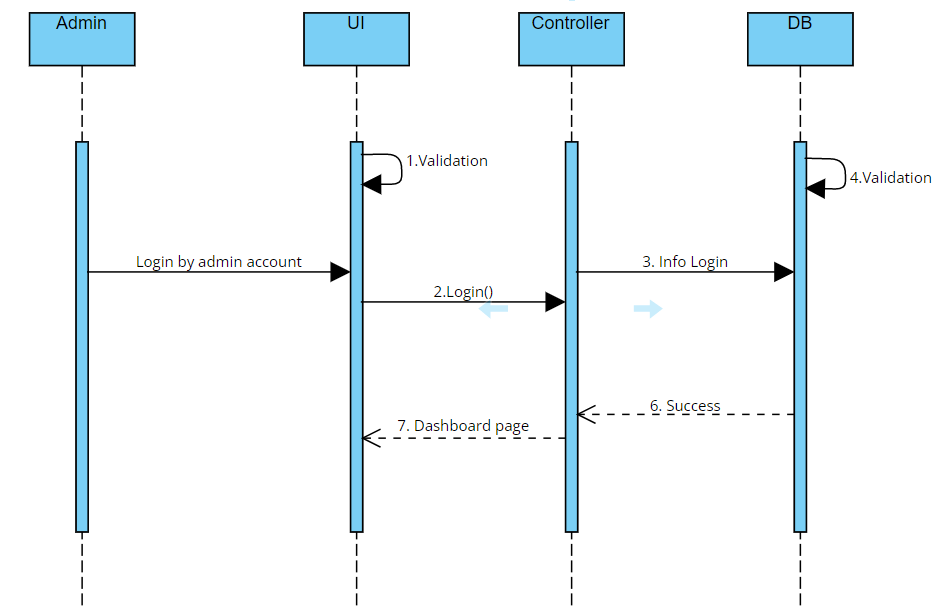


|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | Order | |
| **Actor** | Employee/Admin | |
| **Description** | Employee/Admin wants to order stationery | |
| **Requirements** | Employee/Admin created | |
| **Pre-conditions** |  | |
| **Post -Condition** | Success: Stationery is ordered | |
| **Basic flow** | **Actor action** | **System Responses** |
| 1. Employee/Admin orders stationery | 2. UI posts order to order controller |

|  |  |  |
| --- | --- | --- |
|  | Prepared by (Student) | Approved by (Teacher) |
| Date: 7 July 2022 | All Member | Nguyễn Hữu Trí |

|  |  |  |
| --- | --- | --- |
|  |  | 1. Order controller calls CreateOrder() to insert data into DB with posted order info. 2. Controller get response from DB. 3. Controller return view to UI. |
| **Alternative flow** |  |  |
| **Exception** |  |  |

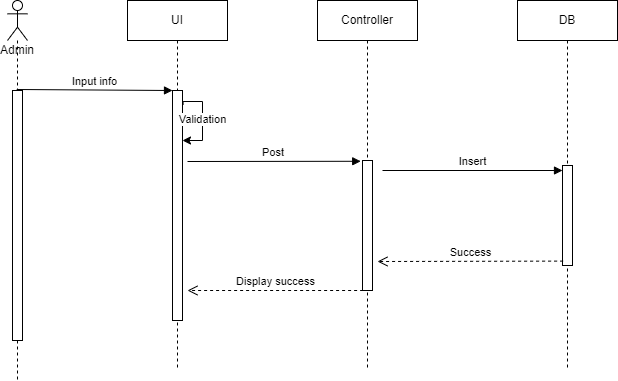
**C. Dashboard (Admin)**

****

|  |  |  |
| --- | --- | --- |
| **Author** |  | |
| **Use case name** | **DashBoard** | |
| **Actor** | Admin | |
| **Description** | Admin view Dashboard about general web active  infomation | |
| **Requirements** | Login by Admin account | |
| **Pre-conditions** |  | |
| **Post -Condition** | Success: Dashboard is display. | |
| **Basic flow** | **Actor action**  1. Login by admin account. | **System Responses**   1. UI Validate 2. System call Login() to controller to get data from DB with info that passenger input. 3. Controller get login info from DB. 4. Validate logon info 5. DB return match logon info to controller.   7 .Controller return view  Dashboard to UI. |
|

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Alternative flow** |  |  |
| **Exception** | Account is not admin | There are message to employee and no result  return |

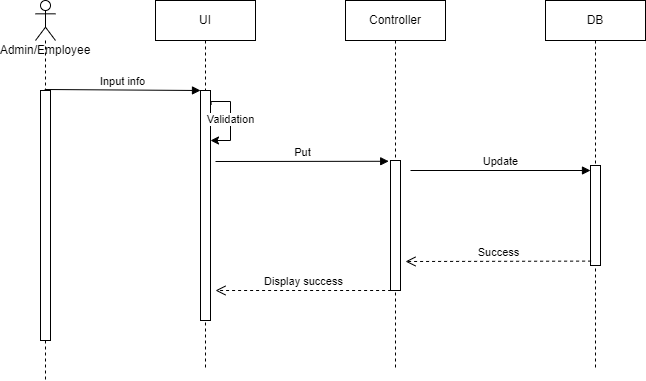
**D. Create (Admin)**



|  |  |
| --- | --- |
| **Author** |  |
| **Use case name** | **Create** |
| **Actor** | Admin |
| **Description** | Admin can to create new product/employee/category/role |
| **Requirements** | Login by Admin account |
| **Pre-conditions** |  |
| **Post -Condition** | Success: New product/employee/category/role is inserted |

|  |  |  |
| --- | --- | --- |
| **Basic flow** | **Actor action** | **System Responses** |
| 1. Admin input information to create new product/employee/category/role | 2. UI validates input information  3. UI post new product/employee/category/role to controller  4. Controller insert new product/employee/category/role into database  5. After inserting successfully, database sent back message to controller.  6. Controller then displays success message to UI |
| **Alternative flow** |  |  |
| **Exception** | Account is not admin, Input value is not valid |  |

**E. Edit Profile (ALL)**

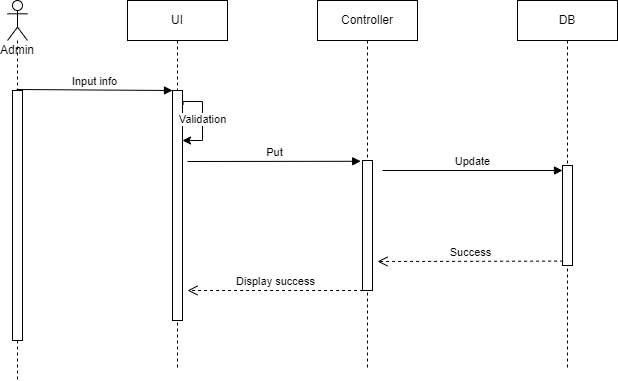


|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | |  | |
| **Use case name** | | **Category** | |
| **Actor** | | Admin/Employee | |
| **Description** | | Admin/Employee can update their profile | |
| **Requirements** | | Login by Admin/Employee account | |
| **Pre-conditions** | |  | |
| **Post -Condition** | | Success: Success message is displayed. | |
| **Basic flow** | | **Actor action** | **System Responses** |
|  | | 1. Adjust information they want to update | 2. UI validates input information |
|  | |  | 3.UI puts updated information to controller |
|  | |  | 4.Controller updates new information to database  5. After updating successfully, database sent back message to controller.  6. Controller then displays success message to UI |
|  | Prepared by (Student) | | Approved by (Teacher) |
| Date: 7 July 2022 | All Member | | Nguyễn Hữu Trí |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Alternative flow** |  |  |
| **Exception** | Input information is not valid | Display requirement in each information field |

|  |  |  |
| --- | --- | --- |
|  | Prepared by (Student) | Approved by (Teacher) |
| Date: 07 July 2022 | All Member | Nguyen Hữu Trí |

