# CHƯƠNG 1 TỔNG QUAN VỀ ĐỀ TÀI QUẢN LÝ ĐỒ ÁN SINH VIÊN KHOA CÔNG NGHỆ THÔNG TIN

Trong chương này, em trình bày tổng quan về đề tài: Mục đích và yêu cầu đặt ra khi xây dựng phần mềm quản lý đồ án sinh viên khoa Công nghệ thông tin. Quá trình khảo sát quản lý đồ án của khoa hiện nay, từ đó đưa ra phương pháp giải quyết bài toán quản lý đồ án sinh viên.

## 1.1 Mục đích, yêu cầu đặt ra khi xây dựng phần mềm quản lý đồ án.

### 1.1.1 Mục đích:

Dự án được xây dựng nhằm mục đích chính là cải thiện và tối ưu hóa quá trình quản lý đồ án sinh viên tại Khoa Công nghệ Thông tin. Em đã đặt ra các mục tiêu cụ thể:

* **Tăng Hiệu Quả Quản Lý:**

Hệ thống sẽ giúp các admin, khoa, bộ môn, giảng viên dễ dàng theo dõi và quản lý các đồ án sinh viên một cách hiệu quả.

* **Minh Bạch và Công Bằng:**

Hệ thống sẽ tạo ra một môi trường minh bạch, nơi mà giảng viên, bộ môn, khoa, admin có thể theo dõi tiến độ và đánh giá công bằng đồ án. Đồ án và các đánh giá sẽ được lưu trữ và quản lý một cách đáng tin cậy.

* **Tương Tác Tốt:**

Hệ thống sẽ hỗ trợ giao tiếp linh hoạt giữa sinh viên và giảng viên thông qua các chức năng như gửi tin nhắn, phản hồi trực tuyến.Sự tương tác này sẽ giúp cải thiện quá trình học tập và quản lý đồ án.

**1.1.2 Yêu cầu đặt ra:**

* **Người Dùng Thân Thiện:**

Giao diện người dùng phải đơn giản, dễ sử dụng, phù hợp với cả người dùng không chuyên về công nghệ.

* **Quản Lý Thông Tin Hiệu Quả:**

Hệ thống cần có khả năng quản lý thông tin đồ án, bao gồm tiến độ, tài liệu, và các thống kê thông báo liên quan.

## Bảo Mật Thông Tin:

## Đảm bảo an toàn thông tin cá nhân và đồ án của sinh viên, ngăn chặn truy cập trái phép.

## Khả Năng Mở Rộng:

## Hệ thống phải có khả năng mở rộng để đáp ứng nhu cầu tăng cường và thay đổi trong quản lý đồ án theo thời gian.

## Tương Thích Đa Nền Tảng:

## Hỗ trợ truy cập từ nhiều thiết bị và trình duyệt khác nhau, đảm bảo tính linh hoạt và tiện ích.

## 1.2 Khảo sát quy trình quản lý đồ án sinh viên khoa Công nghệ thông tin.

### 1.2.1 Quy trình nghiệp vụ:

* **Đăng kí: (Bộ môn, giảng viên)**
* **Đăng nhập và phân quyền: (Admin, khoa, bộ môn, giảng viên)**
* **Quản lý đăng kí đồ án (Bộ môn):**

Bộ môn sẽ tạo mới hoặc cập nhật đề tài hoặc thông tin sinh viên cùng giảng viên hướng dẫn

* **Phân công phản biện (Bộ môn)**
* **Quản lý hội đồng (Bộ môn):**

Thành lập hội đồng, nhập điểm hội đồng.

* **Thống kê (Bộ môn):**

## Danh sách hướng dẫn và phản biện theo đợt

## Danh sách sinh viên không đạt

## Thống kê thông tin giảng viên và hướng nghiên cứu

## Sốl lượng giảng viên hướng dẫn và phản biện của cả bộ môn

* **Cập nhật thông tin cá nhân, hướng nghiên cứu (Giảng viên)**
* **Xác nhận cho sinh viên HD (Giảng viên)**
* **Chấm điểm PB cho sinh viên (Giảng viên)**
* **Thống kê (Giảng viên)**

## Thông tin đề tài đã thực hiện theo: năm học/ nhiều năm học

## Tìm kiếm tên đề tài theo từ khóa

## Sinh viên HD/Phản biện theo năm học

### **Quản lý giao đề tài (Khoa):**

## Khoa sẽ duyệt hoặc không duyệt đề tài từng đợt cho từng sinh viên.

### **Quản lý hội đồng (Khoa):**

## Khoa duyệt hoặc không duyệt hội đồng do bộ môn thành lập theo từng đợt.

* **Thống kê (Khoa):**

## Thống kê chi tiết số lượng giảng viên hướng dẫn và số lượng giảng viên phản biện theo từng bộ môn.

## Thống kê chi tiết sinh viên bảo vệ thành công.

## Biểu đồ sinh viên bảo vệ thành công và khồn thành công theo từng bộ môn.

* **Quản trị (Admin):**

Quản lý bộ môn (sửa thông tin và tài khoản bộ môn, xem chi tiết bộ môn)

Quản lý giảng viên (sửa thông tin và tài khoản giảng viên, xem chi tiết giảng viên)

Cấu hình số lượng tối đa sinh viên hướng dẫn và sinh viên phản biện cho giảng viên.

