

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN MÔN HỌC**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**XÂY DỰNG WEB QUẢN LÝ NHÂN VIÊN**

**CÔNG TY NÓN BẢO HIỂM**

**GVHD: TS. NGUYỄN THÀNH SƠN**

**Sinh viên thực hiện:**

**ĐẶNG NGỌC SƠN 18133046**

**ĐÀO VĂN THẮNG 18133050**

**ĐÀO MINH THY 18133055**

**TRẦN MINH TÚ TRUNG 18133060**

**Tp. Hồ Chí Minh, tháng 12 năm 2020**

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

----\*\*\*----

Tp. Hồ Chí Minh, ngày tháng 12 năm 2020

# **PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên: MSSV:

Ngành:

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

**NHẬN XÉT**

1. Về nội dung đề tài và khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm: (Bằng chữ: )

Tp. Hồ Chí Minh, ngày tháng năm 20

Giáo viên hướng dẫn

*(Ký và ghi rõ họ tên)*

**TS. Nguyễn Thành Sơn**

**LỜI CAM ĐOAN**

Tôi xin cam đoan:

1. Những nội dung trong báo cáo này là do tôi thực hiện dưới sự hướng dẫn trực tiếp của thầy Nguyễn Thành Sơn.

2. Mọi tham khảo dùng trong báo cáo đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố.

3. Mọi sao chép không hợp lệ, vi phạm quy chế đào tạo, hay gian trá, tôi xin chịu hoàn toàn trách nhiệm.

Hồ Chí Minh, ngày 23 tháng 12 năm 2020

Đào Văn Thắng Đặng Ngọc Sơn

Trần Minh Tú Trung Đào Minh Thy

**Mục lục**

[**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN 1**](#_Toc59752060)

[**CHƯƠNG 1. TỔNG QUAN CHUNG 8**](#_Toc59752061)

[**1.1 Giới Thiệu : 8**](#_Toc59752062)

[**1.2 Lý do chọn đề tài: 8**](#_Toc59752063)

[**1.3 Mô tả bài toán : 8**](#_Toc59752064)

[**1.4 Thông tin phần mềm 10**](#_Toc59752065)

[**1.5 Chức năng 10**](#_Toc59752066)

[***1.5.1 Quản lý Tài khoản* 10**](#_Toc59752067)

[***1.5.2 Quản lý Task* 10**](#_Toc59752068)

[***1.5.3 Quản lý Project* 10**](#_Toc59752069)

[***1.5.4 Quản lý Lương theo yêu cầu người quản lý* 10**](#_Toc59752070)

[***1.5.5 Quản lý Nhân viên* 10**](#_Toc59752071)

[***1.5.6 Thống kê Task và Project* 10**](#_Toc59752072)

[***1.5.7 Quản lý Team* 10**](#_Toc59752073)

[**1.6 Kế hoạch thực hiện 10**](#_Toc59752074)

[**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ 13**](#_Toc59752075)

[**2.1 Lược đồ ERD 13**](#_Toc59752076)

[**2.2 Lược đồ quan hệ 13**](#_Toc59752077)

[**2.3 Thiết kế Cơ sở dữ liệu 14**](#_Toc59752078)

[**2.4 Sơ đồ Diagram 18**](#_Toc59752079)

[**CHƯƠNG 3. XÂY DỰNG CƠ SỞ DỮ LIỆU 18**](#_Toc59752080)

[**3.1 Danh sách các Stored Procedure 18**](#_Toc59752081)

[**3.2 Danh sách các Function 19**](#_Toc59752082)

[**3.3 Danh sách các Trigger 20**](#_Toc59752083)

[**3.4 Danh sách các View 20**](#_Toc59752084)

[**CHƯƠNG 4. CÀI ĐẶT SQL 21**](#_Toc59752085)

[**4.1 Store Procedure 21**](#_Toc59752086)

[**4.1.1. SP\_storeUpdateInfoAccount 21**](#_Toc59752087)

[**4.1.2. SP\_addTask 21**](#_Toc59752088)

[**4.1.3. SP\_DeleteTask 22**](#_Toc59752089)

[**4.1.4. SP\_addAccount 22**](#_Toc59752090)

[**4.1.5. SP\_DeleteAccount 22**](#_Toc59752091)

[**4.1.6. SP\_addStaff 23**](#_Toc59752092)

[**4.1.7. SP\_DeleteStaff 23**](#_Toc59752093)

[**4.1.8. SP\_addProject 23**](#_Toc59752094)

[**4.1.9. SP\_DeleteProject 24**](#_Toc59752095)

[**4.1.10. SP\_addTeam 24**](#_Toc59752096)

[**4.1.11. SP\_DeleteTeam 24**](#_Toc59752097)

[**4.1.12. SP\_addSalaryAllYear 24**](#_Toc59752098)

[**4.1.13. SP\_storeStaffOfTeam 25**](#_Toc59752099)

[**4.1.14. SP\_storeSalaryTeam 26**](#_Toc59752100)

[**4.1.15. SP\_checkSalaryMonth 27**](#_Toc59752101)

[**4.1.16. SP\_checkSalaryYearStaff 28**](#_Toc59752102)

[**4.1.17. SP\_checkSalaryYearAdmin 29**](#_Toc59752103)

[**4.1.18. SP\_Login 30**](#_Toc59752104)

[**4.1.19. SP\_InforUser 30**](#_Toc59752105)

[**4.1.20. SP\_ListTaskRunning 30**](#_Toc59752106)

[**4.1.21. SP\_storeTaskName 30**](#_Toc59752107)

[**4.1.22. SP\_Account\_Create 32**](#_Toc59752108)

[**4.1.23. SP\_UpdateStaff 32**](#_Toc59752109)

[**4.1.24. SP\_Account\_Create 33**](#_Toc59752110)

[**4.2 Function 33**](#_Toc59752111)

[**4.2.1. AUTO\_IDSTAFF 33**](#_Toc59752112)

[**4.2.2. AUTO\_IDTEAM 33**](#_Toc59752113)

[**4.2.3. AUTO\_IDTASK 33**](#_Toc59752114)

[**4.2.4. AUTO\_IDPROJECT 34**](#_Toc59752115)

[**4.2.5. FN\_ConvertToUnsign1 34**](#_Toc59752116)

[**4.2.6. FN\_storeListProjectRunning 35**](#_Toc59752117)

[**4.2.7. FN\_storeListProjectTeam 35**](#_Toc59752118)

[**4.2.8. FN\_storeListSalary 35**](#_Toc59752119)

[**4.2.9. FN\_storeTop5\_StaffOfTeam 35**](#_Toc59752120)

[**4.2.10. FN\_storeSumSalaryTeam 35**](#_Toc59752121)

[**4.2.11. FN\_storeTop5\_SalaryOfCompany 36**](#_Toc59752122)

[**4.3 Trigger 36**](#_Toc59752123)

[**4.3.1. TG\_UpDateStaff 36**](#_Toc59752124)

[**4.3.2. TG\_UpdateBonusSalary 37**](#_Toc59752125)

[**4.3.3. TG\_UpdateBonusSalary\_Deleted 38**](#_Toc59752126)

[**4.3.4. TG\_UpdateStatusProject 39**](#_Toc59752127)

[**4.3.4. TG\_UpdateTimeProjectStaff\_Task 39**](#_Toc59752128)

[**4.3.5. TG\_InsertTask 40**](#_Toc59752129)

[**4.4 View 40**](#_Toc59752130)

[**4.4.1. View\_AdminTeam 40**](#_Toc59752131)

[**4.4.2. View\_TeamInfo 41**](#_Toc59752132)

[**4.4.3. View\_ProjectTeam 41**](#_Toc59752133)

[**4.4.4. View\_TaskTeam 41**](#_Toc59752134)

[**4.4.5. View\_Account 41**](#_Toc59752135)

[**4.4.6. View\_TypeStaff 41**](#_Toc59752136)

[**5.1 Phân quyền 42**](#_Toc59752137)

[***5.1.1 Phân quyền Admin* 42**](#_Toc59752138)

[***5.1.2Phân quyền LeaderTeam* 43**](#_Toc59752139)

[***5.1.3 Phân quyền Emp* 43**](#_Toc59752140)

[**CHƯƠNG 5. CÀI ĐẶT CHƯƠNG TRÌNH 45**](#_Toc59752141)

[**5.1 Giao diện trang chủ 45**](#_Toc59752142)

[**5.2 Giao diện trang trang đăng nhập 45**](#_Toc59752143)

[**5.4 Giao diện trang quản lý Project 46**](#_Toc59752144)

[**5.5 Giao diện trang quản lý Team 47**](#_Toc59752145)

[**5.6 Giao diện trang quản lý Task 47**](#_Toc59752146)

[**5.7 Giao diện trang quản lý Salary(Lương) 48**](#_Toc59752147)

[**5.8 Tổ chức các lớp trong chương trình 48**](#_Toc59752148)

[**CHƯƠNG 6. TỔNG KẾT 56**](#_Toc59752149)

[**6.1. bóng góp đề tài 57**](#_Toc59752150)

[**6.2 Các tính năng bổ sung, đặc biệt trong đề tài 57**](#_Toc59752151)

[**6.2.1 Tính Lương nhẩy tự động 57**](#_Toc59752152)

[**6.2.2 Sử dụng Cursor 59**](#_Toc59752153)

[**6.2.3 Thêm login kết nối server 60**](#_Toc59752154)

[**6.2.4 Xóa login kết nối server 60**](#_Toc59752155)

[**6.3 Các hạn chế 61**](#_Toc59752156)

[**6.4 Hướng phát triển 61**](#_Toc59752157)

# **CHƯƠNG 1. TỔNG QUAN CHUNG**

## **1.1 Giới Thiệu :**

Hiện nay, công nghệ thông tin được xem là một ngành mũi nhọn của quốc gia, đặc biệt là các nước đang phát triển, tiến hành công nghiệp hóa hiện đại hóa như nước ta. Nếu không có sự hỗ trợ của tin học, việc quản lý phải cần khá nhiều người, chia thành nhiều khâu mới có thể quản lý nhân viên công ty như: Quản lý nhân viên (nhân viên làm việc theo công việc hay làm dự án ), quản lý Dự án, Công việc… Các công việc này đòi hỏi nhiều thời gian và công sức, mà sự chính xác và hiệu quả không cao, nếu làm bằng thủ công không mang tính tự động. Một số nghiệp vụ như tra cứu, thống kê và hiệu chỉnh thông tin khá vất vả. Ngoài ra còn có một số khó khăn về việc lưu trữ khá đồ sộ, dễ bị thất lạc, tốn kém,…

## **1.2 Lý do chọn đề tài:**

Trong khi đó các nghiệp vụ này có thể tin học hóa một cách đơn giản hơn. Với sự giúp đỡ của tin học, việc quản lý sẽ trở nên thuận tiện, nhanh chóng và hiệu quả hơn rất nhiều. Đó là một hiện trạng cần được giải quyết. Tại các công ty hiện nay với lượng nhân viên ngày càng tăng, để quản lý nhân viên được tốt hơn, chính xác hơn và nhanh chóng hơn thì công ty đã tin hoc hóa các khâu quản lý. Đặt biệt là trong công tác thống kê và quản lýdự án, nhân sự, tình hình công việc . Bởi vì công tác thủ công mà quán đang thực hiện đã bộc lộ nhiều hạn chế:

## **1.3 Mô tả bài toán :**

- Khó khăn trong quá trình thực hiện báo cáo thống kê, nhất là khi có s ự cố đột xuất. Trước tình hình đó vấn đề đặt ra là cần xây dựng một hệ thống thông tin đáp ứng được nhu cầu cơ bản sau:

- Lưu trữ khối lượng lớn thông tin hợp lý giảm ghi chép lưu trữ file giấy.

- Cập nhật dữ liệu nhanh chóng:

+ Thêm thông tin nhân viên, công việc của từng nhân viên.

+ Xóa thông tin , công việc của nhân viên .

+ Sửa thông tin công việc của nhân viên.

+ Truy cập, truy xuất thông tin nhân viên.

- chích khấu thưởng lương theo công việc hoàn thành.

- Có khả năng lưu trữ thông tin lâu dài, đảm bảo tìm kiếm nhanh khi cần thiết.

- Tính tiền lương nhân viên.

- Quản lý dự án theo công việc, người quản lý dự án

- Quản lý có thể kiểm tra tình trạng công việc, nhân viên phụ trách công việc.

- Phân quyền .

- Báo cáo khối lượng công việc hoàn thành trong tháng.

Để sử dụng phần mềm, người dung cần có tài khoản để đăng nhập vào ứng dụng, Mỗi Account sẽ có tên đăng nhập, tên hiển thị và mật khẩu để đăng nhập vào ứng dụng Ngoài ra User được chia thành 2 loại quyền đó là Admin và Staff. Admin được phép truy cập đầy đủ toàn bộ các chức năng của phần mềm, được quyền chỉnh sửa và phần quyền sử dụng cho các nhân viên(Staff). Mỗi nhân viên gồm có tên, địa chỉ, email.

Các Staff được chia thành các team khác nhau (Team), mỗi team sẽ có một người quản lí. Quản lí là người đứng đầu một team, có trách nhiệm theo dõi các dự án ( Project ) và phân chia công việc trong mỗi dự án được giao ( Staff ). Các công việc gồm có tên công việc, nhân viên phụ trách công việc và tình trạng hoàn thành công việc. Mỗi dự án gồm có tên dự án, tên người quản lí dự án và tình trạng hoàn thành dự án, các dự án được đánh dấu “Đã hoàn thành” khi toàn bộ các công việc có trong dự án “Đã hoàn thành”. Nếu đã thực hiện nhiều hơn 1 công việc trong dự án, tình trạng sẽ là “Đang thực hiện”. Nếu không hoàn thành bất kì công việc nào trong dự án, tình trạng sẽ là “Chưa hoàn thành”. Mỗi dự án đều có deadline để có thể xét thưởng phạt cho nhân viên.

Về lương ( Salary ) được tính theo lương cứng của mỗi nhân viên, nếu nhân viên hoàn thành các công việc được giao sẽ được cộng thêm số tiền ( 1% của tiền của tổng dự án ).

## **1.4 Thông tin phần mềm**

* Tên phần mềm: COMPANY HELMET TRUNGLEO’S
* Ngôn ngữ: JAVA , JAVACRIFT, HTML
* Môi trường cài đặt: Intel IJ , Sử dụng SQL Server.

## **1.5 Chức năng**

### ***1.5.1 Quản lý Tài khoản***

* Thêm, sửa, cập nhật thông tin tài khoản người dùng.

### ***1.5.2 Quản lý Task***

* Thêm, sửa, cập nhật thông tin Task.

### ***1.5.3 Quản lý Project***

* Theo dõi, thêm, sửa, cập nhật Project.

### ***1.5.4 Quản lý Lương theo yêu cầu người quản lý***

* Quản lý Lương theo tháng, năm.
* Tính lương thưởng theo dự án hang tháng, Task hằng tháng
* Tính lương thương tháng 13 cho Admin ,Staff

### ***1.5.5 Quản lý Nhân viên***

* Theo dõi, thêm, sửa, cập nhật thông tin Nhân viên.

