**LỜI MỞ ĐẦU**

Trong những năm gần đây, công nghệ thông tin không ngừng phát triển một cách mạnh mẽ và hiện đại. Sự ra đời của công nghệ thông tin làm phong phú bộ mặt xã hội, đời sống con người được nâng cao rõ rệt, đóng góp to lớn cho sự phát triển của nhân loại. Với xu thế toàn cầu hóa nền kinh tế thế giới, đặc biệt là nhu cầu trao đổi hàng hóa của con người ngày càng nhiều về số lượng và chất lượng, nhu cầu sử dụng Internet càng nhiều và hình thức kinh doanh trên mạng ngày càng đa dạng và phong phú được nhiều người sử dụng và ưa chuộng. Vì vậy, nhiều nhà kinh doanh đã áp dụng phương pháp bán hàng qua mạng để khách hàng dễ dàng tìm kiếm thông tin mà không cần tốn kém thời gian và chi phí.

Từ những vấn đề đặt ra ở trên, được sự đồng ý và hướng dẫn tận tình của giảng viên…, nhóm chúng em đã hoàn thành đề tài “***Thiết kế website bán bảo hiểm trực tuyến***”.

Chỉ trong thời gian ngắn học tập và rèn luyện tại trường Học viện Công nghệ Bưu chính Viễn thông chúng em được thầy cô trang bị các kiến thức cơ bản về môn học, đề tài đã cơ bản hoàn thành, song vì thời gian và trình độ hiểu biết còn hẹn hẹp nên không khỏi tránh được những thiếu sót. Vì vậy, chúng em rất mong được sự giúp đỡ, đóng góp của quý thầy cô để nội dung của bài báo cáo được hoàn thiện hơn. Chúng em xin chân thành cảm ơn!

***PHẦN 1: TỔNG QUAN VỀ NGÔN NGỮ KỊCH BẢN PHP VÀ MYSQL***

***PHẦN 1: Giới thiệu về ngôn ngữ PHP***

**1.1) Php là gì?**

PHP là một ngôn ngữ lập trình phía máy chủ (server-side) được sử dụng rộng rãi trong phát triển web. PHP là viết tắt của "PHP: Hypertext Preprocessor" (trước đây là "Personal Home Page"), và nó đã trở thành một trong những ngôn ngữ phát triển web phổ biến nhất trên Internet.

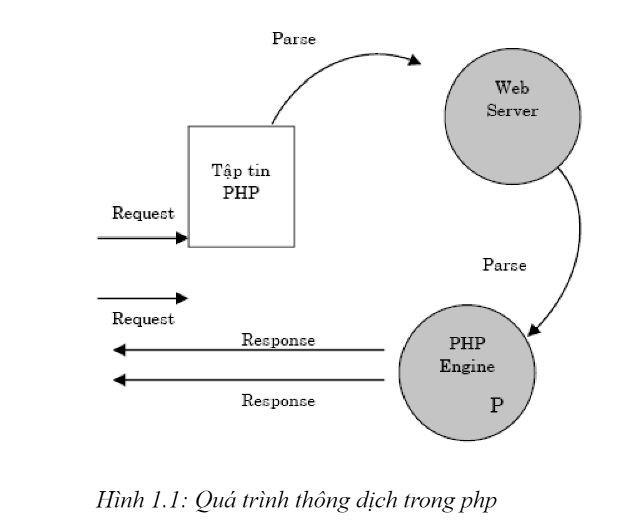
PHP được thiết kế để tạo ra các trang web động và ứng dụng web. Với PHP, bạn có thể thực hiện các tác vụ như tạo, xử lý và lưu trữ dữ liệu từ cơ sở dữ liệu, tương tác với người dùng qua biểu mẫu web, tạo và quản lý phiên làm việc, và xử lý các yêu cầu HTTP.

**1.2) Quá trình thông dịch trang PHP**

PHP là kịch bản trình chủ được chạy trên nền PHP Engine, cùng với ứng dụng Web Server để quản lí chúng.

Khi trang PHP được gọi, Web Server triệu gọi PHP Engine để thông dịch, dịch trang PHP và trả về kết quả cho người sử dụng là một trang thuần HTML.