* **Lưu trữ và quản lý thông tin:**

Kết quả đồ án và tài liệu liên quan được lưu trữ một cách cẩn thận để dễ dàng truy cập và tra cứu trong tương lai.

### 1.2.2 Những khó khăn và thuận lợi trong quá trình Quản lý đồ án cho khoa Công nghệ thông tin

* **Khó khăn:**
* Quản Lý Thời Gian Không Hiệu Quả:

Mô Tả: Khả năng quản lý thời gian không hiệu quả có thể dẫn đến việc sinh viên không đạt được các milestone trong kế hoạch.

* Tương Tác Hạn Chế:

Mô Tả: Thiếu sự tương tác giữa sinh viên và giảng viên có thể làm suy giảm chất lượng hướng dẫn và gây hiểu lầm trong quá trình thực hiện đồ án.

* Thiếu Tài Nguyên Nghiên Cứu:

Mô Tả: Sinh viên có thể gặp khó khăn khi không có đủ tài nguyên nghiên cứu hoặc không có sự hỗ trợ từ giảng viên trong việc xác định nguồn tài liệu.

* Khó Khăn Trong Bảo Vệ:

Mô Tả: Quá trình bảo vệ đồ án có thể là một thách thức, đặc biệt là nếu sinh viên gặp khó khăn trong việc thuyết trình và giải thích sản phẩm của mình.

* **Thuận lợi:**
* Hỗ Trợ Hướng Dẫn Tốt:

Mô Tả: Sự hỗ trợ chặt chẽ và hướng dẫn chi tiết từ giảng viên có thể giúp sinh viên dễ dàng vượt qua các khó khăn và tiến triển một cách hiệu quả.

* Quy Trình Đánh Giá Rõ Ràng:

Mô Tả: Sự minh bạch trong quy trình đánh giá giúp sinh viên hiểu rõ tiêu chí và đạt đến kỳ vọng của giảng viên.

* Cơ Hội Học Hỏi Tốt:

Mô Tả: Việc thực hiện đồ án mang lại cơ hội cho sinh viên học hỏi, nâng cao kỹ năng làm việc nhóm và tự quản lý.

* Hệ Thống Lưu Trữ Đồ Án Hiệu Quả:

Mô Tả: Sự tổ chức tốt và hệ thống lưu trữ đồ án hiệu quả giúp bảo vệ thông tin và kết quả của sinh viên.

## 1.3 Bài toán Quản lý đồ án sinh viên khoa Công nghệ thông tin.

* **Bài toán:**

Tích hợp hệ thống quản lý đồ án sinh viên khoa Công nghệ thông tin.

* **Mô tả:**
* Mục Tiêu:

Xây dựng một hệ thống quản lý đồ án sinh viên cho Khoa Công nghệ Thông tin nhằm tối ưu hóa quá trình theo dõi, đánh giá và quản lý đồ án của sinh viên.

* Chức Năng Cơ Bản:

- Quản Lý Thông Tin Sinh Viên:

Thêm, sửa, xóa thông tin sinh viên.

- Quản Lý Đồ Án:

Bộ môn thêm mới đồ án.

Xem và quản lý kết quả đồ án.

Thêm sửa xóa thông tin đồ án và thông tin sinh viên

Duyệt đồ án theo từng đợt.

- Phân Công Giảng Viên Phản Biện:

- Thành lập hội đồng, nhập điểm hội đồng cho sinh viên, duyệt hội đồng

- Thống kê

- Cập nhật thông tin cá nhân của giảng viên

- Xác nhận cho sinh viên hướng dẫn

- Chấm điểm phản biện cho sinh viên

- Quản lý thêm sửa bộ môn

- Quản lý thêm sửa thông tin giảng viên và tài khoản

- Cấu hình số lượng sinh viên hướng dẫn và bảo vệ cho từng giảng viên

* Tính Năng Mở Rộng:

- Tích Hợp Tìm Kiếm và Thống Kê:

- Cung cấp chức năng tìm kiếm thông tin sinh viên và đồ án.

- Tạo báo cáo thống kê về tiến độ, chất lượng, và đánh giá đồ án.

* Minh Bạch và Theo Dõi:

- Tăng cường minh bạch và theo dõi tiến độ đồ án từng sinh viên.

* Quản Lý Hiệu Quả:

- Hỗ trợ quản lý hiệu quả các hoạt động đồ án, giảm gánh nặng công việc cho giảng viên, bộ môn, khoa và quản trị viên.

* Tiện Lợi và Linh Hoạt:

- Cung cấp sự tiện lợi và linh hoạt trong việc quản lý thông tin và tương tác trực tuyến.

