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

**eProject Document**

Clinic Management

System

|  |  |
| --- | --- |
| **Group 7** | |
| **Members** | Tran Van Hung - Student1247023(Team Leader)  Dinh Hoang Son - Student1240564 |
| **Instructor** | Lai Duc Chung |
| **Batch** | C1908G2 |
| **Semester** | III |

**🙠 Date of issue 08-16-2021 🙢**

**Table of Content**

1. **Problem definition**
   1. ***Introduction***
   2. ***Proposed scenario***
2. **Customer Requirement Specification**
   1. ***Customer Requirement***
   2. ***Hardware – Software Requirements***
3. **Architecture and Design of the System**
4. **Flow Chart**
5. **Data Flow Diagram**
6. **Entity Relationship Diagram**
   1. ***Entity relationship diagram***
   2. ***Entity - Attributes***
7. **Database Design/Structure**
   1. ***Database relationship diagram***
   2. ***Table design***
8. **Some typical interface**
9. **Check List of Validation**
10. **Submission Checklist**
11. **Problem Definition**
    1. **Introduction**

From the beginning in the scientific laboratories and research establishments, computers have enclosed to become essential for modern business and industry. Today their presence is felt in all way of life such as making passenger reservation for the railways, forecasting the weather, monitors and run power plants, helping hospital to render service to patients etc.

* 1. **Proposed Scenario**

Computers have very large storing capacity. Computers also do not make mistakes. Error in computing is generally due to negligence rather than logical faults. The vital characteristics such as each module are to play a center role in ourselves. Since the computer is an electronic data processing machine, it can perform innumerable number of operators in just a few seconds. This processing speed of a computer is usually measured in nanosecond. Computer can perform a wide range of jobs with speed, accuracy and diligence. The areas of application are many:

- Scientific Application

- Medical Application

- Educational Application

- Business Application

The software is developed for providence clinic. This project deals with the task of computerizing the lab and pharmacy of the providence clinic. The software is developed in Visual Basic containing coding programs. It is a very fast high level language developed mainly for business purpose. Some new scientific machine and medicines must be stated in the Home Page. Home Page must be designed with an Introduction to the site and all the navigation links available. Upcoming activities must be flashed in the Home Page. The image must be added with its caption and alt attribute defined.

**ADMIN:** There should be admin Login separate where the entire data of the application can be updated. Admin must maintain the client details and online transaction details. Even keep the track of the all the education activities carried like seminar, practical’s. When the lecture is to be conducted and at what time must be stated in the education link.

1. **Customer Requirement Specification**
   1. **Customer Requirement**
      1. **Users of the system**

* Admin
* Customers
  + 1. **System functions**

|  |  |
| --- | --- |
| **User** | **Functions** |
| **Customers** | * Login, Registration, Logout * Online Purchase * Feedback * View List Orders * View List Order details |
| **Admin** | * Medicine management * Type of medicine management * Employee management * Equipment management * Lecture management * Subject management * View of statistics table * View List Orders * View List Customers * View List Order details |

* 1. **Common functions**

|  |  |
| --- | --- |
| **Login** | |
| Input | Login to system, user must be enter username and password |
| Process | Access information in database |
| Output | Result of login |