Ta có mô hình thông dịch như sau:



Đầu tiên, một đoạn mã PHP luôn được bắt đầu và kết thúc bởi cặp thẻ theo cú pháp:

**<?php**

**Các câu lệnh;**

**?>**

Sau đó, ta cũng có thể nhúng các lệnh của PHP vào trang HTML và đoạn mã PHP có thể đặt bất kì đâu trong tài liệu. Thông thường một trang PHP bao gồm các thẻ HTML như một trang HTML nhưng có thêm các đoạn mã PHP.

***PHẦN 2: Tổng quan về SQL***

**2.1) Giới thiệu SQL:**

SQL (Structured Query Language) là một ngôn ngữ truy vấn có cấu trúc được sử dụng để quản lý và tương tác với cơ sở dữ liệu quan hệ. SQL là tiêu chuẩn được chấp nhận rộng rãi cho các hệ quản trị cơ sở dữ liệu quan hệ như MySQL, Oracle, SQL Server, PostgreSQL và SQLite.

SQL cho phép người dùng thực hiện các thao tác như tạo, truy vấn, cập nhật, xóa và quản lý dữ liệu trong cơ sở dữ liệu quan hệ. Điều này cho phép lưu trữ, truy xuất và xử lý dữ liệu một cách hiệu quả.

**2.2) Kết nối và tạo cơ sở dữ liệu:**

Kết nối vào hệ quản trị cơ sở dữ liệu (HQCSDL) của bạn bằng cách sử dụng công cụ quản lý cơ sở dữ liệu hoặc ngôn ngữ lập trình phù hợp. Ví dụ, nếu bạn sử dụng MySQL, bạn có thể sử dụng phpMyAdmin hoặc kết nối thông qua mã nguồn PHP.

Chọn cơ sở dữ liệu: Sau khi kết nối thành công, chọn cơ sở dữ liệu mà bạn muốn tạo hoặc sử dụng. Nếu cơ sở dữ liệu chưa tồn tại, bạn có thể tạo mới.

Tạo câu lệnh SQL để tạo cơ sở dữ liệu: Sử dụng câu lệnh "CREATE DATABASE" để tạo cơ sở dữ liệu mới. Đặt tên cho cơ sở dữ liệu và thực thi câu lệnh.

*2.2.1)Tạo cơ sở dữ liệu*

Ví dụ với MySQL, bạn có thể sử dụng phpMyAdmin hoặc mã PHP để tạo cơ sở dữ liệu.

*-- Cơ sở dữ liệu: `baohiem`*

*-- Cấu trúc bảng cho bảng `admin`*

CREATE TABLE `admin` (

`id\_admin` int(5) NOT NULL CHECK (octet\_length(`id\_admin`) = 5),

`username` varchar(30) NOT NULL,

`password` varchar(100) NOT NULL,

`email\_admin` varchar(64) NOT NULL,

`sdt\_admin` varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `admin` (`id\_admin`, `username`, `password`, `email\_admin`, `sdt\_admin`) VALUES

(11225, 'admin', '091203', 'admin123@gmail.com', '0156622005');

*-- Cấu trúc bảng cho bảng `chinhsach`*

CREATE TABLE `chinhsach` (

`id\_baohiem` int(11) NOT NULL,

`ten\_baohiem` varchar(100) NOT NULL,

`gia\_baohiem` varchar(50) NOT NULL,

`hinhanh` text NOT NULL,

`mota` longtext NOT NULL,

`ttt` tinyint(4) NOT NULL,

`ttd` tinyint(4) NOT NULL,

`tgck` int(11) NOT NULL,

`id\_danhmuc` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `chinhsach` (`id\_baohiem`, `ten\_baohiem`, `gia\_baohiem`, `hinhanh`, `mota`, `ttt`, `ttd`, `tgck`, `id\_danhmuc`) VALUES