**1.4 Những ưu nhược điểm của phần mềm**

* Ưu điểm:
* Tăng hiệu quả quản lý: Phần mềm giúp tối ưu hóa quá trình quản lý thông tin đồ án, giúp tiết kiệm thời gian và công sức cho việc tổ chức và tìm kiếm thông tin.
* Tra cứu dễ dàng: Khả năng tra cứu nhanh chóng và dễ dàng giúp người dùng tìm kiếm thông tin cần thiết một cách nhanh chóng và hiệu quả.
* Theo dõi Tiến độ: Phần mềm cung cấp chức năng theo dõi tiến độ thực hiện các đồ án, giúp người quản lý nắm bắt được tình hình và đưa ra quyết định kịp thời.
* Hỗ trợ hợp tác: Khả năng tương tác trực tuyến và chia sẻ thông tin giữa các thành viên đội ngũ nghiên cứu giúp tăng cường khả năng hợp tác và giao tiếp.
* Báo cáo thống kê: Phần mềm cung cấp khả năng tạo ra các báo cáo thống kê về hoạt động nghiên cứu, giúp quản lý đưa ra quyết định dựa trên dữ liệu.
* Nhược điểm:
* Phụ thuộc vào Công nghệ: Phần mềm cần dựa vào công nghệ, do đó sự cố kỹ thuật hoặc sự thay đổi trong công nghệ có thể ảnh hưởng đến khả năng hoạt động của phần mềm.
* Thời gian Phát triển: Xây dựng và triển khai phần mềm mất khá nhiều thời gian để phát triển và đưa vào sử dụng.
* Khả năng Tương thích: Cần đảm bảo phần mềm tương thích với các hệ thống và thiết bị khác nhau để đảm bảo sự liên kết thông tin.
* Bảo mật và Quyền Truy cập: Đảm bảo an toàn thông tin và quản lý quyền truy cập có thể là một thách thức khá lớn.
* Khả năng Mở rộng và Tuỳ chỉnh: Hiện tài đề tài đang giới hạn ở một mức nhỏ nên khả năng mở rộng và tùy chỉnh sau khi đưa vào chạy thử nghiệm sẽ không tránh khỏi phát sinh ra những lỗi không mong muốn.

## 1.5 Kết luận chương 1

Chương 1 đã mở đầu cho dự án "Quản Lý Đồ Án Sinh Viên Khoa Công Nghệ Thông Tin" với mục tiêu xây dựng một hệ thống hiệu quả và minh bạch, hỗ trợ quản lý đồ án sinh viên. Bước đầu tiên trong quá trình nghiên cứu này là tìm hiểu sâu rộng về các thách thức hiện tại trong quản lý đồ án và những lợi ích mà một hệ thống quản lý có thể mang lại.

# CHƯƠNG 3: NGÔN NGỮ PHÁT TRIỂN HỆ THỐNG

## 3.1 Ngôn ngữ lập trình giao diện Angular.



*Hình 1: Hình ảnh ngôn ngữ lập trình Angular.*

* **Khái quát:**

Angular là một framework phát triển web được xây dựng và duy trì bởi đội ngũ của Google. Nó là một công cụ mạnh mẽ cho việc xây dựng ứng dụng web động và đa trang. Angular thường được sử dụng để phát triển các ứng dụng đơn trang (Single Page Applications - SPAs) nhờ vào khả năng tương tác cao và cấu trúc tổ chức linh hoạt.

* **Ngôn Ngữ Lập Trình:**

Angular sử dụng TypeScript, một siêu tập hợp của JavaScript, để viết mã nguồn. TypeScript cung cấp kiểu dữ liệu tĩnh và nhiều tính năng khác giúp giảm lỗi và tăng tính dễ bảo trì.

* **Cấu Trúc MVC:**

Angular tuân theo mô hình kiến trúc MVC (Model-View-Controller) hoặc chính xác hơn là MVVM (Model-View-ViewModel). Điều này giúp tạo ra ứng dụng dễ bảo trì và mở rộng.

* **Ưu điểm:**
* Kiến Trúc Mô-đun và Linh Hoạt:

Angular hỗ trợ kiến trúc mô-đun, giúp tạo ra ứng dụng dễ bảo trì và mở rộng.

* Dependency Injection (DI):

Hệ thống Dependency Injection giúp quản lý và tổ chức mã nguồn một cách hiệu quả, tăng tính linh hoạt và khả năng kiểm thử.

* Angular CLI - Tiện Ích Dòng Lệnh:

CLI giúp tạo và quản lý dự án một cách thuận tiện, tự động hóa nhiều công việc như tạo mới component, module, hoặc chạy test.

* Two-Way Data Binding:

Angular cung cấp two-way data binding giữa Model và View, giúp đồng bộ hóa dữ liệu một cách dễ dàng.

* RxJS và Observables:

Sử dụng RxJS giúp xử lý các luồng sự kiện và làm cho việc xử lý bất đồng bộ trở nên thuận lợi.

* TypeScript:

Sử dụng TypeScript giúp tăng tính rõ ràng và bảo mật trong mã nguồn.

* Hiệu Suất Cao:

Angular tối ưu hóa hiệu suất của ứng dụng thông qua các kỹ thuật như Ahead-of-Time (AOT) compilation.

* Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ:

Angular có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn và giải pháp cho các vấn đề phổ biến.

* Bảo Mật Cao:

Angular chú trọng vào bảo mật, với nhiều tính năng như Cross-Site Scripting (XSS) và Cross-Site Request Forgery (CSRF) protection.

* **Nhược điểm:**
* Khả Năng Học Cao Ban Đầu:

Angular có một độ dốc học khá cao, đặc biệt là đối với những người mới vào lập trình web.

* Kích Thước Ứng Dụng Lớn:

Các ứng dụng Angular có thể có kích thước lớn do cần đảm bảo nhiều tính năng và thư viện.

