**Hà Nội, 2025**

**LỜI CẢM ƠN**

Trong quá trình thực tập **tại Du12- khối G1 của CMC Global**, em xin chân thành cảm ơn anh mentor **Lê Thanh Đức** đã dành thời gian hướng dẫn và hỗ trợ em một cách tận tình. Anh không chỉ là người thầy mà còn là một người bạn, luôn sẵn sàng chia sẻ kiến thức và kinh nghiệm quý báu. Sự tận tâm và nhiệt huyết của anh đã giúp em hiểu rõ hơn về quy trình DevOps, từ việc thiết lập môi trường phát triển đến tự động hóa quy trình triển khai. Em đã học được cách sử dụng các công cụ như Docker để container hóa ứng dụng, giúp giảm thiểu sự khác biệt giữa môi trường phát triển và sản xuất. Em cũng đã thực hành thiết lập quy trình CI/CD vớiGilab\_CI, qua đó hiểu rõ hơn về các bước kiểm tra và triển khai tự động, giúp nâng cao hiệu suất làm việc.

Bên cạnh đó, em cũng xin gửi lời cảm ơn đến nhà trường và các giảng viên đã tạo điều kiện cho em được thực tập tại **CMC Global** và cung cấp kiến thức trong suốt thời gian học tập. Những bài giảng và chia sẻ của các thầy cô đã trang bị cho em nền tảng vững chắc để áp dụng vào thực tiễn giúp bản thân em đủ tự tin để thực tập tại **CMC Global**. Ngoài ra em còn biết ơn vì sự quan tâm, săn sóc của các thầy cô phụ trách chúng em trong quá trình thực tâp. Luôn hỏi thăm tiến độ hoàn thành dự án và hỗ trợ trả lời các câu hỏi về phía nhà trường.

**DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT (nếu có)**

|  |  |  |
| --- | --- | --- |
| STT | Ký hiệu viết tắt | Chữ viết đầy đủ |
| 1 | AWS | Amazon Web Services |
| 2 | IAM | Identity and Access Management |
| 3 | EC2 | Elastic Compute Cloud |
| 4 | ECS | Elastic Container Service |
| 5 | S3 | Simple Storage Service |
| 6 | VPC | Virtual Private Cloud |
| 7 | ELB | Elastic Load Balancer |
| 8 | RDS | Relational Database Service |
| 9 | CI/CD | Continuous Integration/Continuous Deployment |
| 10 | IaaS | Infrastructure as a Service |
| 11 | PaaS | Platform as a Service |
| 12 | Saas | Sofware as a Service |
| 13 | ECS | Elastic Container Service |
| 14 | EFS | Elastic File System |
| 15 | DB | Database |
| 16 | BE | Backend |
| 17 | FE | Frontend |

**DANH MỤC CÁC BẢNG, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH (nếu có)**

**MỤC LỤC**

​​

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

**​**

​**​**

**PHẦN MỞ ĐẦU: GIỚI THIỆU**

1. **Giới thiệu.**
2. **Lý do chọn đơn vị thực tập**

* **Vị trí thực tập: Thực tập sinh Devops**
* **Giới thiệu về devops:**
* **DevOps** là viết tắt của Development (Dev) và Operations (Ops). DevOps là một văn hóa làm việc kết hợp giữa kỹ sư phát triển phần mềm (dev) với bộ phận operator (kỹ sư hệ thống, nhân viên bảo mật, kỹ sư mạng, kỹ sư hạ tầng,...) nhằm mục đích rút ngắn vòng đời phát triển sản phẩm (SDLC).
* **Sự phát triển của devops:**
* Chuyên ngành DevOps đang trải qua sự bùng nổ mạnh mẽ với nhiều tiềm năng phát triển trong tương lai. Nhu cầu tăng cao từ các doanh nghiệp muốn phát hành sản phẩm nhanh chóng đã thúc đẩy việc áp dụng DevOps, giúp rút ngắn thời gian từ phát triển đến triển khai.
* Sự phát triển của các công cụ tự động hóa như Jenkins và Docker tối ưu hóa quy trình, giảm thiểu lỗi và nâng cao chất lượng phần mềm. Đồng thời, việc tích hợp trí tuệ nhân tạo và Machine Learning trong DevOps giúp dự đoán và giải quyết sự cố nhanh chóng, nâng cao hiệu suất.
* DevOps cũng thúc đẩy sự hợp tác chặt chẽ giữa các bộ phận, cải thiện quy trình làm việc và tối ưu hóa nguồn lực. Với nhu cầu gia tăng về chuyên gia DevOps, lĩnh vực này đang mở ra nhiều cơ hội việc làm hấp dẫn.
* Với những yếu tố này, DevOps không chỉ là một xu hướng tạm thời mà là một yếu tố then chốt trong chiến lược phát triển bền vững của các tổ chức.
* **Lý do chọn vị trí thực tập:**  Vì trong bối cảnh công nghệ thông tin ngày càng phát triển, **DevOps** nổi lên như một lĩnh vực quan trọng và đầy tiềm năng. Dưới đây là những lý do khiến em nên chọn DevOps để thực tập:
* **Nhu cầu cao trên thị trường:** DevOps đang trở thành một trong những lĩnh vực có nhu cầu lao động lớn nhất trong ngành công nghệ thông tin, mở ra nhiều cơ hội việc làm hấp dẫn.
* **Kỹ năng đa dạng:** Thực tập trong DevOps giúp em vừa phát triển nhiều kỹ năng quan trọng như lập trình, quản lý hệ thống, tự động hóa và làm việc với các công cụ hiện đại.
* **Tăng khả năng thích nghi:** DevOps khuyến khích sự đổi mới và sáng tạo, cho phép tham gia vào các dự án thú vị và thử thách.
* **Tích hợp công nghệ mới:** Bạn sẽ được làm quen với các công nghệ tiên tiến như Docker, CI/CD, giúp ta trở thành ứng viên cạnh tranh hơn trong tương lai.
* **Cải thiện khả năng hợp tác:** DevOps thúc đẩy giao tiếp và hợp tác giữa các nhóm phát triển và vận hành, giúp phát triển kỹ năng làm việc nhóm quan trọng.
* **Tham gia vào quá trình chuyển đổi số:** Làm việc trong DevOps cho phép đóng góp vào các dự án chuyển đổi số, giúp doanh nghiệp thích ứng nhanh hơn với th đổi.

1. **Mối quan hệ giữa Devops và ngành Công nghệ thông tin**: DevOps và ngành công nghệ thông tin (CNTT) có một mối quan hệ chặt chẽ và tương hỗ, dưới đây là những khía cạnh nổi bật của mối quan hệ này:

* **Cải thiện quy trình phát triển phần mềm:** DevOps tối ưu hóa quy trình phát triển, giúp các đội ngũ CNTT phát hành phần mềm nhanh chóng và hiệu quả hơn.
* **Tự động hóa và hiệu suất:** Trong CNTT, việc xử lý khối lượng công việc lớn là rất phổ biến. DevOps sử dụng tự động hóa để giảm thiểu thao tác thủ công, tăng cường hiệu suất và độ chính xác.
* **Hợp tác liên phòng ban:** DevOps thúc đẩy sự hợp tác giữa các nhóm phát triển, vận hành và kiểm thử trong CNTT, giúp giảm thiểu xung đột và cải thiện quy trình làm việc.
* **Khả năng thích ứng với thay đổi:** Ngành CNTT thường xuyên phải đối mặt với sự thay đổi nhanh chóng. DevOps giúp tổ chức thích ứng linh hoạt hơn với các công nghệ và yêu cầu mới.
* **Sử dụng công nghệ tiên tiến:** DevOps thường áp dụng các công nghệ mới như Docker, Kubernetes, và Cloud Computing, điều này giúp ngành CNTT duy trì tính cạnh tranh và hiện đại hóa hạ tầng.
* **Quản lý rủi ro hiệu quả:** DevOps cải thiện khả năng phát hiện và xử lý sự cố, từ đó giảm thiểu rủi ro trong việc triển khai sản phẩm và dịch vụ CNTT.
* **Thúc đẩy chuyển đổi số:** DevOps là một phần quan trọng trong chiến lược chuyển đổi số của nhiều tổ chức CNTT, giúp họ tận dụng công nghệ để cải thiện hiệu suất và đáp ứng nhu cầu thị trường**.**

