# 1. Mở đầu

## 1.1. Giới thiệu

Bài thực hành tuần 05 trong khóa học phát triển ứng dụng web tập trung vào việc xây dựng một hệ thống quản lý công việc và ứng viên. Ứng dụng này được phát triển sử dụng Spring Boot, một framework phổ biến cho việc phát triển ứng dụng Java. Mục tiêu chính của bài thực hành là giúp sinh viên hiểu và thực hành các kỹ thuật phát triển ứng dụng web như ORM, phân trang, và hiển thị dữ liệu trên giao diện người dùng.

Ứng dụng này cho phép các công ty đăng tuyển dụng công việc, yêu cầu các kỹ năng từ ứng viên, và gợi ý các công việc phù hợp với kỹ năng của ứng viên khi họ đăng nhập vào hệ thống. Ngoài ra, các công ty có thể tìm kiếm và mời các ứng viên phù hợp qua email.

## 1.2 Mục tiêu

* Áp dụng các kỹ thuật ORM để tương tác với cơ sở dữ liệu.
* Cài đặt phân trang để tối ưu hóa hiệu suất khi hiển thị danh sách công việc và ứng viên.
* Xây dựng giao diện web để hiển thị dữ liệu và tạo các tính năng cho người dùng.
* Tạo các trang web cho phép công ty đăng tin tuyển dụng với các kỹ năng yêu cầu.
* Gợi ý công việc cho ứng viên dựa trên kỹ năng đã có.
* Cung cấp công cụ tìm kiếm ứng viên phù hợp và gửi email mời tham gia.
* Đề xuất kỹ năng mà ứng viên cần học thêm để đáp ứng yêu cầu công việc.

# 2. Phân tích yêu cầu

Phân tích yêu cầu là bước quan trọng để xác định các tính năng và chức năng mà hệ thống cần có, cũng như cách thức các thành phần trong hệ thống sẽ tương tác với nhau. Dưới đây là phân tích chi tiết các yêu cầu của bài toán:

## 2.1. Các Entity và Mối Quan Hệ

Ứng dụng quản lý công việc và ứng viên cần phải xử lý thông tin liên quan đến công việc và ứng viên. Cụ thể, yêu cầu là:

* Công việc (Job): Mỗi công việc có thể yêu cầu một hoặc nhiều kỹ năng (skills) và có thể có nhiều ứng viên ứng tuyển.
* Ứng viên (Applicant): Mỗi ứng viên có thể có một hoặc nhiều kỹ năng và có thể ứng tuyển vào nhiều công việc.
* Ứng tuyển (Application): Mỗi ứng viên sẽ có thể ứng tuyển vào một công việc và công ty sẽ theo dõi tình trạng ứng tuyển của họ (ví dụ: "Đã nộp hồ sơ", "Phỏng vấn", "Đã thuê").

**Mối quan hệ giữa các entity**:

* Một công việc có thể có nhiều ứng viên thông qua bảng Application.
* Một ứng viên có thể ứng tuyển vào nhiều công việc, mối quan hệ này cũng được lưu trong bảng Application.