(12, 'Bảo hiểm nhân thọ', '100000000', '1685723537\_anh\_bao\_hiem\_nhan\_tho.png', 'Quyền lợi bảo vệ trước thương tật/tử vong.\r\nQuyền lợi bảo vệ trước bệnh ung thư\r\nQuyền lợi chăm sóc y tế\r\nQuyền lợi đầu tư của sản phẩm liên kết chung.\r\nQuyền lợi cho con cái.\r\nQuyền lợi đáo hạn.\r\nQuyền lợi nhận thưởng gia tăng giá trị hợp đồng.', 1, 65, 10, 1),

(13, 'Bảo hiểm xe máy dưới 50cc', '55000', '1685723886\_xe-may-50cc-Cub-81-Viet-Thai.jpg', 'Người tham gia giao thông có thể mua thêm nhằm mang lại quyền lợi chi trả bồi thường tài chính về tài sản hoặc người ngồi trên xe', 18, 65, 1, 2),

(14, 'Bảo hiểm thất nghiệp', '200000', '1685724200\_dieu-kien-nhan-bao-hiem-that-nghiep.jpg', 'Bảo hiểm thất nghiệp là một loại bảo hiểm xã hội hỗ trợ người lao động trong trường hợp người lao động chấm dứt hợp đồng lao động với đơn vị sử dụng lao động. Từ đó, hỗ trợ người lao động học nghề, tìm kiếm việc làm dựa trên cơ sở Qũy bảo hiểm thất nghiệp.\r\nViết cho Đi làm web\r\n', 55, 60, 1, 1),

(15, 'Bảo hiểm cháy nổ', '500000', '1685724408\_bao-hiem-chay-no.png', 'Một hình thức giữ gìn và bảo vệ dành cho tổ ấm thân yêu của người mua và gia đình. Những thiệt hại vật chất do cháy nổ sẽ được đền bù\r\n', 18, 60, 5, 2),

(16, 'Bảo hiểm xe máy trên 50cc', '66000', '1685724538\_xe\_110cc.png', 'Bảo hiểm bắt buộc và bảo hiểm tự nguyện bảo vệ bản thân người sử dụng không may gặp vấn đề đâm đụng xảy ra. Với các phạm vi được bảo hiểm thì công ty bảo hiểm sẽ có thể bù đắp một phần tài chính để bù thiệt hại.', 18, 65, 5, 2);

*-- Cấu trúc bảng cho bảng `danhmuc`*

CREATE TABLE `danhmuc` (

`id\_danhmuc` int(11) NOT NULL,

`tendanhmuc` varchar(100) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `danhmuc` (`id\_danhmuc`, `tendanhmuc`) VALUES

(1, 'Bảo hiểm người'),

(2, 'Bảo hiểm tài sản');

*-- Cấu trúc bảng cho bảng `hopdong`*

CREATE TABLE `hopdong` (

`id\_hopdong` int(11) NOT NULL,

`id\_user` int(11) NOT NULL,

`id\_baohiem` int(11) NOT NULL,

`ten\_baohiem` varchar(100) DEFAULT NULL,

`ten\_khachhang` varchar(50) NOT NULL,

`gioitinh` enum('Nam','Nữ') NOT NULL,

`ngaysinh` date NOT NULL,

`cccd` varchar(12) NOT NULL,

`tuoi` tinyint(4) GENERATED ALWAYS AS (timestampdiff(YEAR,`ngaysinh`,curdate())) VIRTUAL,

`nghenghiep` varchar(50) DEFAULT NULL,

`sodienthoai` varchar(10) NOT NULL,

`diachi` text NOT NULL,

`ngayky\_hopdong` timestamp NOT NULL DEFAULT current\_timestamp(),

`thoihan\_baohiem` int(11) DEFAULT NULL,

`handongtien` date NOT NULL,

`trangthai` enum('0','1') NOT NULL,

`tien\_baohiem` bigint(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

DELIMITER $$

CREATE TRIGGER `trg\_calculate\_handongtien` BEFORE INSERT ON `hopdong` FOR EACH ROW BEGIN

DECLARE v\_id\_danhmuc INT;

DECLARE v\_handongtien DATE;

-- Lấy giá trị id\_danhmuc từ bảng chinhsach dựa trên id\_baohiem

SELECT id\_danhmuc INTO v\_id\_danhmuc FROM chinhsach WHERE id\_baohiem = NEW.id\_baohiem;