* Phức Tạp Cho Ứng Dụng Nhỏ:

Đối với các ứng dụng nhỏ, sử dụng Angular có thể cảm thấy quá nặng nề và không hiệu quả.

* Kéo Dài Thời Gian Phát Triển Ban Đầu:

Việc triển khai Angular có thể mất nhiều thời gian hơn so với một số framework khác.

* Chưa Hoàn Toàn Tương Thích Ngược:

Các phiên bản Angular mới có thể không hoàn toàn tương thích với các phiên bản cũ, đòi hỏi công sức cập nhật khi nâng cấp.

* Phần Đánh Giá SEO Khó Khăn:

Trong môi trường SEO, Angular có thể đối mặt với một số khó khăn, đặc biệt là đối với ứng dụng động và single-page.

## 3.2 Hệ quản trị cơ sở dữ liệu MySQL.



*Hình 2: Mysql*

* **Lịch Sử và Phát Triển:**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở được phát triển bởi một công ty Thụy Điển có tên là MySQL AB, sau đó được Sun Microsystems mua lại vào năm 2008, và hiện nay là một phần của Oracle Corporation. MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.

* **Mã Nguồn Mở:**

MySQL là một dự án mã nguồn mở, điều này có nghĩa là người dùng có thể sử dụng, sửa đổi và phân phối mã nguồn theo các điều khoản của Giấy phép Công cộng GNU (GPL). Điều này giúp tạo ra cộng đồng sáng tạo và tích cực quanh MySQL.

* **Đặc Điểm Chính:**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ, với sự hỗ trợ đầy đủ cho ngôn ngữ truy vấn SQL (Structured Query Language). Nó hỗ trợ nhiều hình thức lưu trữ dữ liệu như InnoDB, MyISAM, và MEMORY, mang lại sự linh hoạt cho các loại ứng dụng khác nhau.

* **Tính Nhẹ và Hiệu Quả:**

MySQL được thiết kế để làm việc hiệu quả và có hiệu suất cao trong các môi trường khác nhau, từ các dự án nhỏ đến các ứng dụng lớn và công nghiệp.

* **Hỗ Trợ Đa Nền Tảng:**

MySQL có sẵn cho nhiều nền tảng, bao gồm Linux, Windows, macOS và nhiều hệ điều hành khác. Điều này giúp cho việc triển khai và tích hợp dễ dàng trên nhiều môi trường khác nhau.

* **Hệ Thống Quyền Truy Cập:**

MySQL cung cấp một hệ thống quản lý quyền mạnh mẽ, cho phép quản trị viên kiểm soát quyền truy cập và thực hiện các biện pháp bảo mật.

* **Cộng Đồng Lớn và Hỗ Trợ Mạnh Mẽ:**

Có một cộng đồng sáng tạo và tích cực xung quanh MySQL, với nhiều tài liệu, diễn đàn trực tuyến, và nguồn lực giúp đỡ người dùng giải quyết vấn đề và tối ưu hóa sử dụng của họ.

* **Hỗ Trợ Nâng Cao:**

MySQL cung cấp nhiều tính năng nâng cao như truy vấn thời gian thực, replication, partitioning, và clustering để đáp ứng các yêu cầu phức tạp của các ứng dụng doanh nghiệp.

* **Sử Dụng Rộng Rãi:**

MySQL đã được sử dụng rộng rãi trong nhiều loại dự án và ứng dụng, từ các trang web cá nhân cho đến các hệ thống doanh nghiệp quy mô lớn.

* **Tương Thích và Mở Rộng:**

MySQL tương thích với nhiều ngôn ngữ lập trình và cung cấp các API và driver

để tương tác với nhiều loại ứng dụng.

* **Nhược Điểm của MySQL:**
* Khả Năng Mở Rộng Giới Hạn:

Trong môi trường có nhu cầu lưu trữ và truy xuất dữ liệu lớn, MySQL có thể gặp hạn chế về khả năng mở rộng so với một số hệ quản trị cơ sở dữ liệu mới.

* Chưa Hỗ Trợ Nhiều Tính Năng SQL Tiên Tiến:

So với một số hệ quản trị cơ sở dữ liệu khác, MySQL có thể không hỗ trợ một số tính năng SQL tiên tiến.

* Thiếu Các Tính Năng ACID Đầy Đủ:

Mặc dù MySQL hỗ trợ các tính năng ACID (Atomicity, Consistency, Isolation, Durability), nhưng trong một số trường hợp, nó có thể không cung cấp đầy đủ tính năng này so với các giải pháp khác.

* Quản Lý Đồng Thời Chưa Hiệu Quả Ở Một Số Tình Huống:

MySQL có thể gặp khó khăn khi quản lý đồng thời ở một số tình huống có nhiều ghi và đọc cùng lúc.

* Cập Nhật và Phát Triển Chậm Hơn Một Số Hệ Quản Trị Cơ Sở Dữ Liệu Mới:

Trong một số trường hợp, quá trình cập nhật và phát triển của MySQL có thể diễn ra chậm hơn so với một số hệ quản trị cơ sở dữ liệu mới hơn.