* 1. **Admin functions**

|  |  |
| --- | --- |
| **Employee management** | |
| Input | Information details of employee such as ID, Name, Birthday, Email, Address, Gender, Phone, Role, Certificate, Status, Professional knowledge, ID Lecture… |
| Process | Access information in database to modify, add new employee to database, delete from database. |
| Output | Display or maintain Information detail of Employee. |
| **Lecture management** | |
| Input | Information details of Lecture such as ID, Title, Date of occurrence, Price, Content, Picture, Status, Id Subject… |
| Process | Access information in database to modify, add new lecture to database, delete from database. |
| Output | Display or maintain Information detail of Lecture. |
| **Subject management** | |
| Input | Information details of Subject such as ID, Name, Status |
| Process | Access information in database to modify, add new Subject to database, delete from database. |
| Output | Display or maintain Information detail of Subject. |
| **Medicine management** | |
| Input | Information details of Medicine such as ID, Name, Status, Quantity, Manufacturing Date, Origin of Production, Producer, **Expiry Date,** Description, **Picture, ID Type of medicine …** |
| Process | Access information in database to modify, add new Medicine to database, delete from database. |
| Output | Display or maintain Information detail of Medicine. |
| **Type of medicine management** | |
| Input | Information details of Type of Medicine such as Name |
| Process | Access information in database to modify, add new Type of Medicine to database, delete from database. |
| Output | Display or maintain Information detail of Type of Medicine. |
| **Equipment management** | |
| Input | Information details of Equipment such as ID, Name, Date Add, Picture, Unit, Price, Description, Quantity… |
| Process | Access information in database to modify, add new equipment to database, delete from database. |
| Output | Display or maintain Information detail of equipment. |
| **View List Order Details** | |
| Input | View List Order Details. |
| Process | Find the Order Details by Order ID or customer name in database |
| Output | Display list Order Details. |
| **View List Customers** | |
| Input | View List Customers. |
| Process | Find customer by Customer ID or customer name in database. |
| Output | Display the list customers. |
| **View List Orders** | |
| Input | View List Orders |
| Process | Find the bill by Order ID or customer name in database |
| Output | Display the list Orders. |
| **View of statistics table** | |
| Input | View of statistics table |
| Process | Find statistics table order by Date, Week, Month, Year |
| Output | Display the statistics table. |

* 1. **Customer functions**

|  |  |
| --- | --- |
| **Place an order** | |
| Input | Customer fills required information: name, address, email, equipment, etc… |
| Process | Send requirement to system. |
| Output | Result of send. |
| **View the bill of him/her** | |
| Input | View the bill |
| Process | Find the bill in database |
| Output | Display detail of bill |
| **Search the order status of him/her** | |
| Input | Customer fills appropriate information to search. |
| Process | Find the order in database. |
| Output | Display the result. |
| **Feedback** | |
| Input | Information which send to company. |
| Process | Maintain the information to database |
| Output | Result of feedback |
| **Registration** | |
| Input | Customer fills required information: account, password, etc… |
| Process | Send requirement to system. |
| Output | Result of send. |

* 1. **Accountant functions**

|  |  |
| --- | --- |
| **View List Bills** | |
| Input | View List Bills. |
| Process | Find the bill by Order ID or customer name in database. |
| Output | Display list Bills. |
| **Generate bills** | |
| Input | Generate bills for Customers every month. |
| Process | System automatic calculated to generate the bill. |
| Output | Send bill to customer’s email and maintain the bill in database. |
| **Update status of bills** | |
| Input | Update status of bill for customer |
| Process | Find the bill by Order ID or customer name in database to update. |
| Output | Status of bill was updated |

* 1. **Technician functions**

|  |  |
| --- | --- |
| **Keep track list orders** | |
| Input | View all orders |
| Process | Find the orders in database |
| Output | Display list orders |
| **Update order status** | |
| Input | Status of order which want to modify |
| Process | Find the order in database to update status |
| Output | Order status was updated |
| **Create new connection** | |
| Input | Information of connection such as type, information of customer… |
| Process | Maintain new connection to database. |
| Output | Result of create new connection |
| **Change the type of connection for order (Dial –up or Broad Band, Land Line…)** | |
| Input | Type of connection |
| Process | Find the connection in database which want to change. |
| Output | Result of change. |

* 1. **Hardware – Software requirement**

### Recommended Hardware Requirements

* CPU: should use core i5 or higher.
* RAM: 4GB RAM or more is recommended.
* Hard Disk: 20 GB.
* OS: Windows 10 and up.
* Chrome new version.
* Internet Information Services (IIS) 10.0.22377.1000

### Software Requirements

### Either or combination of the following Software’s are to be used:

* Environment: Microsoft .NET Framework 4.7.2
* Database Management System: Microsoft SQL Server Management Studio 18
* IDE: Microsoft Visual Studio 2019 preview
* Web Server: Internet Information Services (IIS 10.0.22377.1000)
* Chrome new version.

1. **Architecture and design of the system**

The Application will be made of a Web-based distributed three-tier architecture.