### ***1.5.6 Thống kê Task và Project***

* Thống kê Công việc theo từng nhân viên, Team,theo thời gian và theo dự án.
* Thống kê Dự án theo Team, Theo thời gian , theo số lượng công việc.

### ***1.5.7 Quản lý Team***

* Thêm, sửa, cập nhật thông tin Team

## **1.6 Kế hoạch thực hiện**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TUẦN** | **CÔNG VIỆC THỰC HIỆN** | **NGÀY BD** | **NGÀY KT** | **KẾT QUẢ** |
| 1 | Viết mô tả về đề tài (nêu rõ các thành phần dữ liệu, chức năng và giao diện cần có để project chạy tốt) | 5/10/2020 | 12/10/2020 | Hoàn thành |
| 2 | Thiết kế CSDL và các ràng buộc (dùng ERD) | 12/20/2020 | 19/10/2020 | Hoàn thành |
| 3 | Cài đặt CSDL và các ràng buộc, trigger.  Nhập dữ liệu | 19/10/2020 | 26/10/2020 | Hoàn thành |
| 4 | Thiết kế các view (trình bày bằng mã giả) | 26/10/2020 | 2/11/2020 | Hoàn thành |
| 5 | Cài đặt các view | 2/11/2020 | 9/11/2020 |  |
| 6 | Thiết kế giao diện, xác định các users và quyền sử dụng CSDL của họ | 9/11/2020 | 16/11/2020 | Hoàn thành |
| 7 | Cài đặt giao diện và kết nối chương trình với CSDL, tạo user, phần quyền | 16/11/2020 | 23/11/2020 | Hoàn thành |
| 8 | Thiết kế các hàm, thủ tục thực hiện các chức năng của đề tài | 23/11/2020 | 30/11/2020 | Hoàn thành |
| 9 |  | 30/11/2020 | 07/12/2020 |  |
| 10 | Cài đặt các hàm, thủ tục thực hiện các chức năng của đề tài | 07/12/2020 | 14/12/2020 | Hoàn thành |
| 11 | Viết báo cáo | 14/12/2020 | 23/12/2020 | Hoàn thành |

# **CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ**

## **2.1 Lược đồ ERD**

Hình 1 Lược đồ ERD

## **2.2 Lược đồ quan hệ**

**ACCOUNT** (Username, Displayname, Password, Type)

**STAFF** (Id, Name,DateOfBirth, Address, Phone, Email, WorkTime, IdTeam, UserName,AboutStaff , UserImage , LinkFacebook , status)

**TEAM** (Id, Name, Address, Phone, Email, IdAdmin, status)

**TASK** (Id, Name, Content, StatusTask, TimeFrom , TimeEnd, idStaff, status, IdProject)

**PROJECT** (Id, Name, Content, StatusProject, TimeFrom , TimeEnd, idTeam, status)

**SALARY** ( Month,SYear, IdStaff, Salary, Bonus)

## **2.3 Thiết kế Cơ sở dữ liệu**

* Bảng ACCOUNT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ACCOUNT** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | UserName | nvarchar(100) | PRIMARY KEY | Tên tài khoản ,Khóa chính phân biệt các tài khoản. |
| 2 | DisplayName | nvarchar(50) | NOT NULL | Tên hiện thị của tài khoản |
| 3 | PassWord | nvarchar(30) | NOT NULL | Mật khẩu tài khoản |
| 4 | Type | nvarchar(10) | NOT NULL | Loại Nhân viên |

* Bảng STAFF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STAFF** | | | | | |
| **STT** | **Thuộc tính** | | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | Id | | Int | PRIMARY KEY | Khóa chính phân biệt các Nhân viên. |
| 2 | Name | | NVARCHAR(100) | NOT NULL | Tên nhân viên |
| 3 | DateOfBirth | | DATE | NOT NULL | Ngày sinh của nhân viên |
| 4 | Address | | NVARCHAR(100) | NOT NULL | Địa chỉ của nhân viên |
| 5 | Sex | | nvarchar(10) |  | Giới tính của nhân viên |
| 6 | | Email | nvarchar(100) | NOT NULL,  UNIQUE | Email của nhân viên |
| 7 | Phone | | nvarchar(10) | NOT NULL,  UNIQUE | Số điện thoại của nhân viên |
| 8 | WorkTime | | float |  | Thời gian làm việc hằng tháng |
| 9 |  | | int | NOT NULL, FOREIGN KEY | Id team mà nhân viên đó làm việc |
| 10 | UserName | | Nvarchar(100) | NOT NULL,  UNIQUE | Tên tài khoản dung để đăng nhập |
| 11 | AboutStaff | | Nvarchar(500) |  | Giới thiệu về nhân viên |
| 12 | UserImage | | Nvarchar(200) |  | Đường link ảnh đại diện |
| 13 | LinkFacebook | | Nvarchar(200) |  | Đường link truy cập trang cá nhân facebook |
| 14 | status | | int | NOT NULL | 0 – không còn làm việc  1 – còn đang làm việc |

* Bảng TEAM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEAM** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | id | int | PRIMARY KEY | Khóa chính phân biệt các loại Team |
| 2 | Name | nvarchar(100) | NOT NULL | Tên Team |
| 3 | Address | nvarchar(200) | NOT NULL | Địa chỉ của Team |
| 4 | Phone | nvarchar(10) | NOT NULL | Số điện thoại của Team |
| 5 | Email | nvarchar(100) | NOT NULL | Email của Team |
| 6 | IdAdmin | int | FOREIGN KEY, NOT NULL | Id nhân viên là người quản lý |
| 7 | status | int | NOT NULL | 0 – Không hoạt động  1 – còn hoạt động |

* Bảng TASK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | Id | int | PRIMARY KEY | Khóa chính phân biệt các Task |
| 2 | Name | NVARCHAR(100) | NOT NULL | Tên của Task |
| 3 | Content | NVARCHAR(500) | NOT NULL | Nội dung của Task |
| 4 | StatusTask | int | NOT NULL | 0-chưa hoàn thành  1-Đã hoàn thành |
| 5 | TimeFrom | DateTime | NOT NULL | Thời gian bắt đầu Task |
| 6 | TimeEnd | DateTime | NOT NULL | Thời gian kết thúc Task |
| 7 | idStaff | int | FOREIGN KEY, NOT NULL | Id nhân viên thực hiện Task |
| 8 | status | int | NOT NULL | 0-Task đã bị dừng  1-Task tồn tại |
| 9 | IdProject | int | FOREIGN KEY, NOT NULL | Id Project mà Task thuộc về |

* Bảng PROJECT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PROJECT** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | Id | nchar(10) | PRIMARY KEY | Khóa chính phân biệt các tỉnh, thành phố |
| 2 | Name | NVARCHAR(500) | NOT NULL | Tên tỉnh, thành phố |
| 3 | Content | nvarchar(MAX) | NOT NULL | Nội dung của Project |
| 4 | StatusProject | int | NOT NULL | 0-Project chưa hoàn thành  1-Project hoàn thành |
| 5 | TimeFrom | DateTime | NOT NULL | Thời gian bắt đầu |
| 6 | TimeEnd | DateTime | NOT NULL | Thời gian kết thúc |
|  | idTeam | int | FOREIGN KEY, NOT NULL | Id Team thực hiện Project |
| 7 | status | int | NOT NULL | 0-Project đã bị dừng  1-Project tồn tại |

* Bảng SALARY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SALARY** | | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Ràng buộc** | **Diễn giải** |
| 1 | Month | int | NOT NULL | Khóa chính phân biệt các huyện, quận |
| 2 | SYear | int | NOT NULL | Tên huyện, quận |
| 3 | IdStaff | int | FOREIGN KEY, NOT NULL | Id nhân viên hưởng lương |
| 4 | Salary | real | NOT NULL | Lương theo tháng |
| 5 | Bonus | real | NOT NULL | Thưởng theo tháng |

## **2.4 Diagram**

****

*Hình SEQ Hình \\* ARABIC 2. Diagram*

# **CHƯƠNG 3. XÂY DỰNG CƠ SỞ DỮ LIỆU**

## **3.1 Danh sách các Stored Procedure**

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | SP\_storeUpdateInfoAccount | Thay đổi mật khẩu tài khoản |
| 2 | SP\_addTask | Kiểm tra và thêm Task |
| 3 | SP\_DeleteTask | Kiểm tra và Xóa Task |
| 4 | SP\_addAccount | Kiểm tra và thêm Account |
| 5 | SP\_DeleteAccount | Kiểm tra và xóa Account |
| 6 | SP\_addStaff | Thêm nhân viên vào bảng STAFF |
| 7 | SP\_DeleteStaff | Kiểm tra và xóa nhân viên khỏi bảng STAFF |
| 8 | SP\_addProject | Kiểm tra và them Project |
| 9 | SP\_DeleteProject | Kiểm tra và xóa Project |
| 10 | SP\_addTeam | Thêm 1 team trong công ty |
| 11 | SP\_DeleteTeam | Xóa 1 team trong công ty |
| 12 | SP\_addSalaryAllYear | Them bảng lương cả năm cho tất cả nhân viên trong công ty |
| 13 | SP\_storeStaffOfTeam | Lấy ra nhân viên trong 1 Team |
| 14 | SP\_storeSalaryTeam | Lấy ra lương của nhân viên trong team |
| 15 | SP\_checkSalaryMonth | Tự động tính lương tháng của tất cả nhân viên trong công ty |
| 16 | SP\_checkSalaryYearStaff | Tính lương thưởng năm cho nhân viên |
| 17 | SP\_checkSalaryYearAdmin | Tính lương thưởng năm cho Admin |
| 18 | SP\_Login | Kiểm tra đăng nhập |
| 19 | SP\_InforUser | Thông tin tài khoản |
| 20 | SP\_ListTaskRunning | Lấy ra danh sách Task đang chạy |
| 21 | SP\_storeTaskName | Tìm kiếm Task bằng Tên , Id của Project và Staff |
| 22 | SP\_Account\_Create | Tạo tài khoản sử dụng cả phân quyền DB |
| 23 | SP\_UpdateStaff | Cập nhật lại thông tin Nhân viên |

## **3.2 Danh sách các Function**

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | AUTO\_IDSTAFF | Tự động tính Id tiếp theo của Staff |
| 2 | AUTO\_IDTEAM | Tự động tính Id tiếp theo của Team |
| 3 | AUTO\_IDTASK | Tự động tính Id tiếp theo của Task |
| 4 | AUTO\_IDPROJECT | Tự động tính Id tiếp theo của Project |
| 5 | FN\_ConvertToUnsign1 | Tự động chuyển lỗi chữ về chữ ko dấu |
| 6 | FN\_storeListProjectRunning | Lấy ra danh sách Project đang chạy |
| 7 | FN\_storeListProjectTeam | Lấy ra ds Project theo Team |
| 8 | FN\_storeListSalary | Lấy ra bảng lương |
| 9 | FN\_storeTop5\_StaffOfTeam | Tính ra 5 nhân viên chăm chỉ của Team |
| 10 | FN\_storeSumSalaryTeam | Tính tổng lương nhân viên của từng Team |
| 11 | FN\_storeTop5\_SalaryOfCompany | Tính ra 5 người có số lương cao nhất công ty |

*Table 3. Danh sách Function*

## **3.3 Danh sách các Trigger**

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | TG\_UpDateStaff | Bắt lỗi và update lại Staff khi Update |
| 2 | TG\_UpdateBonusSalary | Tự động tính lại lương thưởng cho nhân viên |
| 3 | TG\_UpdateBonusSalary\_Deleted | Tự động tính lại lương thưởng cho nhân viên |
| 4 | TG\_UpdateStatusProject | Kiểm tra lại Project và set lại cho Task |
| 5 | TG\_UpdateTimeProjectStaff\_Task | Kiểm tra Task và update khi cập nhật Time |
| 6 | TG\_InsertTask | Kiểm tra lại khi Insert Task |

*Table 4. Danh sách Trigger*

## **3.4 Danh sách các View**

|  |  |  |
| --- | --- | --- |
| STT | TÊN | MÔ TẢ |
| 1 | View\_AdminTeam | Xem thông tin Admin của từng Team |
| 2 | View\_TeamInfo | Xem thông tin Info Team |
| 3 | View\_ProjectTeam | Xem danh sách project của team |
| 4 | View\_TaskTeam | Xem danh sách task của team |
| 5 | View\_Account | Xem thông tin các Account |
| 6 | View\_TypeStaff | Xem loại nhân viên |

*Table 5. Danh sách View*

# **CHƯƠNG 4. CÀI ĐẶT SQL**

## **4.1 Store Procedure**

### **4.1.1.** SP\_storeUpdateInfoAccount

- Thay đổi mật khẩu tài khoản: được thực thi mật khẩu nhập vào không được rỗng, mật khẩu mới không được trùng với mật khẩu cũ.

|  |
| --- |
| ALTER PROC [dbo].[SP\_storeUpdateInfoAccount](@userName Nvarchar(20), @passWord\_new Nvarchar(20), @passWord\_old Nvarchar(20))  as  begin  declare @passWord Nvarchar(20)  declare @changeLoginPassword nvarchar(max)  select @passWord = PassWord  from ACCOUNT  where UserName = @userName  if (@passWord = HASHBYTES('MD5', @passWord\_old))  begin  update ACCOUNT  set  PassWord = HASHBYTES('MD5', @passWord\_new)  where  UserName = @userName  end  else  raiserror(N'Wrong password !!!', 16, 1)  end |

### **4.1.2.** SP\_addTask

- Lưu thông tin sản phẩm trong giỏ hàng: kiểm tra nếu giỏ hàng đã có sản phẩm đó thì tăng số lượng Count, ngược lại, nếu chưa có thì thêm sản phẩm mới.

|  |
| --- |
| ALTER PROC SP\_addTask (@Name NVARCHAR(100),  @Content NVARCHAR(500),@IdProject int,  @IdStaff int,@timeFrom DateTime,  @timeEnd DateTime)  as  begin    Declare @TimeEndPro DateTime  Declare @TimeFromPro DateTime  Declare @StatusProject int  Declare @CheckStaff int  Declare @CheckProject int  -- check Project của Task  Select @TimeEndPro = TimeEnd,  @TimeFromPro =TimeFrom ,  @StatusProject = StatusProject,  @checkProject =status  FROM PROJECT WHERE Id = @IdProject    -- check Staff của TASK  Select @CheckStaff = status  from STAFF  Where @IdStaff= STAFF.Id  IF(@timeFrom<@timeEnd and @timeFrom >= @TimeFromPro  and @timeEnd <= @TimeEndPro  and @StatusProject =1 and @CheckStaff =1)  begin  Insert TASK(Name,Content,IdProject,  IdStaff,TimeFrom,  TimeEnd,status)  Values(@Name,@Content,@IdProject,  @IdStaff,@timeFrom,@timeEnd,  @checkProject)  end  else  raiserror(N'Wrong Task !!!',12,1)  end  go |

