



## ĐỒ ÁN CƠ SỞ NGÀNH

ĐỀ TÀI

# XÂY DỰNG WEBSITE TÌM KIẾM VIỆC LÀM BẰNG REACTJS

*Giáo viên hướng dẫn:*  
TS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*  
Họ tên: Đinh Tấn Mãi  
MSSV: 110121063  
Lớp: DA21TTB

# Nội dung

**01** Giới thiệu

**02** Nghiên cứu lý thuyết

**03** Xây dựng website

**04** Kết luận và hướng phát triển

01

# GIỚI THIỆU

## 01 Giới thiệu



Lý do :

- Tình hình xa thải nhân viên hiện nay đang rất phức tạp.
- Nhu cầu tìm kiếm việc làm ngày càng tăng cao.
- Nhiều người làm trái ngành.
- Tìm kiếm việc làm phù hợp với năng lực và sở thích của bản thân khó khăn.

=> Xây dựng website tìm kiếm việc làm là một giải pháp hữu ích giúp giải quyết tình trạng trên.

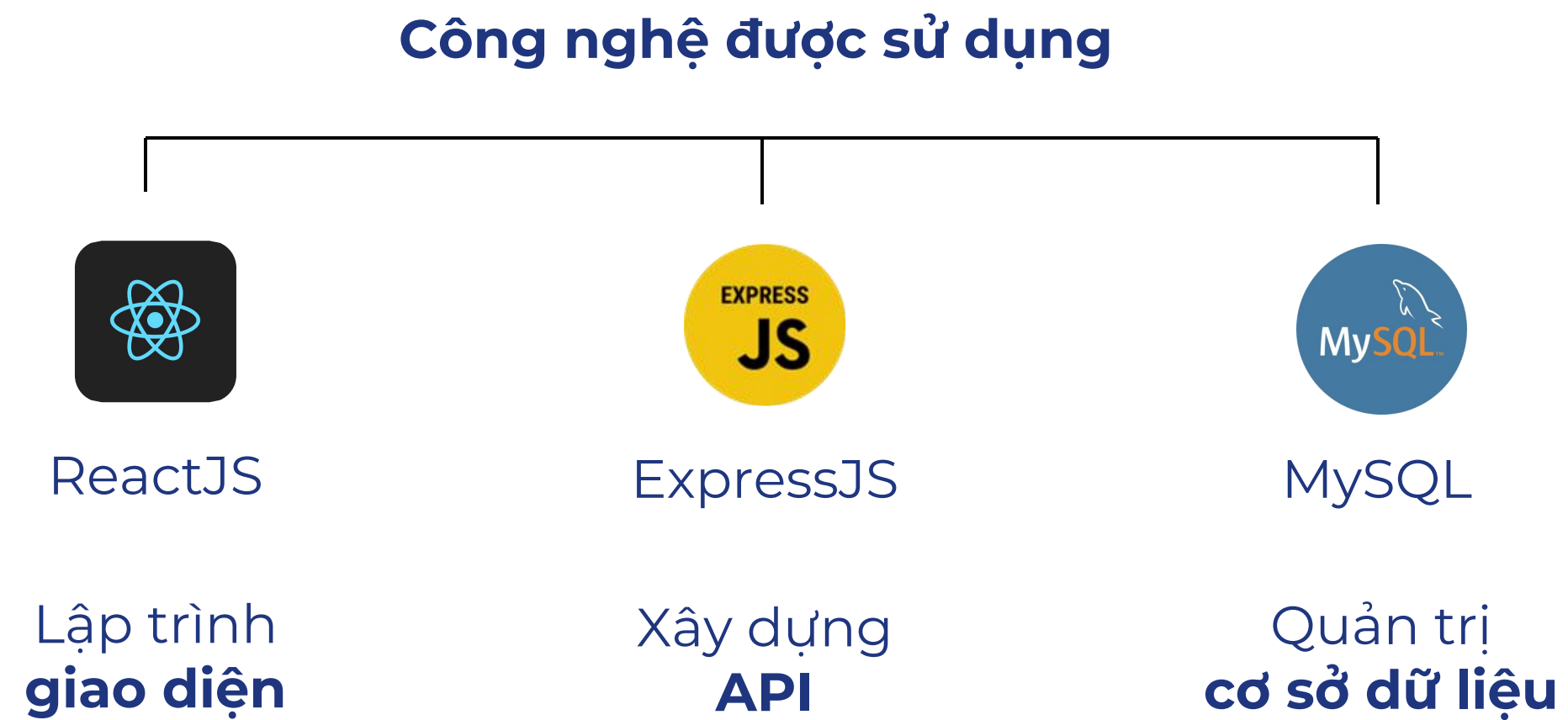
## 01 Giới thiệu



Mục tiêu :

- Tạo ra một website tìm kiếm việc làm sử dụng ReactJS.
- Cung cấp một giao diện người dùng thân thiện và dễ sử dụng.
- Tối ưu hóa trải nghiệm người dùng và tăng cường khả năng tìm kiếm công việc.
- Cho phép dùng đăng ký tài khoản, tìm kiếm việc làm theo các tiêu chí khác nhau, nộp hồ sơ xin việc và theo dõi tình trạng hồ sơ.
- Cho phép nhà tuyển dụng đăng tin tuyển dụng, quản lý tin tuyển dụng và lọc hồ sơ ứng viên.

## 01 Giới thiệu



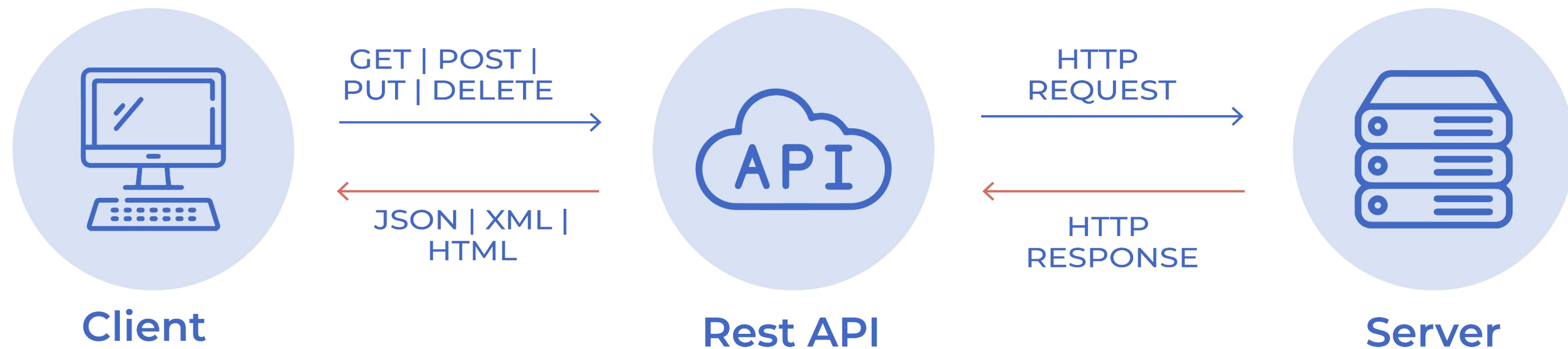
02

# Nghiên cứu lý thuyết



## 02 Nghiên cứu lý thuyết

### Cấu trúc Rest API





## 02 Nghiên cứu lý thuyết



### Frontend

Frontend sử dụng ReactJS, một thư viện Javascript. ReactJS giúp xây dựng giao diện người dùng linh hoạt và hiệu suất cao.



### Backend

Backend sử dụng ExpressJS, cung cấp một cách linh hoạt và đơn giản để xây dựng các ứng dụng web và API.