1. **Mục tiêu**
2. **Mục tiêu chuyên môn**

* **Phát triển kỹ năng kỹ thuật:**
* Học
* Nắm vững quy trình CI/CD (Continuous Integration/Continuous Deployment**).**
* **Tìm hiểu về tự động hóa:**
* Thực hành tự động hóa quy trình phát triển và triển khai phần mềm.
* Xây dựng và tối ưu hóa các script tự động hóa.
* **Cải thiện khả năng làm việc nhóm:**
* Tham gia vào các dự án nhóm, học cách giao tiếp hiệu quả với các bộ phận khác nhau (phát triển, vận hành, QA).
* Tham gia các buổi họp và thảo luận để đóng góp ý kiến.
* **Tham gia vào quy trình phát triển sản phẩm:**
* Theo dõi và hỗ trợ các giai đoạn phát triển phần mềm từ lập kế hoạch đến triển khai.
* Học cách quản lý và giải quyết các sự cố phát sinh trong quá trình triển khai.
* **Tìm hiểu về quản lý dự án:**
* Tham gia vào việc lập kế hoạch và theo dõi tiến độ dự án.
* **Nâng cao kiến thức về bảo mật:**
* Tìm hiểu các nguyên tắc bảo mật trong phát triển và vận hành phần mềm.
* Học cách áp dụng các biện pháp bảo mật vào quy trình DevOps.

1. **Mục tiêu cá nhân**

* **Phát triển kỹ năng tự học:**
* Tìm hiểu tài liệu, khóa học trực tuyến và video hướng dẫn để nâng cao kiến thức về DevOps.
* **Xây dựng mạng lưới chuyên nghiệp:**
* Kết nối với các đồng nghiệp, mentor và chuyên gia trong lĩnh vực để tạo dựng mối quan hệ.
* **Cải thiện kỹ năng giao tiếp:**
* Luyện tập khả năng trình bày ý tưởng và thảo luận trong nhóm để trở thành một thành viên tích cực.
* **Tìm hiểu về văn hóa DevOps:**
* Nắm bắt các nguyên tắc và giá trị cốt lõi của DevOps để hiểu rõ hơn về môi trường làm việc.
* **Thiết lập mục tiêu nghề nghiệp dài hạn:**
* Xác định lộ trình phát triển nghề nghiệp trong lĩnh vực DevOps và công nghệ thông tin.

1. **Thông tin tổng quan đơn vị thực tập**

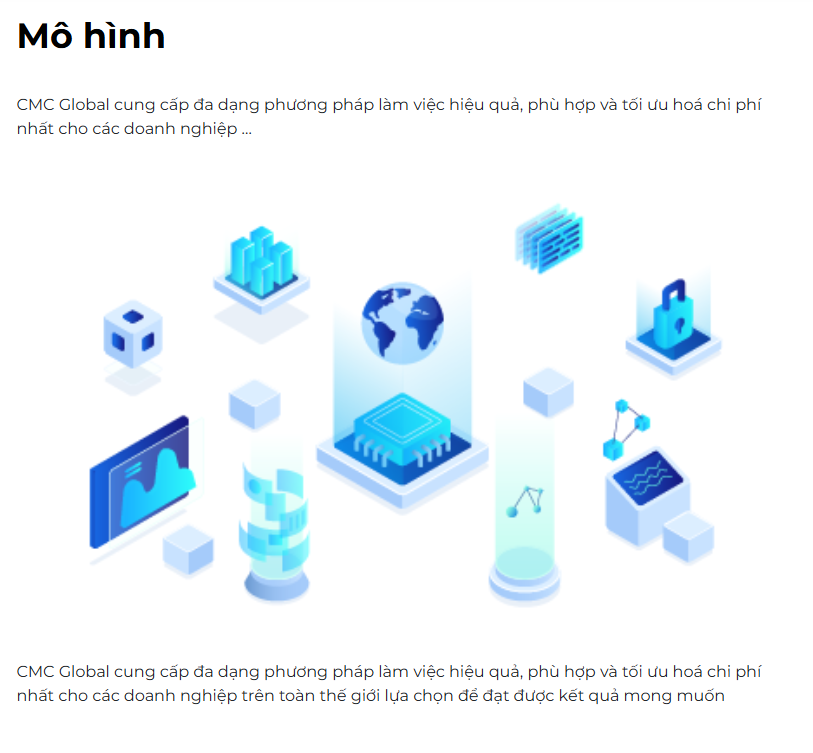
1. **Lịch sử thành thành và phát triển**

**Đơn vị thực tập: CMC GLOBAL**   
**CMC Global** là thành viên của Tập đoàn CMC được thành lập với khát vọng đưa các sản phẩm, giải pháp và dịch vụ CNTT-TT của Việt Nam ra thị trường quốc tế.

Được thành lập vào tháng 3 năm 2017 như một bước đi chiến lược của Tập đoàn để bước ra thị trường toàn cầu, tự hào là công ty hàng đầu tại Việt Nam trong việc cung cấp nhiều Giải pháp & Dịch vụ CNTT toàn diện, từ các dịch vụ Truyền thống đến Chuyển đổi Kỹ thuật số, cũng như Tư vấn cho các tổ chức trên toàn thế giới.



1. **Quy mô, cơ cấu tổ chức**

**Tiêu biểu phải kể đến các mô hình:**

* **Project-based**

Mô hình này tập trung vào việc hoàn thành một dự án cụ thể với mục tiêu, thời gian, và ngân sách xác định. Các đặc điểm chính bao gồm:

Định hướng cụ thể: Mỗi dự án có mục tiêu rõ ràng và các kết quả mong muốn.

Thời gian giới hạn: Dự án có thời gian bắt đầu và kết thúc cụ thể.

Quản lý tài nguyên: Tài nguyên (nhân lực, vật tư, tài chính) được phân bổ cho dự án trong khoảng thời gian nhất định.

Báo cáo và đánh giá: Kết quả được đánh giá dựa trên các chỉ tiêu đã đề ra.

* **Staff Augmentation**

Mô hình này cho phép doanh nghiệp bổ sung nhân lực tạm thời để đáp ứng nhu cầu dự án mà không cần tuyển dụng lâu dài. Các đặc điểm chính bao gồm:

Tăng cường nhân lực: Doanh nghiệp có thể thuê thêm chuyên gia hoặc nhân viên theo thời vụ.

Linh hoạt: Dễ dàng điều chỉnh số lượng nhân viên tùy thuộc vào yêu cầu dự án.

Tiết kiệm chi phí: Giảm thiểu chi phí cho việc tuyển dụng và đào tạo nhân viên mới.

Chuyên môn hóa: Tìm kiếm các chuyên gia có kỹ năng cụ thể cho dự án.

* **Hybrid**

Mô hình Hybrid kết hợp các yếu tố từ cả Project-based và Staff Augmentation. Các đặc điểm chính bao gồm:

Linh hoạt trong quản lý: Kết hợp giữa việc quản lý dự án và tăng cường nhân lực.

Tối ưu hóa nguồn lực: Doanh nghiệp có thể sử dụng cả nhân viên nội bộ và thuê ngoài tùy theo nhu cầu.

Phù hợp với nhiều loại dự án: Có thể áp dụng cho các dự án có quy mô và tính chất khác nhau.

Quản lý hiệu quả: Dễ dàng điều chỉnh dự án và nhân lực để đạt được kết quả tốt nhất.

