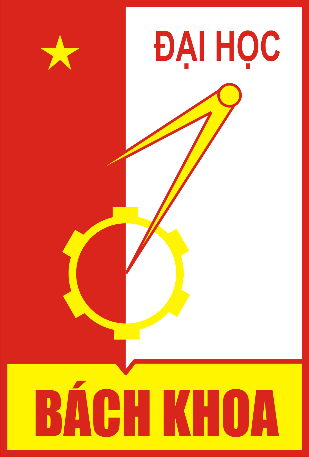
**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

TRƯỜNG ĐIỆN – ĐIỆN TỬ

------🕮------



**BÁO CÁO MÔN HỌC**

**KỸ THUẬT PHẦN MỀM ỨNG DỤNG**

**Đề tài:**

quản lí nhân sự

**Nhóm 22**

**Giảng viên hướng dẫn: TS. Vũ Hải**

**Hà Nội, tháng 6 năm 2025**

**Mục lục**

[**Lời nói đầu** 1](#_Toc200924773)

[**I. Giới thiệu chung** 2](#_Toc200924774)

[*1. Giới thiệu thành viên* 2](#_Toc200924775)

[*2. Mục tiêu dự án* 3](#_Toc200924776)

[*3. Công nghệ và Môi trường phát triển* 4](#_Toc200924777)

[*4. Phạm vi dự án* 5](#_Toc200924778)

[*5. Các công cụ sử dụng* 5](#_Toc200924779)

[**II. Phân tích và thiết kế** 6](#_Toc200924780)

[*1. Yêu cầu bài toán và hiện trạng* 6](#_Toc200924781)

[*2. Lý do chọn đề tài* 7](#_Toc200924782)

[*3. Mục đích đề tài và ý nghĩa thực tiễn* 8](#_Toc200924783)

[*4. Nền tảng sử dụng và các chức năng chính* 8](#_Toc200924784)

[*5. Sơ đồ PCCN* 9](#_Toc200924785)

[*6. Một số Use Case* 10](#_Toc200924786)

[6.1 Cập nhật thông tin (nhân viên) 10](#_Toc200924787)

[6.2 Hiển thị danh sách nhân viên (quản lí) 12](#_Toc200924788)

[6.3 Thêm/xoá nhân viên (quản lí) 14](#_Toc200924789)

[**III. Triển khai cài đặt và thiết kế cơ sở dữ liệu** 17](#_Toc200924790)

[*1. Class diagram* 17](#_Toc200924791)

[*2. Database design* 18](#_Toc200924792)

[*3. Thiết kế giao diện* 22](#_Toc200924793)

[*4. Cài đặt chức năng chính: Tổng quan Backend* 24](#_Toc200924794)

[4.1 Đăng kí 24](#_Toc200924795)

[4.2 Đăng nhập 25](#_Toc200924796)

[4.3 Cập nhật thông tin (nhân viên) 25](#_Toc200924797)

[4.4 Danh sách nhân viên (quản lí) 26](#_Toc200924798)

[4.5 Thêm/xoá nhân viên (quản lí) 26](#_Toc200924799)

# **Lời nói đầu**

Trong bối cảnh hội nhập kinh tế toàn cầu và sự phát triển vượt bậc của công nghệ thông tin, việc ứng dụng các giải pháp phần mềm vào quản lý doanh nghiệp đã trở thành một yếu tố then chốt giúp tối ưu hóa quy trình, nâng cao hiệu suất và tăng cường khả năng cạnh tranh. Đặc biệt, công tác quản lý nhân sự – một trong những trụ cột của mọi tổ chức – luôn đòi hỏi sự chính xác, hiệu quả và linh hoạt.

Nhận thức được tầm quan trọng đó, và với mong muốn vận dụng những kiến thức đã học từ môn **Kỹ thuật Phần mềm Ứng dụng**, nhóm chúng em đã tiến hành nghiên cứu và phát triển đề tài: **"HỆ THỐNG QUẢN LÝ NHÂN SỰ HiStaff”**. Dự án này tập trung vào việc xây dựng một ứng dụng web cơ bản nhằm hỗ trợ các tác vụ quản lý thông tin nhân viên, từ việc đăng ký, đăng nhập hệ thống, cập nhật thông tin cá nhân cho đến các chức năng quản lý danh sách nhân viên chi tiết dành cho cấp quản lý.

Báo cáo này trình bày một cách tổng quan và chi tiết về toàn bộ quá trình phát triển dự án, bao gồm: việc phân tích yêu cầu nghiệp vụ, thiết kế hệ thống (kiến trúc, cơ sở dữ liệu và giao diện), các kết quả triển khai chức năng thực tế, các giải pháp đã áp dụng để đảm bảo tính bảo mật và toàn vẹn dữ liệu, cũng như các phương pháp gỡ lỗi và kiểm thử đã thực hiện. Cuối cùng, báo cáo cũng đúc kết những bài học kinh nghiệm quý báu mà nhóm đã tích lũy được trong quá trình thực hiện dự án và đề xuất các hướng phát triển tiềm năng trong tương lai.