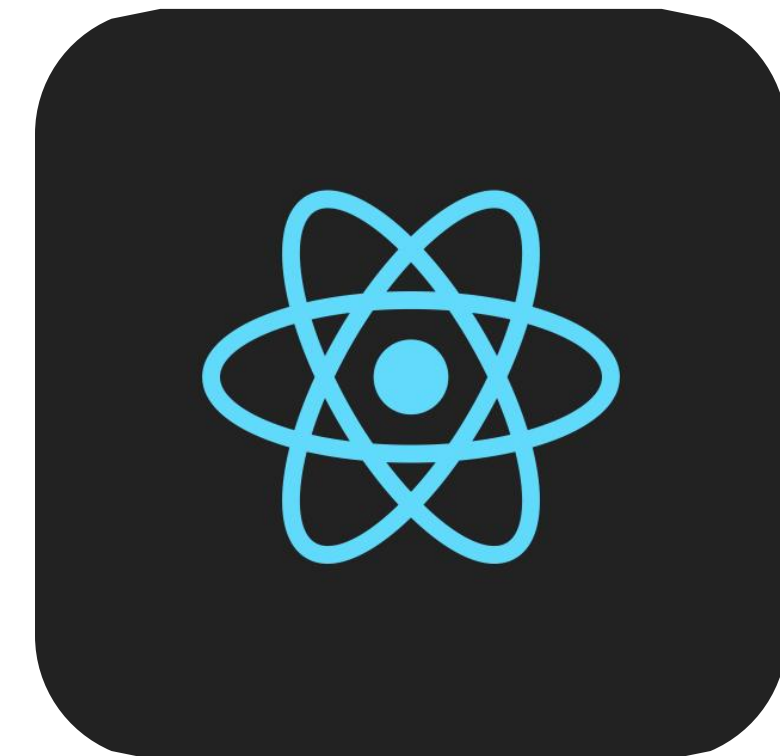
### Xác thực JWT

JWT được sử dụng để xác thực người dùng và chuyển thông tin xác thực giữa client và server một cách an toàn và nhẹ nhàng.

## 02 Nghiên cứu lý thuyết

ReactJS là gì ?

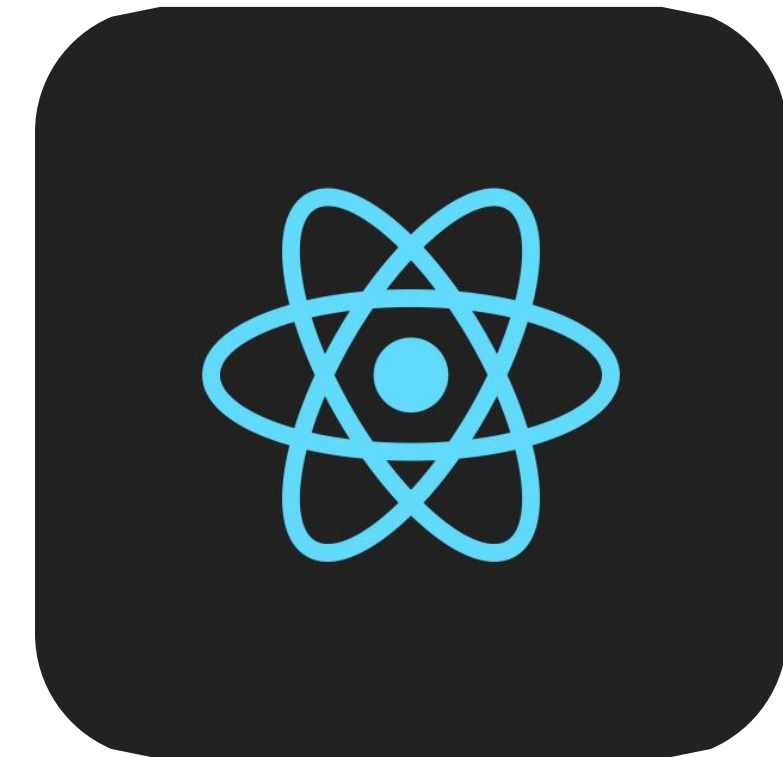
ReactJS là một thư viện JavaScript phổ biến được phát triển bởi Meta, nó chú trọng vào việc xây dựng giao diện người dùng. Sự linh hoạt và hiệu suất của ReactJS làm cho nó trở thành sự lựa chọn hàng đầu cho việc phát triển các ứng dụng web.



## 02 Nghiên cứu lý thuyết

Ưu điểm của ReactJS:

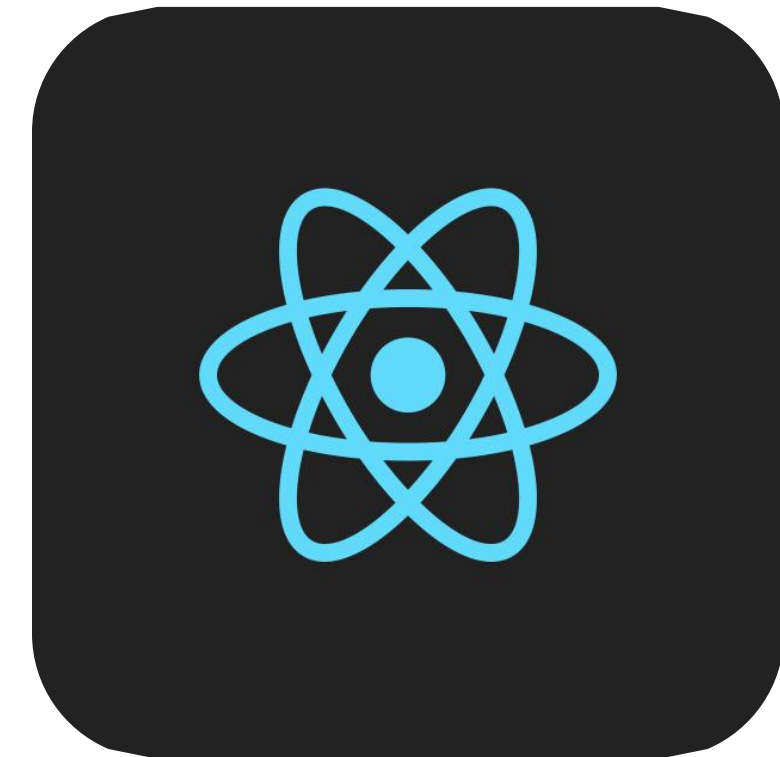
- Tái sử dụng thành phần: giảm độ phức tạp và tăng cường khả năng duy trì.
- Virtual DOM: cải thiện hiệu suất và tối ưu hóa lại giao diện người dùng.
- Thân thiện với SEO: SEO là một phần không thể thiếu để đưa thông tin website của lên top đầu tìm kiếm của Google.



## 02 Nghiên cứu lý thuyết

Một số khái niệm trong ReactJS :

- **Component:** Component là một thành phần giao diện xây dựng cơ bản của React.
- **JSX:** JSX là một phần cú pháp mở rộng của JavaScript, cho phép viết mã HTML bên trong mã JavaScript.
- **State:** Dùng để lưu trữ và quản lý trạng thái của một component.
- **Props:** là các thuộc tính được truyền từ component cha xuống component con, giúp truyền dữ liệu giữa các component.
- **React Router:** Thư viện giúp quản lý điều hướng trong ứng dụng React.

