TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**DỰ ÁN CÔNG NGHỆ THÔNG TIN 1**

**XÂY DỰNG TRANG TÌM KIẾM VIỆC LÀM DỰA TRÊN NỀN TẢNG ANGULAR, NODEJS**

*Giảng viên giám sát:* **MAI VĂN MẠNH**

*Người hướng dẫn*: **BÙI THANH PHƯƠNG**

*Người thực hiện*: **ĐÀO CẨM AN - 51603386**

Lớp **: 16050302**

Khoá  **: 20**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**DỰ ÁN CÔNG NGHỆ THÔNG TIN 1**

**XÂY DỰNG TRANG TÌM KIẾM VIỆC LÀM DỰA TRÊN NỀN TẢNG ANGULAR, NODEJS**

*Giảng viên giám sát:* **MAI VĂN MẠNH**

*Người hướng dẫn*: **BÙI THANH PHƯƠNG**

*Người thực hiện*: **ĐÀO CẨM AN - 51603386**

Lớp **: 16050302**

Khoá  **: 20**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2019**

LỜI CẢM ƠN

Em xin chân thành cảm ơn khoa Công nghệ thông tin trường Đại học Tôn Đức Thắng đã tạo điều kiện để em có cơ hội làm việc với môi trường thực tế.

Em xin cảm ơn các cán bộ hướng dẫn: Bùi Thanh Phương công ty cổ phần AMIT GROUP đã nhiệt tình hướng dẫn, hỗ trợ chúng em trong thời gian vừa qua. Đồng thời, các cán bộ đã tạo môi trường làm việc chuyên nghiệp, năng động giúp chúng em phát huy khả năng của mình, trau dồi thêm kiến thức và kỹ năng thuyết trình. Em chúc các cán bộ luôn khỏe mạnh để tiếp bước cho các thế hệ sau và thành công trong công việc.

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của các cán bộ công ty;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Đào Cẩm An*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Nhu cầu tìm kiếm việc làm bao giờ cũng là vấn đề được xã hội quan tâm trong nhiều thế kỷ. Để đáp ứng nhu cầu đó trong thời đại 4.0, các trang tìm kiếm việc làm xuất hiện nhằm đáp ứng phục vụ, hỗ trợ người dùng tìm kiếm công việc, đăng công việc. Ở đây, em xây dựng trang tìm kiếm việc làm của mình dựa trên các nhu cầu phát sinh của người dùng trong thực tế, đồng thời ứng dụng công nghệ như NodeJS, RestAPI,… vào bài làm của mình.

Trong đợt báo cáo này, em xin trình bày cách xây dựng hệ thống, các công cụ đã dùng và cũng như cách vận hành hệ thống

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc50331884)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iii](#_Toc50331885)

[TÓM TẮT iv](#_Toc50331886)

[MỤC LỤC 1](#_Toc50331887)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 2](#_Toc50331888)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc50331889)

[CHƯƠNG 1 – THIẾT KẾ API (BACKEND) 5](#_Toc50331890)

[1.1 Giới thiệu 5](#_Toc50331891)

[1.2 Các công cụ sử dụng 5](#_Toc50331892)

[1.3 Cơ sở dữ liệu 7](#_Toc50331893)

[1.3.1 Thông tin về cơ sở dữ 7](#_Toc50331894)

[1.3.2 Kết nối cơ sở dữ liệu 7](#_Toc50331895)

[1.3.3 Các bảng trong cơ sở dữ liệu 8](#_Toc50331896)

[1.3.3.1 Bảng User 8](#_Toc50331897)

[1.3.3.2 Bảng Job 8](#_Toc50331898)

[1.4 Danh sách API 9](#_Toc50331899)

[1.4.1 Các API về bảng user 9](#_Toc50331900)

[1.4.2 Các API về bảng job 11](#_Toc50331901)

[1.5 Các hình ảnh chạy API 15](#_Toc50331902)

[1.6 Phương pháp chạy backend 19](#_Toc50331903)

[1.7 Kết luận 21](#_Toc50331904)

[Chương 2 – THIẾT KẾ GIAO DIỆN (FRONTEND -ANGULAR) 21](#_Toc50331905)

[2.1 Giới thiệu 21](#_Toc50331906)

[2.2 Các công cụ sử dụng 22](#_Toc50331907)

[2.3 Cấu trúc thư mục 22](#_Toc50331908)

[2.4 Danh sách module và chức năng hệ thống 23](#_Toc50331909)

[2.5 Hình ảnh về trang web 26](#_Toc50331910)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC KÝ HIỆU**

**CÁC CHỮ VIẾT TẮT**

HTTP HyperText Transfer Protocol (giao thức truyền tải siêu văn bản)

REST Representational State Transfer

JS JavaScript

MVC Model-View-Controller

API Application Programming Interface (giao diện lập trình ứng dụng)

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

[Hình 1.1: Demo API danh sách user 15](#_Toc50332158)

[Hình 1.2: Demo API tạo mới tài khoản 16](#_Toc50332159)

[Hình 1.3: Demo API đăng nhập 16](#_Toc50332160)

[Hình 1.4: Demo API lấy danh sách công việc (khách và admin) 17](#_Toc50332161)

[Hình 1.5: Demo API lấy danh sách công việc (user đăng nhập) 17](#_Toc50332162)

[Hình 1.6: Demo API lấy công việc cụ thể 18](#_Toc50332163)

[Hình 1.7: Demo API tạo một công việc 18](#_Toc50332164)

[Hình 1.8: Demo API cập nhật công việc 19](#_Toc50332165)

[Hình 1.9: Demo API xóa công việc 19](#_Toc50332166)

[Hình 1.10: Console chạy backend 20](#_Toc50332167)

[Hình 1.11: Chạy thử server 20](#_Toc50332168)

[Hình 1.12: Package json hỗ trợ backend 21](#_Toc50332169)

[Hình 2.13: Cây thư mục chính của project 23](#_Toc50332170)

[Hình 2.14: Hình ảnh header với khách 26](#_Toc50332171)