* **Results-based**

Mô hình này tập trung vào kết quả cuối cùng của dự án thay vì quy trình thực hiện. Các đặc điểm chính bao gồm:

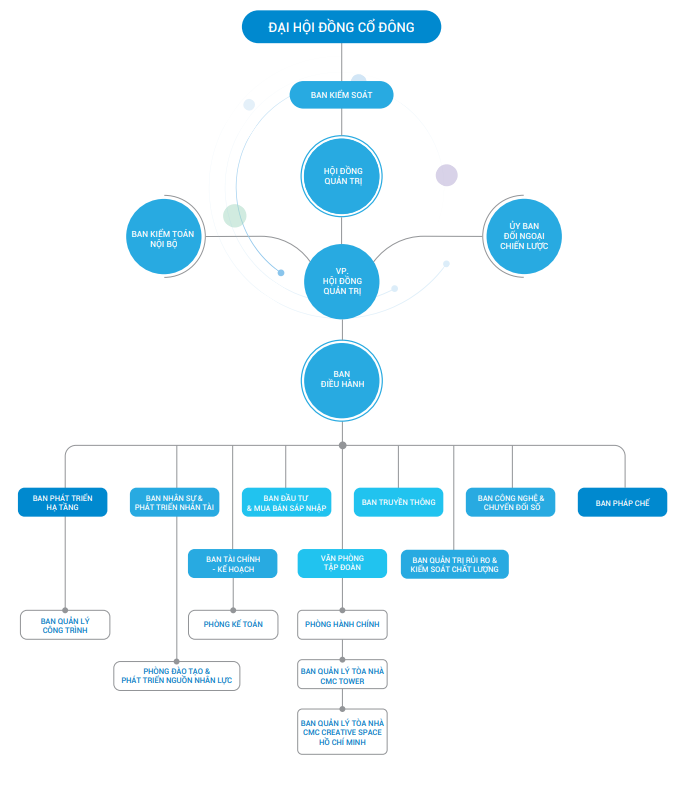
Đánh giá theo kết quả: Thành công được đo lường dựa trên kết quả đạt được.

Khuyến khích hiệu suất: Nhân viên và nhóm làm việc được khuyến khích tạo ra giá trị cao.

Tính minh bạch: Rõ ràng về các mục tiêu và tiêu chí đánh giá.

Tập trung vào giá trị: Chú trọng đến việc tạo ra giá trị cho khách hàng và doanh nghiệp.

**Cơ cấu tổ chức**



1. **Lĩnh vực hoạt động: CMC GLOBAL hoạt động trong nhiều lĩnh vực như:**

* **Tài Chính Ngân Hàng**

Cung cấp các giải pháp công nghệ cho các ngân hàng và tổ chức tài chính.

Tích hợp hệ thống thanh toán, quản lý rủi ro, và dịch vụ ngân hàng số.

* **Công Nghệ Ô Tô**

Phát triển phần mềm và giải pháp công nghệ cho ngành công nghiệp ô tô.

Hỗ trợ các ứng dụng như quản lý sản xuất, tự động hóa và kết nối xe.

* **Logistics**

Cung cấp giải pháp tối ưu hóa chuỗi cung ứng và quản lý vận chuyển.

Tích hợp công nghệ để theo dõi và quản lý hàng hóa.

* **Sản Xuất**

Giải pháp công nghệ cho quy trình sản xuất, bao gồm tự động hóa và quản lý chất lượng.

Hỗ trợ chuyển đổi số trong các nhà máy và xưởng sản xuất.

* **Y Tế**

Phát triển phần mềm quản lý bệnh viện, hệ thống hồ sơ bệnh án điện tử và ứng dụng chăm sóc sức khỏe.

Tích hợp công nghệ để nâng cao chất lượng dịch vụ y tế.

* **Truyền Thông & Giải Trí**

Cung cấp giải pháp công nghệ cho các nền tảng truyền thông và giải trí.

Hỗ trợ phát triển nội dung số, truyền thông đa phương tiện và ứng dụng giải trí.

* **Công Nghệ Tài Chính**

Phát triển các giải pháp fintech, bao gồm ví điện tử, đầu tư trực tuyến và quản lý tài sản.

Hỗ trợ các dịch vụ tài chính tiện ích và an toàn cho người dùng.

1. **Sản phẩm và dịch vụ chính của đơn vị thực tập**

**CMC Global** không chỉ là một trong ba doanh nghiệp gia công phần mềm hàng đầu tại Việt Nam mà còn là biểu tượng cho sự chuyển mình mạnh mẽ trong kỷ nguyên số. Với tầm nhìn chiến lược và các giải pháp công nghệ tiên tiến, công ty đang khẳng định vị thế của mình trên thị trường quốc tế. Dưới đây là những sản phẩm và dịch vụ nổi bật mà CMC Global cung cấp, mang đến cho khách hàng những trải nghiệm công nghệ đột phá:

* **Dịch vụ Gia công Xuất khẩu Phần mềm**
* **Phát triển và Bảo trì Phần mềm**: Giải pháp tùy chỉnh cho doanh nghiệp, từ phát triển ứng dụng mới đến bảo trì hệ thống hiện tại.
* **Kiểm thử Phần mềm**: Đảm bảo chất lượng sản phẩm thông qua quy trình kiểm thử chuyên nghiệp và hiệu quả.
* **Chuyển đổi Hệ thống**: Hỗ trợ chuyển đổi công nghệ thông tin một cách liền mạch, giúp doanh nghiệp tối ưu hóa quy trình làm việc.
* **Giải pháp Chuyển đổi Số (DX)**
* **Công nghệ Đám mây (Cloud)**: Đối tác tư vấn của Amazon Web Services (AWS), cung cấp giải pháp cloud giúp doanh nghiệp vận hành hiệu quả hơn.
* **Tự động hóa Quy trình Kinh doanh (BPA)**: Giải pháp tự động hóa giúp tăng năng suất và giảm thiểu chi phí hoạt động.
* **Phân tích Dữ liệu (Data Analytics)**: Công cụ phân tích hỗ trợ doanh nghiệp đưa ra quyết định dựa trên dữ liệu chính xác.
* **Giải pháp Công nghệ Mới**
* **AI và Dữ liệu Lớn**: Ứng dụng trí tuệ nhân tạo và dữ liệu lớn để giải quyết các thách thức trong quản lý và vận hành.

IoT (Internet of Things): Cung cấp giải pháp IoT cho các lĩnh vực như quản lý chuỗi cung ứng và thương mại điện tử, tối ưu hóa quy trình và nâng cao trải **nghiệm người dùng.**

* **Trung tâm Phát triển Phần mềm Quốc tế (GDC)**
* Với ba GDC tại Hà Nội, TP.HCM và Đà Nẵng, CMC Global đảm bảo chất lượng dịch vụ cao với các chứng chỉ bảo mật thông tin tiêu chuẩn quốc tế.

5. Dịch vụ Hỗ trợ Khách hàng

* CMC Global cam kết cung cấp dịch vụ tận tâm, hỗ trợ khách hàng từ giai đoạn lập kế hoạch đến triển khai, đảm bảo sự hài lòng và thành công trong từng dự án.

Với những sản phẩm và dịch vụ đa dạng này, CMC Global không ngừng nỗ lực nâng cao chất lượng và mở rộng thị trường, tạo ra giá trị gia tăng cho khách hàng trên toàn cầu.

1. **Văn hóa doanh nghiệp**

**PHẦN NỘI DUNG VÀ KẾT QUẢ THỰC TẬP**

**Chương 1: giới thiêu về dự án quản lý kí túc xá**

**1. Tổng quan**   
1.1 Đặt vấn đề

Trong thời đại chuyển đổi số, sự phát triển mạnh mẽ của công nghệ đã ảnh hưởng sâu rộng đến nhiều lĩnh vực, bao gồm cả công tác quản lý ký túc xá (KTX).

Phương pháp quản lý truyền thống với quy trình thủ công thường gặp phải nhiều bất cập như mất nhiều thời gian, dễ xảy ra sai sót và thiếu sự minh bạch trong việc quản lý thông tin sinh viên, phòng ở, cũng như chi phí liên quan. Đồng thời, nhu cầu hiện đại hóa, tối ưu hóa trải nghiệm người dùng và nâng cao hiệu quả quản lý ngày càng trở nên cấp thiết.

Để giải quyết các vấn đề này, việc phát triển một hệ thống web quản lý ký    túc xá thông minh là yêu cầu tất yếu. Hệ thống không chỉ giúp tự động hóa các quy trình quản lý, tăng tính chính xác mà còn tạo sự tiện lợi cho cả ban quản lý và sinh viên. Đây là một giải pháp hữu hiệu giúp nâng cao chất lượng dịch vụ, đáp ứng nhu cầu ngày càng cao của người dùng, đồng thời đóng góp vào sự phát triển bền vững của ký túc xá trong kỷ nguyên công nghệ.