User Interface with HTML

Web - Tier

And Razor pages

Middle Tier

C# file containing

business logic

Database - Tier

SQL Server 2014

**Web – based distributed 3 - Tier Architecture of the Project**

1. **Flow chart**
   1. **Login**

Start

Start

Display confirm exit

Input data

Check Data

Display Error

**false**

Logout

**true**

Stop

Successful login

b. System Logout

Stop

1. System Login
   1. **Add / Update Data**

Start

Start

Load data

Input Data

**false**

Display error

Input Data

Display error

Check Data

**true**

Check Data

Save database

**false**

**true**

Save database

Display messenger

Display messenger

Stop

1. Add data

Stop

b. Update data

* 1. **Search data**

Start

Input keyword

Display error

Check data

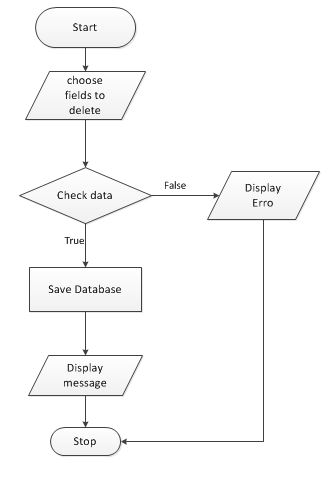
**false**

**true**

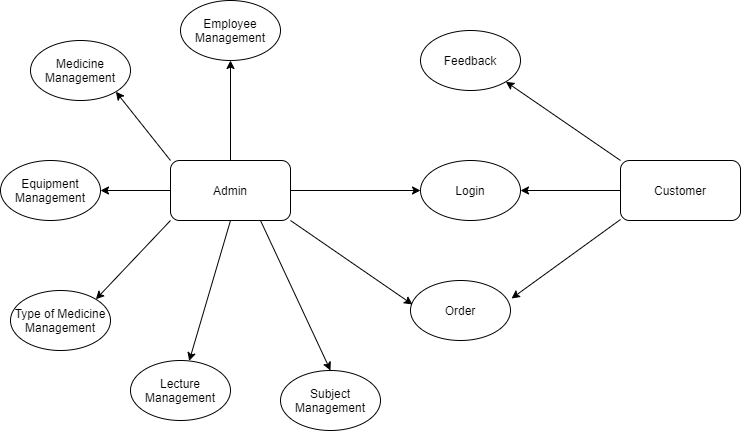
Display result

Stop

* 1. **Delete Data**

****

1. **Data flow diagram**
   1. **General Diagram**

****

* 1. **Use case**

1. Login

Login

Users

Customer

1. Employee Management

Search

Login

Employee

Admin

1. Search

Login

Search

Users

1. Equipment Management

Login

Equipment

Admin

Search

1. View Orders

Search

Login

New Order

Customers

Users

1. Feedback

Customer

Feedback

1. Feedback Management

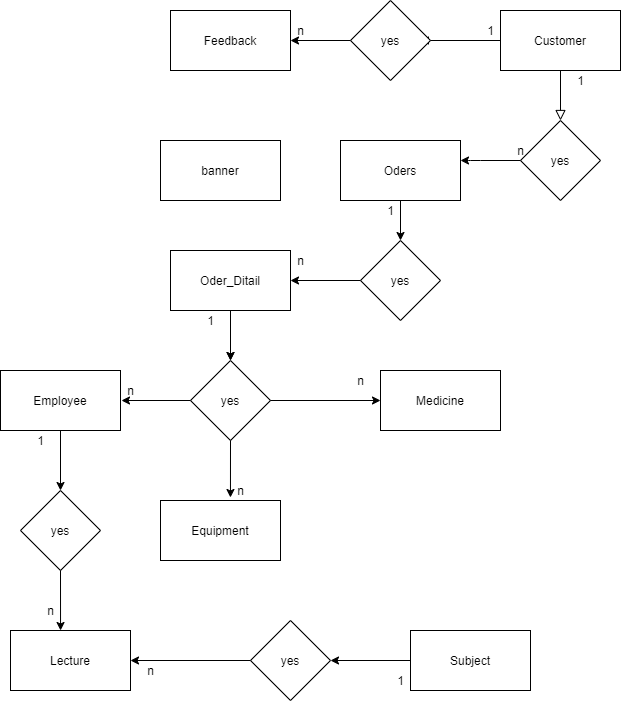
Search

Login

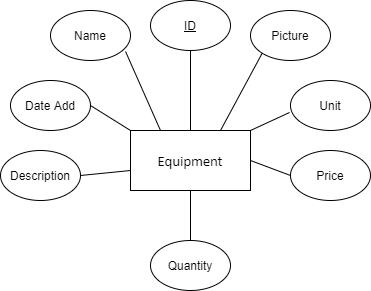
Feedback Manager

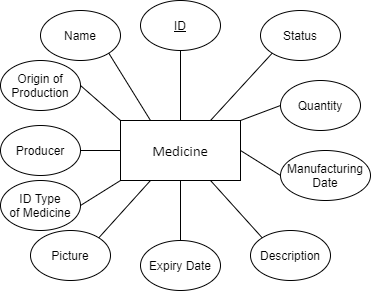
Admin

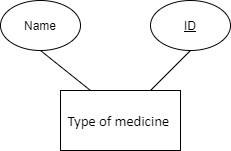
1. **Entity relationship diagram**
   1. **Entity relationship diagram**