## 2.2. Chức Năng và Tính Năng Của Ứng Dụng

1. **Quản lý công việc**:
   * Công ty có thể tạo, sửa, xóa các tin tuyển dụng.
   * Mỗi công việc có các trường như: tiêu đề, mô tả, kỹ năng yêu cầu, và trạng thái (đang tuyển dụng, đã đóng).
   * Có thể tìm kiếm công việc theo các tiêu chí như kỹ năng, địa điểm, mức lương, v.v.
2. **Quản lý ứng viên**:
   * Ứng viên có thể đăng ký tài khoản, nhập các thông tin cá nhân và kỹ năng của mình.
   * Ứng viên có thể tìm kiếm công việc phù hợp với kỹ năng của mình.
   * Ứng viên có thể ứng tuyển vào các công việc mà họ thấy phù hợp.
3. **Gợi ý công việc cho ứng viên**:
   * Hệ thống sẽ tự động gợi ý các công việc phù hợp với kỹ năng của ứng viên khi họ đăng nhập vào hệ thống.
   * Gợi ý sẽ dựa trên các kỹ năng mà ứng viên đã nhập và các kỹ năng yêu cầu trong công việc.
4. **Gửi email mời ứng viên**:
   * Khi công ty tìm thấy ứng viên phù hợp với công việc, hệ thống sẽ gửi email mời ứng viên tham gia phỏng vấn hoặc ứng tuyển.
   * Email sẽ chứa thông tin công việc và lời mời ứng tuyển.
5. **Đề xuất kỹ năng cần học cho ứng viên**:
   * Hệ thống sẽ phân tích kỹ năng của ứng viên và so sánh với các kỹ năng yêu cầu trong công việc.
   * Đề xuất các kỹ năng mà ứng viên thiếu để có thể đáp ứng yêu cầu công việc, giúp ứng viên cải thiện bản thân.
6. **Phân trang và tìm kiếm**:
   * Do số lượng công việc và ứng viên có thể rất lớn, cần có tính năng phân trang để hiển thị dữ liệu một cách hợp lý.
   * Hệ thống cần hỗ trợ tìm kiếm nâng cao cho cả công ty và ứng viên (tìm theo kỹ năng, địa điểm, mức lương, v.v.).
7. **Giao diện người dùng**:
   * Giao diện web phải dễ sử dụng và trực quan, với các tính năng như tìm kiếm công việc, đăng tin tuyển dụng, và đăng ký ứng viên.
   * Cần có các form để công ty đăng công việc và ứng viên nhập thông tin cá nhân, kỹ năng.

## 2.3. Các Kỹ Thuật và Công Nghệ Sử Dụng

Trong quá trình triển khai hệ thống quản lý công việc và ứng viên, các công nghệ sau sẽ được sử dụng để phát triển ứng dụng. Các công nghệ này bao gồm cả frontend và backend, cơ sở dữ liệu, cũng như các dịch vụ hỗ trợ như gửi email.

### Backend

1. **Spring Boot**
   * **Mô tả**: Spring Boot là một framework phổ biến giúp phát triển ứng dụng Java nhanh chóng và dễ dàng. Spring Boot cung cấp các tính năng như cấu hình tự động, tối giản các cấu trúc dự án, giúp tiết kiệm thời gian khi phát triển ứng dụng.
   * **Vai trò**: Đây là framework chính để xây dựng phần backend của hệ thống. Spring Boot sẽ giúp xử lý các API RESTful, quản lý kết nối cơ sở dữ liệu, xác thực người dùng, và các nghiệp vụ backend khác.
2. **Spring Data JPA và Hibernate**
   * **Mô tả**: Spring Data JPA là một phần của Spring Framework, giúp dễ dàng thao tác với cơ sở dữ liệu thông qua ORM (Object Relational Mapping). Hibernate là một thư viện ORM phổ biến, giúp ánh xạ các đối tượng Java vào cơ sở dữ liệu quan hệ.
   * **Vai trò**: Dùng để tạo các repository interface, thao tác với cơ sở dữ liệu thông qua các entity (Job, Applicant, Application). Hibernate sẽ giúp tự động sinh ra SQL từ các câu lệnh JPQL hoặc HQL, giảm bớt sự phức tạp khi làm việc với cơ sở dữ liệu.
3. **Spring Security**
   * **Mô tả**: Spring Security là một framework mạnh mẽ để bảo mật các ứng dụng Java, giúp triển khai các cơ chế bảo mật như xác thực và phân quyền người dùng.
   * **Vai trò**: Đảm bảo bảo mật cho ứng dụng, bao gồm đăng nhập, đăng ký, và kiểm tra quyền truy cập của người dùng (ứng viên và công ty). Spring Security sẽ hỗ trợ các cơ chế xác thực như form login, OAuth, hoặc JWT.
4. **JavaMail API**
   * **Mô tả**: JavaMail API là một thư viện giúp gửi và nhận email trong các ứng dụng Java.
   * **Vai trò**: Được sử dụng để gửi email mời ứng viên tham gia phỏng vấn hoặc ứng tuyển công việc. Ứng dụng sẽ gửi email thông qua SMTP server.
5. **Thymeleaf**
   * **Mô tả**: Thymeleaf là một engine template phổ biến cho ứng dụng web Java. Thymeleaf cho phép kết hợp dữ liệu từ backend vào các file HTML, giúp xây dựng các giao diện người dùng động.
   * **Vai trò**: Được sử dụng để tạo các trang web (frontend) cho ứng dụng. Cụ thể, Thymeleaf sẽ xử lý các form đăng ký công việc, ứng viên, hiển thị danh sách công việc, và các chức năng tìm kiếm.
6. **Spring Boot DevTools**
   * **Mô tả**: Spring Boot DevTools là một công cụ hỗ trợ phát triển ứng dụng nhanh chóng, cung cấp các tính năng như tự động tải lại khi có thay đổi mã nguồn và cải thiện hiệu suất khi phát triển.
   * **Vai trò**: Giúp tăng tốc quá trình phát triển và kiểm thử ứng dụng trong môi trường phát triển.
7. **JUnit và Mockito**
   * **Mô tả**: JUnit là một framework kiểm thử đơn vị cho ứng dụng Java, trong khi Mockito là một thư viện dùng để tạo mock các đối tượng trong các bài kiểm thử.
   * **Vai trò**: Được sử dụng để viết các test cases cho các lớp service và controller, đảm bảo rằng các chức năng của hệ thống hoạt động đúng.