1.2 Mục tiêu và phạm vi của chủ đề

**1.** **Mục tiêu**

1. Phát triển một hệ thống quản lý ký túc xá thông minh giúp tự động hóa quy trình quản lý phòng ở, hợp đồng thuê, và thông tin sinh viên.
2. Cung cấp giao diện thân thiện, dễ sử dụng cho cả ban quản lý và sinh viên.
3. Tăng cường hiệu quả quản lý và giảm thiểu sai sót trong các quy trình vận hành ký túc xá.
4. Đảm bảo hệ thống an toàn, bảo mật thông tin và có khả năng mở rộng để đáp ứng nhu cầu phát triển trong tương lai.

**Phạm vi**

1. Xây dựng hệ thống web quản lý ký túc xá, hỗ trợ hoạt động trên các thiết bị máy tính và di động.
2. Tích hợp các tính năng chính: quản lý phòng ở, quản lý thông tin sinh viên, quản lý hợp đồng thuê, theo dõi và xử lý chi phí, và gửi thông báo tự động. • Hỗ trợ chức năng tìm kiếm và báo cáo, giúp ban quản lý dễ dàng tra cứu và phân tích dữ liệu.
3. Hỗ trợ ngôn ngữ tiếng Việt để phục vụ đối tượng sinh viên và quản trị viên trong nước, với khả năng mở rộng thêm ngôn ngữ quốc tế nếu cần.

1.3 Đối tượng sử dụng

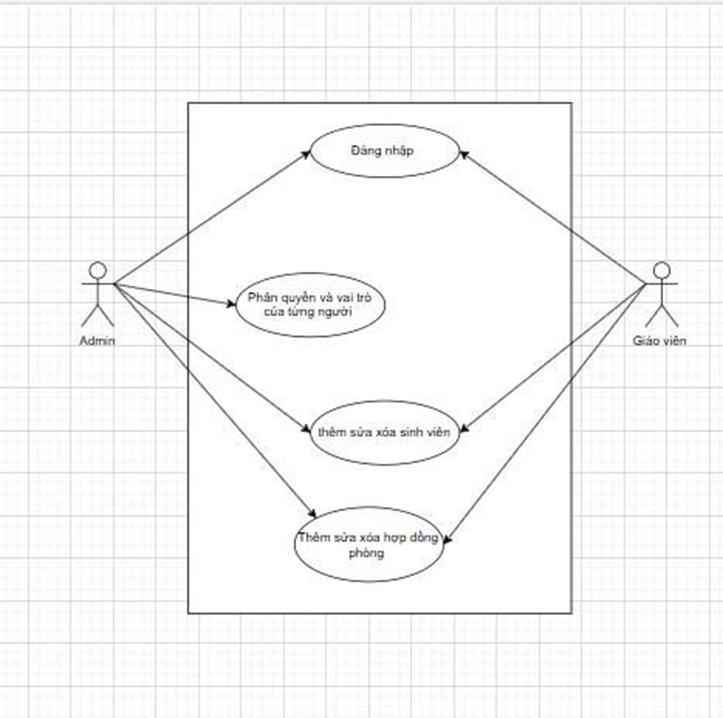
|  |  |
| --- | --- |
| **Đối tượng** | **Đối tượng** |
| **Sinh viên** | 1. Quản lý thông tin cá nhân và phòng ở: o Tra cứu thông tin phòng hiện tại: mã phòng, số lượng người ở, tiện nghi.  * Đăng ký hoặc hủy đăng ký phòng trong một số điều kiện nhất định (như trước thời hạn hợp đồng). * Xem lịch sử đăng ký và trạng thái hợp đồng thuê phòng.  1. Thanh toán chi phí:  * Theo dõi và thanh toán chi phí (tiền phòng, điện, nước) qua mã QR tích hợp với các ví điện tử (Momo, ZaloPay) hoặc chọn phương thức "Thanh toán sau".  1. Nhận email hoặc thông báo xác nhận khi thanh toán thành công. • Nhận thông báo và phản hồi: o Nhận thông báo từ hệ thống liên quan đến thanh toán, gia hạn hợp đồng, hoặc các quy định mới của ký túc xá.  * Gửi phản hồi hoặc yêu cầu hỗ trợ trực tiếp qua hệ thống. |
| **Giảng viên** | 1. Xem thông tin về sinh viên. 2. Kiểm tra tình trạng học tập của sinh viên. 3. Thực hiện các thống kê liên quan đến sinh viên và ký túc xá. • Quản lý các hoạt động giảng dạy và lớp học trong ký túc xá. |
| **Quản lý (Admin)** | 1. Quản lý tài khoản người dùng (Sinh viên, Giảng viên). 2. Quản lý các phòng ký túc xá, cập nhật tình trạng phòng, số lượng phòng trống. 3. Xử lý đăng ký và thanh toán phòng ở của sinh viên. 4. Cập nhật thông tin hóa đơn điện nước cho các phòng. 5. Cập nhật và quản lý các thông tin liên quan đến các cơ sở vật chất, tiện ích trong ký túc xá. 6. Quản lý các quyền truy cập và phân quyền người dùng. |

2. **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

**2.1 Các chức năng của hệ thống**

1. **Đăng nhập**: Người dùng cần có khả năng đăng nhập vào hệ thống để quản lý thông tin cá nhân và tình trạng phòng.
2. **Quản lý tài khoản cá nhân**: Hệ thống cho phép người dùng chỉnh sửa thông tin cá nhân như tên, email, mật khẩu và các thông tin liên quan.
3. **Quản lý phòng ở**: Sinh viên có thể tra cứu và theo dõi tình trạng phòng của mình, yêu cầu chuyển phòng nếu cần.
4. **Quản lý thanh toán**: Hệ thống hỗ trợ sinh viên thanh toán tiền phòng và theo dõi lịch sử thanh toán.
5. **Yêu cầu sửa chữa và bảo trì**: Sinh viên có thể gửi yêu cầu sửa chữa phòng và theo dõi tiến trình xử lý yêu cầu.
6. **Chức năng thống kê**: Hệ thống hỗ trợ các báo cáo thống kê về tình trạng phòng ở, thu phí và các dịch vụ khác.
7. **Bảo mật và phân quyền**: Các quyền truy cập được phân chia rõ ràng giữa các đối tượng người dùng như sinh viên, nhân viên quản lý và quản trị viên.
8. **Tính năng hỗ trợ**: Hệ thống cung cấp các công cụ thông báo về tình trạng phòng, yêu cầu sửa chữa và các thông tin quan trọng khác về ký túc xá.

**2.2 Sơ đồ Use Case**



**3. Phương pháp thực hiện**

3.1.  Giới thiệu về Spring boot

3.1.1   Tổng quan

Spring Boot giúp bạn tạo các ứng dụng Spring độc lập, cấp sản xuất mà  bạn có thể chạy. Chúng tôi có quan điểm chủ quan về nền tảng Spring và các thư viện của bên thứ ba, để bạn có thể bắt đầu với ít rắc rối nhất. Hầu hết các ứng dụng Spring Boot cần rất ít cấu hình Spring.

Bạn có thể sử dụng Spring Boot để tạo các ứng dụng Java có thể được khởi động bằng cách sử dụng java -jar hoặc các triển khai war truyền thống hơn. Chúng tôi cũng cung cấp một công cụ dòng lệnh chạy "spring scripts".

Mục tiêu chính là:

1. Cung cấp trải nghiệm bắt đầu nhanh hơn và dễ tiếp cận hơn cho mọi quá trình phát triển Spring.
2. Hãy nêu ý kiến của mình nhưng hãy nhanh chóng tránh xa khi các yêu cầu bắt đầu khác biệt so với mặc định.
3. Cung cấp một loạt các tính năng không chức năng phổ biến trong các lớp dự án lớn (chẳng hạn như máy chủ nhúng, bảo mật, số liệu, kiểm tra tình trạng và cấu hình bên ngoài).
4. Hoàn toàn không cần tạo mã và không yêu cầu cấu hình XML.