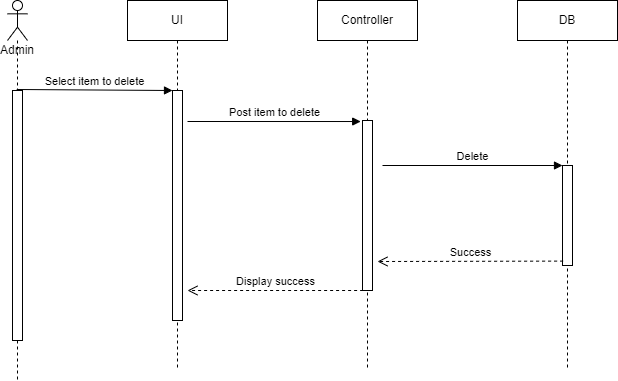
**F. Edit Product/Category/Order/Role/Employee (Admin)**



|  |  |
| --- | --- |
| **Author** |  |
| **Use case name** | **Edit Product/Role/Category/Order/Employee** |
| **Actor** | Admin |
| **Description** | Admin can edit product/employee/category/role/order |
| **Requirements** | Login by Admin account |
| **Pre-conditions** |  |
| **Post -Condition** | Success: New product/employee/category/role/order is updated |

|  |  |  |
| --- | --- | --- |
| **Basic flow** | **Actor action** | **System Responses** |
| 1. Admin adjusts information they want to update | 2. UI validates input information  3.UI puts updated information to controller  4.Controller updates new information to database  5. After updating successfully, database sent back message to controller.  6. Controller then displays success message to UI |
| **Alternative flow** |  |  |
| **Exception** | Account is not admin  Input value is not valid | Navigate back to home page  Display requirement in each information field |

**G. Delete Product/Category/Order/Role/Employee (Admin)**



|  |  |
| --- | --- |
| **Author** |  |
| **Use case name** | **Edit Product/Role/Category/Order/Employee** |
| **Actor** | Admin |
| **Description** | Admin can delete product/employee/category/role/order |
| **Requirements** | Login by Admin account |
| **Pre-conditions** |  |
| **Post -Condition** | Success: Selected product/employee/category/role/order is deleted |

|  |  |  |
| --- | --- | --- |
| **Basic flow** | **Actor action** | **System Responses** |
| 1. Admin select product/employee/category/role they want to delete | 2. UI post selected item to controller  3. Controller delete selected item from database  4. After deleting successfully, database sent back message to controller.  5. Controller then displays success message to UI |

|  |  |  |
| --- | --- | --- |
| **Alternative flow** |  |  |
| **Exception** | Account is not admin | Navigate back to home page |

**Review 2 - Task Sheet**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.: Project title: Stationery Management** | | | **Activity Plan Prepared by: Hoàng** | | **Date of Preparation of Activity Plan** | | |
| **Sr.No.** | **Task** | | **Start Date** | **End Date** | **Actual day** | **Member’s Name** | **status** |
| 1 | Use-Case | | 4 July 2022 | 7 July 2022 | 3 | Thang | Complete |
| 2 | dg DFD | Login | Thang |
| 3 | Create |
| 4 | Delete |
| 5 | Order |
| 6 | Edit |
| 7 | Update Order |
| 8 | ERD | ERD | Thang |
| 9 | ERD Properties |  |
| 10 | SEQUENCE | Login |  |
| 11 | Order | Thang |
| 12 | Dashboard |  |
| 13 | Create | Thang |
| 14 | Edit | Thang |
| 15 | Delete |  |  |  | Thang |  |