### Frontend

1. **Thymeleaf**
   * **Mô tả**: Như đã đề cập, Thymeleaf sẽ xử lý giao diện người dùng của ứng dụng, cho phép hiển thị dữ liệu động từ các controller.
   * **Vai trò**: Dùng để tạo các trang HTML cho các chức năng như đăng công việc, ứng tuyển, tìm kiếm công việc, và gợi ý công việc cho ứng viên.
2. **CSS (Có thể dùng Bootstrap hoặc Tailwind CSS)**
   * **Mô tả**: CSS được sử dụng để định dạng và tạo kiểu cho giao diện người dùng. Các framework như Bootstrap hoặc Tailwind CSS có thể được sử dụng để giúp thiết kế giao diện responsive và đẹp mắt.
   * **Vai trò**: Cung cấp giao diện người dùng đẹp và dễ sử dụng, giúp hiển thị thông tin công việc và ứng viên một cách rõ ràng.

### Cơ Sở Dữ Liệu

1. **MySQL**
   * **Mô tả**: MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, sử dụng ngôn ngữ SQL để thao tác với dữ liệu.
   * **Vai trò**: Cơ sở dữ liệu chính lưu trữ thông tin công việc, ứng viên, và đơn ứng tuyển. Dữ liệu sẽ được ánh xạ vào các bảng qua các entity trong Spring Data JPA.
2. **H2 Database (Chỉ trong môi trường phát triển)**
   * **Mô tả**: H2 là một cơ sở dữ liệu nhúng trong bộ nhớ, được sử dụng trong quá trình phát triển và kiểm thử.
   * **Vai trò**: Được sử dụng trong môi trường phát triển, giúp giảm bớt sự phức tạp khi cấu hình cơ sở dữ liệu ngoài.

### Các Công Cụ và Dịch Vụ Hỗ Trợ

1. **Git**
   * **Mô tả**: Git là hệ thống quản lý phiên bản phân tán giúp theo dõi sự thay đổi của mã nguồn.
   * **Vai trò**: Được sử dụng để quản lý mã nguồn của dự án, giúp các nhà phát triển làm việc cùng nhau và theo dõi lịch sử thay đổi của mã nguồn.
2. **Maven**
   * **Mô tả**: Maven là công cụ quản lý phụ thuộc và xây dựng dự án cho ứng dụng Java.
   * **Vai trò**: Được sử dụng để quản lý các thư viện phụ thuộc trong dự án, cũng như để biên dịch và triển khai ứng dụng.
3. **Docker (Tùy chọn)**
   * **Mô tả**: Docker là một nền tảng container hóa giúp triển khai ứng dụng một cách dễ dàng và nhất quán trên mọi môi trường.
   * **Vai trò**: Nếu cần, Docker có thể được sử dụng để đóng gói ứng dụng và cơ sở dữ liệu vào container, giúp việc triển khai trở nên đơn giản và dễ dàng hơn.

# 3. Triển khai hệ thống

Sau khi phân tích yêu cầu và xác định được các chức năng cần thiết, bước tiếp theo là triển khai hệ thống. Quá trình triển khai bao gồm việc thiết lập cấu trúc dự án, xây dựng các thành phần backend và frontend, cũng như cấu hình hệ thống cơ sở dữ liệu và các dịch vụ liên quan.