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



***Hình 3.1. Logo của Spring boot***

**3.1.2 Ưu điểm của Spring Boot**

1. **Cấu hình đơn giản**: Hỗ trợ nhúng container servlet, chỉ cần môi trường Java để chạy file .jar.
2. **Tích hợp nhanh chóng**: Tự động nhận diện và tích hợp các thư viện bên ngoài mà không cần cấu hình phức tạp.
3. **Giải quyết hạn chế của Spring truyền thống**: Giảm mã, cấu hình XML, và phụ thuộc, tiết kiệm chi phí phát triển.
4. **Dễ dàng phát triển ứng dụng**: Phù hợp cho phát triển nhanh với Java hoặc Groovy.
5. **Tăng hiệu suất**: Giảm mã mẫu và cấu hình mặc định giúp tiết kiệm thời gian.
6. **Tích hợp với hệ sinh thái Spring**: Kết hợp dễ dàng với Spring JDBC, Spring ORM, Spring Data, Spring Security, v.v.
7. **Máy chủ HTTP nhúng**: Hỗ trợ Tomcat, Jetty cho phát triển và thử nghiệm dễ dàng.
8. **Hỗ trợ CLI và công cụ xây dựng**: Cung cấp CLI và plugin cho Maven, Gradle.
9. **Làm việc với cơ sở dữ liệu dễ dàng**: Tích hợp nhiều plugin hỗ trợ.

**3.1.3 Nhược điểm của Spring Boot**

1. **Thiếu kiểm soát**: Cấu hình tự động có thể gây khó khăn cho tùy chỉnh, làm tăng kích thước file triển khai.
2. **Khó khăn với ứng dụng đơn khối**: Không tối ưu cho các ứng dụng lớn.
3. **Giới hạn theo dõi**: Công cụ theo dõi mặc định chưa đầy đủ, cần thêm công cụ hỗ trợ.
4. **Khó nâng cấp mã Spring cũ**: Cập nhật mã cũ lên Spring Boot phức tạp.
5. **Yêu cầu kiến thức nền tảng**: Cần hiểu các khái niệm như proxy, dependency injection, AOP để học hiệu quả.
6. **Tính phù hợp**: Không khuyến nghị dùng cho ứng dụng độc lập lớn mặc dù thiết kế cho dịch vụ vi mô.

**3.2.**  **Angular**

3.2.1 **Angular là gì?**

Angular là một nền tảng và framework mã nguồn mở phổ biến dành cho việc xây dựng các ứng dụng web đơn trang (SPA - Single Page Applications). Nó được phát triển và bảo trì bởi Google, và là một trong những công cụ được ưa chuộng nhất trong giới phát triển phần mềm hiện nay.

Angular cung cấp một bộ công cụ mạnh mẽ và toàn diện cho việc phát triển giao diện người dùng, quản lý trạng thái, tương tác với máy chủ qua HTTP và nhiều tính năng phức tạp khác. Nó hoạt động dựa trên kiến trúc thành phần, nơi mà ứng dụng được chia thành các thành phần nhỏ, tái sử dụng được, giúp quản lý và phát triển ứng dụng trở nên dễ dàng hơn.

  x

**Hình 3.2 Angular**

3.2.2 **Lợi ích sử dụng Angular**

**- Phát triển nhanh chóng**: Angular tự động hóa nhiều tác vụ lập trình, tiết kiệm thời gian cho lập trình viên.

**- Mở rộng dễ dàng**: Hỗ trợ xây dựng ứng dụng web có thể mở rộng và dễ dàng nâng cấp.

**- Bảo trì dễ dàng**: Cấu trúc rõ ràng và mô-đun hóa giúp bảo trì và phát triển thuận tiện hơn.

**- Trải nghiệm người dùng mượt mà**: Tối ưu hóa hiệu suất, mang lại trải nghiệm nhanh chóng cho người dùng.

**3.3.**   **My SQL**



**Hình 3.3 MySQL**

MySQL là một hệ thống quản lý cơ sở dữ liệu nguồn mở hàng đầu, phổ biến trên web và là phần của nền tảng LAMP (Linux, Apache, MySQL, PHP). Nó đa người dùng và đa luồng, chạy trên nhiều hệ điều hành như BSD Unix, Linux, Windows và Mac. Nhiều trang web lớn như Wikipedia, YouTube và Facebook sử dụng MySQL để xử lý hàng triệu truy vấn mỗi ngày.

MySQL có hai phiên bản: máy chủ và hệ thống nhúng, với giấy phép GPL phiên bản 2 và giấy phép độc quyền. Phát triển từ năm 1994 bởi MySQL AB, nó được Sun Microsystems mua lại vào năm 2008 và sau đó thuộc về Oracle vào năm 2010. MySQL được viết bằng C/C++ và có API cho nhiều ngôn ngữ như PHP, Python, Java, và Ruby.

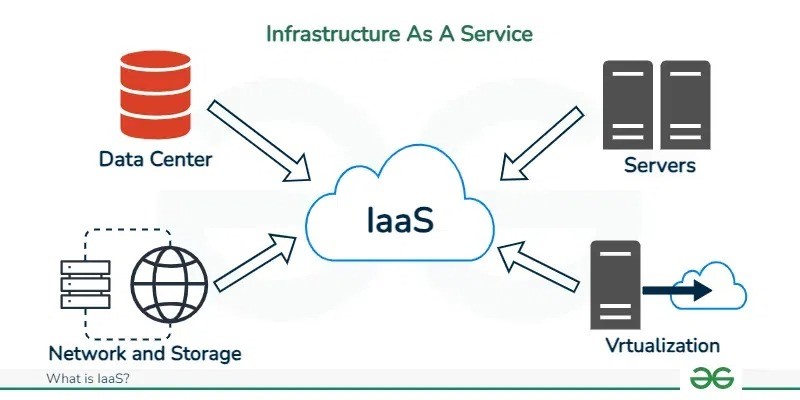
có những phần nào trong dự án    
chức năng của dự án   
sử dụng gì để code    
Chương 2: **Cloud**

3.1 Khái niệm về Cloud Computing

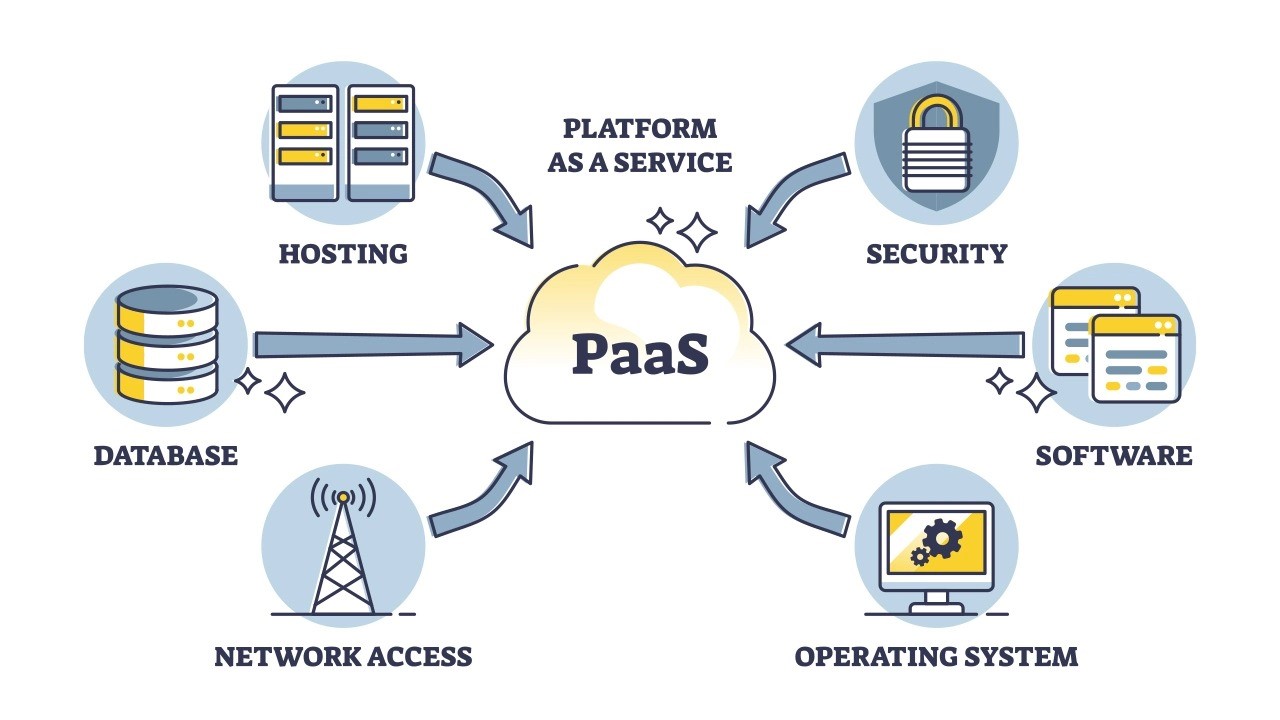
Cloud computing là mô hình cung cấp tài nguyên máy tính như máy chủ, lưu trữ dữ liệu, mạng, phần mềm và các dịch vụ công nghệ thông tin theo yêu cầu. Thay vì phải mua và quản lý cơ sở hạ tầng máy tính riêng, người dùng có thể thuê hoặc sử dụng các dịch vụ này từ các nhà cung cấp dịch vụ đám mây.