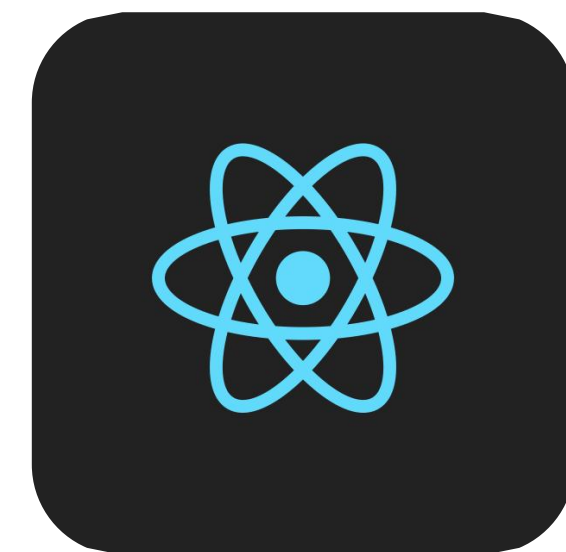


## 02 Nghiên cứu lý thuyết

Kết hợp ReactJS với Sass:

Sass là một ngôn ngữ mở rộng của CSS, được thiết kế để làm cho việc viết mã CSS trở nên dễ dàng hơn và có tổ chức tốt hơn. Sass được biên dịch thành CSS thông thường và hỗ trợ nhiều tính năng và cú pháp mở rộng so với CSS truyền thống. Một số ưu điểm của Sass:

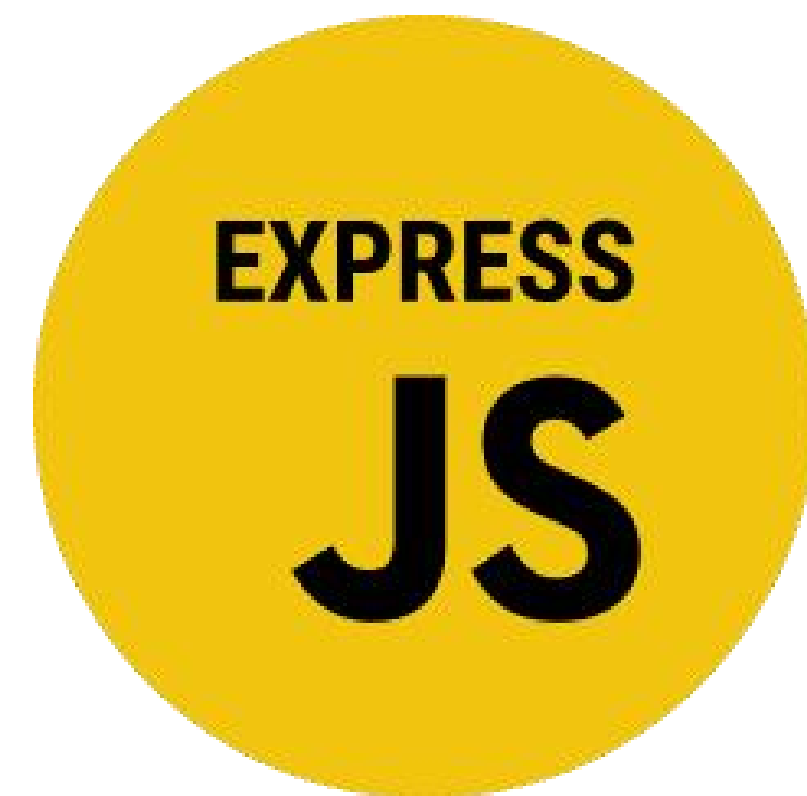
- Biến:** Sass cho phép sử dụng biến để lưu trữ giá trị, giúp tái sử dụng và quản lý dễ dàng.
- Lồng ghép:** giúp mã nguồn trở nên dễ đọc hơn và tổ chức tốt hơn.
- **Mixins:** Cho phép tái sử dụng đoạn mã CSS, giúp giảm độ phức tạp và lặp lại trong mã nguồn.
- Sass hỗ trợ các chỉ thị điều khiển như **@if**, **@for**, và **@each**, giúp kiểm soát dòng mã nguồn tùy thuộc vào điều kiện.



## 02 Nghiên cứu lý thuyết

ExpressJS là gì ?

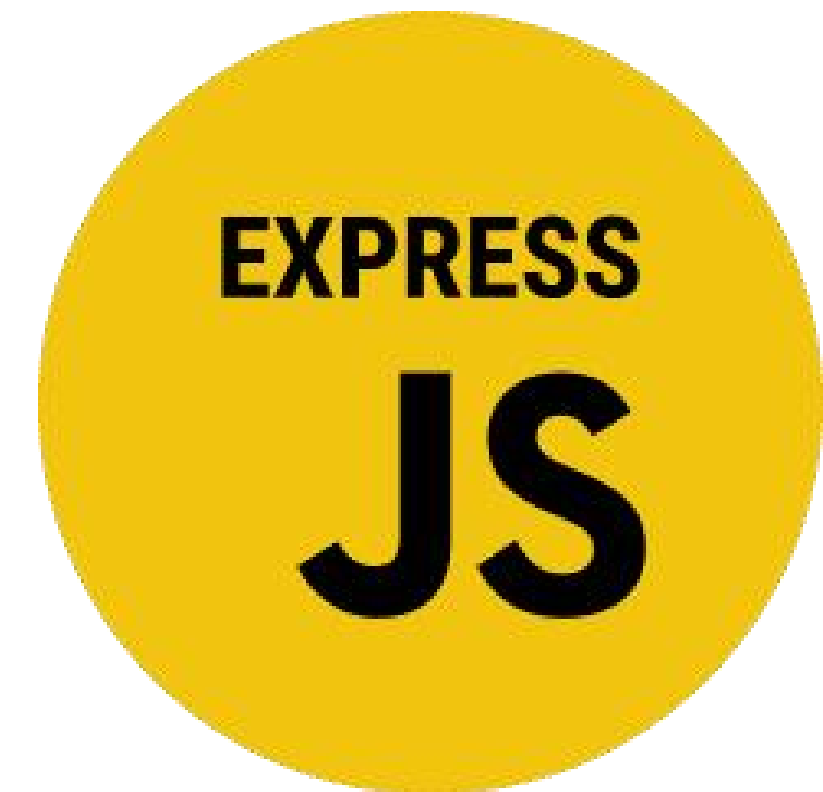
- ExpressJS là một framework của Node.js, cung cấp một cách linh hoạt và đơn giản để xây dựng các ứng dụng web và API.
- Cho phép lập trình viên sử dụng JavaScript để viết mã cho cả phía server và client.
- Có thể sử dụng tất cả các kiến thức JavaScript để phát triển ứng dụng web.



## 02 Nghiên cứu lý thuyết

Ưu điểm của Express.js :

- Có thể xử lý nhiều yêu cầu cùng một lúc một cách hiệu quả, giúp tăng hiệu suất của ứng dụng web.
- Có thể được mở rộng dễ dàng để đáp ứng nhu cầu phát triển của ứng dụng web.
- Cung cấp một số tính năng bảo mật tích hợp, giúp bảo vệ ứng dụng web của bạn khỏi các cuộc tấn công.
- Dễ học và sử dụng, giúp dễ dàng bắt đầu phát triển ứng dụng web.