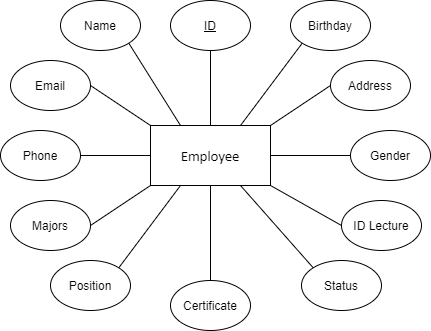
****

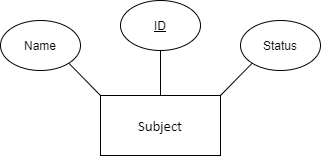
* 1. **Entity - Attributes**

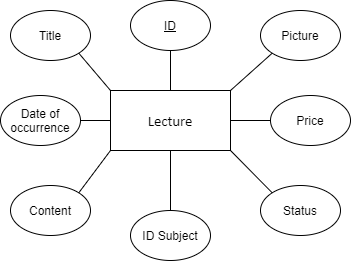
****

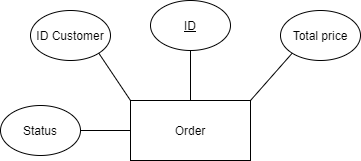
****

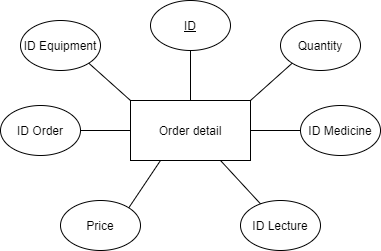
****

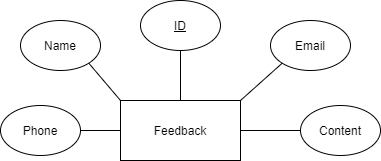
****

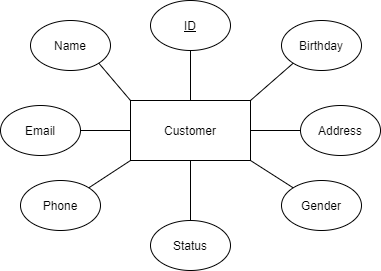
****

****

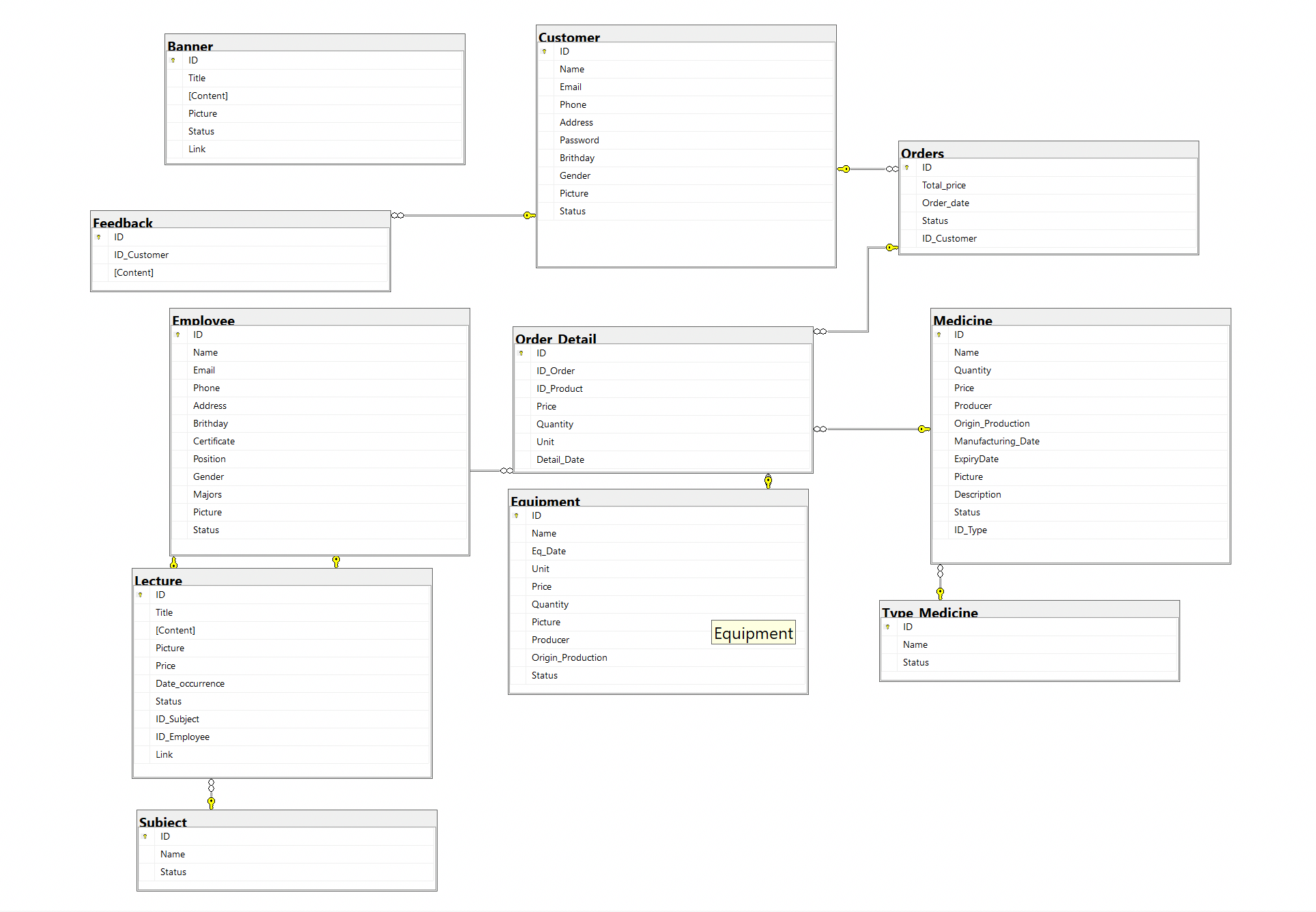
****

****

****

****

1. **Database Design/Structure**
   1. **Database relationship diagram**

****

* 1. **Table Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:Equipment** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | Nvarchar(100) | 🞎 | PK | ID of Equipment |
| Name | Nvarchar(200) | 🞎 |  | Name of Equipment |
| Eq\_Date | Datetime | 🗹 |  | Created date of Equipment |
| Unit | Nvarchar(100) | 🞎 |  | Unit of Equipment |
| Price | Float | 🞎 |  | Price of Equipment |
| Quantity | Int | 🞎 |  | Quantity of Equipment |
| Picture | Nvarchar(200) | 🞎 |  | Picture of Equipment |
| Status | int | 🗹 |  | Status of Equipment |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:Type\_Medicine** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | Int | 🞎 | PK | ID of Type |
| Name | Nvarchar(200) | 🞎 |  | Name of Type |