3.2 Các mô hình dịch vụ

- IaaS (Infrastructure as a Service) là mô hình dịch vụ cung cấp các nguồn lực máy chủ, lưu trữ, mạng và các thành phần cơ sở hạ tầng khác qua internet. Thay vì phải mua và duy trì phần cứng và phần mềm tại nơi làm việc, người dùng có thể thuê các tài nguyên này từ một nhà cung cấp dịch vụ đám mây. Ví dụ Amazon Web Services (AWS), Microsoft Azure, Google Cloud.

Hình : **IaaS (Infrastructure as a Service)**

- **PaaS (Platform as a Service)** cung cấp nền tảng phát triển và triển khai ứng dụng. Người dùng có thể xây dựng, kiểm thử và triển khai ứng dụng mà không cần quản lý cơ sở hạ tầng hoặc hệ điều hành. Ví dụ: Heroku, Google App Engine, Microsoft Azure App Service.

Hình : **PaaS (Platform as a Service)**

- **SaaS (Software as a Service)** cung cấp các ứng dụng phần mềm cho người dùng dưới dạng dịch vụ thông qua internet. Người dùng chỉ cần truy cập và sử dụng phần mềm  mà không phải cài đặt hoặc duy trì phần mềm. Ví dụ: Google Workspace (Gmail, Google Docs), Microsoft Office 365, Dropbox.



**Hình : SaaS (Software as a Service)**

3.3 Mô hình triển khai

* **Public Cloud:** Dịch vụ đám mây được cung cấp cho nhiều khách hàng (người dùng hoặc tổ chức) qua internet. Tài nguyên được chia sẻ và các nhà cung cấp đám mây quản lý hoàn toàn hạ tầng. Ví dụ: AWS, Google Cloud, Microsoft 365.
* **Private Cloud:** Dịch vụ đám mây chỉ dành riêng trong nội bộ của doanh nghiệp hoặc tổ chức, và có thể được triển khai trong cơ sở hạ tầng của công ty hoặc trên môi trường đám mây do một nhà cung cấp riêng biệt quản lý.
* **Hybrid Cloud:** Kết hợp giữa public cloud và private cloud, cho phép doanh nghiệp hoặc tổ chức di chuyển các tài nguyên và dữ liệu giữa hai môi trường này. Điều này giúp tối ưu hóa hiệu suất và bảo mật.

3.4 Lợi ích cloud

* **Tiết kiệm chi phí:** Không cần đầu tư vào hạ tầng phần cứng và chỉ trả tiền cho tài nguyên sử dụng.
* **Khả năng mở rộng:** Có thể tăng giảm tài nguyên theo nhu cầu mà không gặp phải giới hạn vật lý.
* **Tiện lợi và linh hoạt:** Truy cập từ bất kỳ đâu và sử dụng các ứng dụng một cách linh hoạt, không cần cài đặt phức tạp.
* **Độ tin cậy cao:** Các dịch vụ đám mây thường cung cấp tính năng sao lưu và khôi phục dữ liệu giúp đảm bảo sự ổn định.

3.5 AWS và các dịch vụ cơ bản

**Aws (Amazon Web Services)**:  là nền tảng đám mây hàng đầu hiện nay do Amazon phát triển và cung cấp. AWS ra mắt vào năm 2006 và hiện nay cung cấp hơn 200 dịch vụ từ các trung tâm dữ liệu trên toàn thế giới. Doanh nghiệp có thể lựa chọn dịch vụ lưu trữ, phân tích, tính toán, cơ sở dữ liệu, công cụ,... tại Amazon Web Services.

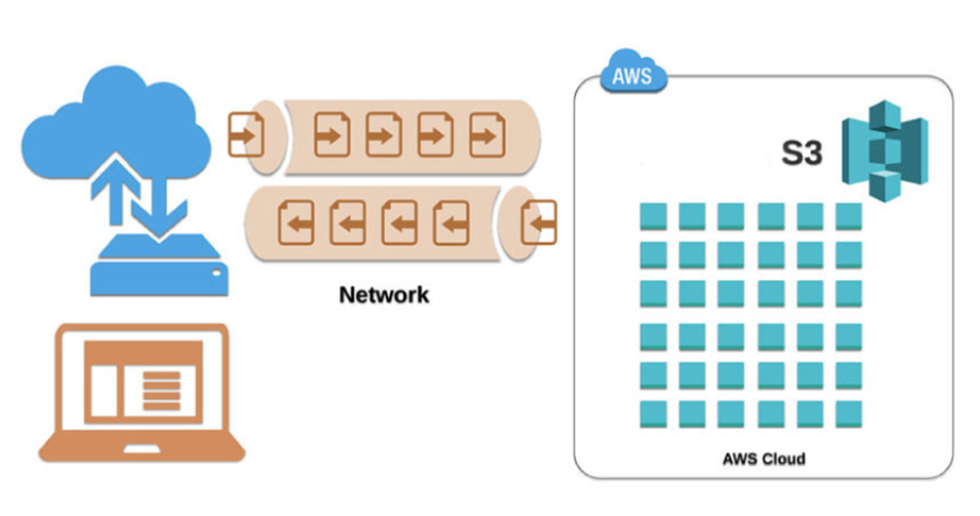
* Điểm mạnh AWS
* **Độ tin cậy cao**: AWS có tính sẵn sàng cao nhờ hạ tầng mạnh mẽ và nhiều trung tâm dữ liệu toàn cầu.
* **Tính linh hoạt**: Dễ dàng tùy chỉnh và mở rộng các dịch vụ theo nhu cầu cụ thể.
* **Đa dạng dịch vụ**: Cung cấp hơn 200 dịch vụ từ lưu trữ, tính toán đến phân tích và trí tuệ nhân tạo.
* **Chi phí hợp lý**: Mô hình thanh toán theo mức sử dụng của AWS giúp doanh nghiệp tiết kiệm đáng kể chi phí so với việc đầu tư vào hạ tầng riêng.
* **Bảo mật cao**: AWS đáp ứng các tiêu chuẩn bảo mật và tuân thủ nghiêm ngặt, đồng thời đạt mức độ khả dụng và đáng tin cậy cao.