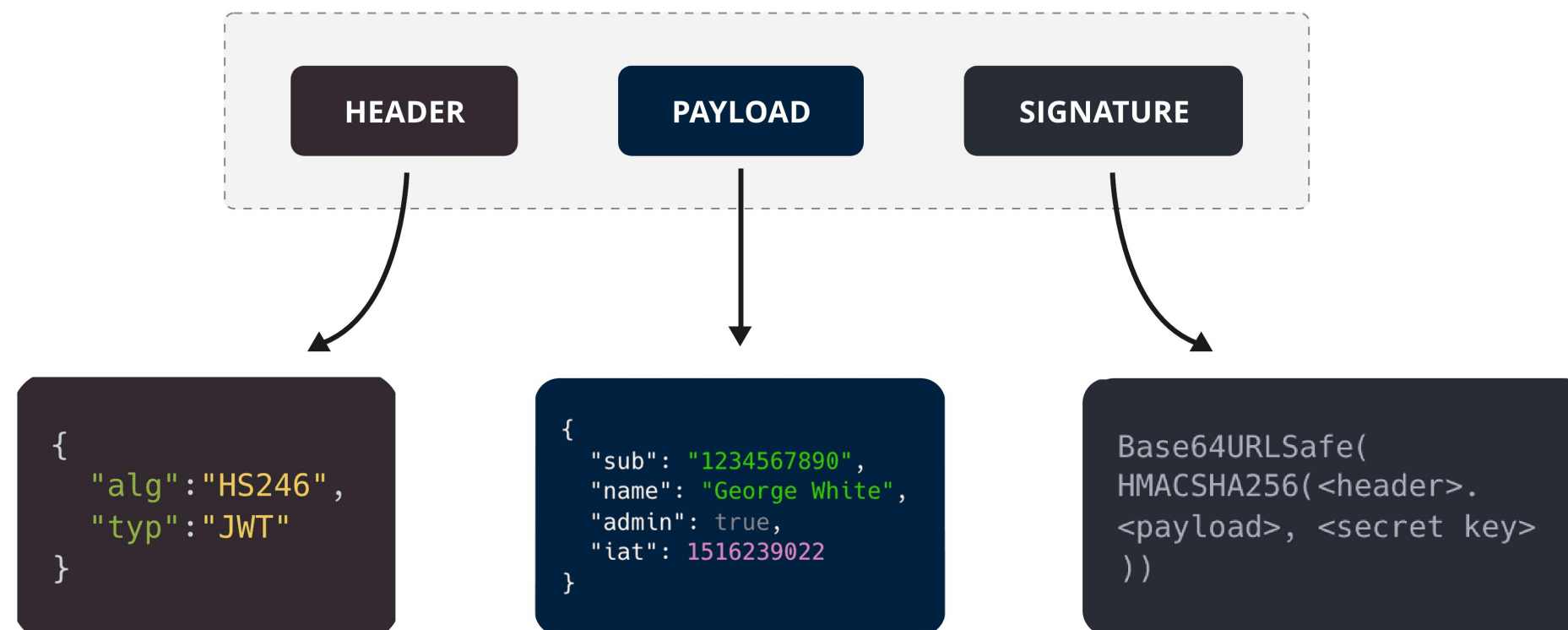




## 02 Nghiên cứu lý thuyết

Xác thực JWT (JSON Web Token ) :

### Structure of a JSON Web Token (JWT)

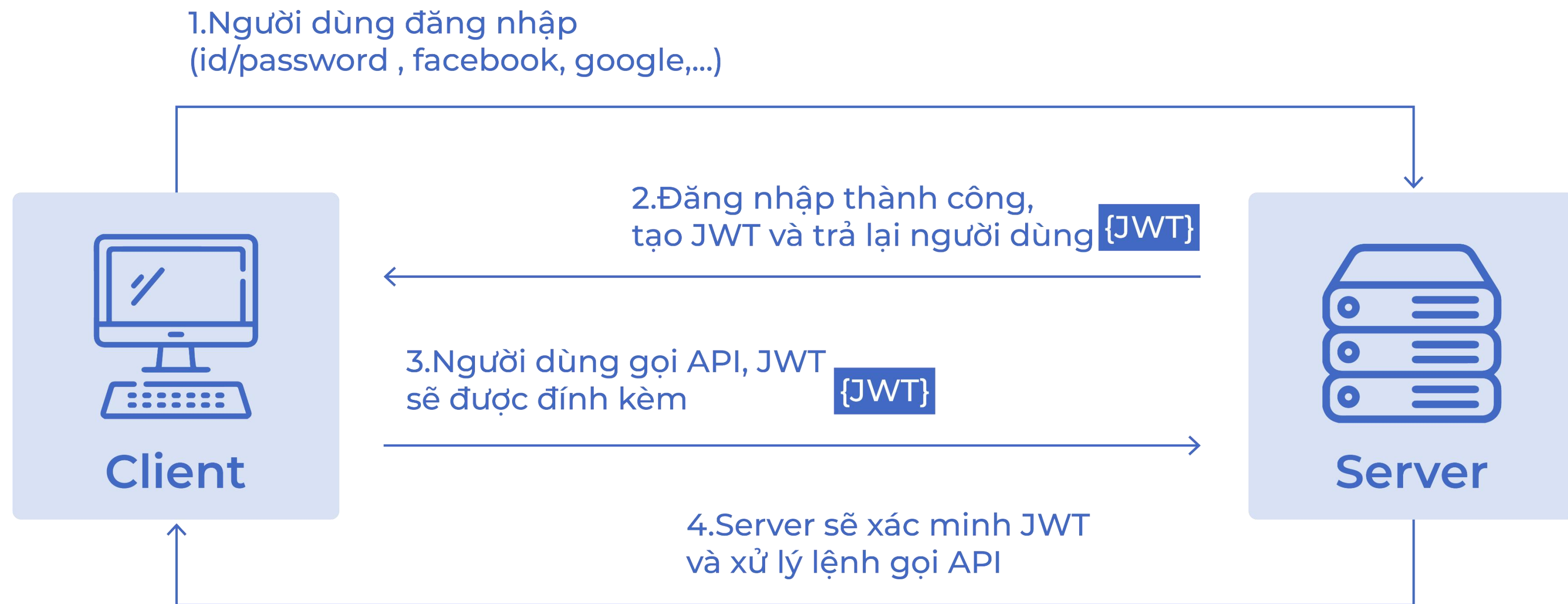


Cấu trúc JWT

JWT (JSON Web Token ) cung cấp khả năng xác thực và đánh dấu tin cậy thông qua “chữ ký”, một chuỗi JSON được mã hóa bằng phương pháp mã hóa, trở thành một chuỗi ký tự lộn xộn. JWT bao gồm ba phần chính: **Header**, **Payload**, và **Signature**.

## 02 Nghiên cứu lý thuyết

Cơ chế hoạt động của JWT :