## 3.3 Ngôn ngữ lập trình back-end Spring boot



*Hình 3: Spring boot*

* Spring Framework:

Spring Boot là một dự án con của Spring Framework, một framework lập trình chính cho Java. Spring Framework cung cấp một cơ sở hạ tầng mạnh mẽ cho việc xây dựng các ứng dụng Java doanh nghiệp.

* Mục Tiêu Chính:

Spring Boot được phát triển với mục tiêu đơn giản hóa và tăng tốc quá trình phát triển ứng dụng Spring. Nó giúp giảm bớt cấu hình phức tạp và các tác vụ mặc định, đồng thời cung cấp các giải pháp mặc định để các vấn đề phổ biến.

* Ưu Điểm:
* Tiết Kiệm Thời Gian và Ghi Công:

Spring Boot giảm đáng kể việc cấu hình thông qua các giả mạo tự động và cung cấp các giải pháp mặc định cho nhiều vấn đề phổ biến.

* Tích Hợp Mạnh Mẽ:

Spring Boot tích hợp nhanh chóng với nhiều công nghệ và framework khác nhau, giúp giảm thời gian tích hợp và tăng tính linh hoạt.

* Tích Hợp Tốt với Spring Ecosystem:

Nó tích hợp tốt với các dự án khác của Spring như Spring Data, Spring Security, Spring Cloud, giúp xây dựng ứng dụng phức tạp một cách dễ dàng.

* Cộng Đồng Mạnh Mẽ:

Với sự hỗ trợ của cộng đồng lớn, Spring Boot có nhiều tài liệu, hướng dẫn và nguồn lực trực tuyến.

* Embedded Server:

Hỗ trợ các máy chủ nhúng như Tomcat, Jetty, Undertow, giúp triển khai ứng dụng mà không cần một máy chủ ngoại vi.

* Tự Động Cấu Hình:

Spring Boot sử dụng các giả mạo tự động để cấu hình ứng dụng, giảm độ phức tạp của các file cấu hình XML.

* Tính Năng:
* Auto-Configuration:

Tự động cấu hình dựa trên các thư viện và framework được thêm vào classpath.

* Spring Boot Starter:

Các Starter cung cấp dependencies và cấu hình mặc định cho các loại ứng dụng cụ thể.

* Spring Boot CLI:

Dòng lệnh CLI giúp phát triển Spring Boot dễ dàng hơn thông qua việc viết script và thực thi nhanh chóng.

* Spring Boot Actuator:

Cung cấp các tính năng giám sát và quản lý ứng dụng một cách dễ dàng.

* Spring Initializr:

Web-based tool giúp tạo và tùy chỉnh dự án Spring Boot một cách nhanh chóng.

* Sử Dụng Phổ Biến:

Spring Boot được sử dụng rộng rãi cho việc phát triển ứng dụng web, microservices, và các dự án Java khác.

* Nhược Điểm:
* Độ Phức Tạp Tăng Lên Với Ứng Dụng Lớn:

Dù Spring Boot giảm độ phức tạp cấu hình, nhưng với các ứng dụng lớn và phức tạp, có thể gặp phải sự khó khăn trong việc duy trì và quản lý.

* Kích Thước Ứng Dụng Lớn:

Ứng dụng Spring Boot có thể có kích thước lớn hơn so với một số framework nhẹ hơn, đặc biệt là khi sử dụng nhiều thư viện và dependencies.

* Thời Gian Khởi Động:

Việc khởi động ứng dụng Spring Boot có thể mất một khoảng thời gian, đặc biệt là khi có nhiều dependencies và phức tạp.

* Chưa Hỗ Trợ Hoàn Toàn Reactive Programming:

Mặc dù Spring Framework 5 đã hỗ trợ Reactive Programming, nhưng Spring Boot có thể không hoàn toàn tận dụng được các tính năng của nó trong mọi tình huống.

* Yêu Cầu Kiến Thức Spring Core:

Đối với những người mới bắt đầu với Spring Boot, kiến thức về Spring Core có thể là một ngưỡng khó khăn.

* Quá Nhiều Tính Năng Mặc Định:

Một số người phát triển có thể cảm thấy không thoải mái với việc sử dụng quá nhiều tính năng mặc định của Spring Boot, và họ muốn có sự linh hoạt lớn hơn trong việc tùy chỉnh.

* Kéo Dài Thời Gian Học:

Dù có nhiều tính năng giúp giảm thời gian phát triển, nhưng với những người mới học, quá trình học về Spring Boot có thể mất một khoảng thời gian đáng kể.

* Quản Lý Phiên Bản:

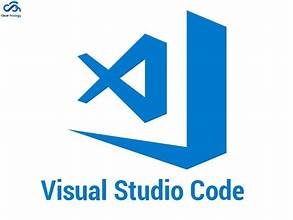
Việc quản lý phiên bản của Spring Boot và các dependencies có thể là một thách thức khi cần nâng cấp phiên bản.

## 3.4 Kết luận chương 3

Mục tiêu chính của chương giới thiệu là xác định ngữ cảnh và lý do nghiên cứu các ngôn ngữ lập trình. Chúng ta đã thấy rằng các ngôn ngữ lập trình không chỉ là một công cụ lập trình mạnh mẽ, mà còn mang đến sự đa dạng và linh hoạt trong việc giải quyết các vấn đề lập trình và phát triển ứng dụng.