## 3.1. Cấu trúc dự án

 **controller/**: Chứa các lớp controller để xử lý các yêu cầu HTTP.

 **model/**: Chứa các lớp entity và model (ví dụ: Job, Applicant, Application).

 **repository/**: Chứa các interface repository để truy cập cơ sở dữ liệu.

 **service/**: Chứa các lớp service để xử lý logic nghiệp vụ.

 **templates/**: Chứa các file Thymeleaf templates cho frontend.

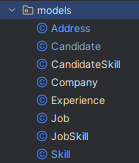
## 3.2. Thiết lập cơ sở dữ liệu

Cơ sở dữ liệu sẽ chứa các bảng liên quan đến công việc, ứng viên và các đơn ứng tuyển.

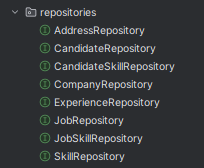
**application.properties**:

spring.application.name=Week05  
spring.datasource.driver-class-name=org.mariadb.jdbc.Driver  
spring.datasource.url=jdbc:mariadb://localhost:3306/works?createDatabaseIfNotExist=true  
spring.datasource.username=root  
spring.datasource.password=root  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
app.rest.path=api/v1  
  
logging.level.root=INFO  
logging.level.org.hibernate.stat=DEBUG  
logging.level.org.hibernate.SQL=DEBUG  
spring.mail.host=smtp.gmail.com  
spring.mail.port=587  
spring.mail.username=thanhnghiem199  
spring.mail.password=b g t i g s r f a v y f h u o c  
spring.mail.properties.mail.smtp.auth=true  
spring.mail.properties.mail.smtp.starttls.enable=true

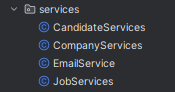
## 3.3. Xây dựng các Entity



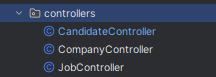
## 3.4. Tạo các Repository



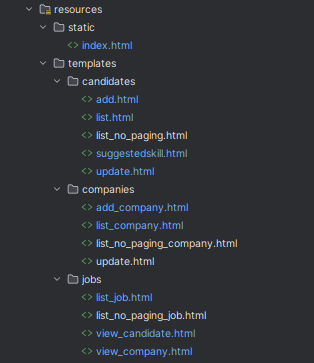
## 3.5. Viết các Service



## 3.6. Tạo các Controller



## 3.7. Frontend với Thymeleaf



## 3.8. Kiểm Tra và Triển Khai

* **Kiểm tra**: Viết các test cases để kiểm tra các chức năng của hệ thống, như đăng công việc, đăng ký ứng viên, tìm kiếm công việc, và gửi email mời ứng viên.
* **Triển khai**: Sau khi phát triển hoàn tất, triển khai ứng dụng lên máy chủ hoặc dịch vụ cloud như **Heroku** hoặc **AWS**.

# 4. Kết quả đạt được

Trong quá trình triển khai và phát triển hệ thống quản lý công việc và ứng viên, hệ thống đã đáp ứng đầy đủ các yêu cầu và mục tiêu đã đề ra từ bài thực hành. Dưới đây là các kết quả đạt được:

## 4.1. Xây dựng thành công hệ thống hỗ trợ công ty và ứng viên:

1. **Quản lý công việc và gợi ý kỹ năng:**
   * **Quản lý công việc**: Hệ thống cho phép công ty đăng tải các công việc với các yêu cầu kỹ năng cụ thể. Các công việc được lưu trữ trong cơ sở dữ liệu và có thể dễ dàng được tìm kiếm, phân trang và lọc theo các kỹ năng yêu cầu.
   * **Gợi ý kỹ năng cho ứng viên**: Ứng viên có thể đăng ký tài khoản, nhập thông tin cá nhân và danh sách kỹ năng của mình. Hệ thống sẽ sử dụng các kỹ năng này để gợi ý các công việc phù hợp, giúp ứng viên tìm kiếm các cơ hội nghề nghiệp tốt hơn.
2. **Hiển thị danh sách ứng viên phân trang hiệu quả:**
   * **Phân trang ứng viên**: Danh sách ứng viên được phân trang để người dùng (công ty) có thể dễ dàng duyệt qua các ứng viên, tìm kiếm và lựa chọn ứng viên phù hợp cho các công việc. Việc phân trang này giúp tối ưu hóa hiệu suất, đặc biệt khi số lượng ứng viên tăng lên.
   * **Tìm kiếm và lọc**: Công ty có thể tìm kiếm ứng viên dựa trên các tiêu chí như kỹ năng, vị trí, hoặc kinh nghiệm, giúp giảm thiểu thời gian tìm kiếm và nâng cao hiệu quả trong việc tuyển dụng.