| Status | int | 🗹 |  | Status of Type |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:Medicine** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | Nvarchar(100) | 🞎 | PK | ID of Medicine |
| Name | Nvarchar(200) | 🞎 |  | Name of Medicine |
| Quantity | Int | 🞎 |  | Quantity of Medicine |
| Price | Float | 🞎 |  | Price of Medicine |
| Producer | Nvarchar(200) | 🞎 |  | Producer of Medicine |
| Origin\_Production | Nvarchar(200) | 🞎 |  | Origin of production of Medicine |
| Manufacturing\_Date | datetime | 🗹 |  | Manufacturing date of Medicine |
| ExpiryDate | datetime | 🗹 |  | Expiry date of Medicine |
| Picture | nvarchar(200) | 🗹 |  | Picture of Medicine |
| Description | text | 🗹 |  | Description of Medicine |
| Status | int | 🗹 |  | Status of Medicine |
| ID\_Type | int | 🞎 | FK | ID of Type\_Medicine |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:Subject** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | Int | 🞎 | PK | ID of Subject |
| Name | nvarchar(200) | 🞎 |  | Name of Subject |
| Status | int | 🗹 |  | Status of Subject |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Employee** | | | | |
| **Column** | **Data Type** | **Allow Nulls** | **Key** | **Description** |
| ID | Nvarchar(200) | 🞎 | PK | ID of Employee |
| Name | Nvarchar(50) | 🞎 |  | Name of Employee |