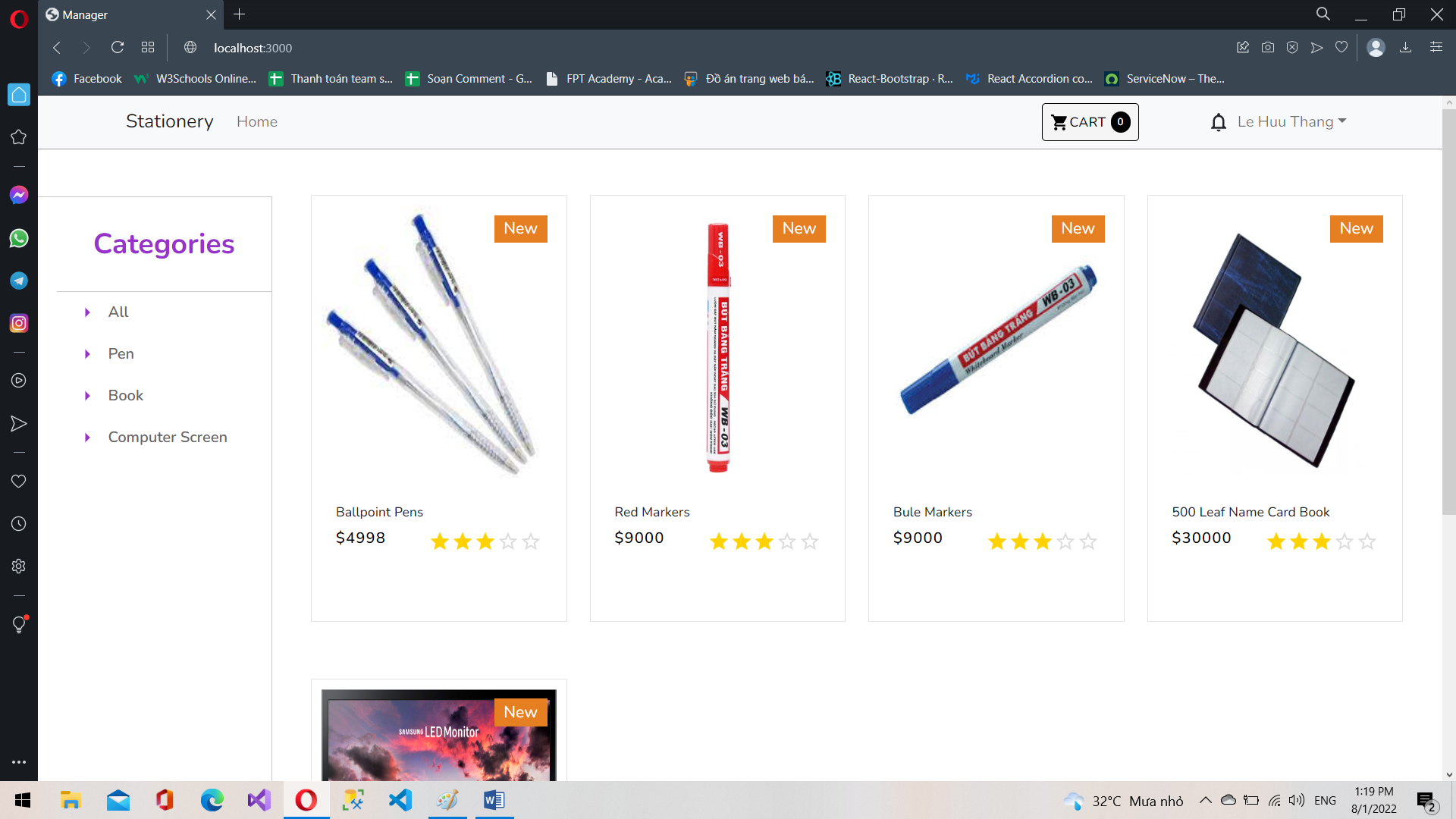
### **Group Leader Faculty**

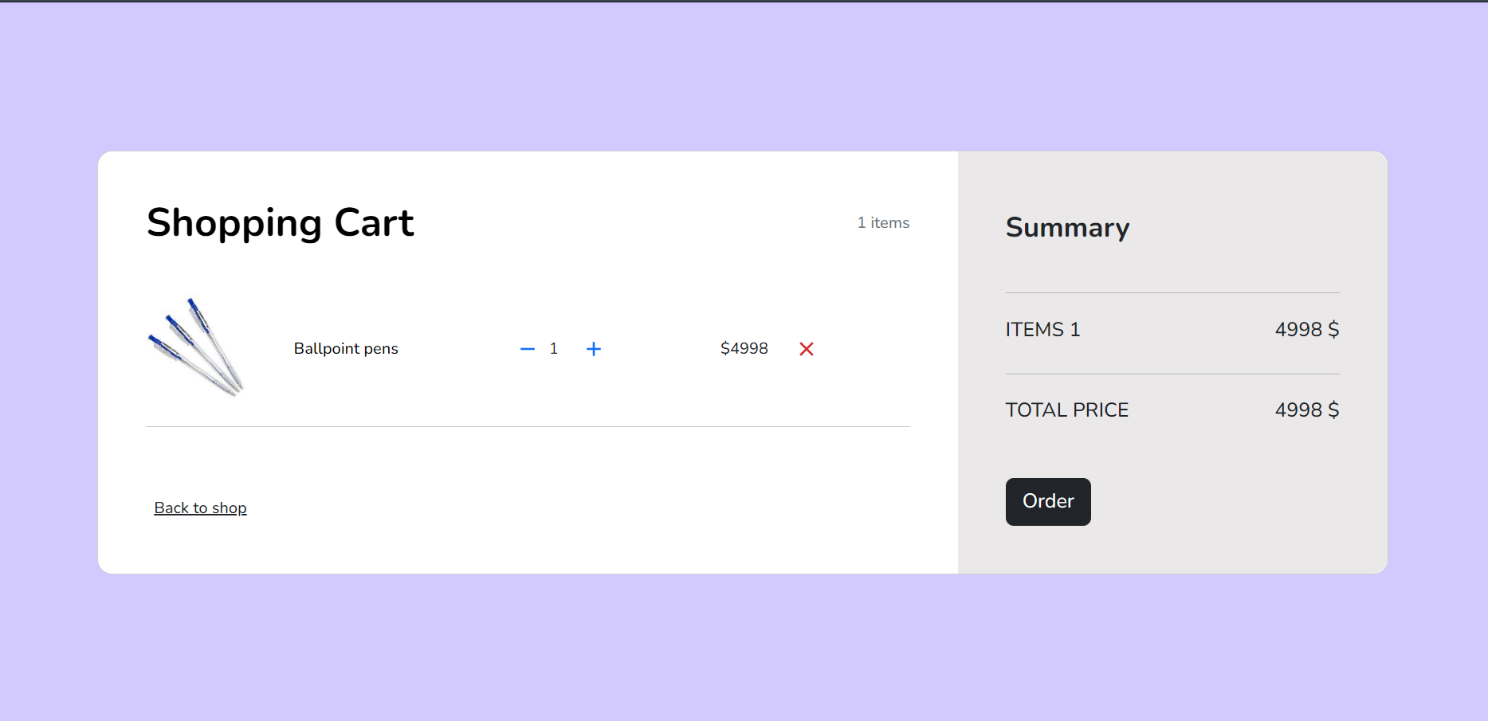
Do Huy Hoang Do Huy Hoang

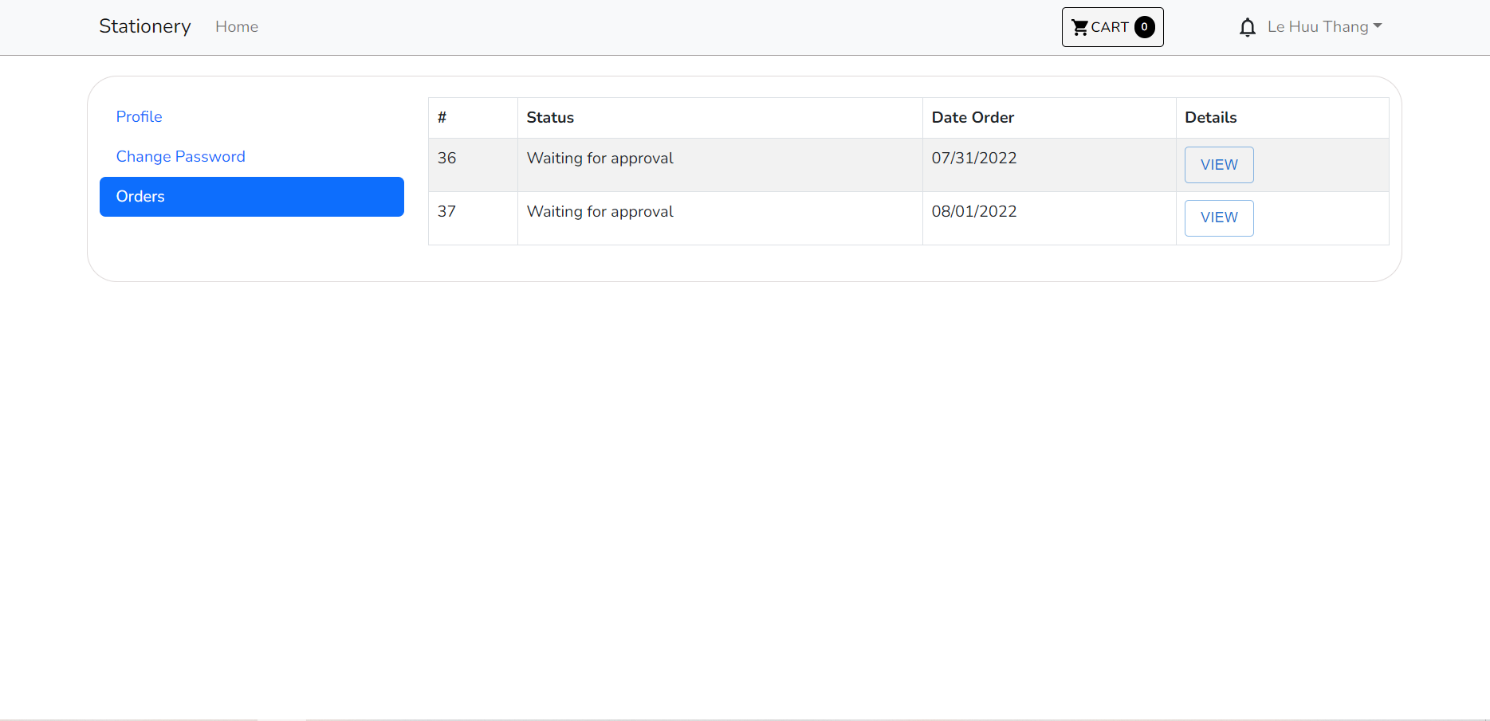
REVIEW 3

|  |
| --- |
| Screen shots |

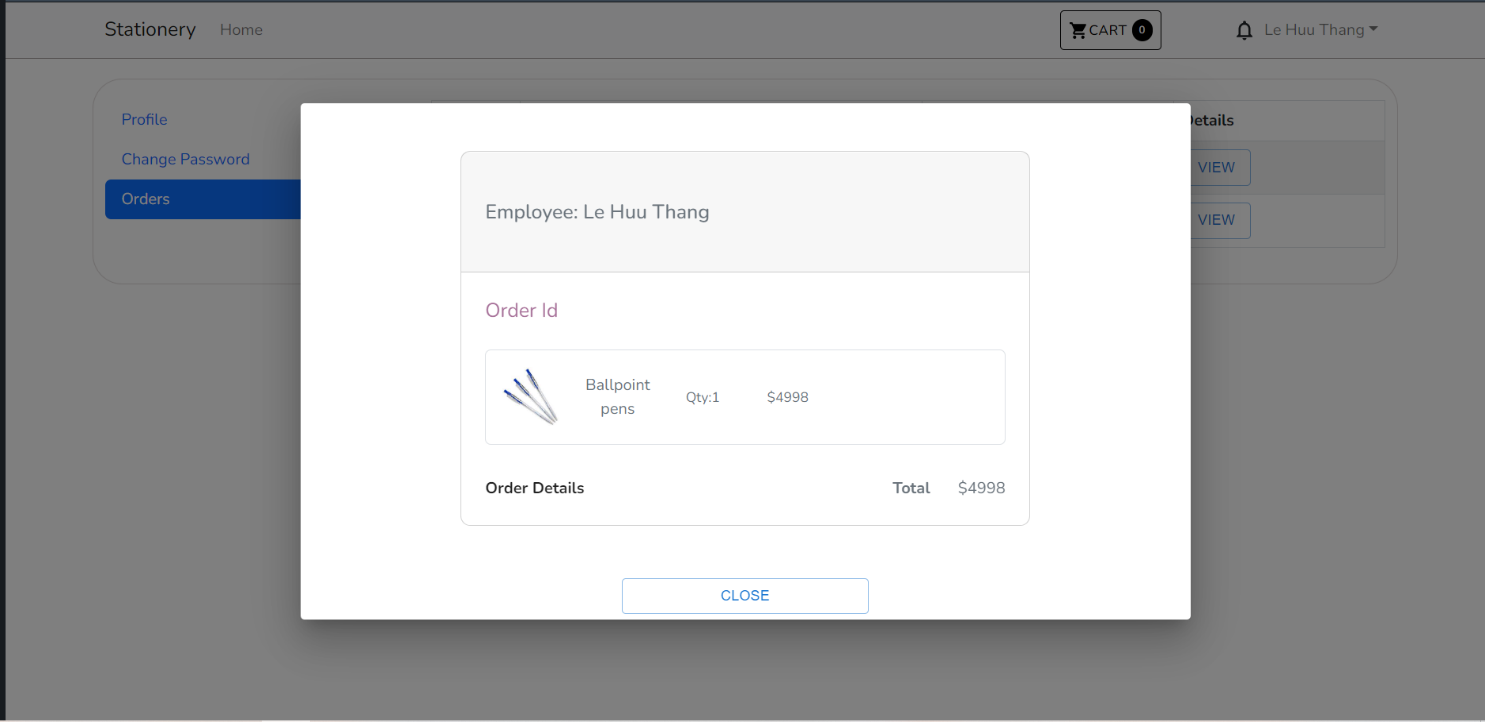
1. **Homepage:**



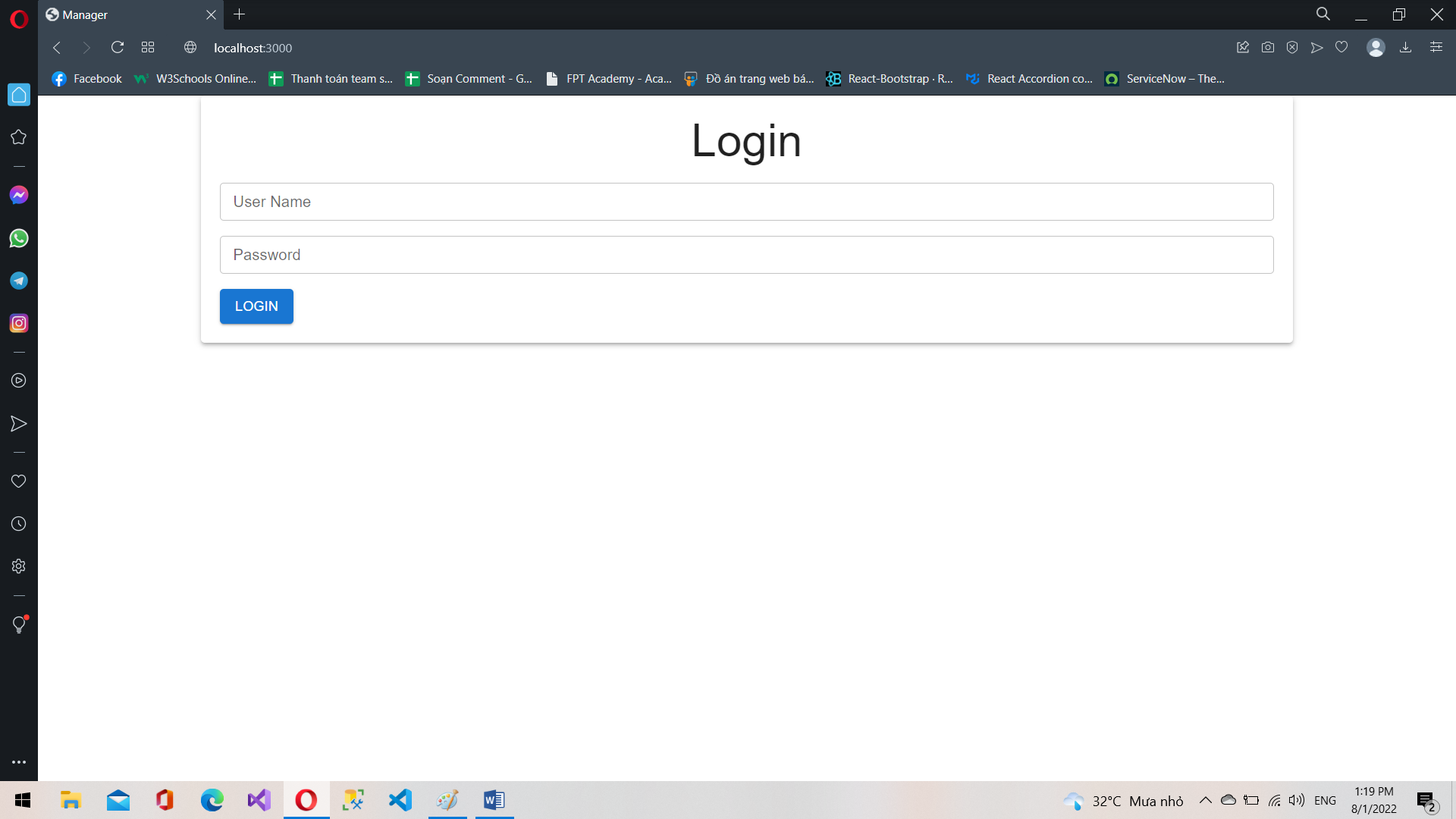
1. **Shopping Cart**
2. **View Orders**