### **4.1.3.** SP\_DeleteTask

- Thay đổi thông tin của tài khoản: tài khoản chỉ được thay đổi thông tin khi nhập đúng username và password.

|  |
| --- |
| ALTER PROC SP\_DeleteTask(@Id int)  as  Begin  Update TASK set status = 0 where Task.Id=@Id  End |

### **4.1.4.** SP\_addAccount

|  |
| --- |
| CREATE PROC SP\_addAccount(@DisplayName NVARCHAR(50),@UserName NVARCHAR(50),@Password NVARCHAR(25),@Type NVARCHAR(15))  as  begin  INSERT ACCOUNT(UserName,DisplayName,PassWord,Type) VALUES (@UserName,@DisplayName,HASHBYTES ('MD5', @Password),@Type)  end |

### 4.1.5. SP\_DeleteAccount

|  |
| --- |
| ALTER PROC SP\_DeleteAccount(@UserName NVARCHAR(50),@status int)  AS  BEGIN  Declare @type\_old NVARCHAR(15),@CK\_Staff int  SELECT @CK\_Staff = COUNT(Id) FROM STAFF WHERE STAFF.UserName = @UserName AND status =1  if(@status = 0 AND @CK\_Staff =0)  Update ACCOUNT set status = 0 where @UserName =UserName  Select Type From ACCOUNT where @UserName =UserName  END |

### **4.1.6. SP\_addStaff**

|  |
| --- |
| ALTER PROC SP\_addStaff(@Name NVARCHAR(100),@DateOfBirth Date, @Address NVARCHAR(100),@Sex NVARCHAR(10),  @Phone NVARCHAR(10),@Email NVARCHAR(50),@IdTeam int ,@UserName NVARCHAR(50))  AS  BEGIN TRANSACTION  DECLARE @StatusTeam int,  @StatusAccount int,  @UserNameAdd NVARCHAR(100),  @Password NVARCHAR(25),  @Type NVARCHAR(15)  SELECT @StatusTeam = status FROM TEAM WHERE TEAM.Id = @IdTeam AND status =1  SELECT @StatusAccount = status FROM ACCOUNT WHERE ACCOUNT.UserName=@UserName AND status =1  IF(@StatusAccount = null)  BEGIN  SELECT @UserNameAdd = REPLACE(@Name, ' ', ''),@Password =N'123',@Type=N'Emp'  EXEC SP\_addAccount @Name,@UserNameAdd,@Password,@Type  SELECT @UserName = @UserNameAdd  END  IF(@StatusTeam=1 and @StatusAccount =1)  INSERT INTO dbo.STAFF(Name,DateOfBirth,Address,Sex,Phone,Email,IdTeam,UserName)  VALUES (@Name,@DateOfBirth, @Address,@Sex,@Phone,@Email,@IdTeam,@UserName)  COMMIT TRANSACTION |

### **4.1.7. SP\_DeleteStaff**

|  |
| --- |
| CREATE PROC SP\_DeleteStaff(@Id INT,@status INT)  AS  BEGIN TRANSACTION  DECLARE @CK\_TeamAdmin INT, @CK\_Task INT  SELECT @CK\_TeamAdmin = COUNT(\*) FROM TEAM WHERE IdAmin=@Id  AND status =1  SELECT @CK\_Task = COUNT(\*) FROM TASK WHERE IdStaff = @Id AND status =1  IF(@CK\_Task = 0 AND @CK\_TeamAdmin=0)  BEGIN  UPDATE STAFF SET status = 0 WHERE Id = @Id  ENDCOMMIT TRANSACTION  **GO** |

### **4.1.8. SP\_addProject**

|  |
| --- |
| CREATE PROC SP\_addProject(@Name NVARCHAR(100),@Content NVARCHAR(500),@IdTeam int,@timeFrom DateTime,@timeEnd DateTime)  AS  BEGIN TRANSACTION  DECLARE @StatusTeam int  SELECT @StatusTeam = status FROM TEAM WHERE TEAM.Id = @IdTeam AND status=1  IF(@StatusTeam !=0 )  BEGIN  INSERT INTO dbo.PROJECT(Name,Content,IdTeam,TimeFrom,TimeEnd) VALUES (@Name,@Content,@IdTeam,@timeFrom,@timeEnd)  END  COMMIT TRANSACTION  **GO** |

### **4.1.9. SP\_DeleteProject**

|  |
| --- |
| CREATE PROC SP\_DeleteProject (@Id int,@Status int)  AS  BEGIN TRANSACTION  DECLARE @CK\_TASK INT  SELECT @CK\_TASK = COUNT(\*) FROM TASK WHERE TASK.IdProject = @Id AND status =1    IF(@CK\_TASK = 0)  UPDATE PROJECT SET status = 0 WHERE Id =@IdCOMMIT TRANSACTION  **GO** |

### **4.1.10. SP\_addTeam**

|  |
| --- |
| CREATE PROC SP\_addTeam(@Name NVARCHAR(100),@Address NVARCHAR(100),@Phone NVARCHAR(10),@Email NVARCHAR(100),@IdAdmin INT)  AS  BEGIN TRANSACTION  DECLARE @CK\_IdAdmin INT  SELECT @CK\_IdAdmin = COUNT(Id) FROM STAFF WHERE IdTeam = @IdAdmin AND status =1  IF(@CK\_IdAdmin=0)  BEGIN  INSERT INTO TEAM (Name,Address,Phone,Email,IdAmin) VALUES (@Name,@Address,@Phone,@Email,@IdAdmin)  END  COMMIT TRANSACTION  **GO** |

### **4.1.11. SP\_DeleteTeam**

|  |
| --- |
| ALTER PROC SP\_DeleteTeam(@Id int, @status int)  AS  BEGIN  DECLARE @CK\_IdStaff int  SELECT @CK\_IdStaff = COUNT(Id) FROM STAFF WHERE STAFF.IdTeam = @Id AND status =1  IF(@CK\_IdStaff >0)  BEGIN  UPDATE STAFF SET status = 0 WHERE Id =@Id  END  END |

### **4.1.12.** SP\_addSalaryAllYear

|  |
| --- |
| ALTER PROC SP\_addSalaryAllYear(@year int)  AS    BEGIN  DECLARE @CK\_Salary int, @Month int,@Salary Real,@Id int,@Type NVARCHAR(20)    DECLARE cursor\_addSalaryAllYear CURSOR FOR  SELECT STAFF.Id,Type  FROM STAFF,ACCOUNT  where STAFF.UserName=ACCOUNT.UserName AND STAFF.status =1  OPEN cursor\_addSalaryAllYear  FETCH NEXT FROM cursor\_addSalaryAllYear INTO @Id,@Type    WHILE @@FETCH\_STATUS = 0  BEGIN  IF(@Type = N'Admin')  SELECT @Salary = 9700  ELSE IF (@Type = N'LeaderTeam')  SELECT @Salary = 8500  ELSE IF(@Type = N'Emp')  SELECT @Salary = 7500  SELECT @Month = 1  While @Month <= 13  BEGIN  SELECT @CK\_Salary = COUNT(IdStaff) FROM SALARY,STAFF WHERE STAFF.Id =IdStaff and SALARY.IdStaff =@Id and STAFF.status = 1 and Month =@Month and SYear =@year  IF(@CK\_Salary =0)  BEGIN  INSERT INTO SALARY (Month, SYear,IdStaff,Salary,Bonus) VALUES (@Month,@year,@Id,@Salary,0)  END  SELECT @Month = @month +1  END    FETCH NEXT FROM cursor\_addSalaryAllYear INTO @Id,@Type  END  CLOSE cursor\_addSalaryAllYear  DEALLOCATE cursor\_addSalaryAllYear  END |

### 4.1.13. SP\_storeStaffOfTeam

|  |
| --- |
| ALTER PROC [dbo].[SP\_storeStaffOfTeam](@IdTeam int)  AS  BEGIN    If(@IdTeam =0)    Select STAFF.Id ,STAFF.Name,  STAFF.Phone, STAFF.Address,  STAFF.Sex,Count(STAFF.Id) As N'NumTask',  ACCOUNT.Type  from STAFF INNER JOIN TASK on STAFF.Id = TASK.IdStaff  INNER JOIN TEAM on TEAM.Id =STAFF.IdTeam  INNER JOIN ACCOUNT on STAFF.UserName = ACCOUNT.UserName  where TASK.status != 0 and 0 <= (Select Count(STAFF.Id)  FROM STAFF,TASK  WHERE STAFF.Id = TASK.IdStaff  and STAFF.Id !=TASK.IdStaff  and STAFF.IdTeam =TASK.Id )  Group By STAFF.Id,STAFF.Name,STAFF.Phone,  STAFF.Name,STAFF.Address,STAFF.Sex,ACCOUNT.Type  If(@IdTeam != 0)  Select STAFF.Id ,STAFF.Name,STAFF.Phone, STAFF.Address,  STAFF.Sex,Count(STAFF.Id) As N'NumTask' ,ACCOUNT.Type  from STAFF INNER JOIN TASK on STAFF.Id = TASK.IdStaff  INNER JOIN TEAM on STAFF.IdTeam = @IdTeam  and TEAM.Id =STAFF.IdTeam  INNER JOIN ACCOUNT on STAFF.UserName =ACCOUNT.UserName  where TASK.status != 0 and 0 <= (Select Count(STAFF.Id)  FROM STAFF,TASK  WHERE STAFF.Id = TASK.IdStaff  and STAFF.IdTeam =TASK.Id)  Group By STAFF.Id,STAFF.Name,STAFF.Phone,  STAFF.Name,STAFF.Address,STAFF.Sex,ACCOUNT.Type  OLD  END |

### **4.1.14. SP\_storeSalaryTeam**

|  |
| --- |
| ALTER PROC SP\_storeSalaryTeam(@IdTeam int,@month int,@year int,@choose int)  as  BEGIN  IF(@IdTeam = 0 and @month <>0 and @choose=1)  Begin  SELECT Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND Month = @month and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary ASC  end  else IF(@IdTeam <> 0 and @month = 0 and @choose=1)  Begin  SELECT Month ,SYear , STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND STAFF.IdTeam =@IdTeam and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary ASC  end  else IF(@IdTeam = 0 and @month = 0 and @choose=1)  Begin  SELECT Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary ASC  end  else IF(@IdTeam = 0 and @month = 0 and @choose=0)  Begin  SELECT Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary DESC  end  else IF(@IdTeam <> 0 and @month <> 0 and @choose=1)  begin  SELECT Month ,SYear , STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND STAFF.IdTeam =@IdTeam AND Month = @month and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary ASC  end  else IF(@IdTeam = 0 and @month <>0 and @choose=0)  Begin  SELECT Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND Month = @month and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary DESC  end  else IF(@IdTeam <> 0 and @month = 0 and @choose = 0)  Begin  SELECT Month ,SYear , STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND STAFF.IdTeam =@IdTeam and SYear = @year and STAFF.status =1  ORDER BY FixedSalary DESC  end    else IF(@IdTeam <> 0 and @month <> 0 and @choose = 0)  begin  SELECT Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND STAFF.IdTeam =@IdTeam AND Month = @month and SYear = @year and STAFF.status =1  Group by Month,SYear,STAFF.Name,Salary,Bonus,TEAM.Name  ORDER BY FixedSalary DESC  end  END |

### **4.1.15.** SP\_checkSalaryMonth

|  |
| --- |
| ALTER PROC SP\_checkSalaryMonth( @month int, @year int)  AS  SET NOCOUNT ON  BEGIN    DECLARE @count int  DECLARE @IdStaff int, @salary real ,@bonus real  ---lay danh sach Salary dua vao con tro cursor\_date  DECLARE cursor\_SALARY\_MONTH CURSOR FOR  SELECT Month , SYear,IdStaff, Salary ,Bonus  FROM SALARY,STAFF  where Month = @month and SYear = @year AND IdStaff=Id  AND STAFF.status=1  OPEN cursor\_SALARY\_MONTH  FETCH NEXT FROM cursor\_SALARY\_MONTH INTO @month,@year,@IdStaff,@salary,@bonus  WHILE @@FETCH\_STATUS = 0  BEGIN    Select @count = Count(\*)  from TASK,STAFF  where TASK.IdStaff = STAFF.Id and STAFF.Id = @IdStaff  and MONTH(TimeEnd) >= @month  and MONTH(TimeFrom) <= @month  and TASK.status = 1  and YEAR(TimeEnd) >= @year  and YEAR(TimeFrom) <= @year  IF(@count > 0)  Begin  update SALARY set Bonus = [Salary\*0.01\*@count](mailto:Salary*0.01*@count)  WHERE IdStaff=@IdStaff AND Month= @month AND SYear = @year  end  else if (@count = 0)  update SALARY set Bonus = Salary\*(-0.005) WHERE IdStaff=@IdStaff AND Month= @month AND SYear = @year  FETCH NEXT FROM cursor\_SALARY\_MONTH INTO @month,@year,@IdStaff,@salary,@bonus  END  CLOSE cursor\_SALARY\_MONTH  DEALLOCATE cursor\_SALARY\_MONTH  END**GO** |

### **4.1.16.** SP\_checkSalaryYearStaff

|  |
| --- |
| ALTER PROC SP\_checkSalaryYearStaff( @year int)  as  Begin    declare @SumSalary real,@SumBonus real,  @CountKPIProject int , @IdStaff int,  @salary real,@bonus real    DECLARE cursor\_SALARY\_Year CURSOR FOR  SELECT SYear,IdStaff, Salary ,Bonus  FROM SALARY,STAFF  where SYear = @year and Month = 13  AND IdStaff=Id AND STAFF.status=1  OPEN cursor\_SALARY\_Year  FETCH NEXT FROM cursor\_SALARY\_Year INTO @year,@IdStaff,@salary,@bonus  WHILE @@FETCH\_STATUS = 0  BEGIN  Select @CountKPIProject = Count(TASK.Id)  from TASK inner join PROJECT on PROJECT.Id = TASK.IdProject and YEAR(PROJECT.TimeEnd) = @year  where TASK.IdStaff = @IdStaff  and PROJECT.StatusProject = 1    Select @SumSalary= Sum(Salary) , @SumBonus= Sum(Bonus) From Salary  Where IdStaff = @IdStaff And SYear = @year AND Month<>13    IF(@CountKPIProject >= 5)  Begin  update SALARY set Salary = @CountKPIProject\*0.2\*@SumBonus  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  end  else if (@CountKPIProject >= 10)  update SALARY set Salary = @CountKPIProject\*0.5\*@SumBonus  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  else if (@CountKPIProject <5 )  update SALARY set Salary = 0  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  FETCH NEXT FROM cursor\_SALARY\_Year INTO @year,@IdStaff,@salary,@bonus  END  CLOSE cursor\_SALARY\_Year  DEALLOCATE cursor\_SALARY\_Year  end |