03

# Xây dựng

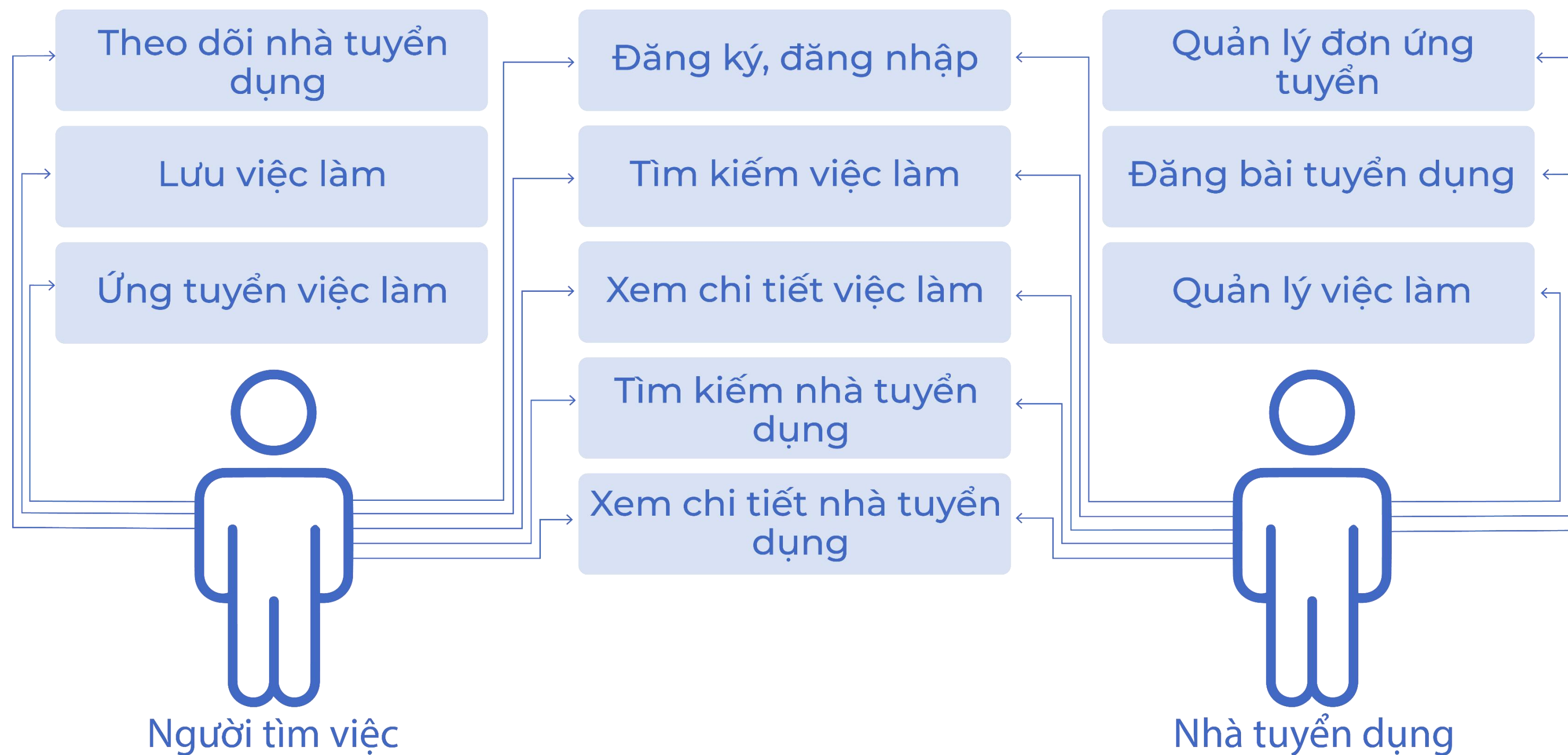
## 03 Xây dựng



Sơ đồ quy trình xây dựng ứng dụng

## 03 Xây dựng

### Xác định chức năng chính :



## 03 Xây dựng

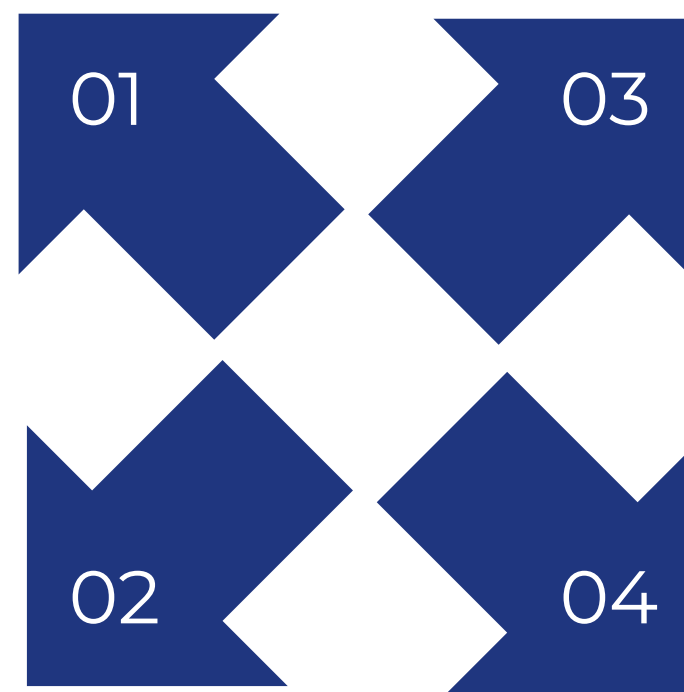
### Yêu cầu phi chức năng :

#### Giao diện thân thiện, dễ sử dụng

Giao diện cần được thiết kế đẹp mắt, dễ sử dụng và phù hợp với người dùng, màu sắc, font chữ phù hợp.

#### Giao diện responsive

Giao diện responsive để có thể sử dụng trên các thiết bị di động và các kích thước màn hình khác nhau.



#### Hiệu suất cao và khả dụng

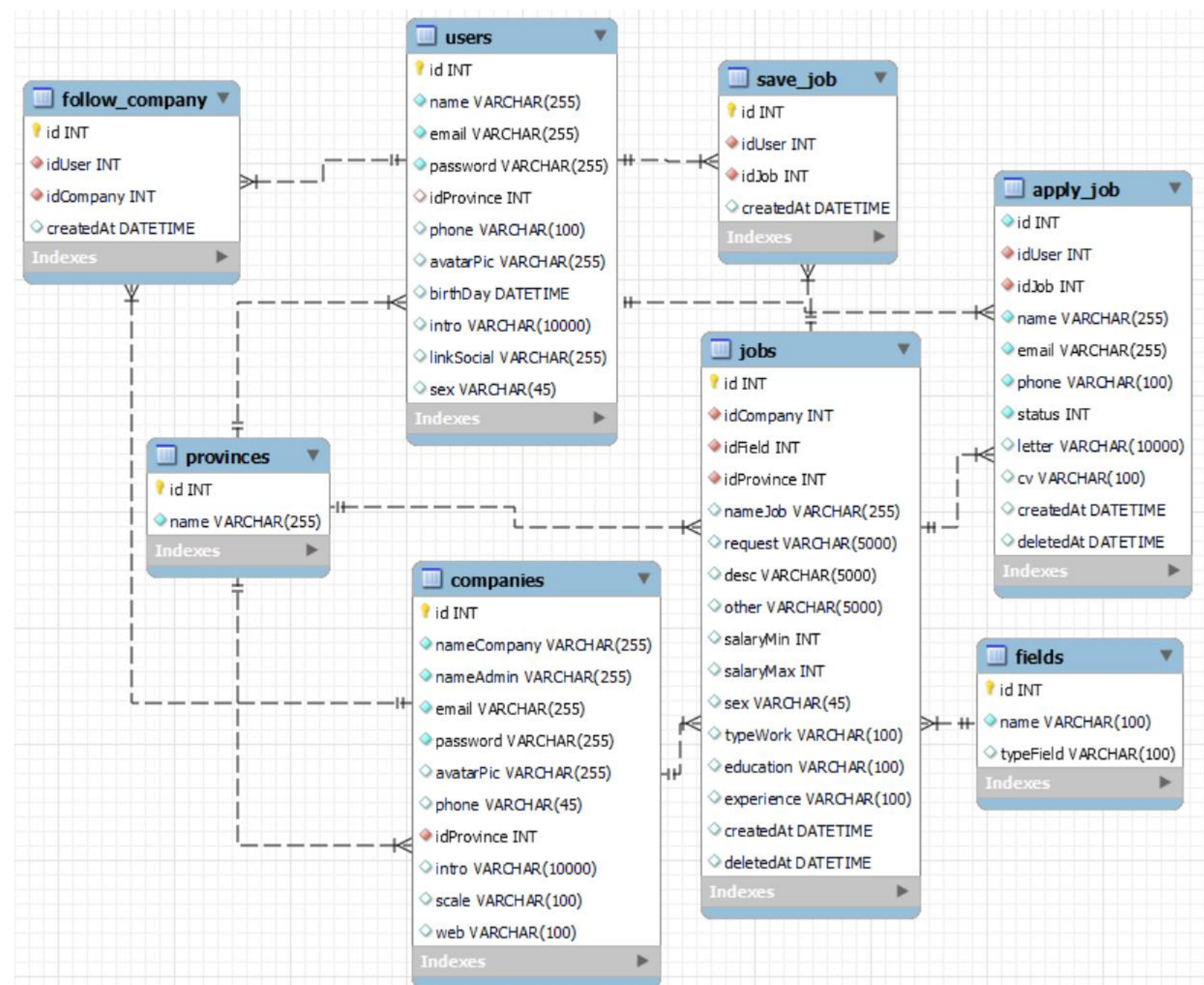
Có thể sử dụng một cách mượt mà, hoạt động ổn định, không bị gián đoạn.