-- Tính toán giá trị handongtien dựa trên id\_danhmuc

IF v\_id\_danhmuc = 1 THEN

SET v\_handongtien = DATE\_ADD(NEW.ngayky\_hopdong, INTERVAL 1 YEAR);

ELSEIF v\_id\_danhmuc = 2 THEN

SET v\_handongtien = DATE\_ADD(NEW.ngayky\_hopdong, INTERVAL 6 MONTH);

END IF;

-- Gán giá trị tính toán cho cột handongtien

SET NEW.handongtien = v\_handongtien;

END

$$

DELIMITER ;

*-- Cấu trúc bảng cho bảng `tb\_cart`*

CREATE TABLE `tb\_cart` (

`id\_cart` int(11) NOT NULL,

`id` int(11) NOT NULL,

`code\_cart` varchar(10) NOT NULL,

`cart\_status` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

*-- Cấu trúc bảng cho bảng `tb\_cart\_detail`*

CREATE TABLE `tb\_cart\_detail` (

`id\_cart\_detail` int(11) NOT NULL,

`code\_cart` varchar(10) NOT NULL,

`id\_baohiem` int(11) NOT NULL,

`soluong` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

*-- Cấu trúc bảng cho bảng `tuvan`*

CREATE TABLE `tuvan` (

`id` int(11) NOT NULL,

`ten` varchar(40) NOT NULL,

`email` varchar(64) NOT NULL,

`phone` varchar(10) DEFAULT NULL,

`comment` longtext NOT NULL,

`time` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8\_unicode\_ci;

*-- Cấu trúc bảng cho bảng `user`*

CREATE TABLE `user` (

`id` int(11) NOT NULL,

`ten\_user` varchar(50) NOT NULL,

`email` varchar(64) NOT NULL,

`matkhau` varchar(100) NOT NULL,

`gioitinh` enum('Nam','Nữ') NOT NULL,

`ngaysinh` date NOT NULL,

`tuoi` int(4) GENERATED ALWAYS AS (timestampdiff(YEAR,`ngaysinh`,curdate())) VIRTUAL,

`nghenghiep` text NOT NULL,

`dienthoai` varchar(10) NOT NULL,

`dia\_chi` text NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

INSERT INTO `user` (`id`, `ten\_user`, `email`, `matkhau`, `gioitinh`, `ngaysinh`, `nghenghiep`, `dienthoai`, `dia\_chi`) VALUES

(11, 'Hồ Thanh Nhật', 'htn@gmail.com', '091203', 'Nam', '2003-12-09', 'Sinh viên', '0706830555', 'An Giang');

DELIMITER $$

CREATE TRIGGER `trg\_before\_insert\_user` BEFORE INSERT ON `user` FOR EACH ROW BEGIN

IF ( NEW.ngaysinh >= CURDATE()) THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Bạn không thể có ngày sinh mà ngày đó lại trước ngày hôm nay ;)))';

END IF;

END

$$

DELIMITER ;

*-- Cấu trúc bảng cho bảng `xacnhan\_thanhtoan`*

CREATE TABLE `xacnhan\_thanhtoan` (

`id\_thanhtoan` int(11) NOT NULL,

`id\_user` int(11) NOT NULL,

`id\_baohiem` int(11) NOT NULL,

`id\_cart\_detail` int(11) NOT NULL,

`ten\_baohiem` varchar(50) NOT NULL,

`chu\_so\_huu` varchar(50) NOT NULL,

`cccd` varchar(12) NOT NULL,

`ngaysinh` date NOT NULL,

`tuoi` tinyint(4) GENERATED ALWAYS AS (timestampdiff(YEAR,`ngaysinh`,curdate())) VIRTUAL,

`sdt\_chu` varchar(10) NOT NULL,

`gioitinh\_chu` enum('Nam','Nữ') NOT NULL,

`diachi\_chu` text NOT NULL,

`han\_bao\_hiem\_chu` int(11) NOT NULL,

`tien\_baohiem` bigint(20) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_general\_ci;