# CHƯƠNG 4: CÔNG CỤ PHÁT TRIỂN HỆ THỐNG

## 4.1 Visual Studio Code



*Hình 4: VS Code*

* Visual Studio Code - Một Sự Nhìn Nhận Toàn Diện:

Visual Studio Code, thường được gọi là VS Code, là một trình soạn thảo mã nguồn mã nguồn mở và miễn phí, phát triển bởi Microsoft. Nó được thiết kế để đáp ứng nhu cầu phát triển phần mềm hiện đại và cung cấp một trải nghiệm phát triển mạnh mẽ và linh hoạt.

* Tính Năng Nổi Bật:
* Dễ Dàng Sử Dụng:

VS Code có giao diện người dùng thân thiện và dễ sử dụng, giúp những người mới bắt đầu và chuyên gia đều cảm thấy thoải mái.

* Hỗ Trợ Nhiều Ngôn Ngữ Lập Trình:

Với hỗ trợ đa dạng cho nhiều ngôn ngữ lập trình, từ JavaScript, Python, đến C# và Java, VS Code là sự lựa chọn đa năng cho các lập trình viên.

* Mở Rộng Bằng Extensions:

VS Code hỗ trợ nhiều tiện ích mở rộng, giúp tùy chỉnh và mở rộng chức năng của nó theo nhu cầu cụ thể của người dùng.

* Debugger Tích Hợp:

Debugger tích hợp mạnh mẽ giúp người phát triển dễ dàng theo dõi và sửa lỗi trong mã nguồn.

* Source Control Integration:

VS Code tích hợp chặt chẽ với các hệ thống quản lý phiên bản như Git, giúp quản lý mã nguồn dễ dàng.

* Hỗ Trợ Terminal Tích Hợp:

Terminal tích hợp giúp người phát triển thực hiện các lệnh hệ thống và quản lý dự án mà không cần chuyển sang ứng dụng bên ngoài.

* Hiệu Suất Cao và Nhẹ Nhàng:

VS Code được xây dựng với sự tập trung vào hiệu suất và tối ưu hóa tài nguyên, giúp chạy mượt mà trên nhiều hệ điều hành và cấu hình máy tính.

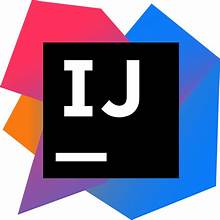
* Hỗ Trợ Cộng Đồng:

Với một cộng đồng lớn và tích cực, VS Code liên tục nhận được cập nhật, bổ sung tính năng mới và sửa lỗi, tạo nên một môi trường phát triển động và linh hoạt.

* Sử Dụng Rộng Rãi:

Visual Studio Code đã trở thành một trong những trình soạn thảo mã nguồn phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

## 4.2 IntelliJ IDEA



*Hình 5: IntelliJ IDEA*

* IntelliJ IDEA - Môi Trường Phát Triển Toàn Diện:

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) được phát triển bởi JetBrains. Nó nổi tiếng với sự linh hoạt và tính năng phong phú, hỗ trợ nhiều ngôn ngữ lập trình và công nghệ.

* Tính Năng Nổi Bật:
* Hỗ Trợ Nhiều Ngôn Ngữ:

IntelliJ IDEA không chỉ hỗ trợ Java, mà còn cung cấp tích hợp tốt với nhiều ngôn ngữ khác như Kotlin, Scala, Groovy, JavaScript, và nhiều ngôn ngữ khác.

* Refactoring Mạnh Mẽ:

IDE cung cấp nhiều công cụ tái cấu trúc mã nguồn (refactoring) mạnh mẽ giúp cải thiện cấu trúc và hiệu suất mã nguồn.

* Debugger và Profiler:

IntelliJ IDEA tích hợp debugger và profiler chất lượng cao, giúp dễ dàng theo dõi và tối ưu hóa mã nguồn.

* Hỗ Trợ Maven và Gradle:

Hỗ trợ tích hợp mạnh mẽ cho các công cụ quản lý dự án như Maven và Gradle, giúp quản lý các phụ thuộc và cấu hình dự án dễ dàng.

* Hỗ Trợ Version Control:

IntelliJ IDEA tích hợp tốt với hệ thống quản lý phiên bản như Git, SVN, Mercurial, cung cấp khả năng làm việc nhóm dễ dàng.

* Đồ Họa Giao Diện (GUI) Builder:

Cung cấp công cụ để xây dựng giao diện người dùng một cách nhanh chóng thông qua GUI Builder.

* Hiệu Suất Cao và Thân Thiện với Người Dùng:

IntelliJ IDEA được tối ưu hóa để chạy mượt mà và tiết kiệm tài nguyên, đồng thời có giao diện người dùng thân thiện, giúp người phát triển làm việc một cách hiệu quả.

* Hỗ Trợ Cộng Đồng:

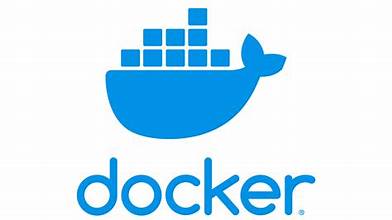
Cộng đồng IntelliJ IDEA rộng lớn, với nhiều plugin và hỗ trợ trực tuyến. JetBrains liên tục cập nhật IDE để đáp ứng nhu cầu mới và sửa lỗi.