Chúng em xin chân thành cảm ơn **TS. Vũ Hải** đã tận tình hướng dẫn, hỗ trợ và cung cấp những kiến thức chuyên môn sâu sắc, giúp chúng em hoàn thành dự án này.

Kính mong nhận được sự đóng góp ý kiến từ thầy để báo cáo và dự án của chúng em được hoàn thiện hơn.

**Nhóm thực hiện**

**I. Giới thiệu chung**

*1. Giới thiệu thành viên*

|  |  |  |
| --- | --- | --- |
| Họ và tên | MSSV | Nhiệm vụ |
| Trịnh Long Vũ - Leader | 20223831 | - Lên kế hoạch, quản lý tiến độ công việc  - Hỗ trợ code Backend  - Tổng hợp báo cáo |
| Đinh Sỹ Đoàn | 20223910 | - Thiết kế Use Case, Activity Diagram  - Thiết kế và xây dựng cơ sở dữ liệu (Database)  - Hỗ trợ báo cáo |
| Vũ Minh Đức | 20233339 | - Xây dựng Frontend và Backend  - Triển khai các tính năng chính của hệ thống  - Hỗ trợ báo cáo |
| Nguyễn Tuấn Anh | 20223855 | - Bẫy lỗi, kiểm thử hệ thống  - Hỗ trợ báo cáo  - Thuyết trình sản phẩm |

*2. Mục tiêu dự án*

Mục tiêu chính của dự án là phát triển một hệ thống quản lý nhân sự cơ bản nhưng hiệu quả trên nền tảng web. Hệ thống này được thiết kế để hỗ trợ tối ưu hóa quy trình quản lý thông tin nhân viên, đảm bảo tính bảo mật, toàn vẹn dữ liệu và cung cấp một giao diện dễ sử dụng cho các đối tượng người dùng khác nhau.

* **Mục tiêu tổng quát:** Xây dựng một hệ thống ứng dụng web quản lý nhân sự, giúp tự động hóa các tác vụ quản lý thông tin nhân viên, phân quyền truy cập, và cung cấp cái nhìn tổng thể về nguồn lực con người trong tổ chức. Hệ thống này hướng tới việc nâng cao hiệu suất quản lý và cung cấp một nền tảng vững chắc cho việc mở rộng trong tương lai.
* **Mục tiêu cụ thể:** Để đạt được mục tiêu tổng quát, dự án tập trung vào việc triển khai các chức năng và yêu cầu sau:
* **Quản lý tài khoản người dùng:** Cho phép người dùng mới đăng ký tài khoản với các vai trò được phân quyền rõ ràng (Quản lý hoặc Nhân viên). Đồng thời, hệ thống phải cung cấp chức năng đăng nhập và đăng xuất an toàn để kiểm soát quyền truy cập.
* **Quản lý thông tin cá nhân:** Cung cấp cho nhân viên khả năng xem và tự cập nhật các thông tin cá nhân cơ bản của mình (như tên, ngày sinh, mật khẩu) một cách thuận tiện.
* **Quản lý danh sách nhân viên:** Hỗ trợ vai trò Quản lý trong việc xem tổng quan danh sách nhân viên, cùng với khả năng thêm mới hoặc xóa bỏ thông tin nhân viên khỏi hệ thống một cách hiệu quả.
* **Quản lý phúc lợi và vi phạm:** Hiển thị rõ ràng các phúc lợi mà nhân viên được hưởng và các vi phạm (nếu có), giúp quản lý có cái nhìn đầy đủ về tình hình nhân sự.
* **Đảm bảo tính bảo mật:** Áp dụng các kỹ thuật lập trình phòng ngừa như băm mật khẩu, sử dụng Prepared Statements để chống SQL Injection, và mã hóa đầu ra để ngăn chặn Cross-Site Scripting (XSS), bảo vệ dữ liệu người dùng khỏi các mối đe dọa phổ biến.
* **Đảm bảo tính toàn vẹn dữ liệu:** Thiết kế cơ sở dữ liệu với các ràng buộc khóa ngoại (Foreign Keys) và sử dụng các cơ chế như ON DELETE CASCADE để duy trì sự nhất quán và chính xác của dữ liệu trên toàn hệ thống.
* **Cung cấp giao diện thân thiện:** Xây dựng một giao diện người dùng trực quan, dễ thao tác và có tính thẩm mỹ, giúp người dùng dễ dàng tương tác với hệ thống.

## *3. Công nghệ và Môi trường phát triển*

Hệ thống được phát triển dựa trên sự kết hợp của các công nghệ web phổ biến và mạnh mẽ, cùng với một môi trường phát triển cục bộ hiệu quả.