## 4.2. Đáp ứng các yêu cầu từ bài thực hành:

1. **Các enities và cơ sở dữ liệu**:
   * Các entity được thiết kế và ánh xạ thành công vào cơ sở dữ liệu, bao gồm các bảng cho công việc, ứng viên và các đơn ứng tuyển. Mối quan hệ giữa các entity được xử lý chính xác, đảm bảo sự đồng bộ và tính toàn vẹn của dữ liệu.
2. **Repository và Service**:
   * Các repository interface được tạo và sử dụng hiệu quả để truy vấn dữ liệu từ cơ sở dữ liệu. Các lớp service đã xử lý các logic nghiệp vụ một cách mạch lạc, từ việc quản lý công việc, ứng viên, đến việc gợi ý công việc và gửi email.
3. **Tạo các trang web cho phép công ty đăng tin tuyển dụng**:
   * Các trang web (dựa trên Thymeleaf) đã được xây dựng cho phép công ty dễ dàng đăng tải các tin tuyển dụng, cũng như hiển thị danh sách công việc, ứng viên, và các thông tin liên quan. Giao diện web đơn giản, dễ sử dụng, hỗ trợ công ty và ứng viên trong việc tìm kiếm và ứng tuyển công việc.
4. **Gợi ý công việc phù hợp cho ứng viên**:
   * Ứng viên khi đăng nhập vào hệ thống sẽ nhận được các gợi ý công việc dựa trên các kỹ năng mà họ đã cung cấp. Điều này giúp ứng viên tìm kiếm các cơ hội nghề nghiệp phù hợp và tiết kiệm thời gian trong việc duyệt qua các công việc không liên quan.
5. **Gửi email mời cho các ứng viên phù hợp**:
   * Sau khi công ty chọn được các ứng viên tiềm năng, hệ thống hỗ trợ gửi email mời ứng viên tham gia phỏng vấn hoặc ứng tuyển. Quá trình gửi email đã được tự động hóa qua JavaMail API, giúp tiết kiệm thời gian cho công ty trong việc tiếp cận ứng viên.
6. **Đề xuất các kỹ năng cần học cho ứng viên**:
   * Hệ thống có thể đề xuất các kỹ năng mà ứng viên chưa có nhưng có thể hữu ích cho công việc tương lai. Điều này giúp ứng viên biết được các kỹ năng cần bổ sung để nâng cao cơ hội nghề nghiệp của mình.

## 4.3. HIển thị giiao diện

### 1. Trang chủ

### 2. Candidates

#### 2.1. Suggest Skills

#### 2.2. Add New Candidate

#### 2.3. Edit Candidate

### 3. Companies

#### 3.1. Add new Company

#### 3.2. Edit Company

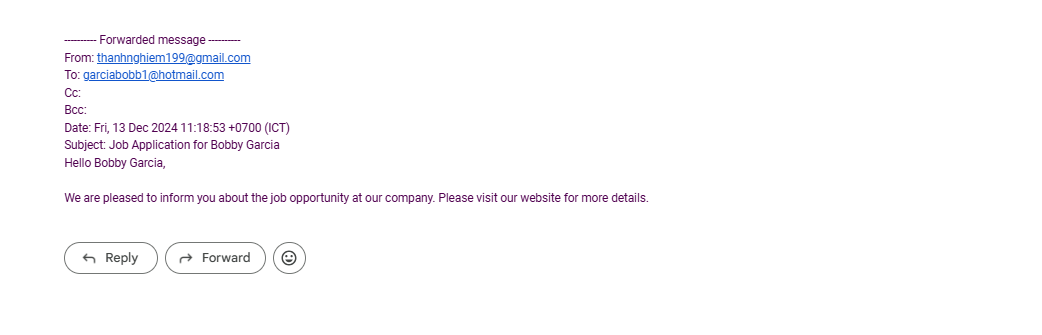
### 4. Jobs

#### 4.1. Add New Job

#### 4.2. View Companies

#### 4.3. View Candidate for Job

### Gửi Email



# . 5. Kết luận

## 5.1. Đánh giá

Hệ thống quản lý công việc và ứng viên đã hoàn thiện về cơ bản và đáp ứng đầy đủ các yêu cầu đề ra. Trong quá trình triển khai, các kỹ thuật quan trọng của Spring Boot như Spring Data JPA, Spring Security, và Thymeleaf đã được áp dụng thành công, giúp xây dựng một hệ thống vững chắc và dễ mở rộng. Các tính năng chính như:

* Quản lý công việc và ứng viên,
* Gợi ý công việc cho ứng viên dựa trên kỹ năng,
* Gửi email mời ứng viên,
* Phân trang danh sách ứng viên,

Được triển khai hiệu quả và hoạt động tốt. Hệ thống cũng hỗ trợ các chức năng cơ bản cần thiết cho cả công ty và ứng viên trong việc tìm kiếm công việc và ứng tuyển. Các thao tác trong hệ thống được thực hiện nhanh chóng và mượt mà, cung cấp một trải nghiệm người dùng hợp lý.

## 5.2. Hướng phát triển

Dù hệ thống đã hoàn thiện về mặt chức năng, vẫn có một số hướng phát triển và cải tiến có thể giúp nâng cao tính năng và hiệu suất của ứng dụng trong tương lai:

1. **Nâng cấp giao diện với các framework hiện đại như React**:
   * **Mô tả**: Giao diện người dùng hiện tại sử dụng Thymeleaf, nhưng có thể nâng cấp lên các framework frontend hiện đại như React để tạo ra một trải nghiệm người dùng linh hoạt và động hơn. React sẽ giúp tăng tính tương tác và mượt mà của giao diện, đặc biệt khi làm việc với dữ liệu lớn hoặc cần thao tác thường xuyên.
   * **Lợi ích**: React giúp dễ dàng quản lý giao diện động, tối ưu hóa hiệu suất bằng cách sử dụng Virtual DOM, và nâng cao trải nghiệm người dùng với các hiệu ứng chuyển động mượt mà. Đồng thời, React có thể dễ dàng tích hợp với backend Spring Boot qua RESTful API.
2. **Thêm chức năng nâng cao như báo cáo thống kê**:
   * **Mô tả**: Thêm các chức năng báo cáo thống kê cho phép công ty và ứng viên có cái nhìn rõ ràng hơn về hoạt động của hệ thống, ví dụ như số lượng công việc đã đăng, số lượng ứng viên đã ứng tuyển, tỷ lệ chấp nhận ứng viên, các kỹ năng được yêu cầu nhiều nhất, v.v.
   * **Lợi ích**: Các báo cáo thống kê sẽ giúp các nhà tuyển dụng theo dõi tiến độ tuyển dụng và cải thiện chiến lược tuyển dụng, trong khi ứng viên có thể nắm được xu hướng thị trường và cải thiện kỹ năng của mình.
3. **Xác thực bảo mật nâng cao**:
   * **Mô tả**: Cải thiện các tính năng bảo mật của hệ thống bằng cách triển khai xác thực bảo mật mạnh mẽ hơn, chẳng hạn như xác thực hai yếu tố (2FA), mã hóa mật khẩu, hoặc sử dụng OAuth2 cho các phương thức đăng nhập bên ngoài.
   * **Lợi ích**: Bảo mật là yếu tố quan trọng đối với mọi ứng dụng web. Việc triển khai xác thực bảo mật nâng cao sẽ giúp bảo vệ dữ liệu người dùng và ngăn chặn các cuộc tấn công mạng như tấn công brute-force hay tấn công giả mạo. Điều này cũng giúp hệ thống đạt tiêu chuẩn bảo mật cao hơn, đảm bảo sự tin tưởng của người dùng.
4. **Tích hợp công nghệ học máy (Machine Learning)**:
   * **Mô tả**: Hệ thống có thể tích hợp các thuật toán học máy để cải thiện khả năng gợi ý công việc cho ứng viên. Ví dụ, sử dụng phân tích dữ liệu để dự đoán công việc mà ứng viên có thể quan tâm dựa trên lịch sử tìm kiếm, ứng tuyển, hoặc các yếu tố khác.
   * **Lợi ích**: Tích hợp học máy sẽ giúp tăng độ chính xác trong việc gợi ý công việc cho ứng viên, đồng thời tối ưu hóa quá trình tuyển dụng cho công ty, nâng cao hiệu quả và giảm thiểu thời gian tìm kiếm công việc và ứng viên phù hợp.