[Hình 2.15: Hình ảnh header với user trong hệ thống 26](#_Toc50332172)

[Hình 2.16: Hình ảnh menu với quyền admin 26](#_Toc50332173)

[Hình 2.17: Hình ảnh menu với quyền user 26](#_Toc50332174)

[Hình 2.18: Hình ảnh footer trang web 27](#_Toc50332175)

[Hình 2.19: Search bar trang home 27](#_Toc50332176)

[Hình 2.19: Danh sách công việc trên trang home 27](#_Toc50332177)

[Hình 2.20: Hình ảnh trên trang home 28](#_Toc50332178)

[Hình 2.21: Trang danh sách công việc cho khách 28](#_Toc50332179)

[Hình 2.22: Form tạo đăng nhập trang đăng ký tài khoản 28](#_Toc50332180)

[Hình 2.23: Form đăng nhập trang login 29](#_Toc50332181)

[Hình 2.24: Form tạo công việc 29](#_Toc50332182)

[Hình 2.25: Form tạo công việc 30](#_Toc50332183)

[Hình 2.26: Form tạo công việc 30](#_Toc50332184)

[Hình 2.27: Danh sách công việc quyền user 31](#_Toc50332185)

[Hình 2.28: Danh sách công việc quyền user 31](#_Toc50332186)

[Hình 2.29: Danh sách user với quyền admin 32](#_Toc50332187)

[Hình 2.30: Trang hiển thị công việc cụ thể 32](#_Toc50332188)

**DANH MỤC BẢNG**

[Bảng 1.1: Danh sách các module sử dụng trong backend 5](#_Toc50330542)

[Bảng 1.2: Table User trong cơ sở dữ liệu 7](#_Toc50330543)

[Bảng 1.3: Bảng Job trong cơ sở dữ liệu 8](#_Toc50330544)

[Bảng 1.4: API lấy danh sách user 9](#_Toc50330545)

[Bảng 1.5: API tạo mới tài khoản 9](#_Toc50330546)

[Bảng 1.6: API đăng nhập 10](#_Toc50330547)

[Bảng 1.7: API lấy danh sách công việc (khách và admin) 10](#_Toc50330548)

[Bảng 1.8: API lấy danh sách công việc (user đăng nhập) 11](#_Toc50330549)

[Bảng 1.9: API lấy công việc cụ thể 12](#_Toc50330550)

[Bảng 1.10: API tạo một công việc 13](#_Toc50330551)

[Bảng 1.11: API cập nhật công việc 13](#_Toc50330552)

[Bảng 1.12: API xóa công việc 14](#_Toc50330553)

[Bảng 2.13: Bảng module Auth 23](#_Toc50330554)

[Bảng 2.14: Bảng module Home 23](#_Toc50330555)

[Bảng 2.14: Bảng module Jobs 24](#_Toc50330556)

[Bảng 2.15: Bảng module post 24](#_Toc50330557)

[Bảng 2.16: Bảng module shared 24](#_Toc50330558)

CHƯƠNG 1 – THIẾT KẾ API (BACKEND)

* 1. Giới thiệu

Web tìm kiếm việc làm được thiết kế dạng REST API. API được thiết kế chính dựa trên nền tảng Node.js sử dụng Express.js để tạo các url truy cập lấy thông tin từ hệ thống và sử dụng cơ sở dữ liệu thuần túy của web MySQL. Việc thiết kế kiến trúc API khiến cho việc kiểm soát code tốt hơn, phân vùng các lỗi theo các đường dẫn, dễ nắm bắt và sửa chữa. Bên cạnh đó việc thiết kế API giúp cho việc thiết kế website được tách biệt giữa thiết kế giao diện (frontend) và thiết kế, truy xuất dữ liệu (backend). Nhờ đó việc xây dựng website sẽ không bị ảnh hưởng nhiều từ backend và frontend. Đây cũng là mô hình nhiều website lớn, dự án lớn được xây dựng giảm thiểu thời gian và đạt hiểu qua cao.

* 1. Các công cụ sử dụng

API được viết dựa trên ngôn ngữ Javascript là ngôn ngữ chính. IDE hỗ trợ chính trong quá trình thiết kế là Visual studio code. Cũng như sử dụng công Xampp hỗ trợ thiết kế cơ sở dữ liệu cho website. Ngoài ra còn sử dụng một số module khác được cung cấp bởi node.

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên module | Phiên bản | Mục đích |
| 1. | Express.js | 4.17.1 | Xây dựng API cho web site, các đường dẫn route truy xuát thông tin |
| 2. | MySQL | 2.18.1 | Giúp giao tiếp với hệ cơ sở dữ liệu MySQL từ ngôn ngữ JS. |
| 3. | Body-parser | 1.19.0 | Hỗ trợ lấy các thông tin từ các request từ website, các dữ liệu để tiến hành truy xuất database và trả về thông tin phù hợp |
| 4. | Nodemoon | 2.0.4 | Hỗ trợ việc reload lại server trong việc debug và thay đổi file trong hệ thống. |
| 5. | Cors | 2.8.5 | Hỗ trợ việc giải quyết vấn đề cors từ request. |
| 6. | Crypto-js | 4.0 | Hỗ trợ việc mã hóa mật khẩu để không bị đánh cắp trên cơ sở dữ liệu. Mã hóa MD5 được sử dụng chính. |

Bảng 1.1: Danh sách các module sử dụng trong backend

* 1. Cơ sở dữ liệu

1.3.1 Thông tin về cơ sở dữ

- Tên cơ sở dữ liệu: job

- User: root

- Password:

- Có tổng cổng 2 bảng chính (user, job)

1.3.2 Kết nối cơ sở dữ liệu

+ API được khởi chạy từ localhost với cổng 3000

|  |
| --- |
| const http = require('http');  const app = require('./app');  const port = process.env.PORT || 3000;  const server = http.createServer(app);  server.listen(port); |