* **Ngôn ngữ lập trình:**
* **Backend:** PHP (được lựa chọn vì tính linh hoạt, cộng đồng lớn và khả năng tương thích cao với MySQL).
* **Frontend:** HTML5 (cấu trúc nội dung), CSS (tạo phong cách và bố cục giao diện), JavaScript (tương tác người dùng và xử lý động).
* **Cơ sở dữ liệu:** MySQL (hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, mạnh mẽ và phổ biến).
* **Kiến trúc hệ thống:** Ứng dụng được xây dựng theo kiến trúc web cơ bản (Client-Server), nơi Frontend gửi các yêu cầu HTTP (GET/POST) đến Backend PHP. Backend xử lý logic nghiệp vụ, tương tác với CSDL MySQL, và trả về phản hồi dưới dạng HTML hoặc dữ liệu để Frontend hiển thị.
* **Môi trường phát triển cục bộ:**
* **Laragon:** Được sử dụng như một môi trường phát triển server ảo (WAMP/LAMP stack) trên máy tính cá nhân. Laragon cung cấp một cách nhanh chóng và dễ dàng để thiết lập máy chủ Apache/Nginx, MySQL, PHP, giúp quản lý các dịch vụ một cách trực quan và hỗ trợ tạo các tên miền ảo (pretty URLs) tự động, tối ưu hóa quá trình phát triển.

## *4. Phạm vi dự án*

Dự án này tập trung vào việc xây dựng các chức năng cốt lõi của một hệ thống quản lý nhân sự. Các chức năng đã triển khai trong phiên bản này bao gồm:

* Module đăng ký và đăng nhập người dùng với hai vai trò chính: Quản lý và Nhân viên.
* Chức năng xem và cập nhật thông tin cá nhân dành cho Nhân viên.
* Các chức năng quản lý danh sách nhân viên cơ bản cho Quản lý, bao gồm xem thông tin chi tiết, thêm nhân viên mới và xóa nhân viên.
* Hiển thị thông tin liên quan đến phúc lợi và vi phạm của từng nhân viên.

Trong phạm vi của dự án này, một số chức năng phức tạp hơn như quản lý chi tiết phúc lợi/vi phạm (thêm/sửa/xóa từng mục), module chấm công chuyên sâu, quản lý yêu cầu nghỉ phép, hệ thống bảng lương tự động, và các báo cáo thống kê phức tạp chưa được triển khai. Đây sẽ là những định hướng phát triển trong các phiên bản tương lai.

## *5. Các công cụ sử dụng*

Để hỗ trợ quá trình phát triển dự án, nhóm đã sử dụng một số công cụ chính:

* **Môi trường phát triển tích hợp (IDE):** Visual Studio Code (VS Code) – để viết và quản lý mã nguồn.
* **Thiết kế và mô hình hóa:**
* PlantUML – để tạo các biểu đồ UML như Use Case Diagram và Activity Diagram.
* MySQL Workbench – để thiết kế cấu trúc cơ sở dữ liệu (ERD) và quản lý dữ liệu.
* **Quản lý cơ sở dữ liệu:** MySQL Workbench.
* **Công cụ cộng tác:**
* GitHub – để quản lý mã nguồn, kiểm soát phiên bản và cộng tác nhóm hiệu quả.
* Zalo, Messenger – để liên lạc, trao đổi thông tin và tổ chức họp nhóm thường xuyên.

**II. Phân tích và thiết kế**

*1. Yêu cầu bài toán và hiện trạng*

Trong bối cảnh nền kinh tế số hóa và sự phát triển không ngừng của công nghệ thông tin, việc ứng dụng các giải pháp phần mềm vào quản lý doanh nghiệp đã trở thành một yếu yếu tố sống còn để tối ưu hóa hoạt động và nâng cao năng lực cạnh tranh. Đặc biệt, quản lý nguồn nhân lực là một trong những trụ cột quan trọng nhất của mọi tổ chức, bởi lẽ con người là tài sản quý giá nhất.

Hiện nay, nhiều doanh nghiệp, đặc biệt là các doanh nghiệp vừa và nhỏ, vẫn đang đối mặt với những thách thức đáng kể trong công tác quản lý nhân sự. Các phương pháp quản lý truyền thống như sử dụng bảng tính Excel, hồ sơ giấy tờ thủ công, hay trao đổi thông tin qua email bộc lộ nhiều hạn chế:

* **Thủ công và Tốn thời gian:** Việc nhập liệu, cập nhật thông tin nhân viên, phúc lợi, hay vi phạm một cách thủ công tiêu tốn rất nhiều thời gian và nguồn lực.
* **Khó khăn trong lưu trữ và truy xuất:** Dữ liệu phân tán, thiếu tập trung khiến việc lưu trữ, tìm kiếm và truy xuất thông tin trở nên phức tạp, dễ gây sai sót và trùng lặp.
* **Hạn chế về cập nhật và đồng bộ:** Khi có sự thay đổi về thông tin nhân viên hoặc các chính sách phúc lợi/vi phạm, việc cập nhật thủ công trên nhiều tài liệu khác nhau dễ dẫn đến sự không nhất quán và thiếu đồng bộ của dữ liệu.
* **Thiếu cơ chế phân quyền rõ ràng:** Các phương pháp thủ công thường không có cơ chế phân quyền chặt chẽ, tiềm ẩn rủi ro về bảo mật thông tin nhạy cảm và khó kiểm soát quyền truy cập của từng đối tượng.
* **Khó khăn trong thống kê và báo cáo:** Việc tổng hợp, phân tích dữ liệu để lập các báo cáo thống kê về nhân sự (ví dụ: số lượng nhân viên, phúc lợi được hưởng, vi phạm mắc phải) trở nên chậm chạp và kém hiệu quả, gây khó khăn cho việc ra quyết định của cấp quản lý.