| Birthday | date | 🞎 |  | Birthday of Employee |
| Gender | bit | 🞎 |  | Gender of Employee |
| Address | nvarchar(100) | 🞎 |  | Address of Employee |
| Position | nvarchar(100) | 🗹 |  | Position of Employee |
| Certificate | nvarchar(100) | 🗹 |  | Certificate of Employee |
| Email | varchar(100) | 🞎 |  | Email of Employee |
| Phone | varchar(100) | 🞎 |  | Phone of Employee |
| Majors | nvarchar(100) | 🗹 |  | Majors of Employee |
| Status | int | 🞎 |  | Status of Employee |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Lecture** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | int | 🞎 | PK | ID of Lecture |
| Title | Nvarchar(200) | 🞎 |  | Title of Lecture |
| Content | Text | 🞎 |  | Content Lecture |
| Price | float | 🞎 |  | Price of Lecture |
| Picture | Int | 🗹 |  | Picture Lecture |
| Date\_occurrence | datetime | 🗹 |  | Date of occurrence of Lecture |
| Status | int | 🗹 |  | Status of Lecture |
| ID\_Subject | Int | 🞎 | FK | ID of Subject |
| ID\_Employee | nvarchar(200) | 🞎 | FK | ID of Employee |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table:Feedback** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | Int | 🞎 | PK | ID of Feedback |
| ID\_Customer | int | 🞎 | FK | ID of Customer |
| Content | Text | 🞎 |  | Content of Feedback |