#### Khả năng mở rộng

Cho phép mở rộng thêm các tính năng mới hoặc tăng quy mô hoạt động.



## 03 Xây dựng Thiết kế cơ sở dữ liệu :



Lược đồ cơ sở dữ liệu.



## 03 Xây dựng

### Thiết kế giao diện :

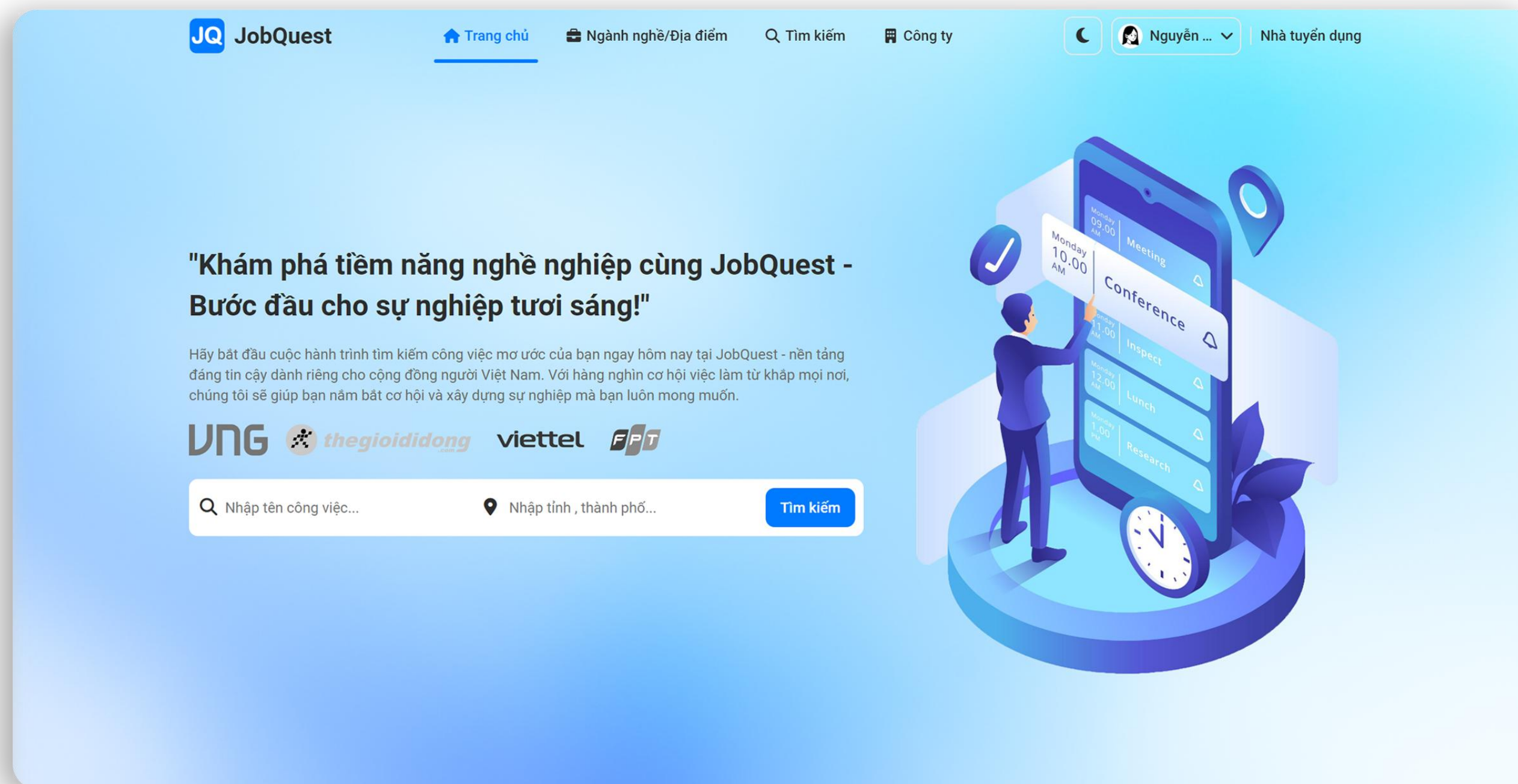
- **Thư mục Components** : chứa các thành phần của ReactJS có thể tái sử dụng được. Các thành phần này được chia thành các thư mục con theo chức năng hoặc tính năng của chúng.
- **Thư mục Pages**: chứa các trang cụ thể của ứng dụng.
- **Thư mục Layout** sẽ chứa bố cục dùng chung của ứng dụng.
- **Tập tin App.js**: thành phần gốc của ứng dụng ReactJS. Tập tin này sẽ import thư viện React và các thành phần cần thiết khác.

```
1  src/  
2    components/  #  
3      Header.js  
4      Footer.js  
5      Sidebar.js  
6      ...  
7    pages/  
8      Home.js  
9      Job.js  
10     Search.js  
11     ...  
12    layout/  
13      MainLayout.js  
14     ...  
15    App.js
```