Chính vì những lý do trên, nhu cầu về một **hệ thống web tập trung** nhằm quản lý nhân sự một cách nhanh chóng, chính xác và hiệu quả trở nên cấp thiết hơn bao giờ hết. Hệ thống này không chỉ giải quyết các vấn đề hiện tại mà còn tạo nền tảng cho sự phát triển bền vững của doanh nghiệp.

## *2. Lý do chọn đề tài*

Xuất phát từ yêu cầu thực tiễn và định hướng ứng dụng kiến thức vào giải quyết vấn đề, nhóm chúng em đã quyết định lựa chọn đề tài **"HỆ THỐNG QUẢN LÝ NHÂN SỰ HiStaff"** với những lý do chính sau:

* **Tính thiết thực và gần gũi:** Quản lý nhân sự là nghiệp vụ cốt lõi trong mọi loại hình doanh nghiệp, từ đó giúp đề tài có tính ứng dụng cao và gần gũi với môi trường làm việc thực tế. Việc giải quyết một vấn đề thực tiễn giúp chúng em hiểu sâu hơn về quy trình nghiệp vụ và cách phần mềm hỗ trợ doanh nghiệp.
* **Phù hợp với kiến thức chuyên môn:** Đề tài cho phép chúng em vận dụng một cách tổng hợp và thực hành các kiến thức đã được học trong môn **Kỹ thuật Phần mềm Ứng dụng** như:
* **Phân tích và thiết kế hệ thống:** Sử dụng các công cụ như UML (Use Case Diagram, Activity Diagram) để mô hình hóa yêu cầu và luồng xử lý.
* **Thiết kế cơ sở dữ liệu:** Áp dụng kiến thức về SQL để xây dựng và quản lý cơ sở dữ liệu quan hệ (ERD).
* **Lập trình web:** Thực hành lập trình Backend với PHP và Frontend với HTML, CSS, JavaScript.
* **Bảo mật và kiểm thử:** Triển khai các kỹ thuật lập trình phòng ngừa và kiểm thử để đảm bảo chất lượng phần mềm.
* **Khả năng mở rộng và phát triển:** Mặc dù trong phạm vi đồ án, hệ thống được xây dựng ở mức cơ bản, nhưng nó có nền tảng tốt để dễ dàng mở rộng các module phức tạp hơn trong tương lai như: quản lý chấm công, tính lương tự động, quản lý khen thưởng, hệ thống đánh giá hiệu suất, v.v., cho thấy tiềm năng phát triển lâu dài của dự án.
* **Tăng cường kỹ năng mềm và kỹ năng chuyên môn:** Quá trình thực hiện dự án không chỉ giúp củng cố kiến thức lập trình mà còn phát triển các kỹ năng mềm quan trọng như: làm việc nhóm (teamwork), phân công và quản lý công việc, giải quyết vấn đề, và thiết kế hệ thống một cách toàn diện.

Những yếu tố trên là động lực chính để nhóm chúng em thực hiện dự án này, với hy vọng mang lại một sản phẩm có giá trị ứng dụng thực tiễn và củng cố kiến thức chuyên môn đã học.

## *3. Mục đích đề tài và ý nghĩa thực tiễn*

Mục đích đề tài:

* Xây dựng hệ thống web cho:
* Cập nhật thông tin nhân viên
* Quản lý nghỉ phép, vi phạm, phúc lợi
* Phân quyền vai trò: nhân viên – quản lý
* Thiết kế cơ sở dữ liệu chuẩn hóa, dễ mở rộng, dễ bảo trì

Ý nghĩa thực tiễn:

* Giảm tải cho bộ phận nhân sự
* Dữ liệu tập trung, minh bạch, dễ tra cứu
* Có thể ứng dụng vào doanh nghiệp, phòng ban, tổ chức nhỏ
* Giúp sinh viên áp dụng kiến thức kỹ thuật phần mềm vào sản phẩm thật

## *4. Nền tảng sử dụng và các chức năng chính*

Nền tảng sử dụng- Web:

* Khả năng truy cập rộng rãi.
* Phát triển và bảo trì tương đối dễ dàng.
* Chi phí phát triển thường thấp hơn so với ứng dụng di động.
* Phù hợp với nhiều loại ứng dụng, từ đơn giản đến phức tạp.