****

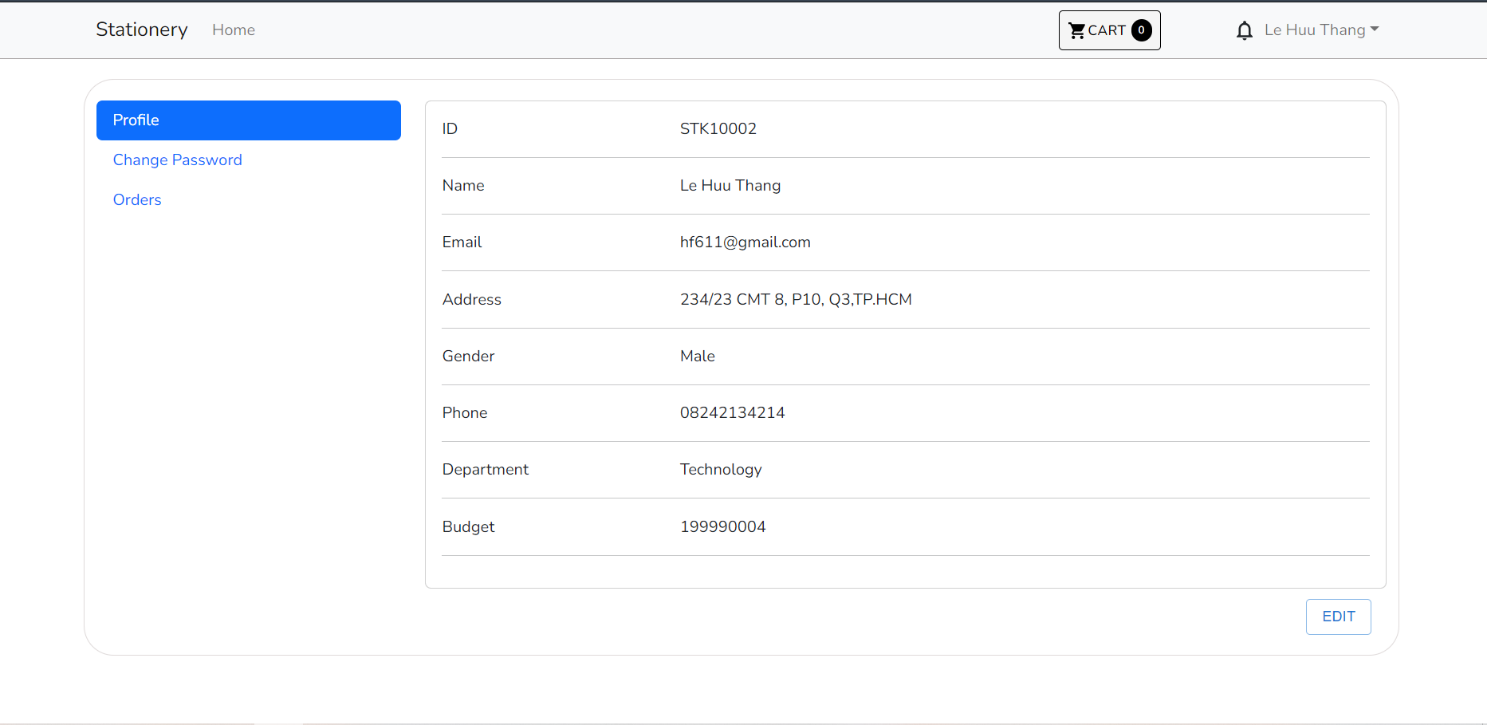
1. **View Orders Details**

****

1. **Login**

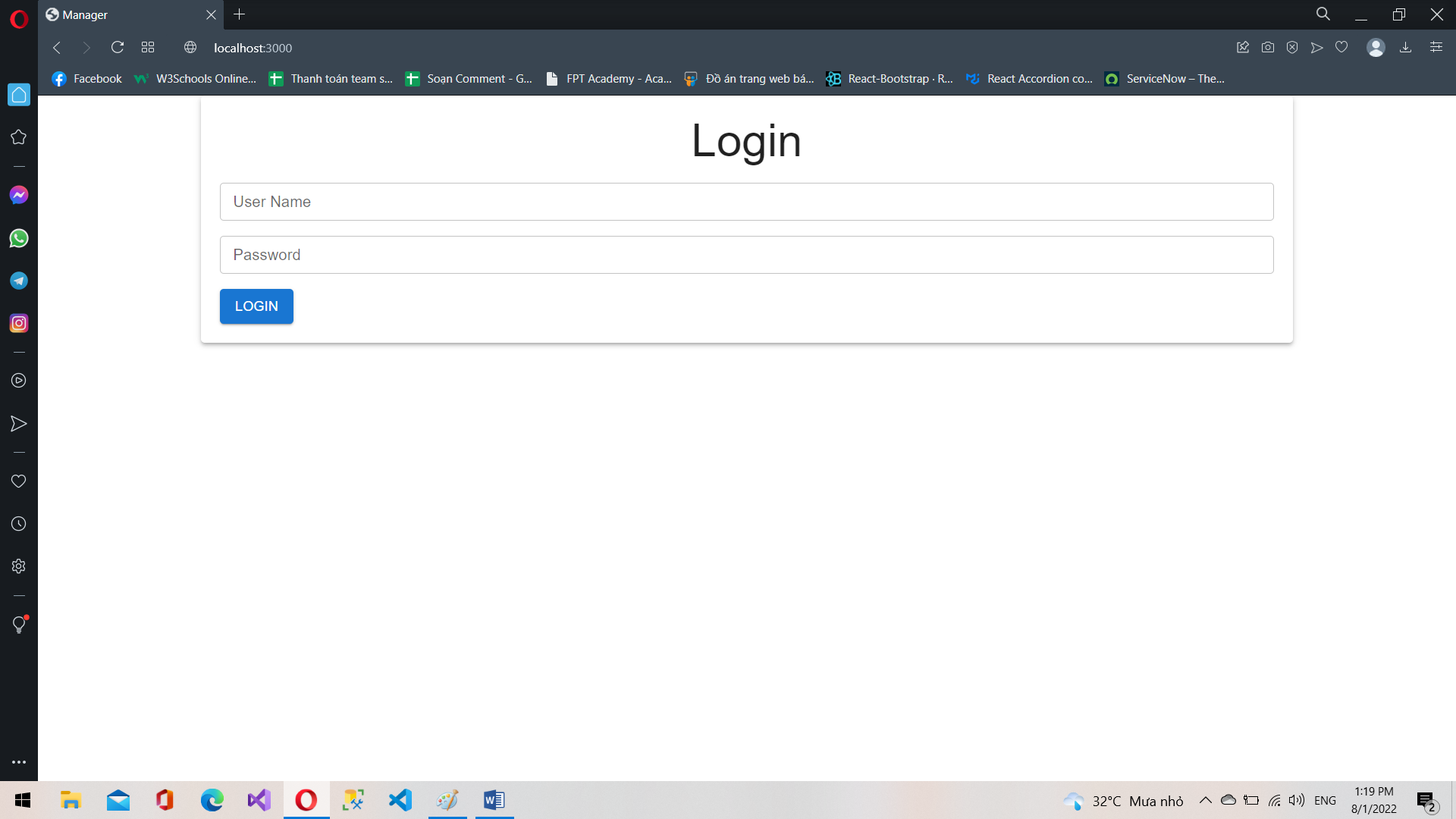


1. **Profile**

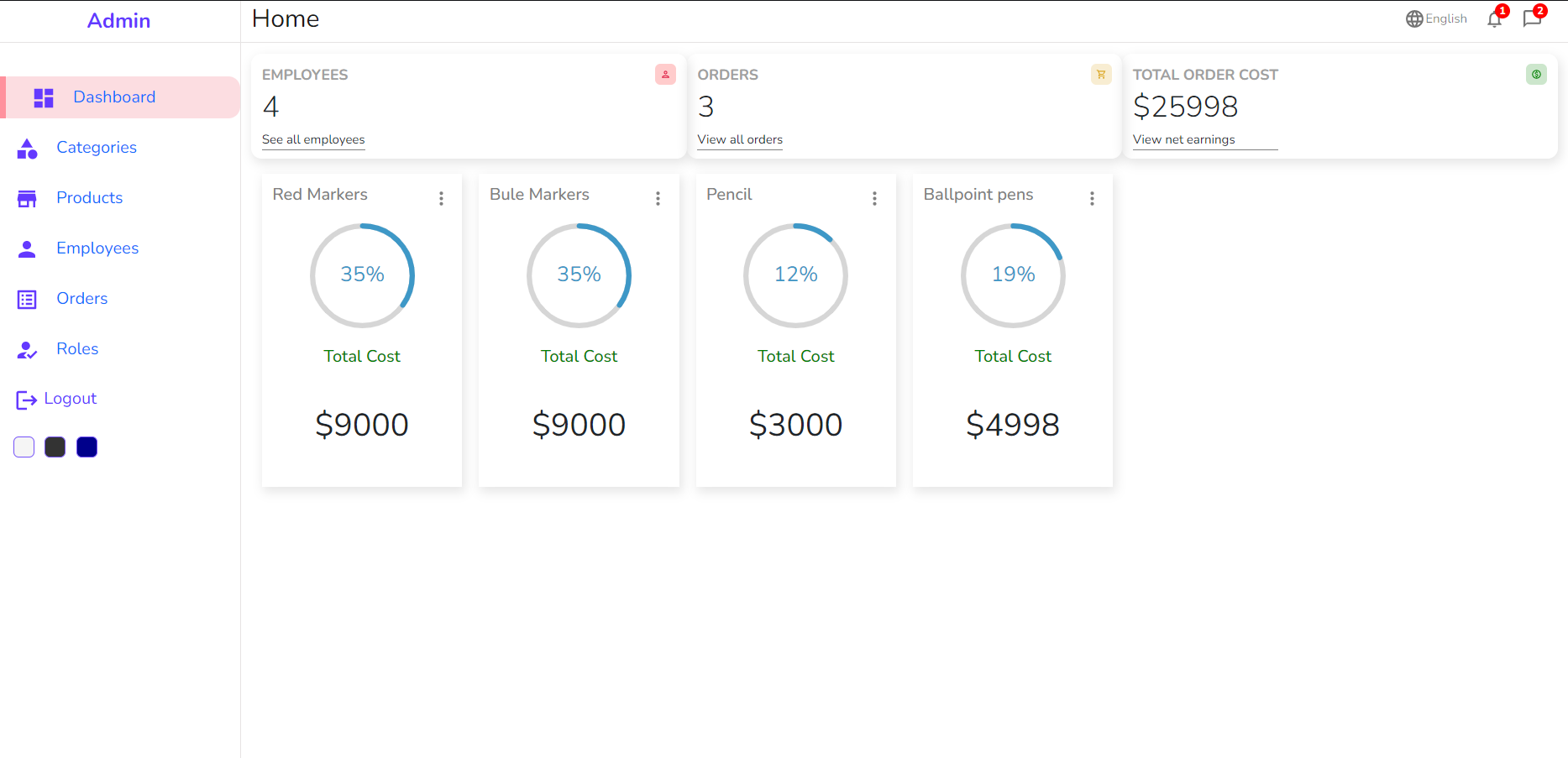
****

1. **Admin**

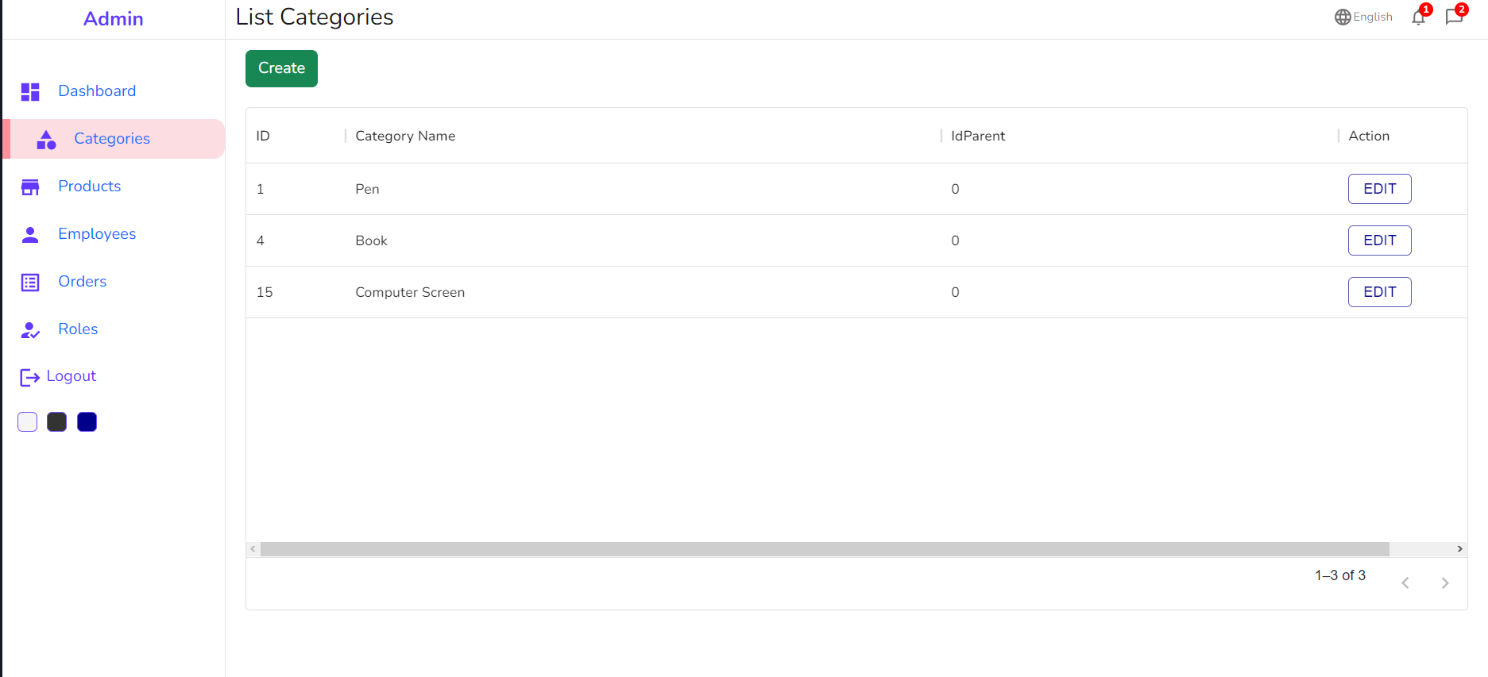
**6.1. Login**



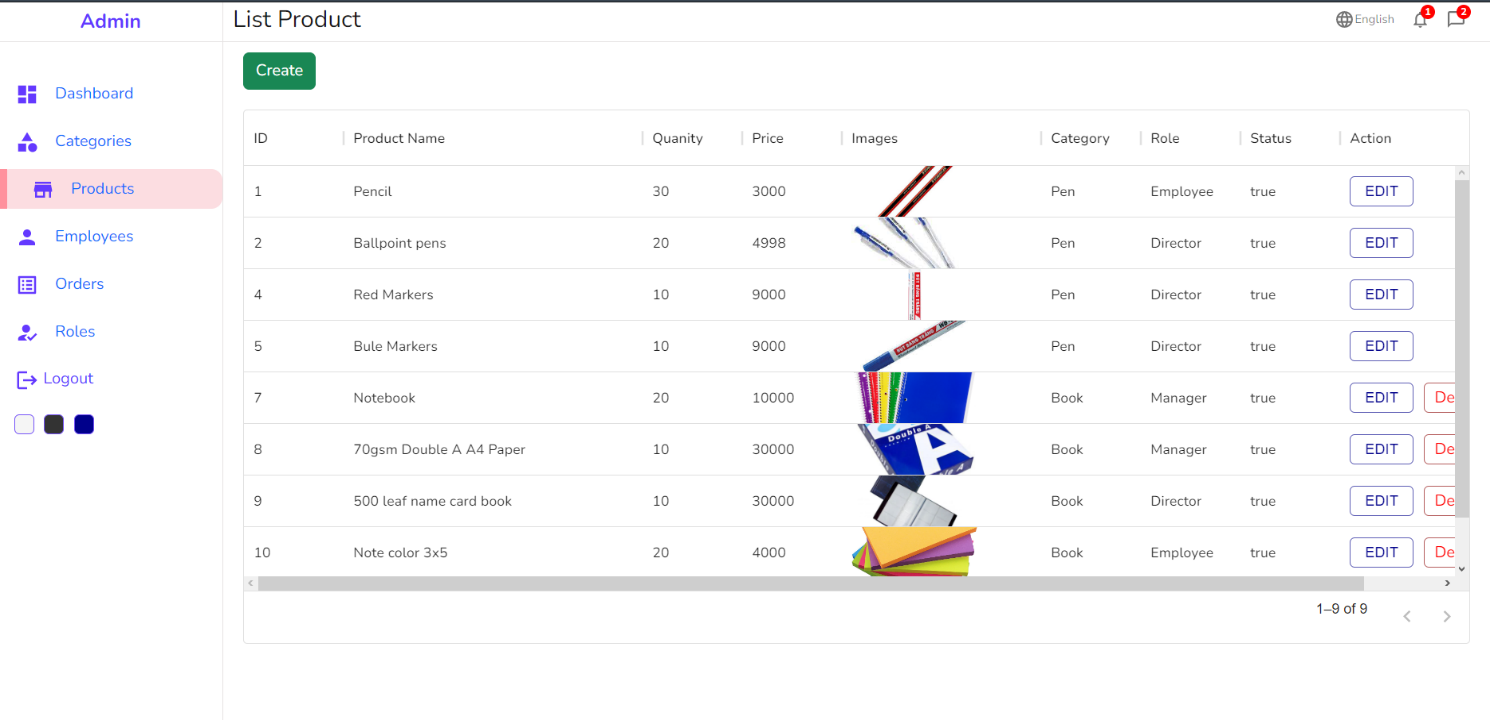
**6.2. Dashboard**

****

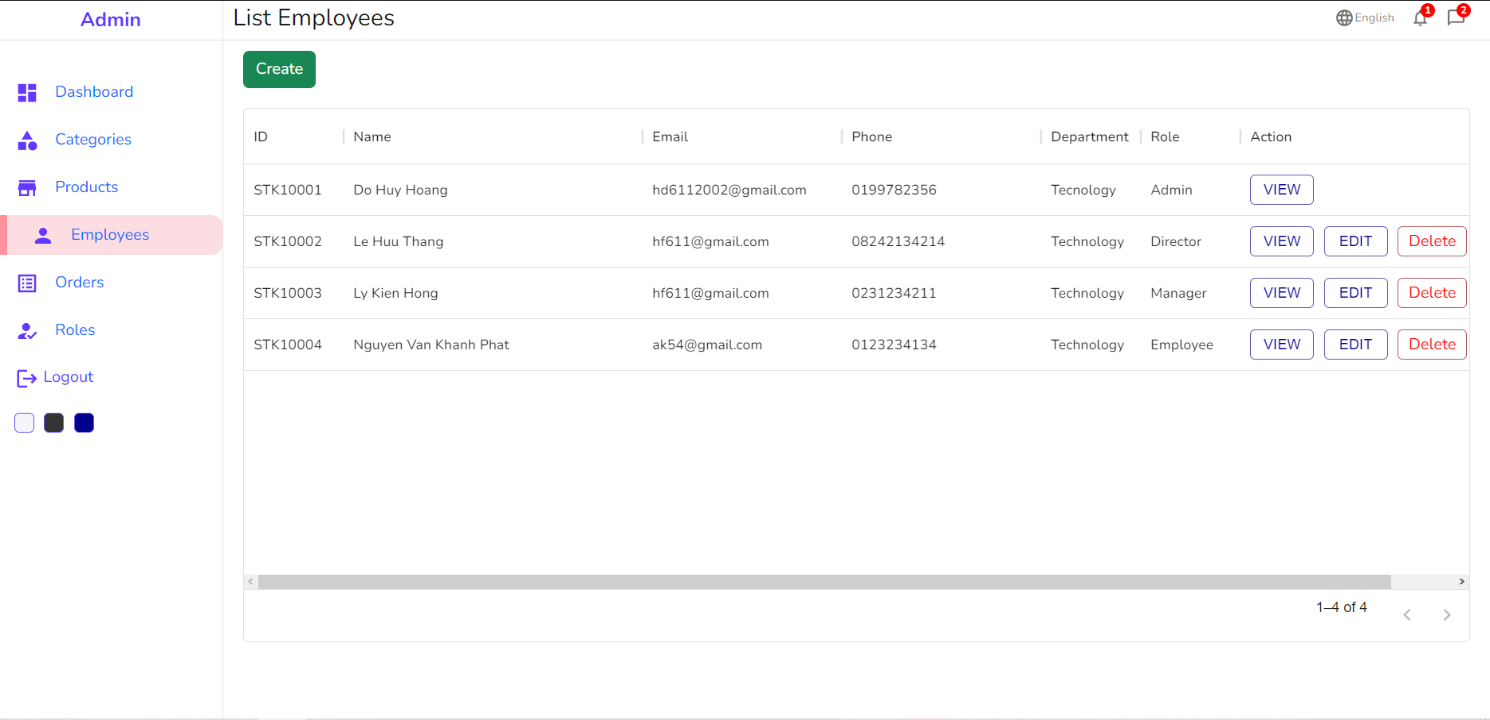