Cấu trúc thư mục ReactJS

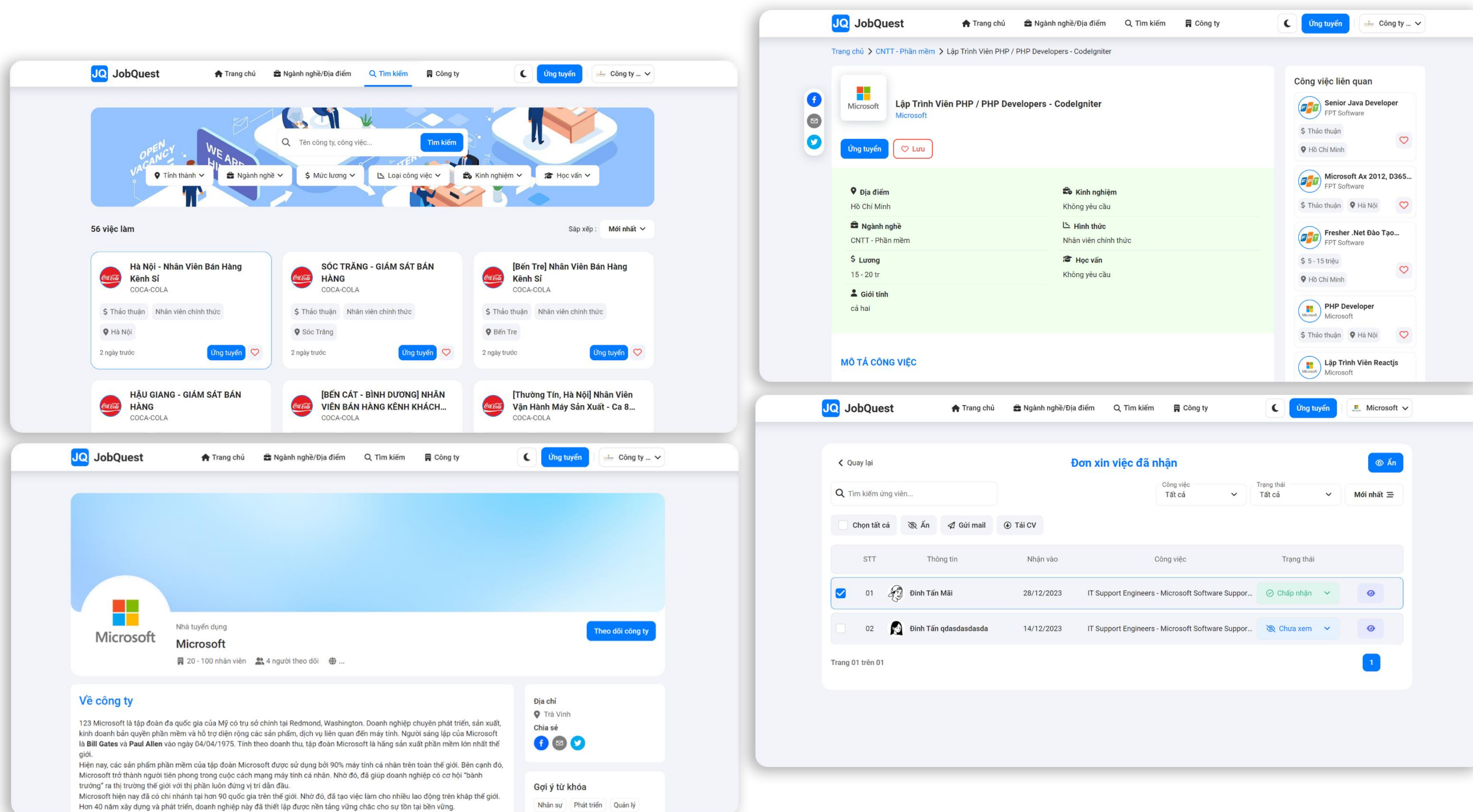
## 03 Xây dựng

### Một số giao diện của chương trình :



Giao diện trang chủ

## 03 Xây dựng



Một số giao diện khác



## 03 Xây dựng

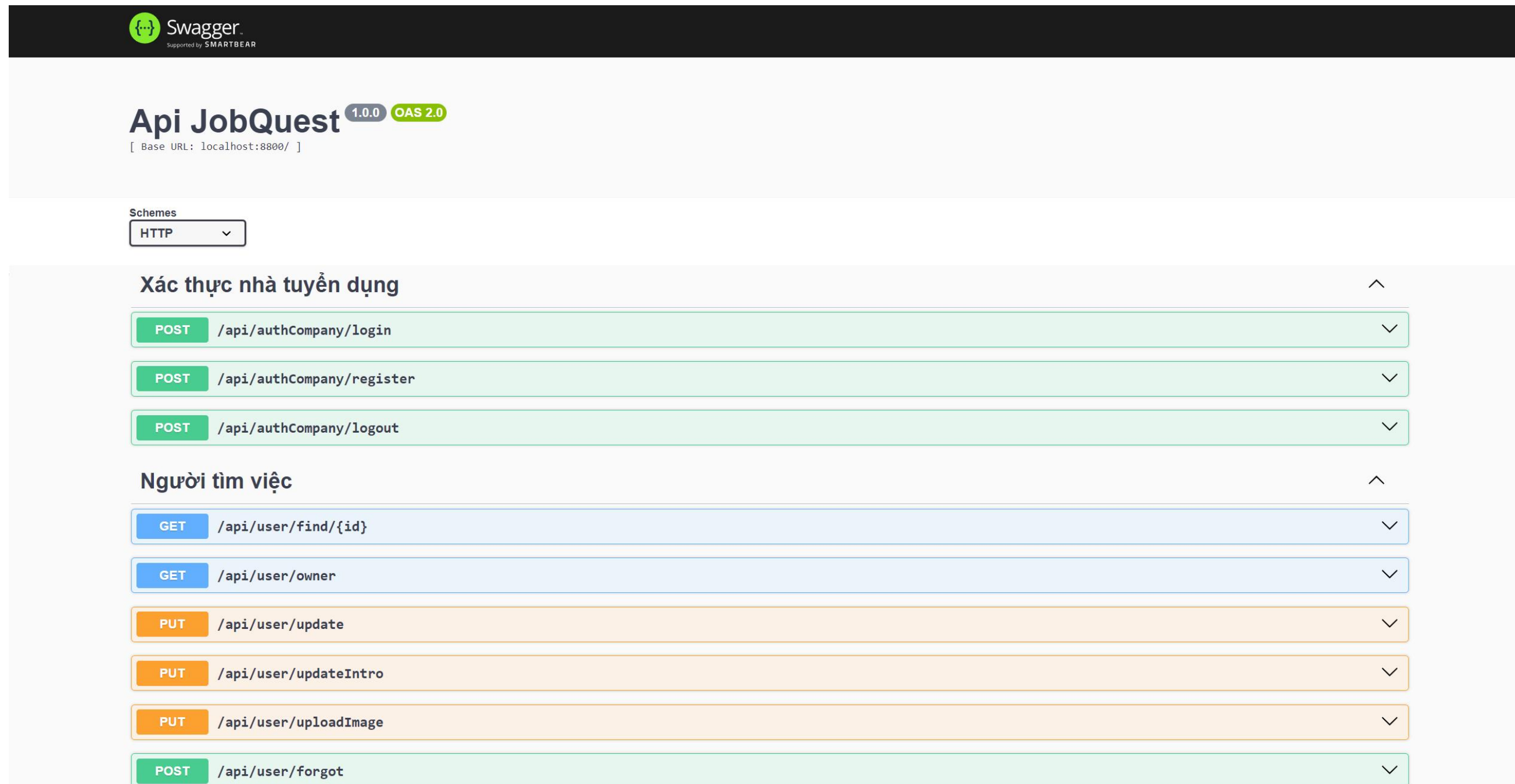
### Thiết kế API :

- Tập tin index.js:** các API sẽ được định nghĩa trong tập tin index.js để xử lý các yêu cầu từ phía người dùng gửi lên.
- Thư mục routes:** Mỗi tập tin route sẽ định tuyến các URL và gọi hàm xử lý tương ứng .
- Thư mục controllers:** Mỗi tập tin controller sẽ chứa các hàm xử lý các yêu cầu của người dùng liên quan đến một chức năng cụ thể.
- Thư mục middlewares:** Mỗi tập tin middlewares sẽ kiểm tra giá trị đầu vào trước khi xử lý.

```
1 controllers/  
2   user.controller.js  
3   company.controller.js  
4   job.controller.js  
5   ...  
6 middlewares/  
7   checkEmail.middleware.js  
8   checkPassword.middleware.js  
9   ...  
10 routes/  
11   user.routes.js  
12   company.routes.js  
13   ...  
14 index.js
```

Cấu trúc thư mục API

## 03 Xây dựng Thiết kế API :



The image shows the Swagger UI for an API named "Api JobQuest". The UI is dark-themed with a black header bar containing the Swagger logo and "Supported by SMARTBEAR". Below the header, the API title "Api JobQuest" is displayed with version tags "1.0.0" and "OAS 2.0". The base URL is "[ Base URL: localhost:8800/ ]". A "Schemes" dropdown menu is set to "HTTP". The API endpoints are organized into two sections: "Xác thực nhà tuyển dụng" (Employer authentication) and "Người tìm việc" (Job seekers). The "Xác thực nhà tuyển dụng" section contains three POST endpoints: "/api/authCompany/login", "/api/authCompany/register", and "/api/authCompany/logout". The "Người tìm việc" section contains six endpoints: two GET endpoints ("/api/user/find/{id}" and "/api/user/owner"), three PUT endpoints ("/api/user/update", "/api/user/updateIntro", and "/api/user/uploadImage"), and one POST endpoint ("/api/user/forgot"). Each endpoint is represented by a colored bar with the HTTP method, the path, and a dropdown arrow.