+ Kết nối cơ sở dữ liệu

|  |
| --- |
| var mysql = require('mysql');  var connection = mysql.createConnection({  host : 'localhost',  user : 'root',  password : '',  database : 'job'  });  // connect to database  connection.connect(function(err) {if (err) throw err;});  module.exports = connection; |

1.3.3 Các bảng trong cơ sở dữ liệu

1.3.3.1 Bảng User

|  |  |  |
| --- | --- | --- |
| STT | Tên | Kiểu dữ liệu |
| 1 | Id | Int(11), primary key, auto increment |
| 2 | First\_name | Varchar(50) |
| 3 | Last\_name | Varchar(50) |
| 4 | Email | Varchar(100) |
| 5 | Password | Varchar(100) |
| 6 | Type | Int(11) |

Bảng 1.2: Table User trong cơ sở dữ liệu

1.3.3.2 Bảng Job

|  |  |  |
| --- | --- | --- |
| STT | Tên | Kiểu dữ liệu |
| 1. | Id | Int(11) primary key, auto increment |
| 2. | Id\_user | Int(11) |
| 3. | Job\_title | Varchar(100) |
| 4. | Job\_des | Longtext |
| 5. | Type | Int(11) |
| 6 | Salary | Int(11) |
| 7 | Location | Varchar(50) |
| 8 | Category | Int(11) |
| 9 | Expire\_date | Date |
| 10 | Company\_name | Varchar(30) |
| 11 | Company\_des | Longtext |
| 12 | Web\_site | Varchar(50) |
| 13 | Email | Varchar(50) |

Bảng 1.3: Bảng Job trong cơ sở dữ liệu

* 1. Danh sách API

1.4.1 Các API về bảng user

- API lấy danh sách user:

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/user/list” |
| Phương thức | POST |
| Dữ liệu đầu vào | {‘page’:1, ‘display’:5} |
| Dữ liệu đầu ra | {  message: "ok",  users: [],  \_meta: {  total: 1,  num\_pages: 1,  current\_page: 1,  previous\_page: 1,  next\_page: 1,  num\_per\_page: 5  } |
| Mô tả | Dùng để lấy danh sách user |

Bảng 1.4: API lấy danh sách user

- API tạo mới user:

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/user/create” |
| Phương thức | POST |
| Dữ liệu đầu vào | user |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  user: user  } |
| Thất bại:{message: 'Tài khoản đã tồn tại'} |
| Mô tả | Dùng để tạo mới tài khoản để đăng bài tuyển dụng |

Bảng 1.5: API tạo mới tài khoản

- API đăng nhập:

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/user/login” |
| Phương thức | POST |
| Dữ liệu đầu vào | user |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  user: user  } |
| Thất bại không tồn tài khoản  {message: ‘Không tồn tại tài khoản'} |
| Thất bại sai mật khẩu  { message: ‘Mật khẩu không đúng'} |
| Mô tả | Dùng để đăng nhập vào hệ thống |

Bảng 1.6: API đăng nhập

1.4.2 Các API về bảng job

- API lấy tất cả job (phía người dùng không đăng nhập, admin)

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/list” |
| Phương thức | POST |
| Dữ liệu đầu vào | {‘page’:1, ‘display’:5, text: ‘finding’} |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  jobs: [],  \_meta: {  total: 1,  num\_pages: 1,  current\_page: 1,  previous\_page: 1,  next\_page: 1,  num\_per\_page: 5  }  } |
| Mô tả | Dùng để lấy danh sách công việc |

Bảng 1.7: API lấy danh sách công việc (khách và admin)

- API lấy tất cả job (phía người sử dụng có đăng nhập)

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/list/user/:userId” |
| Phương thức | POST |
| Dữ liệu đầu vào | {‘page’:1, ‘display’:5 }, |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  jobs: [],  \_meta: {  total: 1,  num\_pages: 1,  current\_page: 1,  previous\_page: 1,  next\_page: 1,  num\_per\_page: 5  }  } |
| Mô tả | Dùng để lấy danh sách công việc |

Bảng 1.8: API lấy danh sách công việc (user đăng nhập)

- API lấy một công việc cụ thể

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/list/:id” |
| Phương thức | GET |
| Dữ liệu đầu vào |  |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  job: job  } |
| Mô tả | Dùng để lấy công việc cụ thể |

Bảng 1.9: API lấy công việc cụ thể

- API tạo một công việc

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/create” |
| Phương thức | POST |
| Dữ liệu đầu vào | job |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  job: job  } |
| Mô tả | Dùng để tạo một job trên hệ thống |

Bảng 1.10: API tạo một công việc

- API lấy một công việc cụ thể

|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/update” |
| Phương thức | PUT |
| Dữ liệu đầu vào | user |
| Dữ liệu đầu ra | Thành công:  {  message: "ok",  } |
| Mô tả | Cập nhật công việc |