DELIMITER $$

CREATE TRIGGER `check\_ngaysinh` BEFORE INSERT ON `xacnhan\_thanhtoan` FOR EACH ROW BEGIN

IF NEW.ngaysinh >= CURDATE() THEN

SIGNAL SQLSTATE '45000'

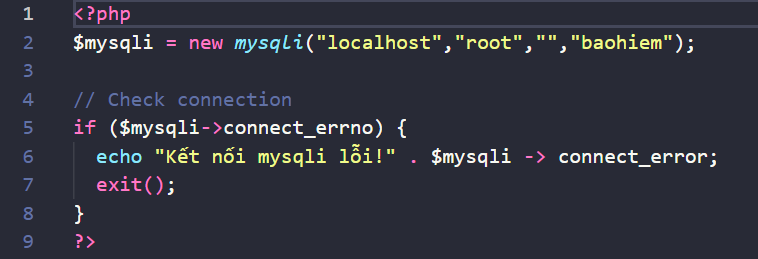
SET MESSAGE\_TEXT = 'Ngaysinh phai nho hon ngay hien tai.';

END IF;

END

$$

DELIMITER ;

*2.2.2) Kết nối cơ sở dữ liệu:* 

*2.2.3) Các câu lệnh SQL:*

2.2.3.1) Câu lệnh SELECT:

Câu SELECT lệnh được sử dụng để chọn dữ liệu từ cơ sở dữ liệu. Dữ liệu trả về được lưu trữ trong một bảng kết quả, được gọi là tập hợp kết quả.

**SELECT** *column1, column2, ...*

**FROM** *table\_name***;**

Ở đây, cột1, cột2, ... là tên trường của bảng mà bạn muốn chọn dữ liệu. Nếu muốn chọn tất cả các trường có trong bảng thì sử dụng cú pháp sau:

**SELECT \* FROM** *table\_name***;**

2.2.3.2) Câu lệnh INSERT:

Câu INSERT INTO lệnh được sử dụng để chèn các bản ghi mới vào một bảng.

**INSERT INTO** *table\_name (column1, column2, column3, ...)*

**VALUES** (*value1, value2, value3, ...*)**;**

2.2.3.3) Câu lệnh UPDATE:

Câu UPDATE lệnh được sử dụng để sửa đổi các bản ghi hiện có trong một bảng.

**UPDATE** *table\_name*

**SET** *column1 = value1, column2 = value2, ...*

**WHERE** *condition;*

2.2.3.4) Câu lệnh DELETE:

Câu DELETE lệnh được sử dụng để xóa các bản ghi hiện có trong một bảng.

**DELETE FROM** *table\_name* **WHERE** *condition;*

2.2.3.5) Câu lệnh JOIN:

Một JOIN mệnh đề được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng, dựa trên một cột có liên quan giữa chúng.

Dưới đây là các loại JOIN khác nhau trong SQL:

**(INNER) JOIN**: Trả về các bản ghi có giá trị khớp trong cả hai bảng

**LEFT JOIN**: Trả về tất cả các bản ghi từ bảng bên trái và các bản ghi phù hợp từ bảng bên phải

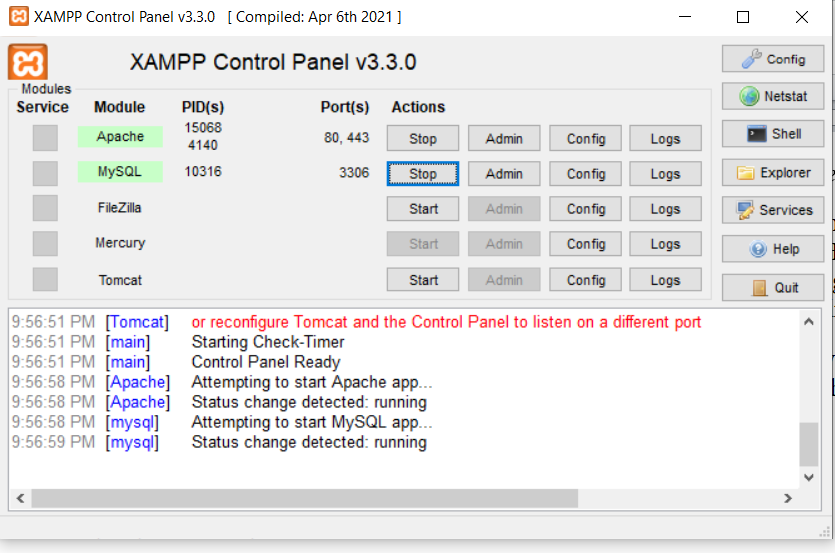
**RIGHT JOIN**: Trả về tất cả các bản ghi từ bảng bên phải và các bản ghi phù hợp từ bảng bên trái