Các tính năng chính:

* Cập nhật thông tin (Nhân viên)
* Hiển thị ds nhân viên (Quản lí)
* Thêm, xoá nhân viên (Quản lí)
* …

## *5. Sơ đồ PCCN*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Nội dung do AI tạo ra có thể không chính xác.

## *6. Một số Use Case*

### 6.1 Cập nhật thông tin (nhân viên)

Ảnh có chứa văn bản, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Use Case diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Sequence diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

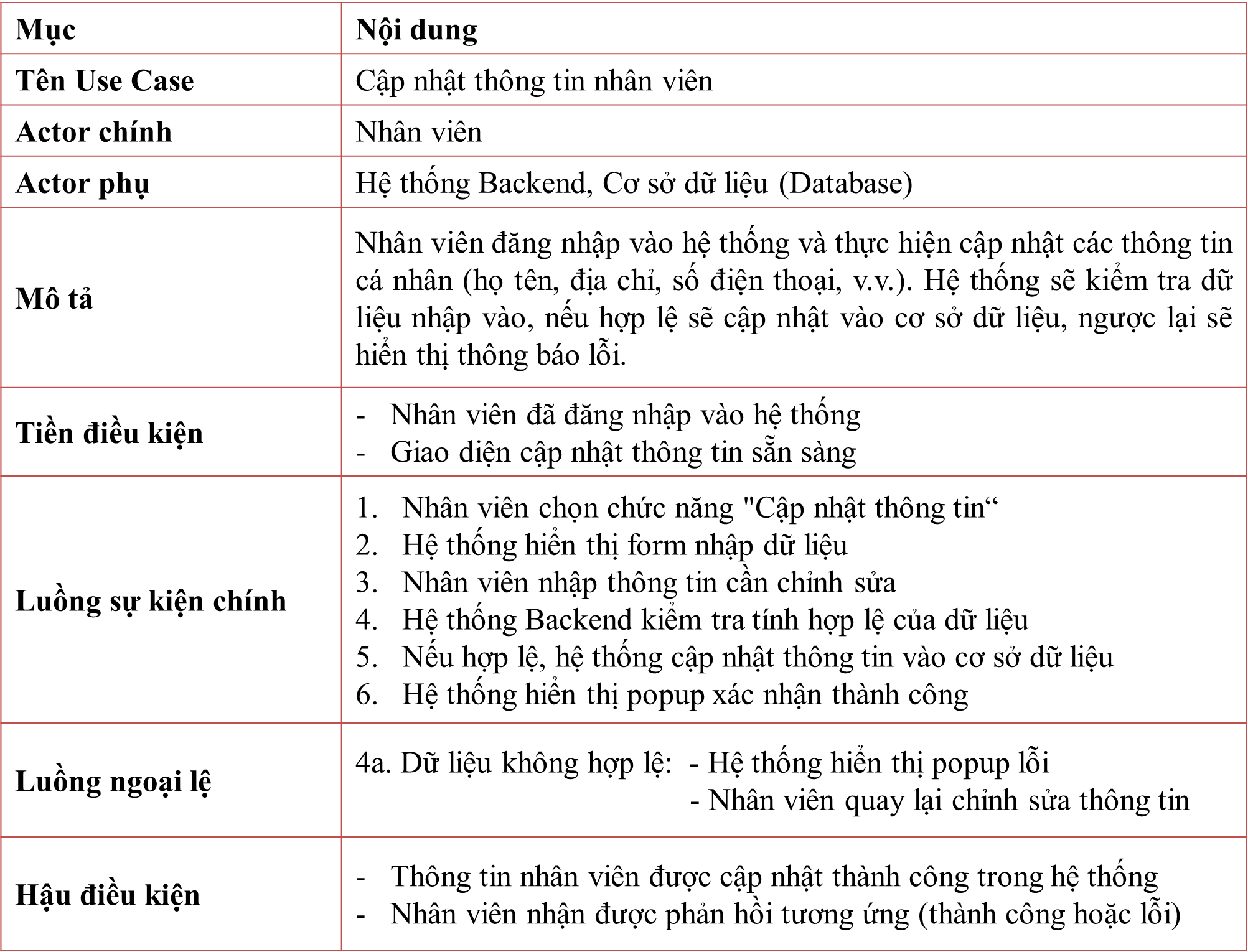
Nội dung do AI tạo ra có thể không chính xác.

Activity diagram

Ảnh có chứa văn bản, biểu đồ, biên lai, hàng

Nội dung do AI tạo ra có thể không chính xác.

Swimlane diagram



### 6.2 Hiển thị danh sách nhân viên (quản lí)

Ảnh có chứa văn bản, Phông chữ, biểu đồ, chữ viết tay

Nội dung do AI tạo ra có thể không chính xác.

Use Case diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, biểu đồ

Nội dung do AI tạo ra có thể không chính xác.

Sequence diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Activity diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Swimlane diagram

Ảnh có chứa ảnh chụp màn hình, màu đen, hàng

Nội dung do AI tạo ra có thể không chính xác.

### 6.3 Thêm/xoá nhân viên (quản lí)

Ảnh có chứa văn bản, hàng, biểu đồ, biên lai

Nội dung do AI tạo ra có thể không chính xác.

Use Case diagram

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

Sequence diagram

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Activity diagram

Ảnh có chứa văn bản, biểu đồ, biên lai, hàng

Nội dung do AI tạo ra có thể không chính xác.

Swimlane diagram

Ảnh có chứa ảnh chụp màn hình, hàng, Hình chữ nhật, Song song

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa ảnh chụp màn hình, màu đen, hàng

Nội dung do AI tạo ra có thể không chính xác.