Swagger  
Supported by SMARTBEAR

Api JobQuest 1.0.0 OAS 2.0  
[ Base URL: localhost:8800/ ]

Schemes  
HTTP

Xác thực nhà tuyển dụng ^

- POST /api/authCompany/login
- POST /api/authCompany/register
- POST /api/authCompany/logout

Người tìm việc ^

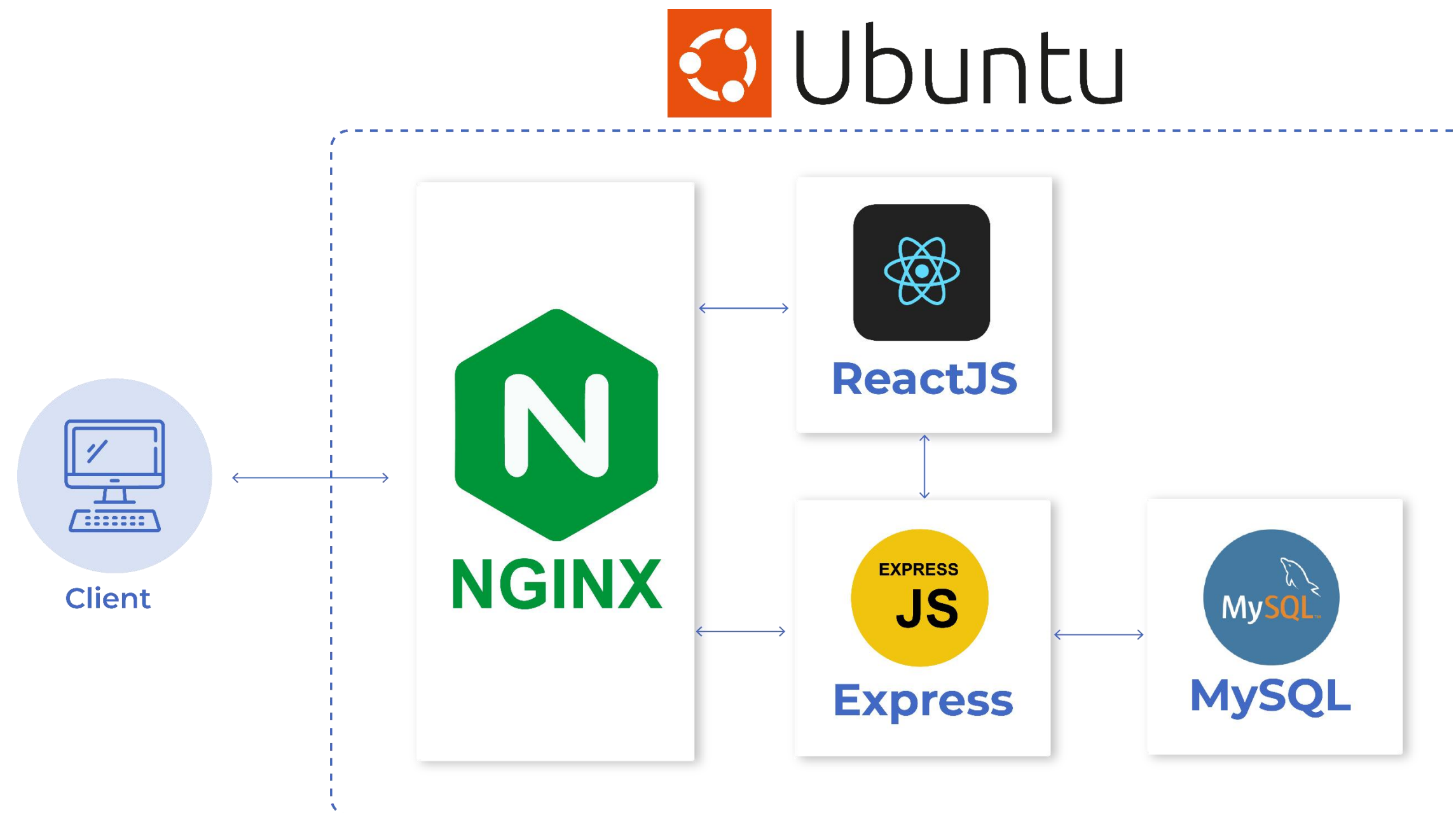
- GET /api/user/find/{id}
- GET /api/user/owner
- PUT /api/user/update
- PUT /api/user/updateIntro
- PUT /api/user/uploadImage
- POST /api/user/forgot

Giao diện API

## 03 Xây dựng

### Triển khai ứng dụng :

Ứng dụng được triển khai lên máy chủ VPS với cấu hình 2 core CPU, 2 GB RAM, 40GB SSD.



04

# Kết luận và hướng phát triển



## 04 Kết luận và hướng phát triển

### Kết quả đạt được :

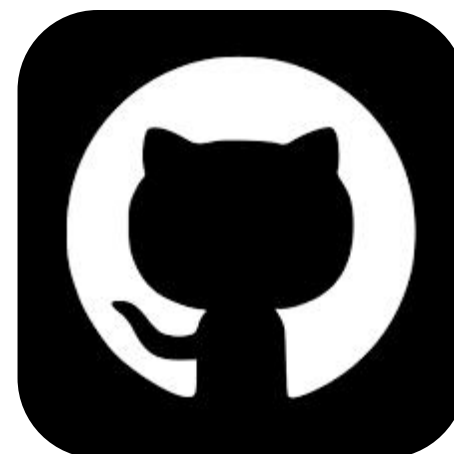
- ✓ Tạo ra một website tìm kiếm việc làm sử dụng ReactJS.
- ✓ Cung cấp một giao diện người dùng thân thiện và dễ sử dụng.
- ✓ Tối ưu hóa trải nghiệm người dùng và tăng cường khả năng tìm kiếm công việc.
- ✓ Cho phép dùng đăng ký tài khoản, tìm kiếm việc làm theo các tiêu chí khác nhau, nộp hồ sơ xin việc và theo dõi tình trạng hồ sơ.
- ✓ Cho phép nhà tuyển dụng đăng tin tuyển dụng, quản lý tin tuyển dụng và lọc hồ sơ ứng viên.

## 04 Kết luận và hướng phát triển

### **Hướng phát triển :**

- Phát triển ứng dụng đa nền tảng.
- Hệ thống thông báo và gợi ý.
- Xây dựng trang admin quản lý website.
- Thực hiện chức năng chat và phỏng vấn trực tuyến.
- Thêm nhiều lựa chọn đăng nhập.
- Thêm chức năng đánh giá nhà tuyển dụng và đánh giá ứng viên.
- Xác thực nhà tuyển dụng.

Toàn bộ mã nguồn của dự án  
được lưu trữ và quản lý trên  
Github



 [github.com/tanmaiii/csn-da2lttb-dinh-tanmai-tim-viec-nodejs](https://github.com/tanmaiii/csn-da2lttb-dinh-tanmai-tim-viec-nodejs)

Ứng dụng được triển khai với  
địa chỉ



 <https://jobquest.tanmai.id.vn>

# CẢM ƠN THẦY CÔ

Rất mong sự đóng góp từ thầy cô

**Demo** >

 ĐINH TẤN MÃI