### 4.1.17. SP\_checkSalaryYearAdmin

|  |
| --- |
| ALTER PROC SP\_checkSalaryYearAdmin( @year int)  as  Begin  declare @count int,@IdStaff int,@salary real,@bonus real    DECLARE cursor\_SALARY\_YearAdmin CURSOR FOR  SELECT SYear,IdStaff, Salary ,Bonus  FROM SALARY,TEAM  where SYear = @year and Month = 13 AND IdStaff = TEAM.IdAmin  OPEN cursor\_SALARY\_YearAdmin  FETCH NEXT FROM cursor\_SALARY\_YearAdmin INTO @year,@IdStaff,@salary,@bonus  WHILE @@FETCH\_STATUS = 0  BEGIN  Select @count = Count(PROJECT.Id)  from PROJECT,TEAM  where PROJECT.IdTeam = TEAM.Id and TEAM.IdAmin = @IdStaff  and YEAR(TimeEnd) = @year AND StatusProject = 1    IF(@count > 0)  Begin  update SALARY set Bonus = @count\*2000 where IdStaff=@IdStaff AND Month= 13 And SYear = @year  end  else if (@count =0)  update SALARY set Bonus = 0 where IdStaff=@IdStaff AND Month= 13 And SYear = @year  FETCH NEXT FROM cursor\_SALARY\_YearAdmin INTO @year,@IdStaff,@salary,@bonus  END  CLOSE cursor\_SALARY\_YearAdmin  DEALLOCATE cursor\_SALARY\_YearAdmin    end |

## **4.1.18.** SP\_Login

|  |
| --- |
| CREATE proc dbo.SP\_Login(@username Nvarchar(50) , @password Nvarchar(20))  as  begin  select \*  from STAFF  where UserName = @username and PassWord = HASHBYTES('MD5', @password)  end |

### **4.1.19.** SP\_InforUser

|  |
| --- |
| CREATE PROC SP\_InforUser  @username NVARCHAR(100)  AS  BEGIN  SELECT \* FROM dbo.STAFF WHERE UserName = @username  END |

### **4.1.20.** SP\_ListTaskRunning

|  |
| --- |
| ALTER PROC SP\_ListTaskRunning  as  BEGIN  SELECT \* FROM TASK Where TimeEnd > GETDATE()AND status = 1  GROUP by Id,Name,Content,StatusTask,TimeFrom,TimeEnd,IdStaff,IdProject,status  END |

### **4.1.21.** SP\_storeTaskName

|  |
| --- |
| ALTER PROC SP\_storeTaskName(@CHAR nvarchar(50),@IdProject int,@IdStaff int)  AS  begin TRANSACTION  if(@IdProject = 0 and @IdStaff = 0 and @CHAR != N'')  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE CHARINDEX(@CHAR , TASK.Name)>0 AND TASK.IdStaff= STAFF.Id and TASK.IdProject = PROJECT.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end  else if(@IdProject = 0 and @IdStaff != 0 and @CHAR != N'')  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE CHARINDEX(@CHAR , TASK.Name)>0 AND TASK.IdStaff= @IdStaff and TASK.IdProject = PROJECT.Id AND TASK.IdStaff= STAFF.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end  else if(@IdStaff = 0 and @IdProject!= 0 and @CHAR != N'')  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE CHARINDEX(@CHAR , TASK.Name)>0 AND TASK.IdStaff= STAFF.Id and TASK.IdProject = @IdProject and TASK.IdProject = PROJECT.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end  else if(@IdProject = 0 and @IdStaff != 0 and @CHAR = N'')  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE TASK.IdStaff= @IdStaff and TASK.IdProject = PROJECT.Id AND TASK.IdStaff= STAFF.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end  else if (@IdStaff = 0 and @IdProject != 0 and @CHAR = N'')  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE TASK.IdStaff= STAFF.Id and TASK.IdProject = @IdProject and TASK.IdProject = PROJECT.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end  else --if(@CHAR = null and @IdProject != 0 and @IdStaff != 0 )  begin  SELECT TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name as N'nameStaff', PROJECT.Name AS N'nameProject'  FROM TASK, STAFF,PROJECT  WHERE TASK.IdStaff= @IdStaff and TASK.IdProject = @IdProject AND TASK.IdStaff= STAFF.Id and TASK.IdProject = PROJECT.Id  Group By TASK.Id,TASK.Name,TASK.Content,TASK.TimeFrom,TASK.TimeEnd,TASK.StatusTask,TASK.status,STAFF.Name , PROJECT.Name  end    COMMIT TRANSACTION |

### **4.1.22.** SP\_Account\_Create

|  |
| --- |
| Create proc SP\_Account\_Create(@username nvarchar(20),@password nvarchar(max),@namerole nvarchar(50)) as  BEGIN  DECLARE @createUser nvarchar(max)  DECLARE @createLogin nvarchar(max)  DECLARE @addLeaderRole nvarchar(max)  DECLARE @addEmpRole nvarchar(max)  DECLARE @addAdminRole nvarchar(max)  DECLARE @error int  SET @error=0  SET @createLogin='create login '+ @username + ' with password = ' + Char(39) + @password + Char(39)  SET @createUser='create user '+ @username + ' for login '+ @username  SET @addLeaderRole='Sp\_addRoleMember' + Char(39) +'projectsRole' + Char(39) + ',' + Char(39) + @username + Char(39)  SET @addAdminRole='Sp\_addRoleMember' + Char(39) +'db\_owner' + Char(39) + ',' + Char(39) + @username + Char(39)  SET @addEmpRole='Sp\_addRoleMember' + Char(39) +'empRole' + Char(39) + ',' + Char(39) + @username + Char(39)  BEGIN TRY  BEGIN TRANSACTION    EXEC (@createLogin)  EXEC (@createUser)  IF(@namerole = N'Admin')  EXEC (@addAdminRole)  IF(@namerole = N'LeaderTeam')  EXEC (@addLeaderRole)  IF(@namerole = N'Emp')  EXEC (@addEmpRole)  COMMIT  END TRY  BEGIN CATCH    IF (XACT\_STATE()) = -1  BEGIN  ROLLBACK TRANSACTION;  END;  END CATCH  END |

### **4.1.23.** SP\_UpdateStaff

|  |
| --- |
| ALTER PROC SP\_UpdateStaff(@Name NVARCHAR(100),@DateOfBirth Date,@Address NVARCHAR(100),@Sex NVARCHAR(10),  @Phone NVARCHAR(10),@Email NVARCHAR(50),@IdTeam int ,@UserName NVARCHAR(50) )  AS  BEGIN  IF(YEAR(@DateOfBirth)<(YEAR(GETDATE())-18))  UPDATE STAFF SET Name=@Name, DateOfBirth = @DateOfBirth,  Address =@Address, Sex = @sex,  Phone = @Phone,Email=@Email,IdTeam = @IdTeam  where UserName = @UserName    END |

### **4.1.24.** SP\_Account\_Create

|  |
| --- |
| ALTER PROC SP\_UpdateStaff(@Name NVARCHAR(100),@DateOfBirth Date,@Address NVARCHAR(100),@Sex NVARCHAR(10),  @Phone NVARCHAR(10),@Email NVARCHAR(50),@IdTeam int ,@UserName NVARCHAR(50) )  AS  BEGIN  IF(YEAR(@DateOfBirth)<(YEAR(GETDATE())-18))  UPDATE STAFF SET Name=@Name, DateOfBirth = @DateOfBirth,  Address =@Address, Sex = @sex,  Phone = @Phone,Email=@Email,IdTeam = @IdTeam  where UserName = @UserName    END |

## **4.2 Function**

### **4.2.1.** AUTO\_IDSTAFF

|  |
| --- |
| CREATE FUNCTION AUTO\_IDSTAFF()  RETURNS int  AS  BEGIN  DECLARE @ID int  IF (SELECT COUNT(Id) FROM STAFF) = 0  SET @ID = 0  ELSE  SELECT @ID = MAX(Id)+1 FROM STAFF    RETURN @ID  END |

### **4.2.2. AUTO\_IDTEAM**

|  |
| --- |
| CREATE FUNCTION AUTO\_IDTEAM()  RETURNS int  AS  BEGIN  DECLARE @ID int  IF (SELECT COUNT(Id) FROM TEAM) = 0  SET @ID = 0  ELSE  SELECT @ID = MAX(Id)+1 FROM TEAM    RETURN @ID  END |

### **4.2.3. AUTO\_IDTASK**

|  |
| --- |
| CREATE FUNCTION AUTO\_IDTASK()  RETURNS int  AS  BEGIN  DECLARE @ID int  IF (SELECT COUNT(Id) FROM TASK) = 0  SET @ID = 0  ELSE  SELECT @ID = MAX(Id)+1 FROM TASK    RETURN @ID  END |

### **4.2.4.** AUTO\_IDPROJECT

|  |
| --- |
| CREATE FUNCTION AUTO\_IDPROJECT()  RETURNS int  AS  BEGIN  DECLARE @ID int  IF (SELECT COUNT(Id) FROM PROJECT) = 0  SET @ID = 0  ELSE  SELECT @ID = MAX(Id)+1 FROM PROJECT    RETURN @ID  END |

### **4.2.5. FN\_ConvertToUnsign1**

|  |
| --- |
| CREATE FUNCTION [dbo].[FN\_ConvertToUnsign1] ( @strInput NVARCHAR(4000) ) RETURNS NVARCHAR(4000)  AS  BEGIN  IF @strInput IS NULL  RETURN @strInput  IF @strInput = ''  RETURN @strInput  DECLARE @RT NVARCHAR(4000)  DECLARE @SIGN\_CHARS NCHAR(136)  DECLARE @UNSIGN\_CHARS NCHAR (136)  SET @SIGN\_CHARS = N'ăâđêôơưàảãạáằẳẵặắầẩẫậấèẻẽẹéềểễệế ìỉĩịíòỏõọóồổỗộốờởỡợớùủũụúừửữựứỳỷỹỵý ĂÂĐÊÔƠƯÀẢÃẠÁẰẲẴẶẮẦẨẪẬẤÈẺẼẸÉỀỂỄỆẾÌỈĨỊÍ ÒỎÕỌÓỒỔỖỘỐỜỞỠỢỚÙỦŨỤÚỪỬỮỰỨỲỶỸỴÝ' +NCHAR(272)+ NCHAR(208)  SET @UNSIGN\_CHARS = N'aadeoouaaaaaaaaaaaaaaaeeeeeeeeee iiiiiooooooooooooooouuuuuuuuuuyyyyy AADEOOUAAAAAAAAAAAAAAAEEEEEEEEEEIIIII OOOOOOOOOOOOOOOUUUUUUUUUUYYYYYDD'  DECLARE @COUNTER INT  DECLARE @COUNTER1 int  SET @COUNTER = 1  WHILE (@COUNTER <=LEN(@strInput))  BEGIN  SET @COUNTER1 = 1  WHILE (@COUNTER1 <=LEN(@SIGN\_CHARS)+1)  BEGIN IF UNICODE(SUBSTRING(@SIGN\_CHARS, @COUNTER1,1)) = UNICODE(SUBSTRING(@strInput,@COUNTER ,1) )  BEGIN IF @COUNTER=1 SET @strInput = SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)-1)  ELSE SET @strInput = SUBSTRING(@strInput, 1, @COUNTER-1) +SUBSTRING(@UNSIGN\_CHARS, @COUNTER1,1) + SUBSTRING(@strInput, @COUNTER+1,LEN(@strInput)- @COUNTER)  BREAK END SET @COUNTER1 = @COUNTER1 +1 END SET @COUNTER = @COUNTER +1 END SET @strInput = replace(@strInput,' ','-')  RETURN @strInput  END |

### **4.2.6. FN\_storeListProjectRunning**

|  |
| --- |
| CREATE FUNCTION FN\_storeListProjectRunning ()  RETURNS TABLE  AS RETURN (  SELECT \* FROM PROJECT Where TimeEnd > GETDATE() AND status = 1  GROUP BY Id, Name,Content,StatusProject,TimeFrom,TimeEnd,IdTeam,status  ) |

### **4.2.7.** FN\_storeListProjectTeam

|  |
| --- |
| CREATE FUNCTION FN\_storeListProjectTeam(@IdTeam INT)  RETURNS TABLE  AS RETURN  (SELECT \* FROM PROJECT Where IdTeam = @IdTeam AND status = 1 And StatusProject =1  ) |

### **4.2.8.** FN\_storeListSalary

|  |
| --- |
| CREATE FUNCTION FN\_storeListSalary()  RETURNS TABLE  AS RETURN (  Select Month ,SYear, STAFF.Name ,Salary as 'FixedSalary' ,Bonus,TEAM.Name AS 'Team'  From STAFF,SALARY,TEAM  WHERE STAFF.Id = SALARY.IdStaff AND STAFF.IdTeam = TEAM.Id AND Month = MONTH(GETDATE())  ) |

### **4.2.9.** FN\_storeTop5\_StaffOfTeam

|  |
| --- |
| ALTER FUNCTION FN\_storeTop5\_StaffOfTeam(@IdTeam INT)  RETURNS TABLE  AS RETURN  (  SELECT top 5 COUNT(TASK.Id) AS N'NumTask' , STAFF.Id,STAFF.Name  FROM TASK INNER JOIN STAFF ON STAFF.Id = TASK.IdStaff  INNER JOIN TEAM ON STAFF.IdTeam = TEAM.Id AND TEAM.Id = @IdTeam  where TASK.status =1  GROUP BY STAFF.Id,STAFF.Name  ORDER BY COUNT(TASK.Id) DESC  )  go |

|  |
| --- |
| **4.2.10.** FN\_storeSumSalaryTeam |
| ALTER FUNCTION FN\_storeSumSalaryTeam()  RETURNS TABLE  AS RETURN  (  SELECT TOP 5 SUM(SALARY.Salary + SALARY.Bonus) AS N'SumSalary',  TEAM.Id, TEAM.Name, STAFF.Name as N'Name Admin'  FROM STAFF INNER JOIN SALARY ON IdStaff = STAFF.Id  INNER JOIN TEAM ON TEAM.Id = IdStaff AND TEAM.IdAmin = STAFF.Id  GROUP BY TEAM.Id, TEAM.Name,STAFF.Name  ORDER BY SUM(SALARY.Salary + SALARY.Bonus) DESC  ) |

|  |
| --- |
| **4.2.11.** FN\_storeTop5\_SalaryOfCompany |
| CREATE FUNCTION FN\_storeTop5\_SalaryOfCompany(@Year int)  RETURNS TABLE  AS RETURN  (  SELECT TOP 5 SUM(SALARY.Salary + SALARY.Bonus) AS N'SumSalary',  STAFF.Id,STAFF.Name,TEAM.Name AS N'Team'  FROM SALARY INNER JOIN STAFF ON STAFF.Id = IdStaff  INNER JOIN TEAM ON STAFF.IdTeam = TEAM.Id  WHERE SALARY.SYear = @Year  GROUP BY STAFF.Id,STAFF.Name,TEAM.Name  ORDER BY SUM(SALARY.Salary+ SALARY.Bonus) DESC  ) |