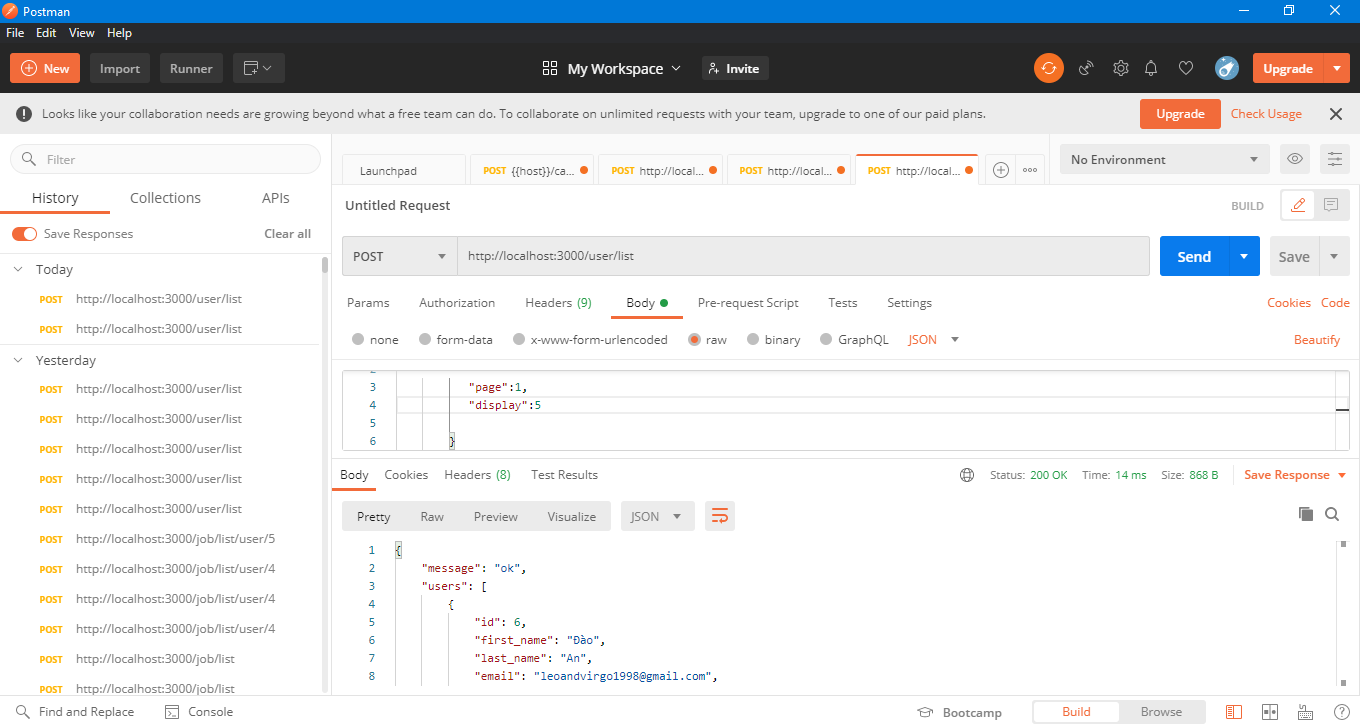
Bảng 1.11: API cập nhật công việc

- API xóa một công việc cụ thể

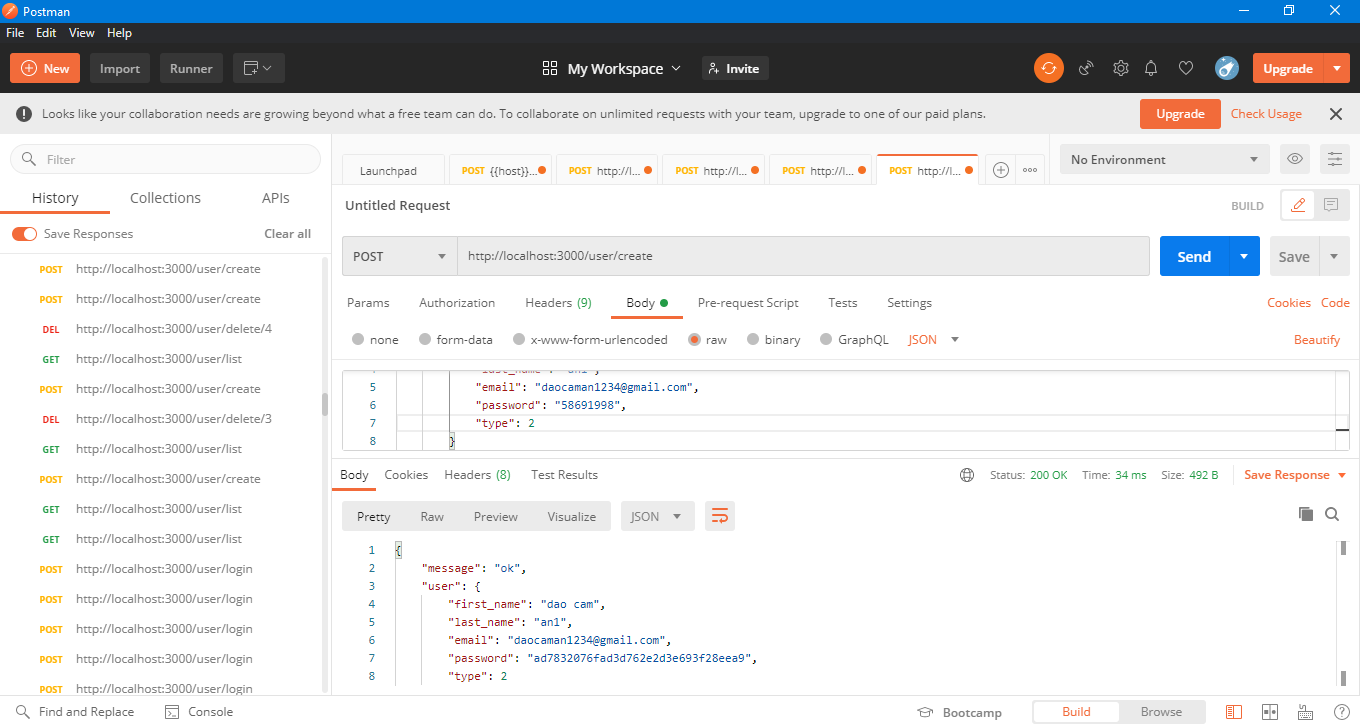
|  |  |
| --- | --- |
| Đường dẫn | “localhost:3000/job/delete/:id” |
| Phương thức | DELETE |
| Dữ liệu đầu vào |  |
| Dữ liệu đầu ra | Thành công:  {  message: "delete success",  } |
| Mô tả | Xóa công việc |