**III. Triển khai cài đặt và thiết kế cơ sở dữ liệu**

*1. Class diagram*

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa ảnh chụp màn hình, hàng, màu đen, hình vuông

Nội dung do AI tạo ra có thể không chính xác.

RELATION SHIP

*2. Database design*

CREATE TABLE `role` (

`Id` int NOT NULL,

`Name` varchar(100) DEFAULT NULL,

PRIMARY KEY (`Id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `user` (

`Id` varchar(20) NOT NULL,

`Name` varchar(100) DEFAULT NULL,

`Email` varchar(100) DEFAULT NULL,

`Password` varchar(255) DEFAULT NULL,

`RoleId` int DEFAULT NULL,

`DateOfBirth` date DEFAULT NULL,

PRIMARY KEY (`Id`),

UNIQUE KEY `Email` (`Email`),

KEY `FK\_User\_Role` (`RoleId`),

CONSTRAINT `FK\_User\_Role` FOREIGN KEY (`RoleId`) REFERENCES `role` (`Id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `manager` (

`ManagerId` varchar(20) NOT NULL,

PRIMARY KEY (`ManagerId`),

CONSTRAINT `FK\_Manager\_User` FOREIGN KEY (`ManagerId`) REFERENCES `user` (`Id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `employee` (

`EmployeeId` varchar(20) NOT NULL,

`Salary` decimal(10,2) DEFAULT NULL,

`Workdays` int DEFAULT NULL,

PRIMARY KEY (`EmployeeId`),

CONSTRAINT `FK\_Employee\_User` FOREIGN KEY (`EmployeeId`) REFERENCES `user` (`Id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `violation` (

`ViolationID` varchar(20) NOT NULL,

`ViolationName` varchar(100) NOT NULL,

`Fine` decimal(10,2) DEFAULT NULL,

PRIMARY KEY (`ViolationID`),

UNIQUE KEY `ViolationName` (`ViolationName`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `welfare` (

`WelfareID` varchar(20) NOT NULL,

`WelfareName` varchar(100) NOT NULL,

`Bonus` decimal(10,2) DEFAULT NULL,

PRIMARY KEY (`WelfareID`),

UNIQUE KEY `WelfareName` (`WelfareName`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `employeewelfares` (

`EmployeeId` varchar(20) NOT NULL,

`WelfareID` varchar(20) NOT NULL,

PRIMARY KEY (`EmployeeId`,`WelfareID`),

KEY `FK\_employeewelfares\_welfare` (`WelfareID`),

CONSTRAINT `FK\_employeewelfares\_employee` FOREIGN KEY (`EmployeeId`) REFERENCES `employee` (`EmployeeId`) ON DELETE CASCADE,

CONSTRAINT `FK\_employeewelfares\_welfare` FOREIGN KEY (`WelfareID`) REFERENCES `welfare` (`WelfareID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `employeeviolations` (

`EmployeeId` varchar(20) NOT NULL,

`ViolationID` varchar(20) NOT NULL,

PRIMARY KEY (`EmployeeId`,`ViolationID`),

KEY `FK\_employeeviolations\_violation` (`ViolationID`),

CONSTRAINT `FK\_employeeviolations\_employee` FOREIGN KEY (`EmployeeId`) REFERENCES `employee` (`EmployeeId`) ON DELETE CASCADE,

CONSTRAINT `FK\_employeeviolations\_violation` FOREIGN KEY (`ViolationID`) REFERENCES `violation` (`ViolationID`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

CREATE TABLE `leaverequest` (

`LeaveRequestId` int NOT NULL AUTO\_INCREMENT,

`EmployeeId` varchar(20) NOT NULL,

`Reason` varchar(255) NOT NULL,

`FormDate` date NOT NULL,

`ToDate` date NOT NULL,

`Status` varchar(50) NOT NULL,

PRIMARY KEY (`LeaveRequestId`),

KEY `FK\_LeaveRequest\_Employee` (`EmployeeId`),

CONSTRAINT `FK\_LeaveRequest\_Employee` FOREIGN KEY (`EmployeeId`) REFERENCES `employee` (`EmployeeId`) ON DELETE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4\_0900\_ai\_ci

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, Song song

Nội dung do AI tạo ra có thể không chính xác.

EER DIAGRAM

*3. Thiết kế giao diện*

Ảnh có chứa văn bản, thiết kế đồ họa, thiết kế, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

MÀN HÌNH ĐĂNG NHẬP

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.

MÀN HÌNH ĐĂNG KÍ

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

GIAO DIỆN CỦA QUẢN LÍ

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

GIAO DIỆN CỦA NHÂN VIÊN

*4. Cài đặt chức năng chính: Tổng quan Backend*

**Ngôn ngữ lập trình:** PHP (VS Code)

**Cơ sở dữ liệu:** MySQL

**Môi trường phát triển cục bộ:** Laragon

**Kiến trúc:** Web ứng dụng cơ bản (Frontend gửi yêu cầu HTTP

(POST/GET) đến Backend PHP. Backend PHP xử lý logic, tương tác với MySQL, và trả về phản hồi HTML/JavaScript).