Dịch **vụ nổi bật của**AWS

EC2 (Elastic Compute Cloud)

* EC2 cung cấp các ứng dụng máy tính ảo hóa mở rộng, cho phép tùy chọn linh hoạt về phần cứng ảo (RAM, vi xử lý) và lưu trữ dữ liệu, đảm bảo an toàn nhờ kiến trúc đám mây mạnh mẽ của AWS.
* EC2 cho phép triển khai nhanh chóng các máy chủ ảo và kết hợp chúng, đảm bảo tính sẵn sàng cao. Người dùng cũng dễ dàng theo dõi chi phí dựa trên tài nguyên sử dụng.
* **Storage Amazon S3**

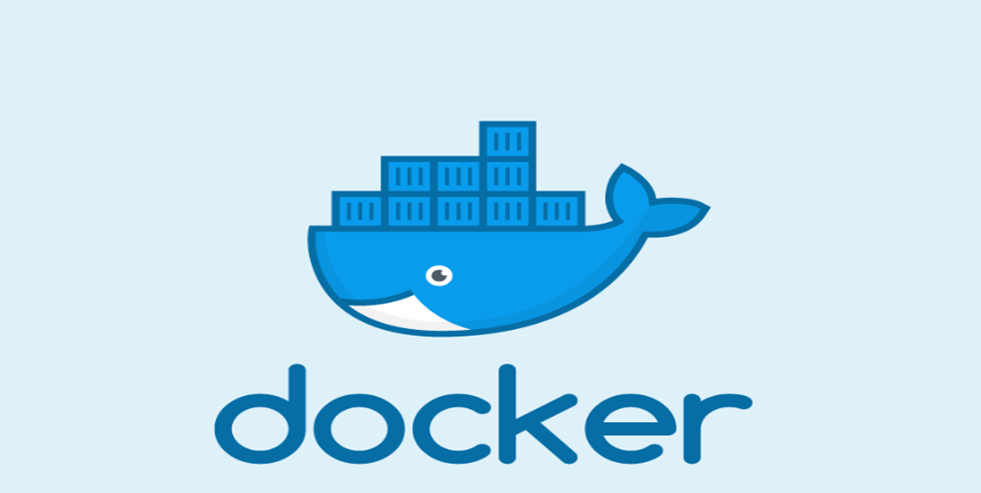
Giải pháp Storage Amazon S3 cung cấp một môi trường an toàn, đáng tin cậy cho doanh nghiệp lưu trữ thông tin dữ liệu của mình trên Internet. Điểm nổi bật của giải pháp này là có tính sẵn sàng cao, chi phí thấp và phù hợp cho nhiều doanh nghiệp tại nhiều khu vực địa lý khác nhau. Storage Amazon S3 phù hợp cho các nhà phát triển, các doanh nghiệp có nhu cầu lưu trữ hướng đối tượng với tính bảo mật cao, khả năng mở rộng linh hoạt.



Doanh nghiệp có thể lưu trữ bất kỳ loại dữ liệu nào, trên bất cứ format nào. Storage Amazon S3 đảm bảo an toàn cho dữ liệu bằng cách mã hóa dữ liệu, lưu lại các phiên bản dữ liệu để phục hồi khi cần thiết, có các cơ chế thường xuyên xác minh tính toàn vẹn của dữ liệu,... Khi cần mở rộng tài nguyên bộ nhờ, doanh nghiệp chỉ cần tăng dung lượng lưu trữ qua vài click đơn giản và chỉ trả tiền cho bộ nhớ đã dùng.

**Chương 3. Docker**

3.1 Khái niệm về Docker   
 **Docker** là một nền tảng phần mềm mã nguồn mở giúp tự động hóa việc triển khai, chạy và quản lý các ứng dụng trong các container. **Container** là những đơn vị phần mềm nhẹ, độc lập và có thể chạy trên bất kỳ môi trường nào mà không phụ thuộc vào hệ điều hành hay phần cứng cụ thể.



3.2 Các đặc điểm chính của Docker:

3.2.1. Containerization (Đóng gói trong container):

* Docker sử dụng container để đóng gói ứng dụng cùng tất cả các thành phần cần thiết để chạy ứng dụng đó (bao gồm mã nguồn, thư viện, môi trường chạy, v.v.) vào trong một container duy nhất. Điều này đảm bảo rằng ứng dụng sẽ chạy giống nhau trên mọi môi trường, từ máy tính của nhà phát triển cho đến môi trường sản xuất (production).

3.2.2. Nhẹ và nhanh:

* Containers rất nhẹ, vì chúng chia sẻ kernel (hạt nhân hệ điều hành) của máy chủ mà không phải tải lại hệ điều hành riêng biệt như các máy ảo (VM). Điều này giúp Docker khởi động và dừng nhanh chóng, tối ưu hóa tài nguyên hệ thống.

3.2.3. Khả năng di chuyển giữa các môi trường:

* Docker cho phép các ứng dụng được đóng gói trong container di chuyển và chạy một cách nhất quán trên các môi trường khác nhau (từ máy tính cá nhân, máy chủ, đến môi trường đám mây). Điều này giải quyết được vấn đề về sự khác biệt giữa môi trường phát triển và môi trường sản xuất.

3.2.4. Quản lý container dễ dàng:

Docker cung cấp các công cụ giúp tạo, triển khai và quản lý container. Công cụ nổi bật là Docker CLI (dòng lệnh) và Docker Compose (cho phép định nghĩa và chạy các ứng dụng đa container).

3.3 Cấu trúc của Docker:

3.3.1. Docker Engine:

* Là phần mềm chủ yếu của Docker, chịu trách nhiệm chạy các container. Docker Engine bao gồm hai thành phần:
* Docker Daemon: Là tiến trình chạy nền trên máy chủ và chịu trách nhiệm quản lý các container và các hình ảnh Docker.
* Docker CLI: Là giao diện dòng lệnh, nơi người dùng có thể tương tác với Docker Daemon để thực hiện các lệnh như tạo, chạy, dừng, v.v.

3.3.2. Docker Images:

* Là các mẫu (template) không thay đổi của các ứng dụng và môi trường hệ thống, được tạo ra từ các Dockerfile. Một Docker image chứa tất cả các phần mềm cần thiết để chạy ứng dụng. Các images có thể được lưu trữ và chia sẻ qua Docker Hub, một kho lưu trữ công cộng cho Docker images.

3.3.3. Docker Containers:

* Là các phiên bản đang chạy của Docker image. Một container là một quá trình đang chạy trên hệ điều hành, được cách ly hoàn toàn với các container khác và hệ điều hành bên ngoài.

3.3.4. Docker Compose:

* Là công cụ cho phép người dùng định nghĩa và chạy các ứng dụng Docker đa container. Với Docker Compose, bạn có thể sử dụng một tệp cấu hình (thường là docker-compose.yml) để định nghĩa các container, mạng và volume cần thiết cho ứng dụng của mình.

3.4 Lợi ích của Docker:

3.4.1. Tính nhất quán giữa các môi trường:

* Docker giúp giảm thiểu các vấn đề phát sinh khi chạy ứng dụng trong các môi trường khác nhau. Một ứng dụng đã được đóng gói trong container sẽ chạy như nhau ở mọi nơi.

3.4.2. Tính di động cao:

* Các container Docker có thể chạy trên bất kỳ hệ điều hành nào hỗ trợ Docker, bao gồm Linux, macOS, Windows, và các môi trường đám mây như AWS, Google Cloud và Microsoft Azure.

3.4.3. Tiết kiệm tài nguyên và hiệu suất:

* Các container Docker chia sẻ hệ điều hành và tài nguyên phần cứng, giúp tiết kiệm tài nguyên so với việc sử dụng máy ảo, đồng thời cho phép chạy nhiều container trên cùng một máy chủ.

3.4.4. Quản lý ứng dụng dễ dàng hơn:

* Docker giúp đơn giản hóa việc triển khai, cập nhật và mở rộng ứng dụng. Bạn có thể dễ dàng tạo, triển khai, dừng, và quản lý nhiều container ứng dụng với các công cụ dòng lệnh và các công cụ quản lý như Kubernetes.

3.4.5. Khả năng tự động hóa:

* Docker có thể dễ dàng tích hợp với các công cụ DevOps, CI/CD để tự động hóa quy trình phát triển, kiểm tra, triển khai ứng dụng.

3.5 Docker Hub:

* Docker Hub là dịch vụ lưu trữ các Docker images, nơi bạn có thể tìm kiếm, tải về và chia sẻ các images với cộng đồng. Đây là kho chứa hình ảnh chính thức của Docker, bao gồm các images của nhiều ứng dụng phổ biến.

3.6 Các công cụ và dịch vụ bổ sung của Docker:

* Docker Swarm: Là công cụ điều phối (orchestrator) của Docker, cho phép quản lý các container Docker trong môi trường phân tán.
* Kubernetes: Một công cụ phổ biến hơn giúp điều phối các container Docker trong môi trường sản xuất quy mô lớn.