## **4.3 Trigger**

### **4.3.1.** TG\_UpDateStaff

|  |
| --- |
| ALTER TRIGGER TG\_UpDateStaff  ON dbo.STAFF FOR UPDATE,insert  AS  BEGIN  DECLARE @id int,  @userName NVARCHAR(50),  @name NVARCHAR(50),  @status INT,  @address NVARCHAR(100),  @email NVARCHAR(50),  @phone NVARCHAR(10),  @idTeam INT,  @dateOfBirth DATE,  @sex NVARCHAR(10),  @aboutStaff NVARCHAR(500)    --  SELECT @id = Id, @userName = UserName, @status=status , @email= Email ,@phone =Phone,@idTeam =IdTeam FROM Inserted  --  DECLARE @CK\_TASK int  DECLARE @CK\_TEAM int  Select @CK\_TASK = Count(\*)  From TASK  Where TASK.IdStaff = @id    Select @CK\_TEAM = Count(\*)  From TEAM  Where @idTeam = TEAM.IdAmin  -- check Task and TEAM admin  IF(@status = 0)  BEGIN  If (@CK\_TASK != 0 or @CK\_TEAM != 0)  Update STAFF set status=1 where Id=@id  END    -- check Id\_Team  DECLARE @CK\_IdTeam int  SELECT @CK\_IdTeam = status  FROM TEAM where TEAM.Id = @idTeam  IF (@CK\_IdTeam = 0 or @CK\_IdTeam= NULL)  BEGIN  Update STAFF set status=0 where Id=@id  END  --- check email, phone  IF(@email = N'' or len(@phone)<>10 or @idTeam is null )  BEGIN  Update STAFF set Email = N'CELLO'+ convert(nvarchar(10), Id) +N'@gmail.com' ,  Phone = '012345678'+ convert(nvarchar(10), @id)+ convert(nvarchar(10),  Id)  where UserName=@userName    END  DECLARE @Year int  SELECT @Year = YEAR(GETDATE())  EXEC SP\_addSalaryAllYear @Year    END |

### **4.3.2. TG\_UpdateBonusSalary**

|  |
| --- |
| ALTER TRIGGER TG\_UpdateBonusSalary  ON dbo.TASK FOR INSERT,update  AS  BEGIN  Declare @IdStaff int,  @IdStask int,  @salary int,  @month int,  @year int,  @timeEnd DateTime,  @status int,  @count int  Select @IdStaff= IdStaff , @timeEnd=TimeEnd ,@status=status From inserted  Select @count = Count(\*) From TASK where IdStaff=@IdStaff and YEAR(@timeEnd)=YEAR(TimeEnd)  and MONTH(@timeEnd)=MONTH(TimeEnd)  and status = 1  and StatusTask = 1  SELECT @month = MONTH(@timeEnd);  SELECT @year = YEAR(@timeEnd);  If(@count > 0 and @month != 12 )  begin  update SALARY set Bonus = Salary\*0.01\*@count where IdStaff=@IdStaff AND Month= @month AND SYear =@year  end  else if(@count = 0 and @month != 12)  begin  update SALARY set Bonus = Salary\*(-0.005) where IdStaff=@IdStaff AND Month= @month AND SYear =@year  end  EXEC SP\_checkSalaryYearStaff @year  EXEC SP\_checkSalaryYearAdmin @year  END |

### **4.3.3. TG\_UpdateBonusSalary\_Deleted**

|  |
| --- |
| ALTER TRIGGER TG\_UpdateBonusSalary\_Deleted  ON dbo.TASK FOR update  AS  BEGIN    Declare @IdStaff int,  @IdStask int,  @salary int,  @month int,  @year int,  @timeEnd DateTime,  @status int,  @count int  Select @IdStaff= IdStaff , @timeEnd=TimeEnd ,@status=status From deleted  Select @count = Count(\*) From TASK where IdStaff=@IdStaff and YEAR(@timeEnd)=YEAR(TimeEnd)  and MONTH(@timeEnd)=MONTH(TimeEnd)  and status = 1  and StatusTask = 1  SELECT @month = MONTH(@timeEnd);  SELECT @year = YEAR(@timeEnd);  If(@count > 0 and @month != 12 )  begin  update SALARY set Bonus = Salary\*0.01\*@count where IdStaff=@IdStaff AND Month= @month AND SYear =@year  end  else if(@count = 0 and @month != 12)  begin  update SALARY set Bonus = Salary\*(-0.005) where IdStaff=@IdStaff AND Month= @month AND SYear =@year  end  END |

### 4.3.4. TG\_UpdateStatusProject

|  |
| --- |
| ALTER Trigger TG\_UpdateStatusProject  On dbo.PROJECT FOR UPDATE  as  BEGIN  DECLARE @IdProject int,  @IdTask int,  @StatusPro int,  @Status int,  @StatusTask int,  @year int  SELECT @IdProject = Id ,@Status = status,@StatusPro =StatusProject, @year = YEAR(TimeEnd) FROM inserted  IF(@Status = 0)  BEGIN  UPDATE TASK SET status = 0 where IdProject =@IdProject  END  IF(@StatusPro = 0)  BEGIN  UPDATE PROJECT SET StatusProject =0 where Id = @IdProject  END  EXEC SP\_checkSalaryYearStaff @year  EXEC SP\_checkSalaryYearAdmin @year    END |

### 4.3.4. TG\_UpdateTimeProjectStaff\_Task

|  |
| --- |
| ALTER TRIGGER TG\_UpdateTimeProjectStaff\_Task  ON dbo.TASK FOR UPDATE  AS  BEGIN  DECLARE @IdTask int,  @TimeFromTask DateTime,  @TimeEndTask DateTime,  @IdProject int,  @CK\_StatusPro int,  @CK\_DonePro int,  @TimeFromPro DateTime,  @TimeEndPro DateTime,  @IdStaff int,  @CK\_StatusTask int,  @CK\_Status int  SELECT @IdTask = Id ,  @TimeFromTask= TimeFrom,  @TimeEndTask = TimeEnd,  @IdStaff = IdStaff,  @IdProject =IdProject,  @CK\_StatusTask = status  FROM inserted  SELECT @TimeFromPro = TimeFrom ,@TimeEndPro = TimeEnd,@CK\_StatusPro = StatusProject  FROM PROJECT  WHERE Id=@IdProject  SELECT @CK\_Status = status FROM STAFF WHERE Id =@IdStaff  IF(@TimeFromTask < @TimeFromPro)  begin  UPDATE TASK SET TimeFrom =@TimeFromPro where Id= @IdTask  end  IF(@TimeFromTask > @TimeEndPro)  begin  UPDATE TASK SET TimeFrom =@TimeFromPro where Id= @IdTask  end  IF(@TimeEndTask > @TimeEndPro)  UPDATE TASK SET TimeEnd =@TimeEndPro where Id= @IdTask  IF(@TimeEndTask < @TimeFromPro)  UPDATE TASK SET TimeEnd =@TimeEndPro where Id= @IdTask  IF(@CK\_Status != 1 or @CK\_StatusPro != 1 )  BEGIN  UPDATE TASK SET status = 0 where Id= @IdTask  END  IF(@CK\_StatusTask=0)  UPDATE TASK SET StatusTask = 0 where Id= @IdTask  EXEC SP\_checkSalaryYearStaff @year  EXEC SP\_checkSalaryYearAdmin @year  END |

### 4.3.5. TG\_InsertTask

|  |
| --- |
| CREATE TRIGGER TG\_InsertTask  ON dbo.TASK FOR INSERT  AS  BEGIN  DECLARE @IdTask int,  @IdProject int,  @TimeEndPro DateTime,  @CK\_StatusPro int  SELECT @IdTask = Id,  @IdProject =IdProject  FROM inserted  SELECT @TimeEndPro = TimeEnd, @CK\_StatusPro = StatusProject  FROM PROJECT  WHERE Id=@IdProject  If(@TimeEndPro < GETDATE())  UPDATE TASK SET StatusTask =0 , status = 0 where Id= @IdTask  END |

### **4.4 View**

### **4.4.1.** View\_AdminTeam

|  |
| --- |
| CREATE VIEW View\_AdminTeam  AS  select STAFF.Id,STAFF.Name,DateOfBirth,STAFF.Address,Sex,STAFF.Phone,STAFF.Email,STAFF.UserName  from STAFF,TEAM  where STAFF.Id = IdAmin  GO |

### **4.4.2.** View\_TeamInfo

|  |
| --- |
| CREATE VIEW View\_TeamInfo  AS  (  SELECT \* FROM TEAM  ) |

### **4.4.3.** View\_ProjectTeam

|  |
| --- |
| CREATE VIEW View\_ProjectTeam  AS  (  SELECT PROJECT.Name,PROJECT.StatusProject,PROJECT.TimeFrom,PROJECT.TimeEnd,TEAM.Name AS N'TEAM'  FROM PROJECT INNER JOIN TEAM ON IdTeam = TEAM.Id  ) |

### **4.4.4.** View\_TaskTeam

|  |
| --- |
| CREATE VIEW View\_TaskTeam  As(  SELECT TASK.Id,TASK.Name,TASK.TimeFrom,TASK.TimeEnd , TEAM.Name AS N'TEAM'  FROM TASK INNER JOIN STAFF ON TASK.IdStaff=STAFF.Id INNER JOIN TEAM ON STAFF.IdTeam =TEAM.Id  ) |

### **4.4.5.** View\_Account

|  |
| --- |
| CREATE VIEW View\_Account  AS(  SELECT \* FROM ACCOUNT  ) |

### **4.4.6.** View\_TypeStaff

|  |
| --- |
| CREATE VIEW View\_TypeStaff  AS(  SELECT STAFF.Id,STAFF.Name, ACCOUNT.Type,TEAM.Name AS N'TEAM'  FROM STAFF INNER JOIN ACCOUNT ON STAFF.UserName = ACCOUNT.UserName,TEAM  WHERE TEAM.Id =STAFF.IdTeam  ) |

## **5.1 Phân quyền**

### ***5.1.1 Phân quyền Admin***

* Các thao tác được phép truy cập:
  + Xem, sửa, xóa các bảng trong CSDL
  + Xem các view trong CSDL
  + Thực thi các thủ tục và hàm
* Code thực thi

|  |
| --- |
| CREATE LOGIN Admin WITH PASSWORD = '123'  CREATE USER AdminUser FOR LOGIN Admin  CREATE ROLE adminRole  ALTER ROLE db\_owner ADD MEMBER adminUser  GRANT SELECT, INSERT, UPDATE ON dbo.ACCOUNT TO adminRole  GRANT SELECT, INSERT, UPDATE ON dbo.PROJECT TO adminRole  GRANT SELECT, INSERT, UPDATE ON dbo.STAFF TO adminRole  GRANT SELECT, INSERT, UPDATE ON dbo.SALARY TO adminRole  GRANT SELECT, INSERT, UPDATE ON dbo.TASK TO adminRole  GRANT SELECT, INSERT, UPDATE ON dbo.TEAM TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_addTask TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteTask TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_addAccount TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteAccount TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_addStaff TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteStaff TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_addProject TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteProject TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteProject TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteTeam TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_addSalaryAllYear TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_storeStaffOfTeam TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_storeSalaryTeam TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryMonth TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearStaff TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearAdmin TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_Login TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_InforUser TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_ListTaskRunning TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_storeTaskName TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_Account\_Create TO adminRole  GRANT EXECUTE ON OBJECT:: SP\_UpdateStaff TO adminRole  GRANT EXECUTE ON OBJECT:: FN\_ConvertToUnsign1 TO adminRole  GRANT EXECUTE ON OBJECT:: FN\_storeListProjectRunning TO adminRole  GRANT EXECUTE ON OBJECT:: FN\_storeTop5\_SalaryOfCompany TO adminRole  GRANT EXECUTE ON OBJECT:: FN\_storeSumSalaryTeam TO adminRole  ALTER ROLE adminRole ADD MEMBER adminUser |

### ***5.1.2Phân quyền LeaderTeam***

* Các thao tác được phép truy cập:
  + Xem, sửa một số bảng trong CSDL
  + Thực thi một số thủ tục và hàm
* Code thực thi

|  |
| --- |
| CREATE LOGIN LeaderTeam WITH PASSWORD = '123'  CREATE USER projectUser FOR LOGIN LeaderTeam  GRANT SELECT, UPDATE ON dbo.ACCOUNT TO projectsRole  GRANT SELECT, INSERT, UPDATE ON dbo.PROJECT TO projectsRole  GRANT SELECT, INSERT, UPDATE ON dbo.STAFF TO projectsRole  GRANT SELECT,INSERT, UPDATE ON dbo.SALARY TO projectsRole  GRANT SELECT, INSERT, UPDATE ON dbo.TASK TO projectsRole  GRANT SELECT ON dbo.TEAM TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_addTask TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteTask TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_addStaff TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteStaff TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_addProject TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteProject TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_addSalaryAllYear TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_storeStaffOfTeam TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_storeSalaryTeam TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryMonth TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearStaff TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearAdmin TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_Login TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_InforUser TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_ListTaskRunning TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_storeTaskName TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_Account\_Create TO projectsRole  GRANT EXECUTE ON OBJECT:: SP\_UpdateStaff TO projectsRole  GRANT EXECUTE ON OBJECT:: FN\_ConvertToUnsign1 TO projectsRole  GRANT EXECUTE ON OBJECT:: FN\_storeListProjectRunning TO projectsRole  GRANT EXECUTE ON OBJECT:: FN\_storeTop5\_SalaryOfCompany TO projectsRole  GRANT EXECUTE ON OBJECT:: FN\_storeSumSalaryTeam TO projectsRole  ALTER ROLE projectRole ADD MEMBER projectUser |

### ***5.1.3 Phân quyền Emp***

* Các thao tác được phép truy cập:
  + Xem, sửa một số bảng trong CSDL
  + Thực thi một số thủ tục và hàm
* Code thực thi

|  |
| --- |
| CREATE LOGIN Emp WITH PASSWORD = '123'  CREATE USER EmpUser FOR LOGIN Emp  CREATE ROLE empRole  ALTER ROLE employeesRole ADD MEMBER employeeUser  GRANT SELECT, UPDATE ON dbo.ACCOUNT TO empRole  GRANT SELECT ON dbo.PROJECT TO empRole  GRANT SELECT, UPDATE ON dbo.STAFF TO empRole  GRANT SELECT ON dbo.SALARY TO empRole  GRANT SELECT, INSERT, UPDATE ON dbo.TASK TO empRole  GRANT SELECT ON dbo.TEAM TO empRole  GRANT EXECUTE ON OBJECT:: SP\_addTask TO empRole  GRANT EXECUTE ON OBJECT:: SP\_DeleteTask TO empRole  GRANT EXECUTE ON OBJECT:: SP\_storeStaffOfTeam TO empRole  GRANT EXECUTE ON OBJECT:: SP\_storeSalaryTeam TO empRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryMonth TO empRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearStaff TO empRole  GRANT EXECUTE ON OBJECT:: SP\_checkSalaryYearAdmin TO empRole  GRANT EXECUTE ON OBJECT:: SP\_Login TO empRole  GRANT EXECUTE ON OBJECT:: SP\_InforUser TO empRole  GRANT EXECUTE ON OBJECT:: SP\_ListTaskRunning TO empRole  GRANT EXECUTE ON OBJECT:: SP\_storeTaskName TO empRole  GRANT EXECUTE ON OBJECT:: SP\_UpdateStaff TO empRole  GRANT EXECUTE ON OBJECT:: FN\_ConvertToUnsign1 TO empRole  GRANT EXECUTE ON OBJECT:: FN\_storeListProjectRunning TO empRole  GRANT EXECUTE ON OBJECT:: FN\_storeTop5\_SalaryOfCompany TO empRole  GRANT EXECUTE ON OBJECT:: FN\_storeSumSalaryTeam TO empRole  ALTER ROLE employeesRole ADD MEMBER employeeUser |