**Nguyên tắc bảo mật:**

* Sử dụng Prepared Statements để chống SQL Injection.
* Băm mật khẩu (hashing) trước khi lưu vào CSDL.
* Sử dụng session để quản lý trạng thái đăng nhập và phân quyền.

**Tương tác với CSDL:** Sử dụng thư viện MySQLi của PHP.

### 4.1 Đăng kí

***Mô tả Backend (signup.php):***

**Nhận dữ liệu:** Thu thập Tên, Email, Mật khẩu, Ngày sinh, Vai trò (Manager/Employee) từ form.

**Xác thực:**

* Kiểm tra tính khớp của mật khẩu.
* Đảm bảo Email chưa tồn tại trong hệ thống.

**Xử lý dữ liệu:**

* Tự động tạo ID duy nhất (EP... hoặc MN...) dựa trên vai trò.
* **Băm mật khẩu** (password\_hash()) để bảo mật.

**Lưu vào CSDL:**

* Sử dụng **Prepared Statements** để chèn thông tin vào bảng User.
* Nếu là Employee, tiếp tục chèn vào bảng Employee với lương và ngày công mặc định.

**Phản hồi:** Hiển thị thông báo thành công hoặc lỗi trên giao diện.

### 4.2 Đăng nhập

***Mô tả Backend (signin.php):***

**Nhận dữ liệu:** Thu thập Email và Mật khẩu từ form đăng nhập.

**Xác thực đầu vào:** Kiểm tra Email/Mật khẩu không rỗng và định dạng Email hợp lệ.

**Kiểm tra xác thực:**

* Truy vấn User theo Email bằng **Prepared Statement**.
* Dùng password\_verify() để so sánh mật khẩu nhập vào với mật khẩu đã băm lưu trong CSDL.

**Quản lý phiên:**

* Nếu xác thực thành công: Lưu user\_id và role vào $\_SESSION.
* Chuyển hướng người dùng đến trang phù hợp (Nhân viên/Quản lý).

**Phản hồi:** Hiển thị pop-up thông báo lỗi nếu xác thực thất bại.

### 4.3 Cập nhật thông tin (nhân viên)

***Mô tả Backend (update-information.php):***

**Xác thực truy cập:** Đảm bảo chỉ nhân viên đã đăng nhập mới có thể truy cập.

**Hiển thị dữ liệu cũ:** Lấy và hiển thị thông tin hiện tại (tên, ngày sinh) của nhân viên từ CSDL.

**Xử lý cập nhật:**

* Nhận Tên mới, Mật khẩu mới (tùy chọn), Xác nhận mật khẩu, Ngày sinh mới từ form.
* Kiểm tra Mật khẩu mới và Xác nhận mật khẩu có khớp nhau.
* **Băm mật khẩu** **mới** (password\_hash()) nếu có.
* Sử dụng **Prepared Statement** để cập nhật thông tin vào bảng User dựa trên ID của nhân viên.

**Phản hồi:** Hiển thị pop-up thông báo thành công hoặc lỗi trên giao diện.

### 4.4 Danh sách nhân viên (quản lí)

***Mô tả Backend (employee-list.php):***

**Xác thực truy cập:** Đảm bảo chỉ quản lý đã đăng nhập mới có thể xem.

**Truy vấn dữ liệu:**

* Lấy tất cả thông tin nhân viên (User và Employee) từ CSDL.
* Truy vấn riêng các phúc lợi và vi phạm của từng nhân viên.
* Sử dụng **Prepared Statements** cho các truy vấn có tham số.

**Hiển thị:**

* Tạo bảng hiển thị danh sách nhân viên với thông tin cơ bản.
* Cung cấp tính năng xem chi tiết thông tin nhân viên (bao gồm phúc lợi, vi phạm) thông qua một modal (pop-up) sử dụng JavaScript.

### 4.5 Thêm/xoá nhân viên (quản lí)

***Mô tả Backend (add-remove-employee.php):***

**Xác thực truy cập:** Đảm bảo chỉ quản lý đã đăng nhập mới có quyền.

**Thêm nhân viên:**

* Nhận thông tin (Tên, Email, Mật khẩu, Ngày sinh, Lương) từ form.
* Xác thực dữ liệu (không rỗng, email duy nhất, **băm mật khẩu**).
* Tạo ID duy nhất (EP...).
* **Sử dụng Transaction:** Chèn vào cả bảng User và Employee để đảm bảo toàn vẹn dữ liệu.

**Xóa nhân viên:**

* Nhận ID nhân viên cần xóa.
* Xóa bản ghi khỏi bảng User (sẽ tự động xóa khỏi Employee, EmployeeWelfares, EmployeeViolations nhờ ON DELETE CASCADE).

**Phản hồi:** Hiển thị thông báo thành công hoặc lỗi (qua pop-up/JS).

**IV. Kết quả thực hiện**

*1. Các chức năng đã triển khai & Liên kết CSDL*

***Chức năng chính****:* Đăng ký, Đăng nhập, Cập nhật thông tin, Quản lý nhân viên (Xem, Thêm, Xóa).

***Nhất quán CSDL:***

* Triển khai theo ERD với PHP và MySQL.
* Dữ liệu nhập/cập nhật lưu chính xác vào các bảng User, Employee...
* Sử dụng ON DELETE CASCADE đảm bảo toàn vẹn dữ liệu khi xóa.