**6.3. Categories**

****

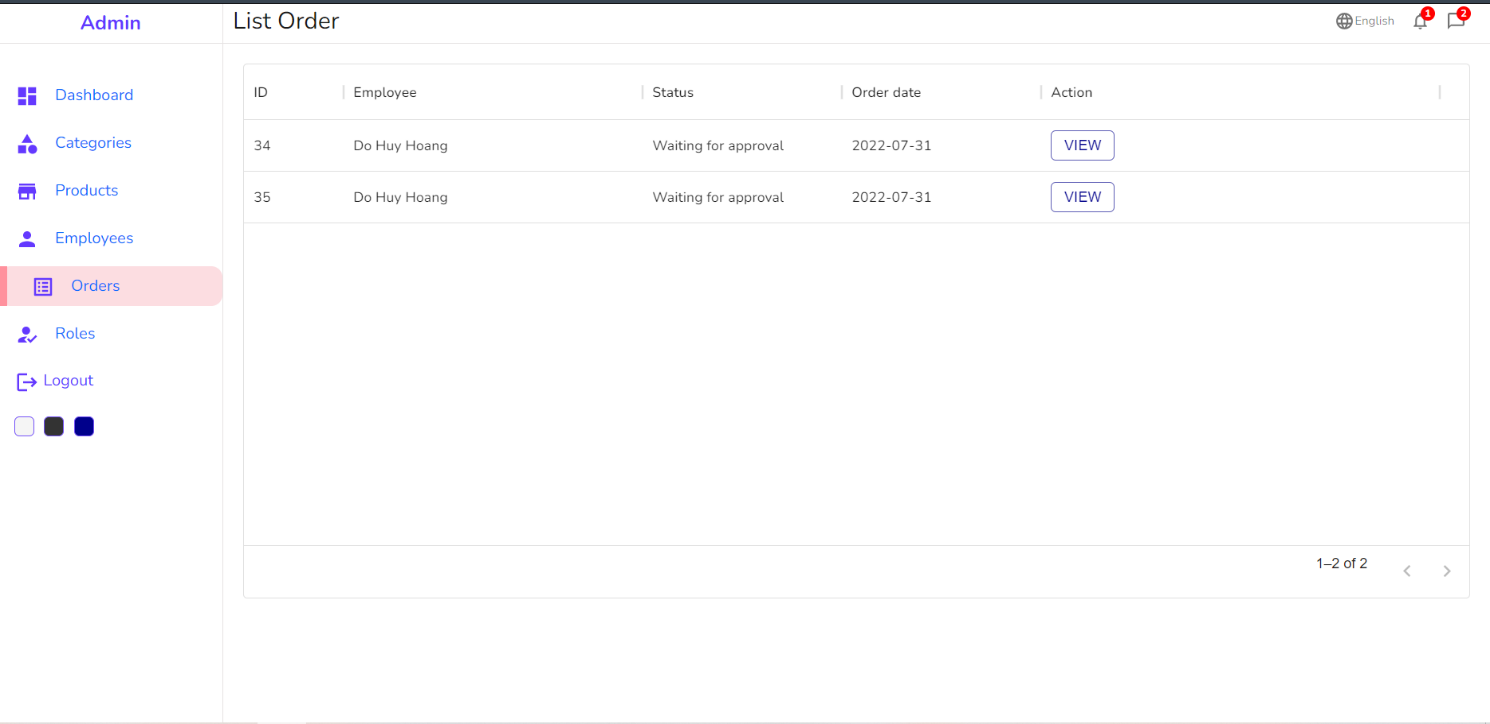
**6.4. Products**

****

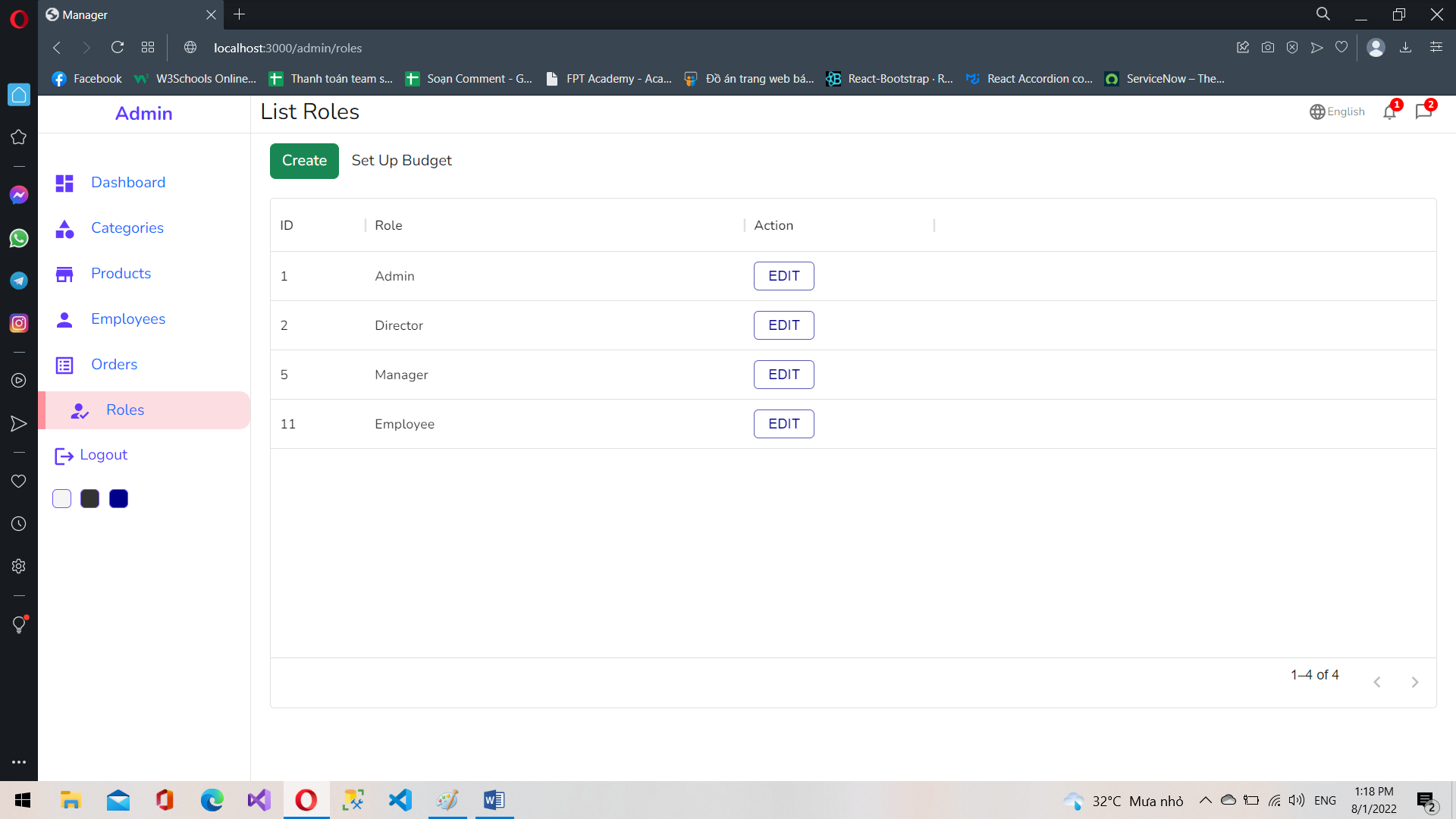
**6.5. Employees**

****

**6.6 . Orders**

****

**6.7. Roles**



|  |
| --- |
| TaskSheet |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.: Project title: Stationery Management** | | **Activity