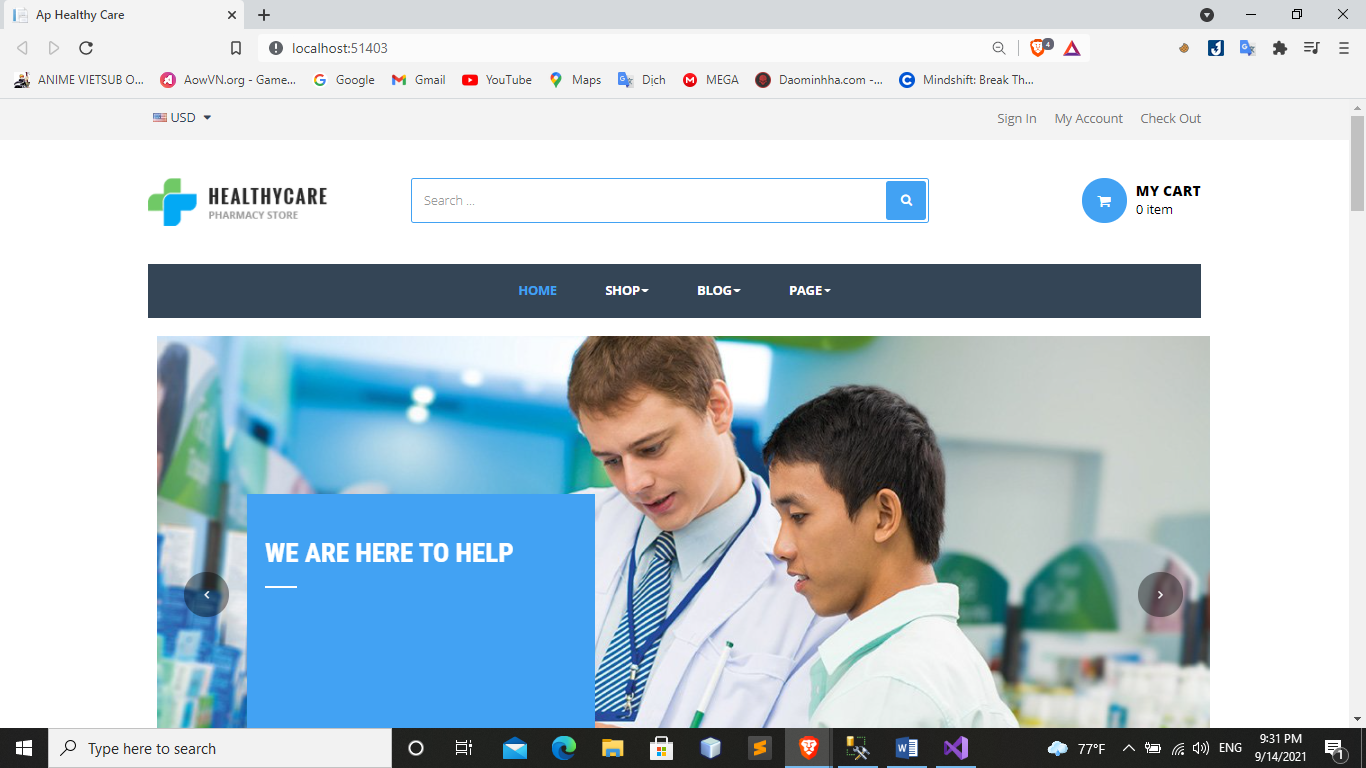
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Customer** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | int | 🞎 | PK | ID of Customer |
| Name | Nvarchar(200) | 🞎 |  | Name of Customer |
| Email | varchar(100) | 🞎 |  | Email of Customer |
| Phone | varchar(100) | 🞎 |  | Phone of Customer |
| Address | text | 🞎 |  | Address of Customer |
| Brithday | text | 🗹 |  | Brithday of Customer |
| Gender | bit | 🞎 |  | Gender of Customer |
| Picture | nvarchar(200) | 🞎 |  | Picture of Customer |
| Status | int | 🗹 |  | Status of Customer |