* Sử Dụng Rộng Rãi:

IntelliJ IDEA là một trong những IDE phổ biến nhất và được sử dụng rộng rãi trong cộng đồng phát triển phần mềm, đặc biệt là trong lĩnh vực phát triển Java.

IntelliJ IDEA không chỉ là một công cụ phát triển mà còn là một nền tảng linh hoạt và mạnh mẽ, mang lại trải nghiệm lập trình chuyên nghiệp cho những người phát triển.

## 4.3 Docker



*Hình 6: Docker*

* Docker - Giải Pháp Container Hóa Hiệu Quả:

Docker là một nền tảng container hóa mở và được sử dụng rộng rãi, giúp đơn giản hóa việc triển khai ứng dụng, tạo môi trường phát triển và quản lý các container một cách hiệu quả.

* Tính Năng Nổi Bật:
* Containerization:

Docker sử dụng khái niệm container để đóng gói ứng dụng và tất cả các phụ thuộc của nó, giúp đảm bảo sự di động và khả năng chạy ổn định trên mọi môi trường.

* Khả Năng Di Động:

Containers Docker có thể chạy trên bất kỳ máy chủ hoặc môi trường nào hỗ trợ Docker, từ máy tính cá nhân đến các môi trường điện toán đám mây.

* Linh Hoạt và Tương Thích:

Docker có thể tích hợp với nhiều công nghệ và dịch vụ khác nhau, tạo nên một hệ sinh thái đa dạng và linh hoạt.

* Quản Lý Dễ Dàng:

Docker cung cấp các công cụ quản lý mạnh mẽ như Docker Compose để đơn giản hóa việc triển khai và quản lý các container.

* Hỗ Trợ Cộng Đồng Lớn:

Docker có một cộng đồng lớn, với hàng ngàn container và hình ảnh sẵn sàng cho việc sử dụng, cùng với tài liệu phong phú.

* Ứng Dụng Đa Dạng:

Docker được sử dụng rộng rãi trong các lĩnh vực như phát triển phần mềm, triển khai ứng dụng, kiểm thử, và cả trong các quy trình CI/CD.

* Hiệu Suất Cao và Nhẹ Nhàng:

Containers Docker chia sẻ nhân hệ điều hành và tài nguyên với máy host, giúp giảm thiểu overhead và tăng hiệu suất.

* Bảo Mật và Tính Cô Lập:

Docker cung cấp cơ chế cô lập và bảo mật cho từng container, đảm bảo an toàn khi triển khai ứng dụng.

* Đóng Gói Hóa và Phân Phối:

Docker giúp đóng gói ứng dụng và môi trường phát triển thành một container, giúp dễ dàng chia sẻ và triển khai mọi nơi.

Docker đã trở thành một công nghệ quan trọng trong việc tạo, triển khai, và quản lý ứng dụng, mang lại sự linh hoạt và hiệu quả trong quá trình phát triển và vận hành phần mềm.

## 4.4 Navicat



*Hình 7: Navicat*

* Navicat - Quản Lý Cơ Sở Dữ Liệu Đa Nền Tảng:

Navicat là một ứng dụng quản lý cơ sở dữ liệu đa nền tảng, được phát triển bởi PremiumSoft CyberTech Ltd. Nó cung cấp một giao diện đồ họa dễ sử dụng để quản lý và tương tác với các hệ quản lý cơ sở dữ liệu phổ biến như MySQL, PostgreSQL, Oracle, SQLite, và nhiều hệ quản lý khác.

* Tính Năng Nổi Bật:
* Hỗ Trợ Nhiều Cơ Sở Dữ Liệu:

Navicat hỗ trợ nhiều hệ quản lý cơ sở dữ liệu như MySQL, MariaDB, SQL Server, SQLite, PostgreSQL, Oracle, và nhiều loại khác.

* Giao Diện Người Dùng Đồ Họa:

Giao diện đồ họa trực quan giúp người sử dụng dễ dàng thao tác, thiết lập và quản lý cơ sở dữ liệu mà không cần sử dụng câu lệnh SQL.

* Tích Hợp Xem Dữ Liệu:

Cung cấp các công cụ xem dữ liệu và truy vấn dữ liệu một cách thuận tiện, giúp người sử dụng hiểu rõ cấu trúc và nội dung của cơ sở dữ liệu.

* Tích Hợp Nhập/Xuất Dữ Liệu:

Navicat hỗ trợ các tính năng nhập và xuất dữ liệu, giúp dễ dàng chuyển đổi dữ liệu giữa các hệ quản lý cơ sở dữ liệu khác nhau.

* Synchronization và Backup:

Cung cấp tính năng đồng bộ hóa dữ liệu giữa các cơ sở dữ liệu và hỗ trợ sao lưu dữ liệu một cách an toàn.

* Môi Trường Lập Trình SQL:

Navicat cung cấp một môi trường lập trình SQL chuyên nghiệp cho phép người sử dụng thực hiện các tác vụ phức tạp.

* Hỗ Trợ Đám Mây và Điều Khiển Phiên:

Hỗ trợ kết nối và quản lý cơ sở dữ liệu trên các nền tảng đám mây và điều khiển phiên làm việc đồng thời.