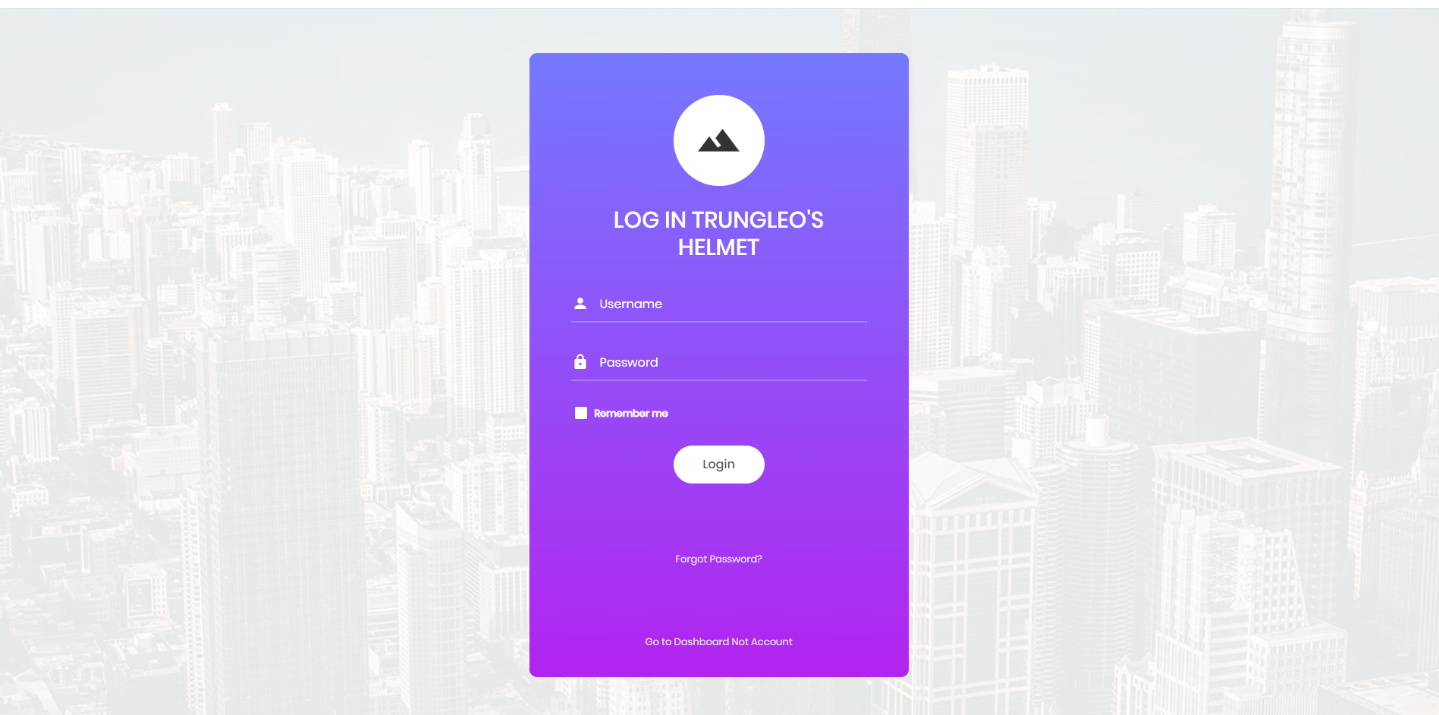
# **CHƯƠNG 5. CÀI ĐẶT CHƯƠNG TRÌNH**

## **5.1 Giao diện trang chủ**

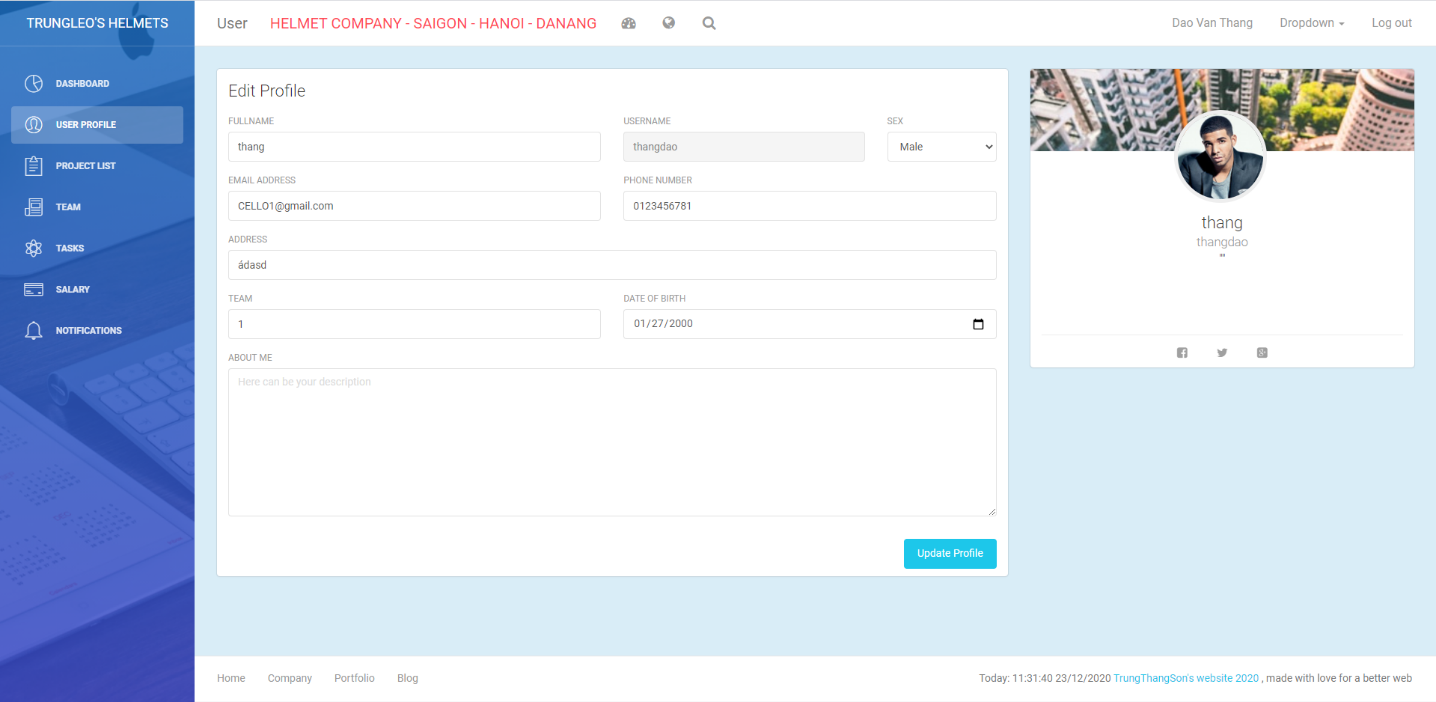
## **5.2 Giao diện trang trang đăng nhập**

*Hình SEQ Hình \\* ARABIC 3. Giao diện trang mua hàng*

*Hình SEQ Hình \\* ARABIC 4. Giao diện trang mua hàng*

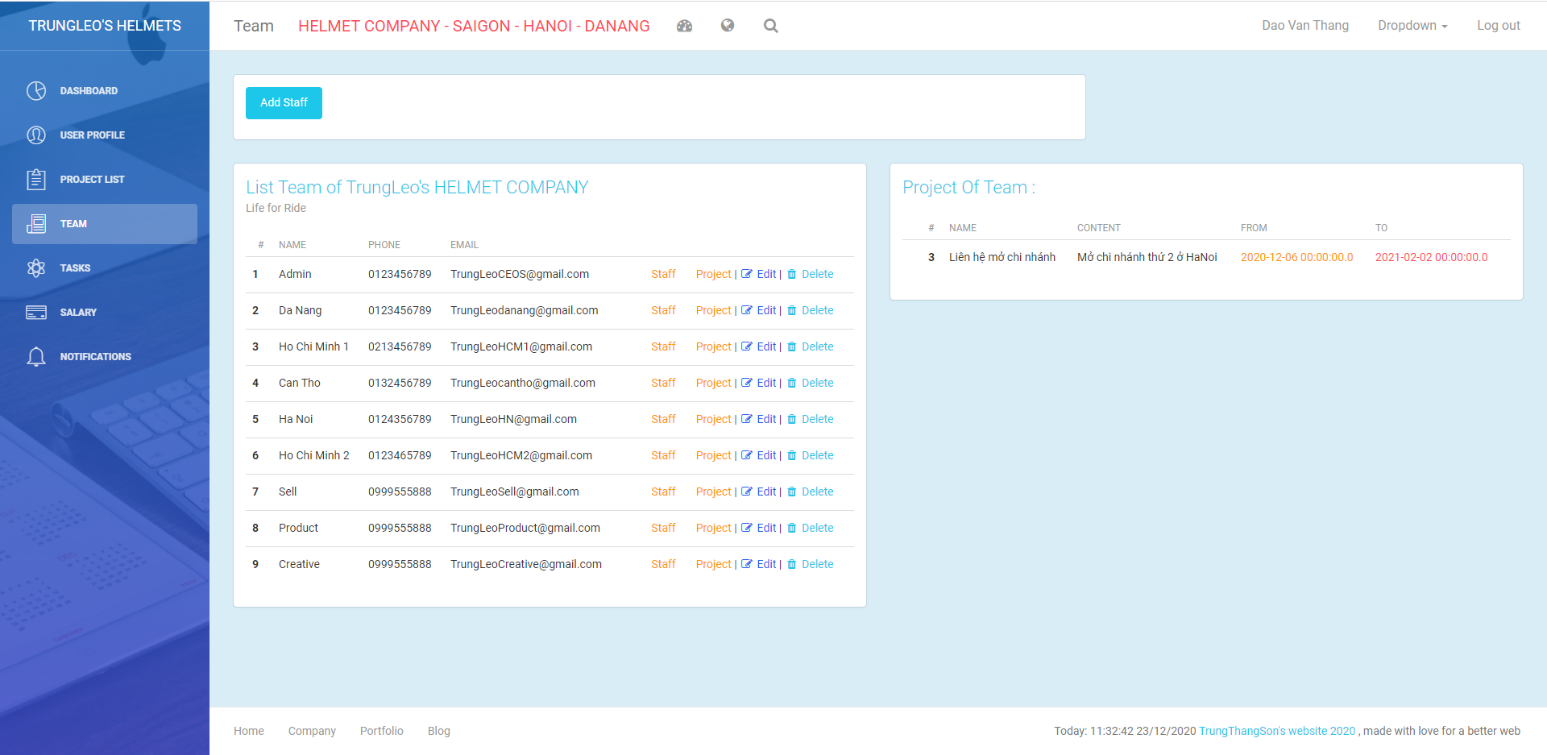


*Hình SEQ Hình \\* ARABIC 5. Đơn hàng*

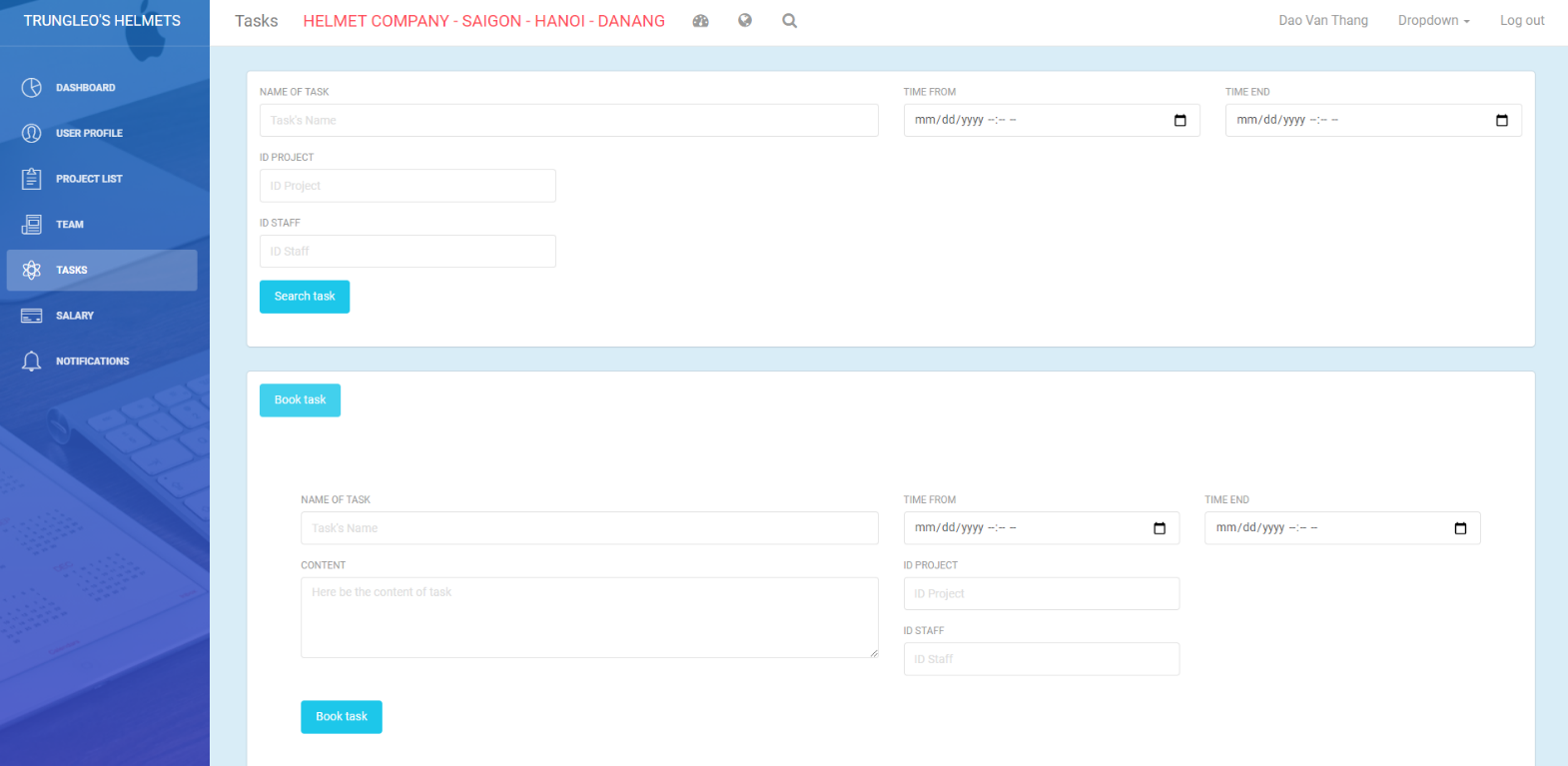
****5.3 Giao diện trang thông tin cá nhân cho phép chỉnh sửa**