**FULL JOIN**: Trả về tất cả các bản ghi khi có sự trùng khớp trong bảng bên trái hoặc bên phải

***PHẦN 3: Giới thiệu XAMPP – công cụ hỗ trợ hoàn thành website***

Để chạy được PHP chúng ta phải thiết lập môi trường web Server. Tuy nhiên, không phải lúc nào chúng ta cũng đủ kinh phí để mua một Server về phục vụ cho việc học. Điều này vô cùng tốn kém và không cần thiết. Vậy làm thế nào để ta có thể chạy được PHP? Trong bài tập nhóm này, chúng em tìm hiểu về một công cụ giúp giải quyết được vấn đề trên, đó chính là **XAMPP**.

XAMPP là một công cụ tích hợp đầy đủ các thành bao gồm Apache, Mysql, PHP, Perl. Giúp chúng ta tạo ra môi trường web server trên máy tính của mình, để có thể chạy được kịch bản PHP.



***PHẦN 2: KHẢO SÁT VÀ PHÂN TÍCH HỆ THỐNG WEBSITE***

**PHẦN 1: PHẦN CHUNG**

* 1. **Yêu cầu chức năng:**

Đối với website bán bảo hiểm, nhóm nghiên cứu đã tìm hiểu và phân tích tất cả các chức năng cần có để phục vụ nhu cầu mua của khách hàng một cách tốt nhất. Mục tiêu đặt ra với há thống website sẽ có các chức năng sau:

- Hiển thị thông tin các bảo hiểm hàng đầu, các chức năng của website trên trang chủ

- Hiển thị các công dụng hàng đầu của bảo hiểm trong trang tin tức.

- Cho phép tìm kiếm thông tin chi tiết bảo hiểm(ảnh, giá, loại, mô tả,..)

- Cho phép đánh giá, hiển thị đánh giá, bình luận của người mua lên trang chi tiết.

- Trang giỏ hàng hiển thị các loại bảo hiểm khách hàng muốn mua và chuyển sang trang thanh toán.

- Trang account cho phép khách hàng đăng ký và đăng nhập tài khoản để mua hàng.

- Hiển thị thống kê số lượng bảo hiểm, liên hệ, tin tức, tài khoản admin trên trang quản trị.

- Cho phép cập nhật (thêm, sửa, xóa) các loại bảo hiểm bán.