Bảng 1.12: API xóa công việc

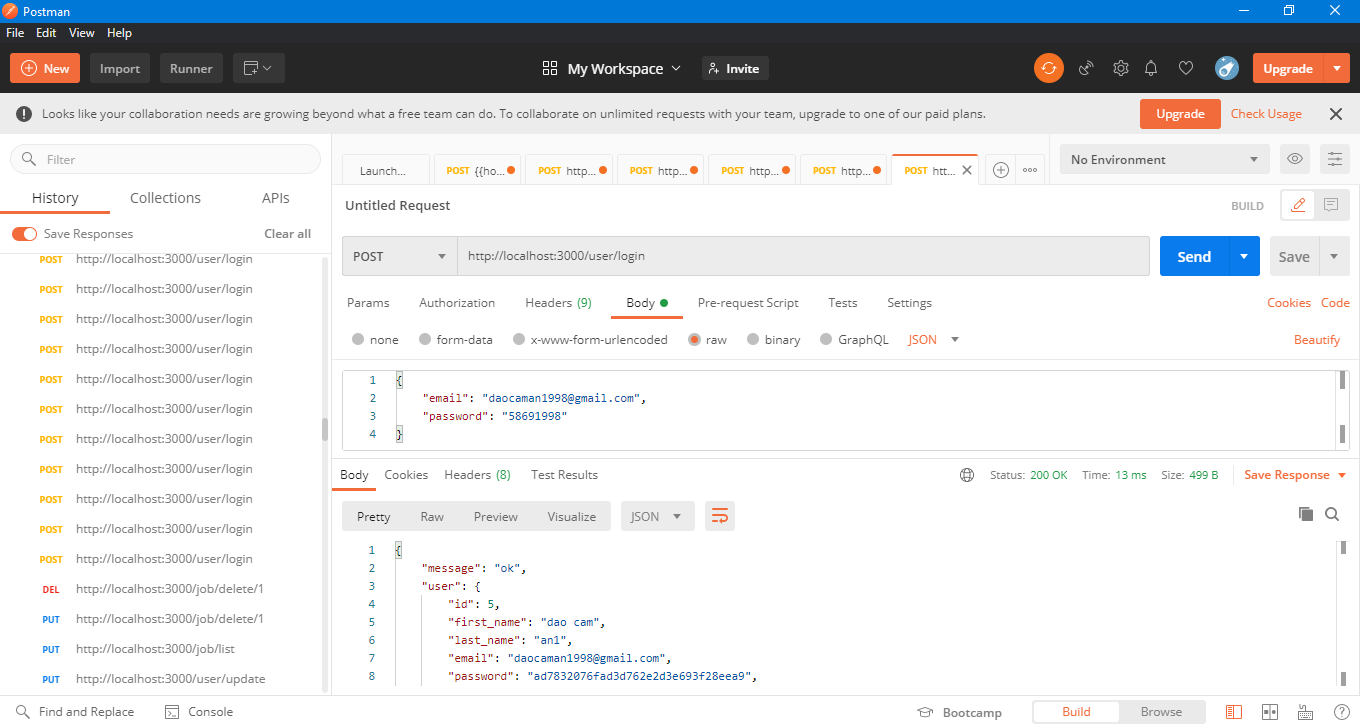
* 1. Các hình ảnh chạy API



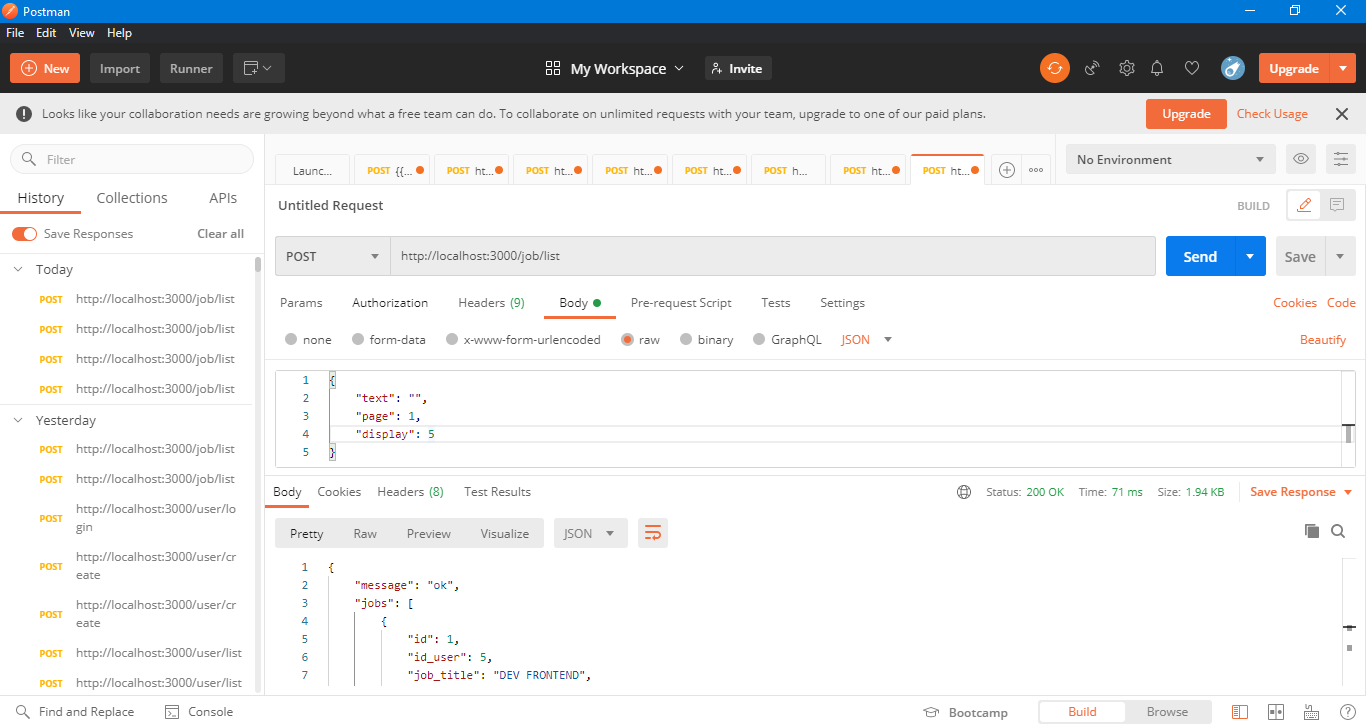
Hình 1.1: Demo API danh sách user