## **5.4 Giao diện trang quản lý Project**

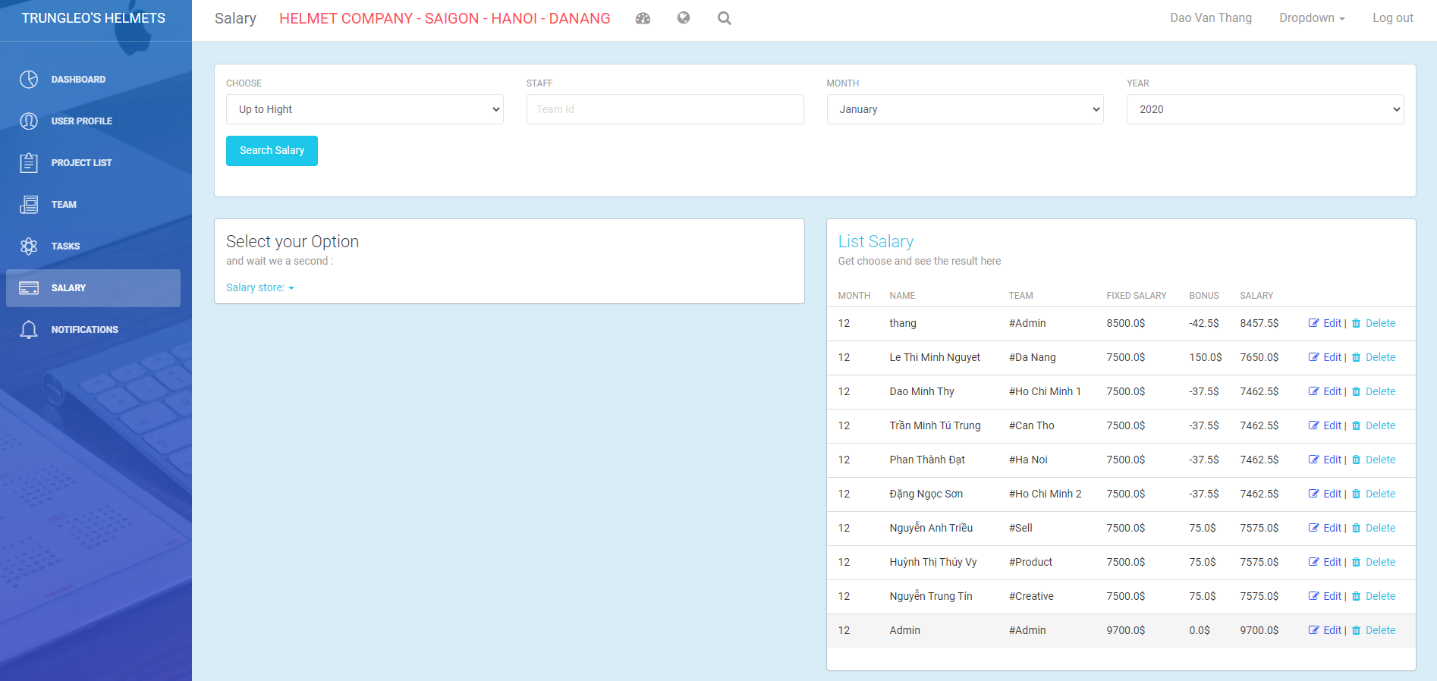
## **5.5 Giao diện trang quản lý Team**

****

## **5.6 Giao diện trang quản lý Task**

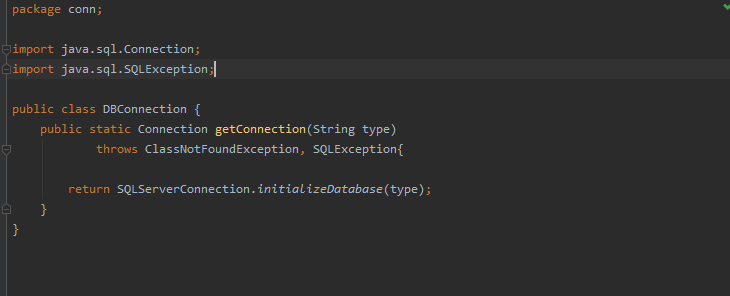


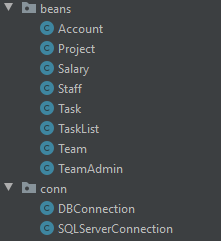
## **5.7 Giao diện trang quản lý Salary(Lương)**

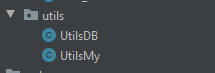
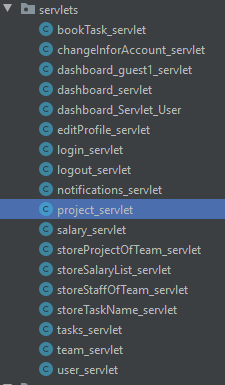


*Hình SEQ Hình \\* ARABIC 7. Giao diện trang đăng nhập*

* 1. **Tổ chức các lớp trong chương trình**





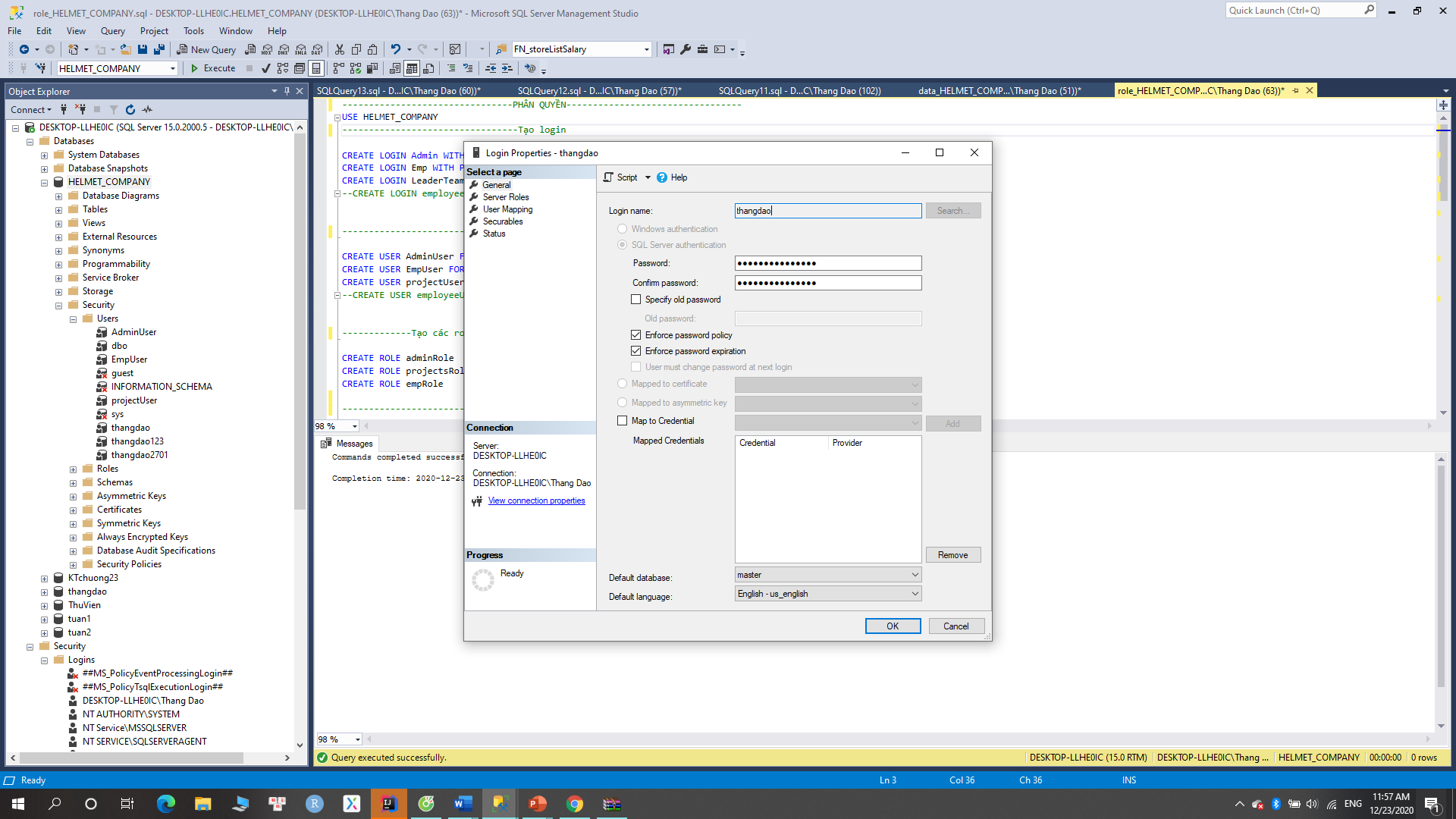
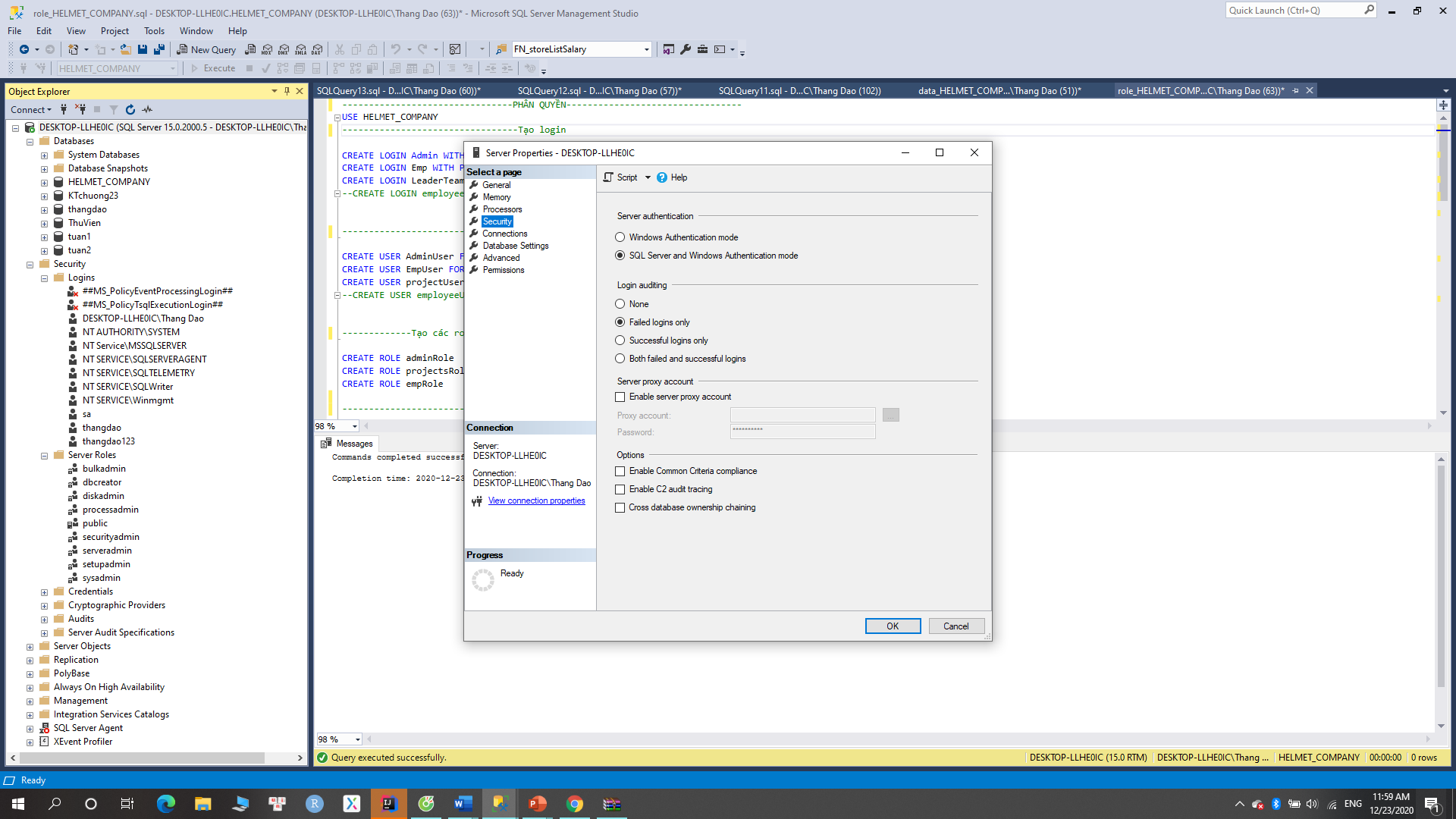
 