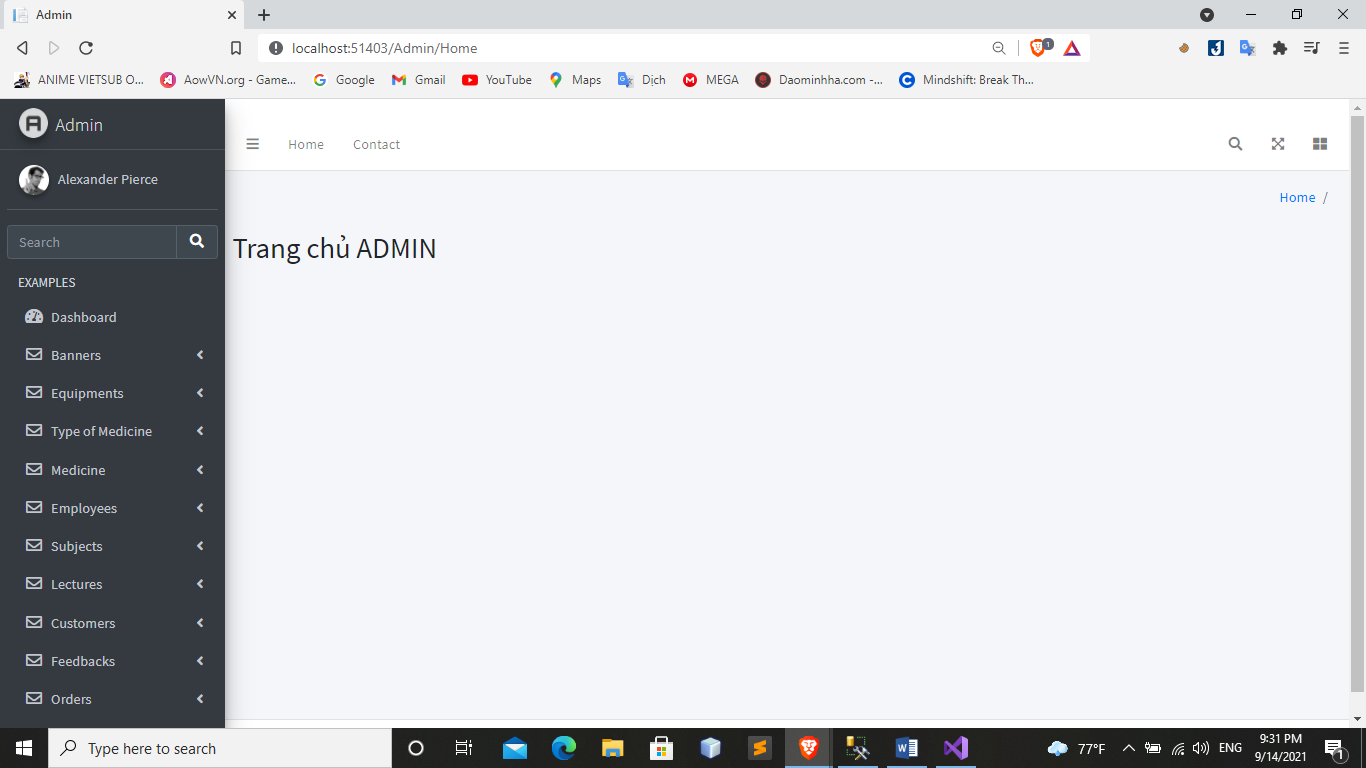
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Orders** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | int | 🞎 | PK | ID of Order |
| Order\_date | datetime | 🞎 |  | Order Date of Order |
| Total\_price | float | 🞎 |  | Total price of Order |
| Status | int | 🗹 |  | Status of order |
| ID\_Customer | int | 🞎 | FK | ID of Customer |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Order\_Detail** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | int | 🞎 | PK | ID of Order\_detail |
| ID\_Order | int | 🞎 |  | ID of Order |
| ID\_Product | Varchar(50) | 🗹 | PK | ID of Lecture, ID of Equipment, ID of Medicine |
| Price | float | 🞎 |  | Price |
| Quantity | int | 🞎 |  | Quantity |
| Unit | Nvarchar(100) | 🗹 |  | Unit |
| Detail\_date | datetime | 🞎 |  | Date of Order Detail |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table: Banner** | | | | |
| **Column** | **Data type** | **Allow Nulls** | **Key** | **Description** |
| ID | int | 🞎 | PK | ID of Banner |
| Title | Nvarchar(200) | 🞎 |  | Title of Banner |
| Content | Text | 🞎 |  | Content of Banner |
| Picture | nvarchar(200) | 🗹 |  | Picture of Banner |
| Link | nvarchar(200) | 🗹 |  | Link of Banner |
| Status | int | 🗹 |  | Status of Banner |

1. **Some typical interface**





1. **Check list of validation**

|  |  |
| --- | --- |
| Option | Validate |
| Can a new user who get registered, enter the application after logging in ? | Yes |
| Do all the option present in the application display the correct result | Yes |
| Does the application’s functionality resolve the user problem and satisfy their needs ? | Yes |
| Has the hardware and software been correctly chosen ? | Yes |

1. **Submission checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.No | Particular | Yes | No | Comments |
| 1 | Are the users able to login to application after validations is performed on the on the user name and password ? | Yes |  |  |
| 2 | Are the user able to modify their detail after getting registered | Yes |  |  |
| 3 | Are all the screen contents devoid spelling mistakes | Yes |  |  |
| 4 | Is the application user-friendly | Yes |  |  |

**THE END**