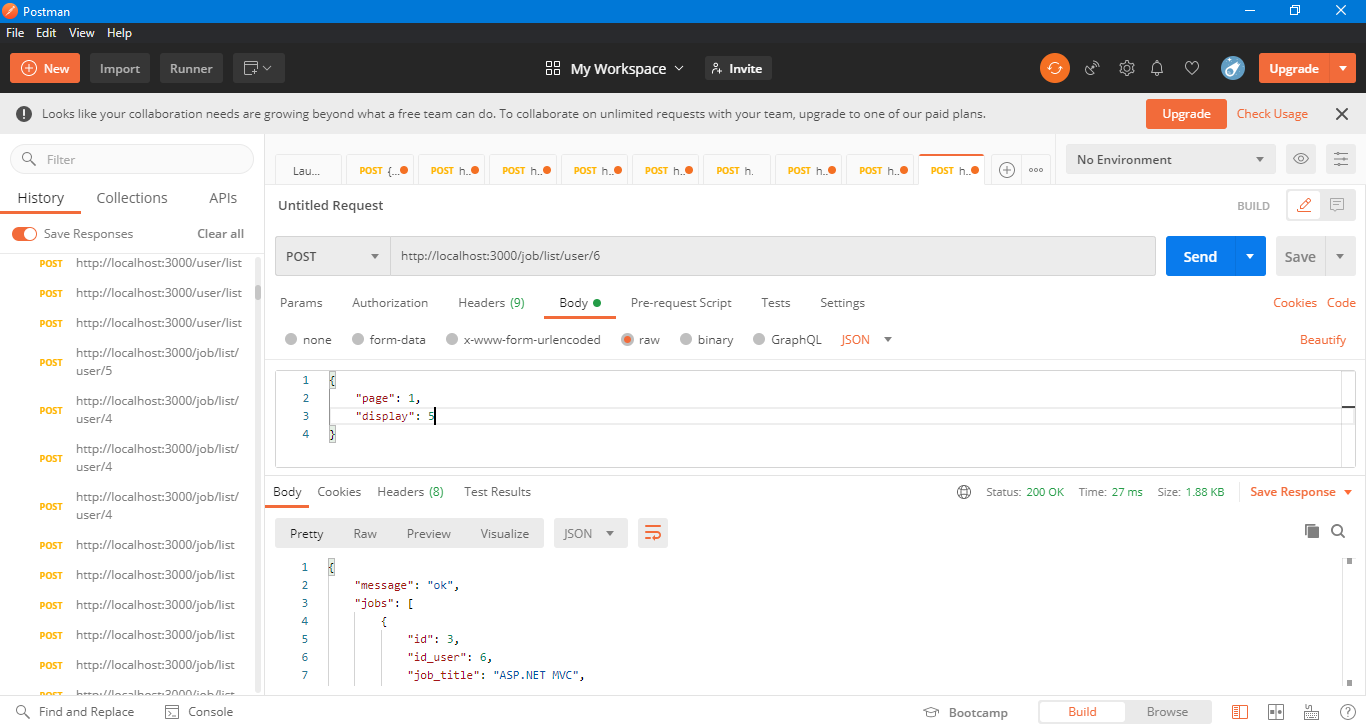
Hình 1.2: Demo API tạo mới tài khoản



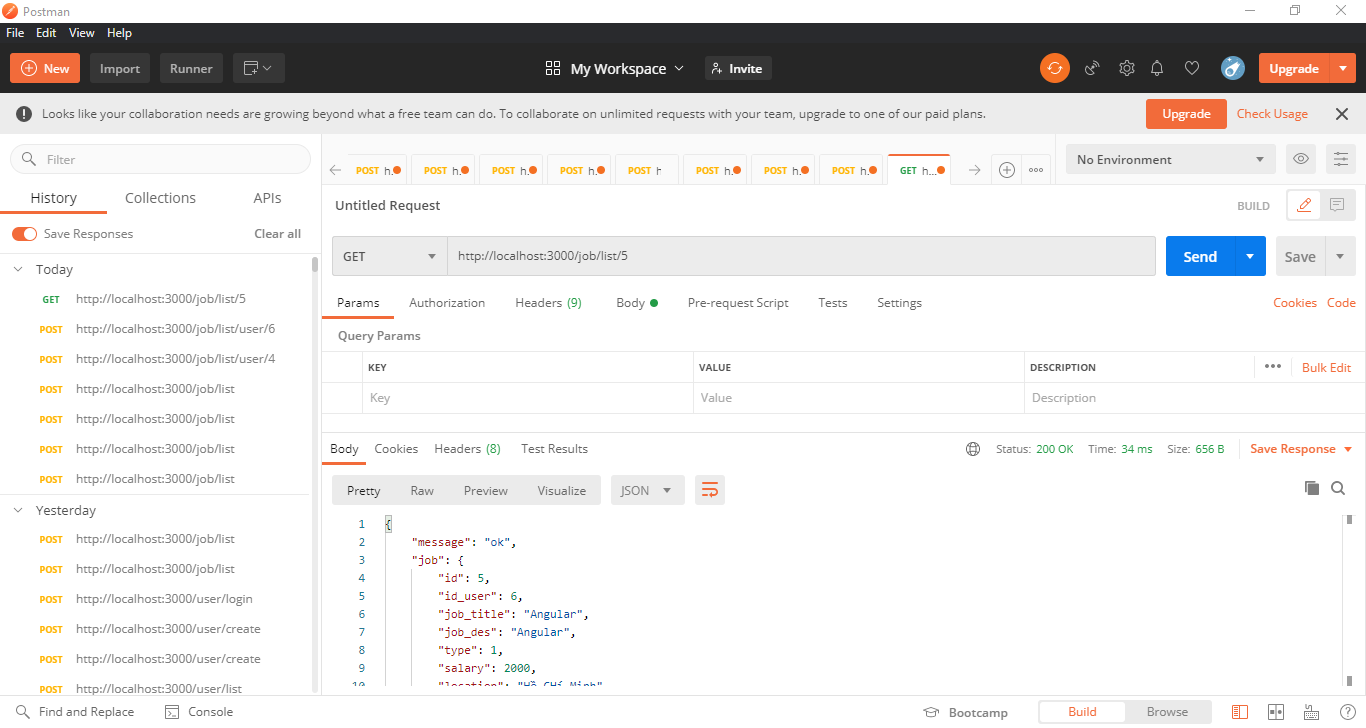
Hình 1.3: Demo API đăng nhập



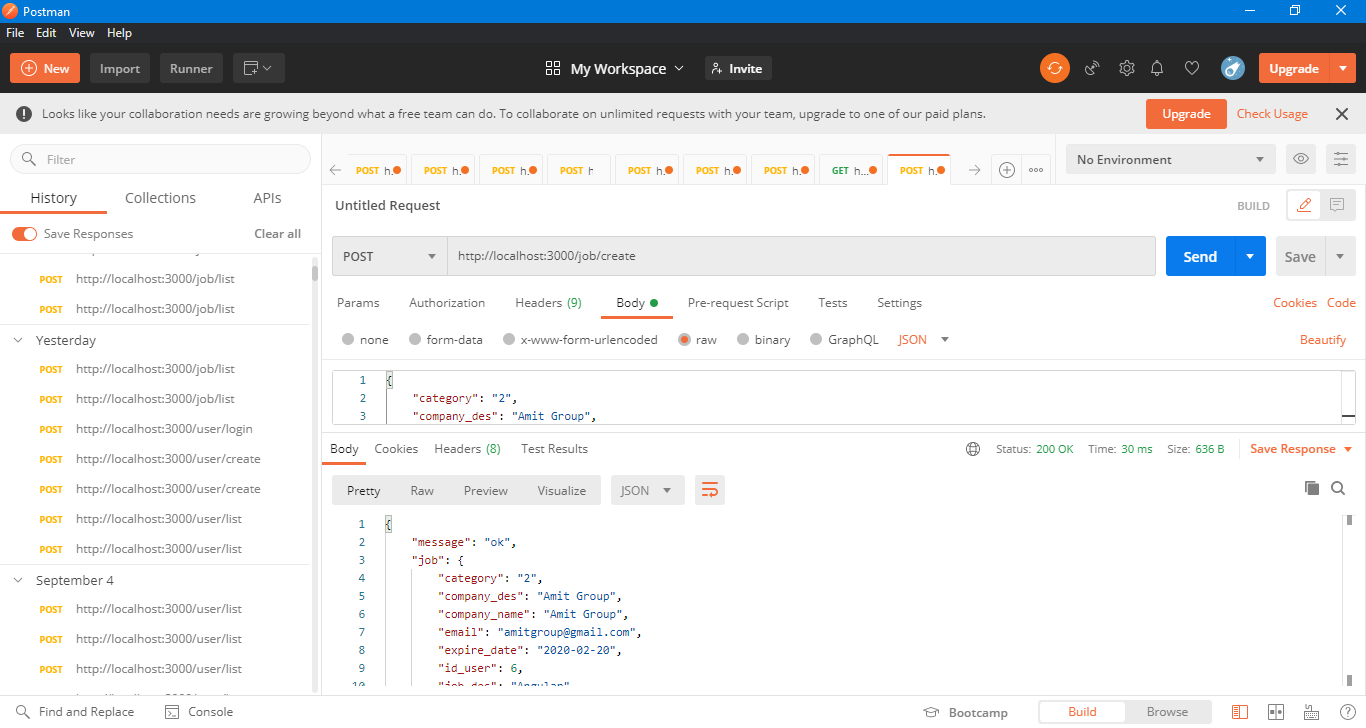