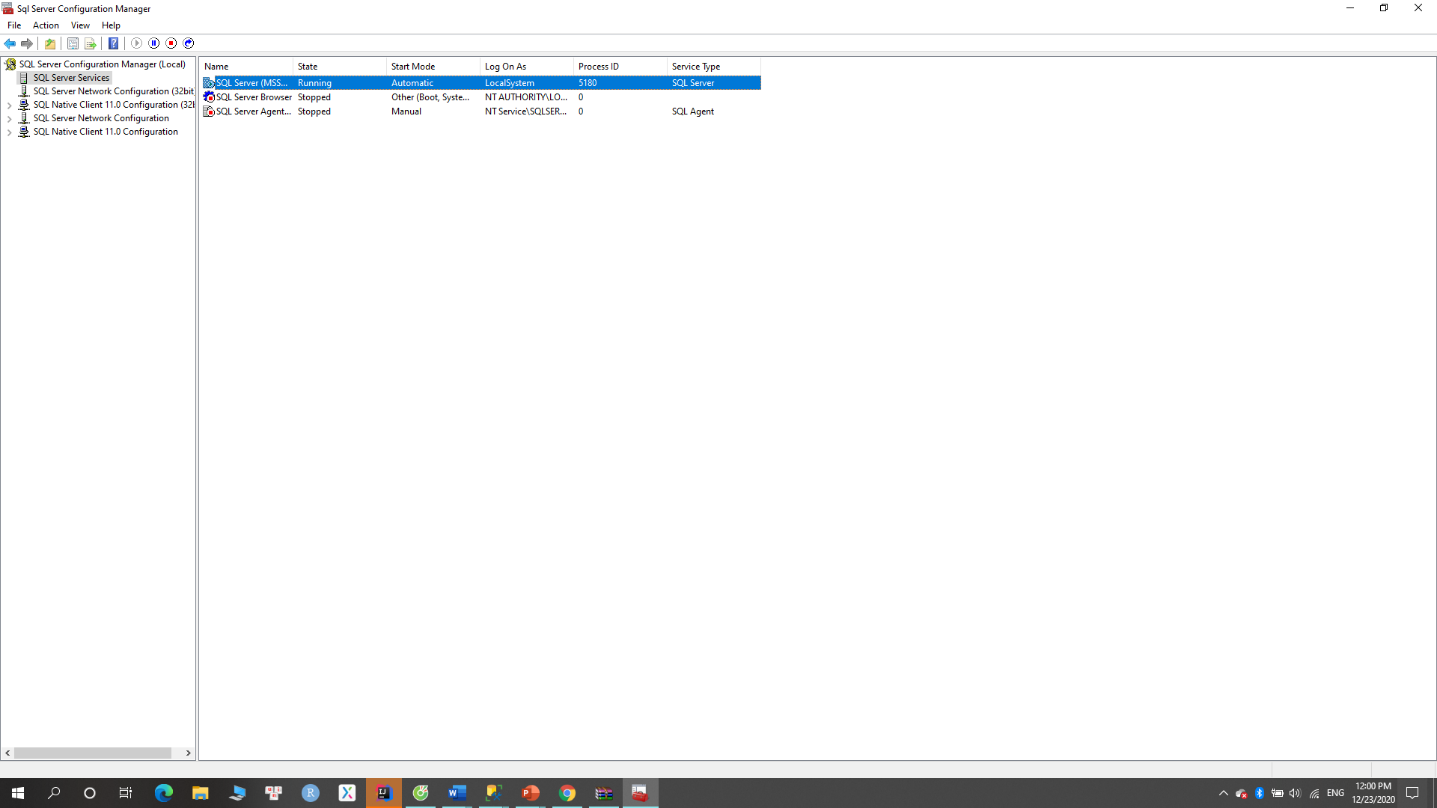
*Hình SEQ Hình \\* ARABIC 12. Models*

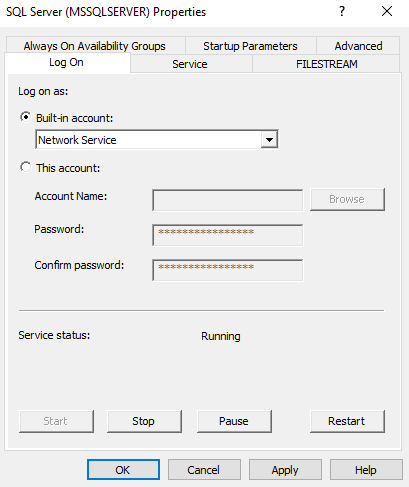
* Views: tầng giao diện hiển thị cho người dùng nhìn thấy, hiển thị dữ liệu được truy xuất từ tầng Servlets
* Servlets : tầng giúp kết nối giữa tầng conn và Utils và tầng Views
* Kết nối CSDL thông qua Connection strings trong conn và Utils:

*Hình SEQ Hình \\* ARABIC 13. Chuỗi ConnectionStrings*

**5.9 Kết nối CSDL từ xa**

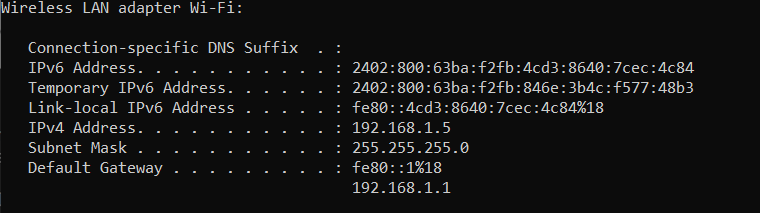
* Tạo login từ SQL Server
* Cho phép đăng nhập bằng tài khoản
* Mở SQL Server Configuration Manager

Chọn Properties của MSSQLSERVER, chọn Built-in account – Network Service

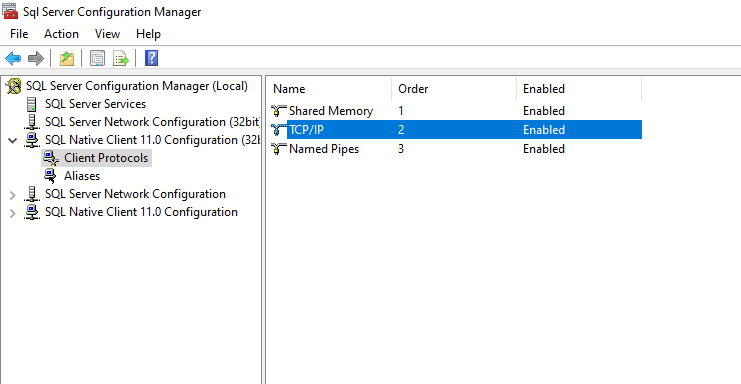


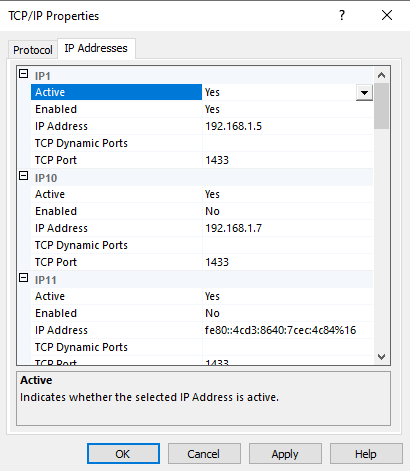
* Cấu hình TCP/IP – IP Address:

Mở CMD, gõ ipconfig để lấy địa chỉ Ipv4



Chọn Properties của TCP/IP



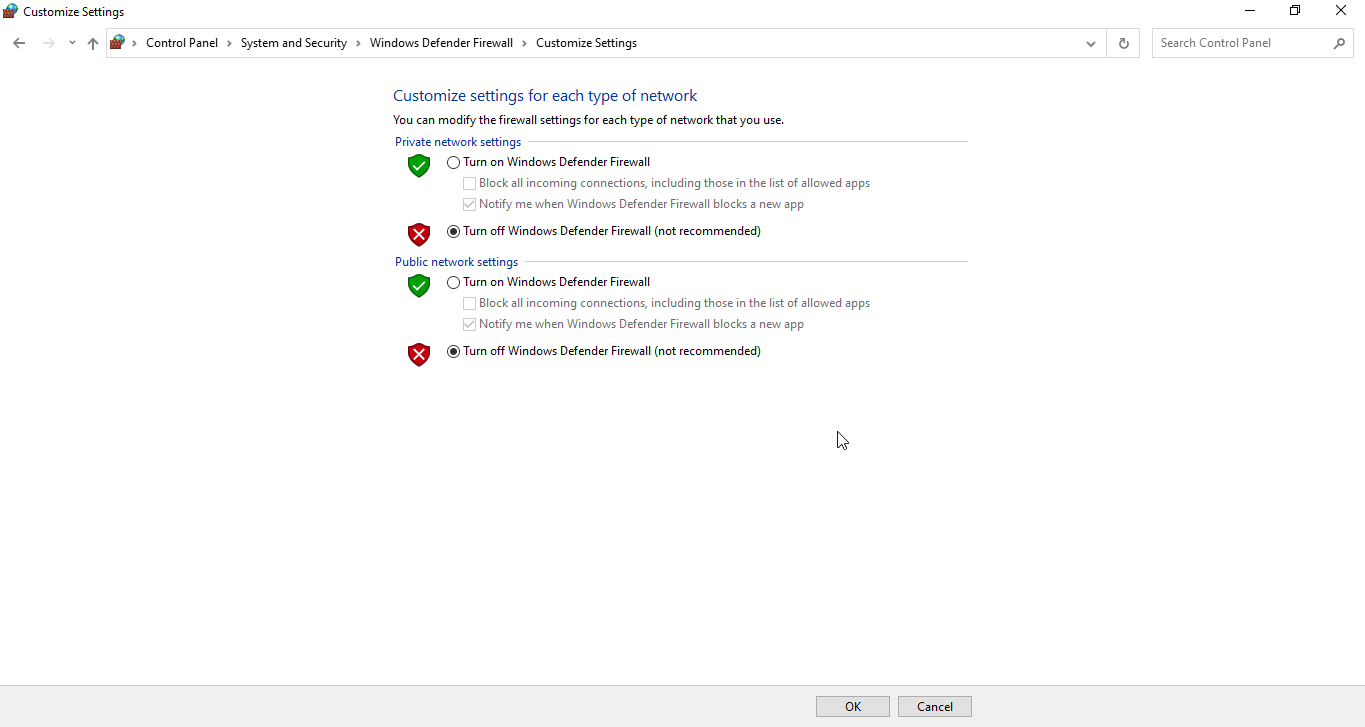


Protocol -> Enable -> Yes

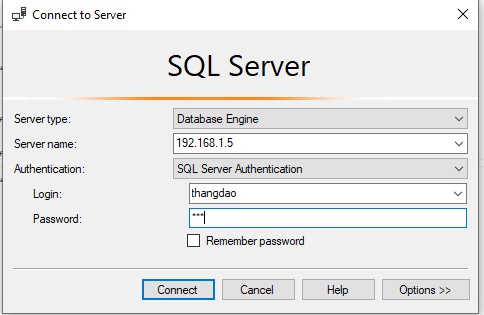
Cấu hình IP Address



* Tắt Firewall



* Đăng nhập SQL Server bằng account được tạo



# **CHƯƠNG 6. TỔNG KẾT**

## **6.1. bóng góp đề tài**

Xây dựng trang đẹp mắt, trực quan, thân thiện với người dùng. Trang web được áp dụng các kiến thức của Hệ quản trị CSDL, cụ thể:

* Phân tích, nắm bắt được nhu cầu sử dụng trang web để thiết kế CSDL phù hợp.
* Xây dựng CSDL chặt chẽ từ lược đồ ERD.
* Phân cấp, phân quyền cho các đối tượng sử dụng CSDL.
* Tìm hiểu và nắm rõ về các khái niệm trong SQL Server.
* Có khả năng mở rộng CSDL khi có nhu cầu.
* Kết nối và chạy được trên nền tảng web.
* Tạo thống kê với Chart để dễ dàng quản lý.

## **6.2 Các tính năng bổ sung, đặc biệt trong đề tài**

### **6.2.1 Tính Lương nhẩy tự động**

|  |
| --- |
| ALTER PROC SP\_checkSalaryYearStaff( @year int)  as  Begin    DECLARE @SumSalary real,@SumBonus real,  @CountKPIProject int , @IdStaff int,  @salary real,@bonus real    DECLARE cursor\_SALARY\_Year CURSOR FOR  SELECT SYear,IdStaff, Salary ,Bonus  FROM SALARY,STAFF  where SYear = @year and Month = 13 AND IdStaff=Id AND STAFF.status=1  OPEN cursor\_SALARY\_Year  FETCH NEXT FROM cursor\_SALARY\_Year INTO @year,@IdStaff,@salary,@bonus  WHILE @@FETCH\_STATUS = 0  BEGIN  Select @CountKPIProject = Count(TASK.Id)  from TASK inner join PROJECT on PROJECT.Id = TASK.IdProject and YEAR(PROJECT.TimeEnd) = @year  where TASK.IdStaff = @IdStaff  and PROJECT.StatusProject = 1    Select @SumSalary= Sum(Salary) , @SumBonus= Sum(Bonus) From Salary  Where IdStaff = @IdStaff And SYear = @year AND Month<>13    IF(@CountKPIProject >= 5)  Begin  update SALARY set Salary = @CountKPIProject\*0.2\*@SumBonus  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  end  else if (@CountKPIProject >= 10)  update SALARY set Salary = @CountKPIProject\*0.5\*@SumBonus  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  else if (@CountKPIProject <5 )  update SALARY set Salary = 0  where IdStaff=@IdStaff AND Month= 13 And SYear = @year  FETCH NEXT FROM cursor\_SALARY\_Year INTO @year,@IdStaff,@salary,@bonus  END  CLOSE cursor\_SALARY\_Year  DEALLOCATE cursor\_SALARY\_Year  end |

|  |
| --- |
|  |

### **6.2.2 Sử dụng Cursor**

Sử dụng Cursor thay đổi Status của bảng Discounts, mã giảm giá có thời gian quy định sử dụng trong khoảng thời gian nhất định, dùng cursor để duyệt từng mã giảm giá có trong CSDL để kiểm tra HSD.

### **6.2.3 Thêm login kết nối server**

|  |
| --- |
| ALTER proc [dbo].[SQLAccount\_Create](@username nvarchar(20),@password nvarchar(max),@namerole nvarchar(50)) as  BEGIN  DECLARE @createUser nvarchar(max)  DECLARE @createLogin nvarchar(max)  DECLARE @addSellerRole nvarchar(max)  DECLARE @addAdminRole nvarchar(max)  DECLARE @error int  SET @error=0  SET @createLogin='create login '+ @username + ' with password = ' + Char(39) + @password + Char(39)  SET @createUser='create user '+ @username + ' for login '+ @username  SET @addSellerRole='Sp\_addRoleMember' + Char(39) +'seller\_role' + Char(39) + ',' + Char(39) + @username + Char(39)  SET @addAdminRole='Sp\_addRoleMember' + Char(39) +'db\_owner' + Char(39) + ',' + Char(39) + @username + Char(39)  BEGIN TRY  BEGIN TRANSACTION    EXEC (@createLogin)  EXEC (@createUser)  IF(@namerole = N'Admin')  EXEC (@addAdminRole)  IF(@namerole = N'Bán hàng')  EXEC (@addSellerRole)  COMMIT  END TRY  BEGIN CATCH  IF (XACT\_STATE()) = -1  BEGIN  ROLLBACK TRANSACTION;  END;  END CATCH  END |

## **6.2.4 Xóa login kết nối server**

|  |
| --- |
| CREATE PROC [dbo].[SQLAccount\_KillConnection] @username nvarchar(20)  AS  BEGIN  DECLARE @killConnection NVARCHAR(MAX) = N'';  SELECT @killConnection += N'KILL ' + CONVERT(VARCHAR(11), session\_id) + N';'  FROM sys.dm\_exec\_sessions  WHERE login\_name = @username  EXEC sys.sp\_executesql @killConnection  END  GO |

## **6.3 Các hạn chế**

* Các chức năng trong trang web chưa hoàn thiện nên chưa thể sử dụng được tất cả các câu truy vấn.
* Một số truy vấn vẫn còn chưa được tối ưu.
* Chưa đảm bảo tất cả các ngoại lệ có thể xảy ra trong thực tế.

## **6.4 Hướng phát triển**

Nhóm đã cố gắng phát triển và xây dựng nền tảng đề tài theo yêu cầu. Do thời gian có hạn chỉ mới đi vào các chức năng cơ bản. Với đề tài này, nhóm có hướng phát triển xây dựng được toàn bộ các chức năng cần có, chỉnh sửa, mở rộng và tối ưu chức năng để hoàn thiện chương trình.

**TÀI LIỆU THAM KHẢO**

**PHỤ LỤC**

**Hướng dẫn sử dụng**

(File đính kèm)