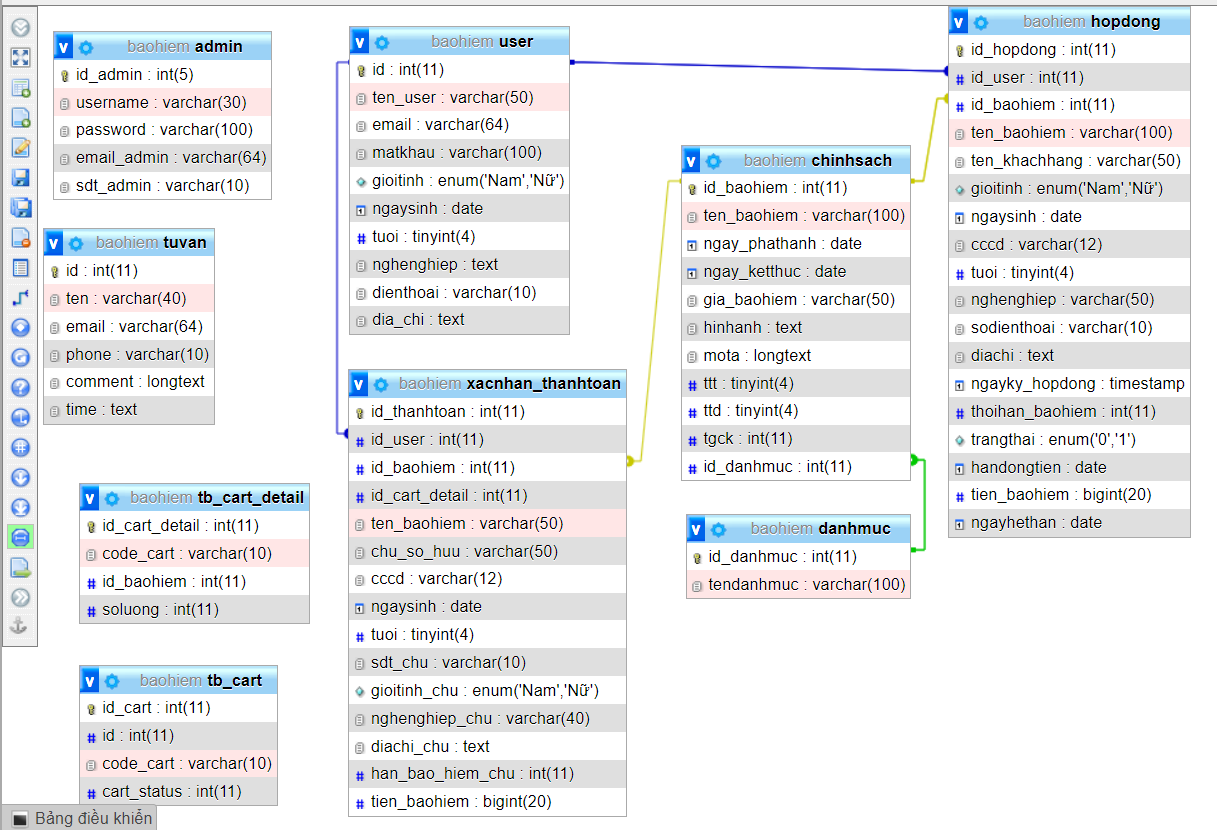
- Cho phép cập nhật (đăng nhập, đổi mật khẩu) thông tin liên hệ, trợ giúp, thông tin admin.

- Hiển thị thống kê sản phẩm, nhà cung cấp, tin tức, tài khoản khách hàng, người dùng,..

Nhóm đã sử dụng MySQL để tạo CSDL cho website bảo hiểm trực tuyến



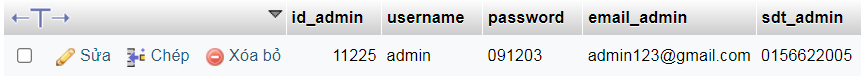
Sơ đồ quan hệ giữa các bảng:



**PHẦN 2: ADMIN**

* 1. **Đăng nhập:**

Khi thêm mới một admin cho trang quÁn trị, bÁng này dùng để lưu trữ thông tin của admin mới đó:



Gồm các thuộc tính:

- id\_admin:lưu lại id của admin khi đăng nhập

- username: tên admin khi đăng nhập

- password: mật khẩu đăng nhập của người dùng

- email\_admin: email đăng nhập của người dùng

- sdt\_admin:số điện thoại đăng nhập.

Các luồng sự kiện:

-Tác nhân: Chọn admin để đăng nhập hệ thống.

- Hệ thống: Hiển thị thông báo chưa đăng nhập và hiện giao diện đăng nhập

- Tác nhân: Nhập email, mật khẩu rồi chọn button Đăng nhập.

- Hệ thống: Kiểm tra username, password admin nhập nếu sai hiển thị lại. Nếu đúng hiển thị thông báo đăng nhập thành công và hiển thị giao diện

- Kết quả: Quá trình đăng nhập thành công, admin được sử dụng các chức năng trong trang quản trị

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

*Hình 1: giao diện admin chưa đăng nhâp và đã đăng nhập*

* 1. **Đổi mật khẩu admin:**