Hình 1.4: Demo API lấy danh sách công việc (khách và admin)



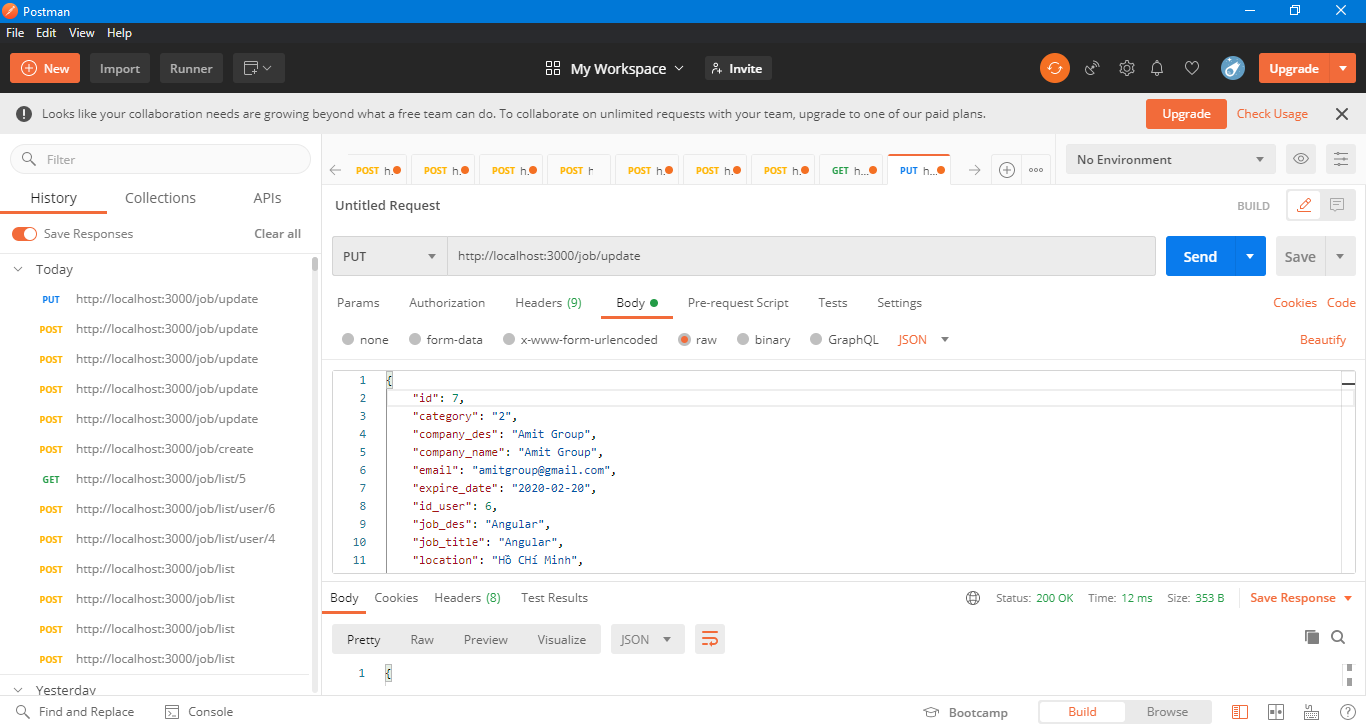
Hình 1.5: Demo API lấy danh sách công việc (user đăng nhập)