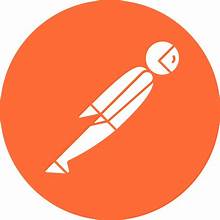
* Sử Dụng Rộng Rãi:

Navicat được sử dụng rộng rãi trong cộng đồng phát triển và quản trị cơ sở dữ liệu, đặc biệt là trong môi trường doanh nghiệp và dự án lớn.

* Bảo Mật và Quản Lý Phiên Bản:

Cung cấp các tính năng bảo mật mạnh mẽ và khả năng quản lý phiên bản để giúp người sử dụng duy trì tính toàn vẹn của dữ liệu.

## 4.5 Postman



*Hình 8: Postman*

* Postman - Công Cụ Kiểm Thử và Phát Triển API:

Postman là một ứng dụng giúp kiểm thử, phát triển và quản lý API (Application Programming Interface). Nó cung cấp một giao diện người dùng đồ họa để tạo, gửi, và kiểm thử các yêu cầu HTTP và HTTPS đối với các API.

* Tính Năng Nổi Bật:
* Xây Dựng và Gửi Yêu Cầu HTTP:

Postman cho phép người sử dụng dễ dàng xây dựng và gửi các yêu cầu HTTP và HTTPS với các phương thức khác nhau như GET, POST, PUT, DELETE.

* Quản Lý Biến Môi Trường và Collection:

Hỗ trợ quản lý biến môi trường và sưu tập (collection) để tổ chức và chia sẻ các yêu cầu liên quan.

* Kiểm Thử API:

Cung cấp các tính năng kiểm thử API mạnh mẽ như kiểm thử nhanh, kiểm thử đơn vị, và kiểm thử tích hợp.

* Automate và Kiểm Thử Tự Động:

Postman hỗ trợ tự động hóa các bước kiểm thử và cho phép tạo các bộ kiểm thử tự động.

* Giao Diện Người Dùng Thân Thiện:

Giao diện đồ họa thân thiện và dễ sử dụng giúp người phát triển và kiểm thử viên tận dụng hết các tính năng.

* Đồ Họa Hóa Dữ Liệu:

Cho phép đồ họa hóa dữ liệu trả về từ các yêu cầu API, giúp người sử dụng dễ dàng hiểu và xử lý dữ liệu.

* Chia Sẻ và Hợp Tác:

Hỗ trợ chia sẻ các bộ kiểm thử và yêu cầu thông qua link chia sẻ, giúp tăng cường hợp tác trong đội ngũ phát triển.

* Sử Dụng Đa Nền Tảng:

Postman có phiên bản trực tuyến và ứng dụng desktop, hỗ trợ nhiều nền tảng như Windows, macOS và Linux.

* Hỗ Trợ Đa Ngôn Ngữ:

Giao diện của Postman được hỗ trợ trong nhiều ngôn ngữ khác nhau, giúp người dùng từ các quốc gia và khu vực khác nhau sử dụng dễ dàng.

## 4.6 MySQL Workbench

## IMG_256

*Hình 9: MySQL Workbench*

* Thiết Kế Cơ Sở Dữ Liệu:

ER Modeling: Cho phép bạn tạo và quản lý mô hình thực thể-quan hệ (ER) một cách trực quan.

Tạo Bảng và Quan hệ: Dễ dàng tạo và quản lý các bảng, quan hệ và các ràng buộc khóa.

* Thực Thi Truy Vấn SQL:

Query Editor: Cung cấp một trình soạn thảo SQL mạnh mẽ với các tính năng như kiểm tra cú pháp, hoàn thành từ khóa và định dạng mã nguồn.

Execution Plan: Hiển thị kế hoạch thực hiện truy vấn để tối ưu hóa hiệu suất.

* Quản Lý Người Dùng và Quyền Truy Cập:

User Administration: Cho phép tạo, xóa và quản lý người dùng và quyền truy cập một cách dễ dàng.

Role Management: Hỗ trợ quản lý vai trò để đơn giản hóa việc quản lý quyền truy cập.

* Định Dạng và Xuất Dữ Liệu:

Export/Import Data: Cung cấp chức năng xuất và nhập dữ liệu từ và vào nhiều định dạng khác nhau như CSV, SQL, JSON.

Data Dump: Tạo các bản sao dự phòng hoặc dump dữ liệu một cách nhanh chóng.

* Quản Lý Kết Nối và Server:

Connection Management: Dễ dàng quản lý và kết nối đến các máy chủ MySQL khác nhau.

Server Administration: Cung cấp các công cụ quản lý máy chủ như kiểm tra trạng thái, quản lý thiết lập, và kiểm tra logs.

* Visualization Tools:

Database Relationship Graphs: Hiển thị mối quan hệ giữa các bảng trong một biểu đồ.

Performance Dashboard: Đánh giá và theo dõi hiệu suất của cơ sở dữ liệu.

* SQL Code Snippets:

Cung cấp một thư viện các đoạn mã SQL mẫu để tăng tốc quá trình phát triển.

* Công Cụ Điều Khiển Phiên Bản:

Hỗ trợ tích hợp với các hệ thống quản lý phiên bản như Git để theo dõi các thay đổi trong cơ sở dữ liệu.