*2. Gỡ rối và cơ chế bắt lỗi*

***Mục đích:*** Đảm bảo hệ thống ổn định và thân thiện người dùng.

***Cơ chế bắt lỗi:***

* Kết nối CSDL: Kiểm tra và thông báo lỗi kết nối ("Hệ thống đang bảo trì").
* Xác thực dữ liệu: Kiểm tra rỗng, định dạng, khớp mật khẩu, email trùng lặp.
* Lỗi nghiệp vụ/SQL: Phát hiện lỗi truy vấn (qua **Prepared Statements**), hiển thị cảnh báo alert() rõ ràng.

***Kỹ thuật gỡ rối:*** Sử dụng display\_errors và error\_log() để theo dõi lỗi.

*3. Lập trình phòng ngừa*

***Mục đích:*** Hạn chế lỗi và tăng cường bảo mật ứng dụng.

***Các kỹ thuật chính:***

* Chống SQL Injection: Sử dụng **Prepared Statements** cho mọi truy vấn dữ liệu đầu vào.
* Bảo mật mật khẩu: **Băm mật khẩu** (password\_hash()) khi lưu và xác minh an toàn (password\_verify()) khi đăng nhập.
* Quản lý phiên: Dùng **Session** ($\_SESSION) để kiểm soát trạng thái đăng nhập và phân quyền.
* Chống XSS: Sử dụng **htmlspecialchars()** khi hiển thị dữ liệu từ CSDL.
* Toàn vẹn dữ liệu: Thiết lập **ON DELETE CASCADE** trong CSDL để tự động xóa dữ liệu liên quan.

*4. Kiểm thử*

***Mục đích:*** Đảm bảo chất lượng và độ tin cậy của hệ thống.

***Các hình thức kiểm thử đã thực hiện (Thủ công):***

* Kiểm thử chức năng (Black-box Testing):
  + Kiểm tra hoạt động đúng của từng chức năng từ góc độ người dùng: Đăng ký/đăng nhập thành công/thất bại, cập nhật thông tin, thêm/xóa nhân viên, xem danh sách.
  + Kiểm tra các trường hợp dữ liệu hợp lệ, không hợp lệ, biên.
* Kiểm thử đơn vị/tích hợp (White-box/Integration Testing):
  + Kiểm tra các thành phần mã nguồn: Kết nối CSDL, logic băm/kiểm tra mật khẩu, tạo ID, xử lý các điều kiện if/else.
  + Kiểm tra sự phối hợp giữa Frontend và Backend, và giữa Backend với CSDL.
  + (Ví dụ: Sau khi đăng ký thành công, tài khoản có thể đăng nhập được không? Sau khi xóa, nhân viên có thực sự biến mất khỏi danh sách không?).

***Kết quả:*** Các chức năng chính đã hoạt động ổn định và đáp ứng yêu cầu.

**V. Kết luận và hướng phát triển tương lai**

*1. Kết luận*

***Chúng em đã học được:***

* Phân tích & Thiết kế: Nền tảng vững chắc giúp tiết kiệm thời gian, tránh lỗi lớn.
* Bảo mật mã nguồn: Hiểu biết Prepared Statements, băm mật khẩu, htmlspecialchars() là thiết yếu để chống SQL Injection, XSS.
* Quản lý phiên (Session): Quan trọng cho việc kiểm soát đăng nhập và phân quyền.
* Toàn vẹn dữ liệu: Sử dụng ON DELETE CASCADE đảm bảo dữ liệu sạch và nhất quán.
* Gỡ lỗi & Kiểm thử: Phát hiện và sửa lỗi sớm, chủ động kiểm thử giúp nâng cao chất lượng sản phẩm.
* Quản lý dự án: Chia nhỏ công việc giúp triển khai hiệu quả.

*2. Hướng phát triển tương lai*

***Các định hướng chính:***

* Mở rộng chức năng quản lý: Thêm module quản lý chi tiết phúc lợi, vi phạm, chấm công, nghỉ phép và tạo báo cáo thống kê.
* Cải thiện UX/UI: Nâng cấp giao diện hiện đại hơn, thiết kế responsive và sử dụng thông báo thân thiện (thay vì alert()).
* Nâng cao bảo mật: Triển khai xác thực 2 yếu tố (2FA) và hệ thống ghi log (audit log).
* Tối ưu hiệu suất: Đánh giá truy vấn CSDL, tích hợp AJAX để tối ưu tốc độ.
* Phát triển tính năng cho nhân viên: Cho phép xem lịch sử làm việc, thông tin lương, và hệ thống thông báo nội bộ.

**VI. Tài liệu tham khảo và liên quan**

1. Slide bài giảng Giảng viên Vũ Hải

2. <https://github.com/id140123/Project-KTPMUD-nhom-22>