Plan Prepared by: Trí** | | **Date of Preparation of Activity Plan** | | |
| **Sr.No.** | **Task** | **Start Date** | **End Date** | **Actual day** | **Member’s Name** | **status** |
| **1** | Draw Entity Relationship Diagram |  |  |  |  |  |
| **2** | Describing Table |  |  |  |  |  |
| **3** | FrontEnd- HomePage implementation |  |  |  |  |  |
| **4** | FrontEnd- Cart implementation |  |  |  |  |  |
| **5** | FrontEnd- Category implementation |  |  |  |  |  |
| **6** | FrontEnd- Profile implementation |  |  |  |  |  |
| **7** | FrontEnd- Checkout implementation |  |  |  |  |  |
| **8** | FrontEnd- Product implementation |  |  |  |  |  |
| **9** | BackEnd- Login implementation |  |  |  |  |  |
| **10** | BackEnd- Dashboard implementation |  |  |  |  |  |
| **11** | BackEnd- Categories implementation |  |  |  |  |  |
| **12** | BackEnd- Orders implementation |  |  |  |  |  |
| **13** | BackEnd- Products implementation |  |  |  |  |  |
| **14** | BackEnd- Roles implementation |  |  |  |  |  |
| **15** | BackEnd- Employees implementation |  |  |  |  |  |