3.7 Dockerfile của dự án

3.7.1 Dockerfile của Database

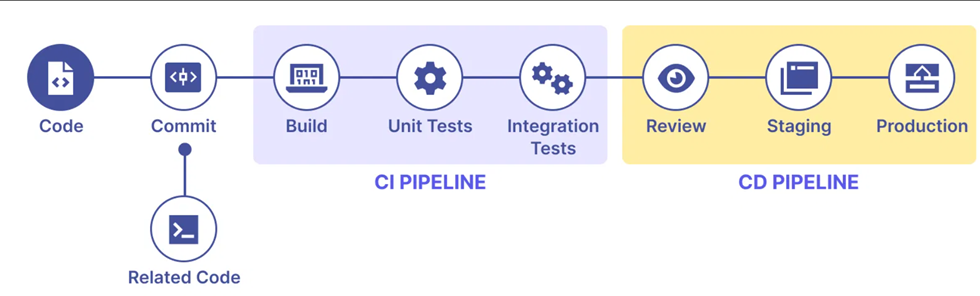
3.7.2 Dockerfile của Backend

3.7.3 Dockerfile của Frontend

**CHƯƠNG 4: TRIỂN KHAI CI/CD CHO DỰ ÁN**

4.1 Tìm hiểu về CI/CD

**CI/CD** là viết tắt của **Continuous Integration** (Tích hợp liên tục) và **Continuous Delivery/Continuous Deployment** (Phân phối/ Triển khai liên tục). Đây là một tập hợp các phương pháp phát triển phần mềm nhằm tự động hóa các giai đoạn trong quy trình phát triển phần mềm để cải thiện tốc độ và chất lượng của việc phát hành phần mềm. CI/CD giúp tăng cường hiệu quả trong việc triển khai, kiểm tra, và phát hành phần mềm một cách nhanh chóng và an toàn.

Hình 2.4.2. CICD WorkFlow

4.1.1 Continuous Integration (CI) - Tích hợp liên tục:

CI là phương pháp trong đó các nhà phát triển tích hợp (hoặc "gửi") mã nguồn của họ vào một kho lưu trữ chung nhiều lần trong ngày. Mỗi lần tích hợp sẽ kích hoạt một chu trình tự động kiểm tra mã nguồn để đảm bảo rằng mã mới không gây ra lỗi hoặc xung đột với mã hiện tại.

Các bước trong CI:

* Tích hợp mã thường xuyên: Các nhà phát triển thường xuyên gửi mã vào kho lưu trữ mã (như Git) để tích hợp với mã của các thành viên khác.
* Kiểm tra tự động: Sau mỗi lần tích hợp, một bộ kiểm thử tự động (unit tests, integration tests) sẽ chạy để xác nhận rằng mã mới không làm hỏng ứng dụng hoặc gây ra lỗi.
* Kiểm tra chất lượng mã: Công cụ phân tích mã có thể được sử dụng để đảm bảo mã tuân thủ các quy chuẩn và không có lỗi tiềm ẩn.

Lợi ích của CI:

* Phát hiện lỗi sớm: Các lỗi được phát hiện ngay lập tức khi tích hợp mã, giúp giảm thiểu chi phí và thời gian sửa lỗi.
* Tăng cường chất lượng mã: Các kiểm thử tự động và phân tích mã giúp đảm bảo chất lượng mã cao hơn.
* Tiết kiệm thời gian và công sức: Quá trình kiểm thử và xây dựng tự động giúp đội ngũ tiết kiệm thời gian kiểm tra thủ công.

4.1.2. Continuous Delivery (CD) - Phân phối liên tục:

* Continuous Delivery: là một phương pháp trong đó mã nguồn sau khi vượt qua các kiểm tra tự động (trong CI) sẽ được chuẩn bị sẵn sàng để triển khai vào môi trường sản xuất. Tuy nhiên, việc triển khai có thể vẫn yêu cầu sự can thiệp của con người. Mục tiêu là để mã luôn sẵn sàng cho việc phát hành vào bất kỳ thời điểm nào

Các bước trong CD:

* Triển khai mã vào môi trường thử nghiệm: Mã sẽ được triển khai vào các môi trường thử nghiệm hoặc staging để kiểm tra thêm về tính năng, hiệu suất và bảo mật.
* Kiểm tra cuối cùng: Các bài kiểm thử như kiểm tra hồi quy, kiểm tra tải, và kiểm tra bảo mật có thể được thực hiện trong môi trường staging.
* Chuẩn bị cho phát hành: Sau khi mã vượt qua tất cả các bài kiểm tra, nó được đóng gói và sẵn sàng cho việc phát hành vào môi trường sản xuất

Lợi ích của Continuous Delivery:

* Giảm thiểu rủi ro: Các bản phát hành nhỏ và thường xuyên giúp giảm rủi ro và dễ dàng khắc phục các vấn đề phát sinh.
* Tăng tốc thời gian đưa ra thị trường: Việc chuẩn bị mã sẵn sàng để phát hành giúp tăng tốc quy trình ra mắt các tính năng mới hoặc sửa lỗi.
* Tính linh hoạt: Do tính sẵn sàng của mã, các đội phát triển có thể nhanh chóng triển khai khi có yêu cầu.

4.1.3. Continuous Deployment (CD) - Triển khai liên tục:

Continuous Deployment: là một bước tiến xa hơn của Continuous Delivery. Trong mô hình này, mã đã qua các bài kiểm tra và kiểm duyệt tự động sẽ được triển khai trực tiếp vào môi trường sản xuất mà không cần sự can thiệp của con người. Mỗi thay đổi mã sau khi tích hợp và kiểm tra sẽ được đưa ngay vào môi trường sản xuất.

Các bước trong Continuous Deployment:

* Triển khai tự động vào môi trường sản xuất: Sau khi mã được kiểm tra thành công trong các môi trường thử nghiệm, nó sẽ được tự động triển khai vào môi trường sản xuất mà không cần sự can thiệp của con người.
* Theo dõi và phản hồi: Sau khi mã được triển khai, hệ thống sẽ tiếp tục theo dõi hiệu suất và các vấn đề có thể xảy ra trong môi trường sản xuất.

Lợi ích của Continuous Deployment:

* Phát hành nhanh chóng: Các tính năng hoặc sửa lỗi có thể đến tay người dùng nhanh chóng mà không cần phải chờ đợi một chu kỳ phát hành lớn.
* Giảm sự can thiệp của con người: Quy trình hoàn toàn tự động giúp giảm thiểu các lỗi do con người và tăng hiệu quả.
* Phản hồi tức thì: Nhờ vào triển khai nhanh chóng, các đội có thể nhận được phản hồi từ người dùng nhanh chóng, giúp cải thiện sản phẩm

4.1.4. CI/CD Pipeline - Quy trình CI/CD:

Quy trình CI/CD là một chuỗi các bước tự động hóa từ việc phát triển mã, kiểm tra, xây dựng, và triển khai mã vào môi trường sản xuất. Một pipeline CI/CD bao gồm các công đoạn chính sau:

* Source: Tải mã từ kho lưu trữ (ví dụ: Git).
* Build: Xây dựng và biên dịch mã nguồn.
* Test: Kiểm tra mã để phát hiện lỗi và đảm bảo chất lượng.
* Deploy: Triển khai mã vào môi trường thử nghiệm hoặc sản xuất.

Các công cụ phổ biến hỗ trợ CI/CD pipeline:

* Jenkins: Công cụ mã nguồn mở phổ biến để tự động hóa quá trình CI/CD.
* GitLab CI: Một giải pháp CI/CD tích hợp với GitLab.
* CircleCI: Công cụ CI/CD hỗ trợ việc kiểm thử và triển khai tự động.
* Travis CI: Công cụ CI phổ biến cho các dự án mã nguồn mở.
* Azure DevOps: Giải pháp của Microsoft cho CI/CD.

**KẾT LUẬN VÀ KIẾN NGHỊ**

1. **Kết luận**
2. **Kiến nghị**

**DANH MỤC TÀI LIỆU THAM KHẢO**