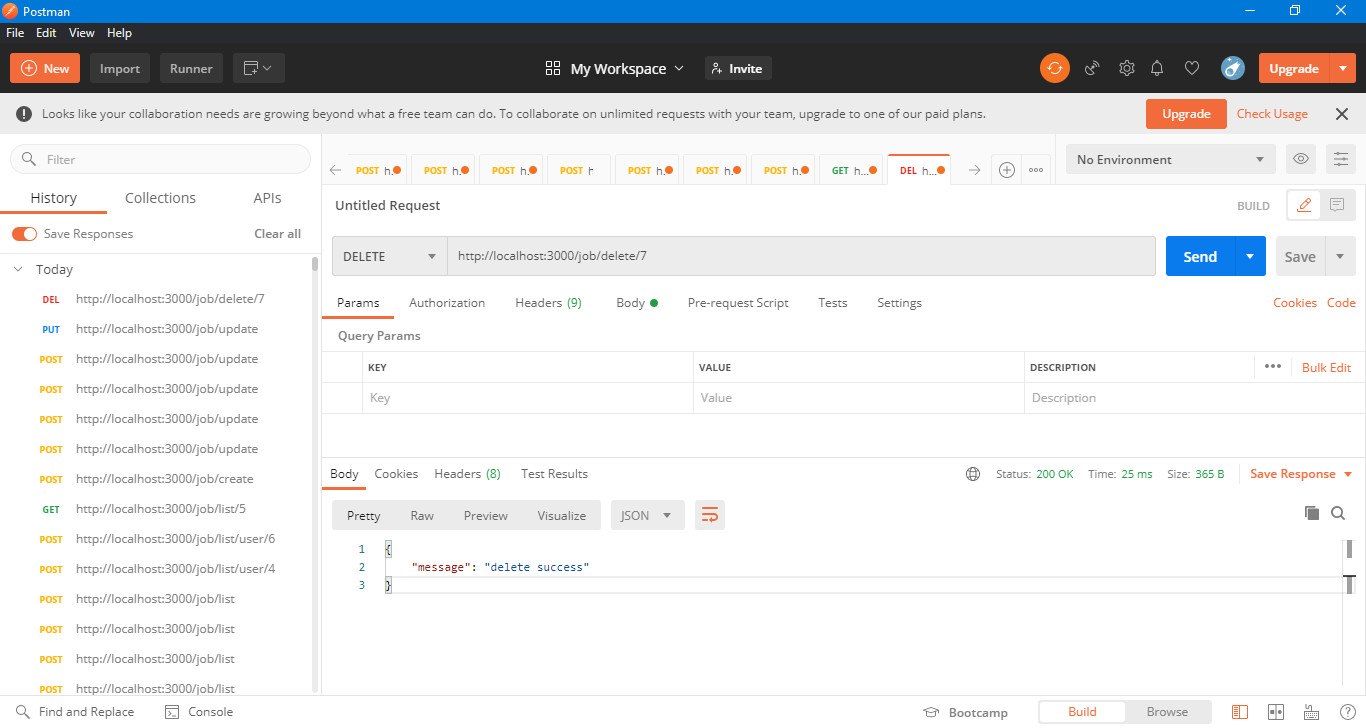
Hình 1.6: Demo API lấy công việc cụ thể



Hình 1.7: Demo API tạo một công việc



Hình 1.8: Demo API cập nhật công việc

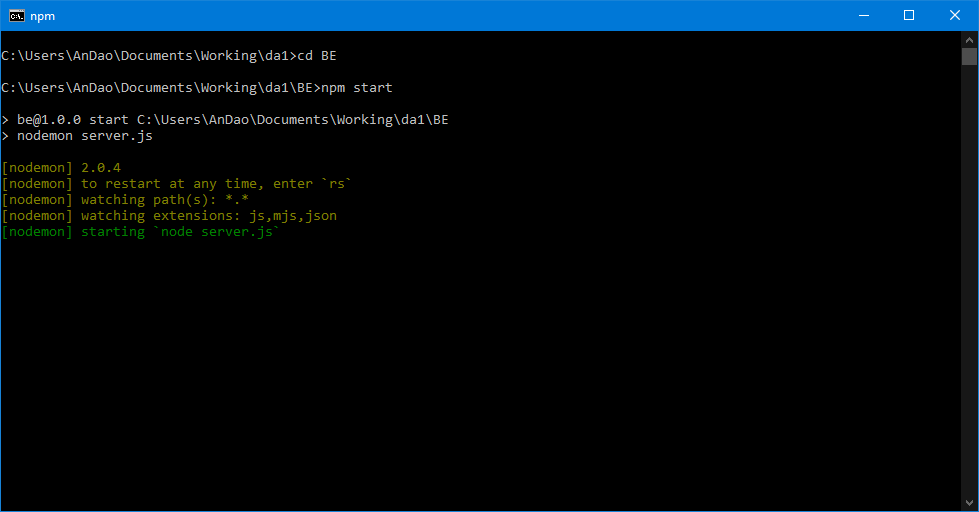


Hình 1.9: Demo API xóa công việc

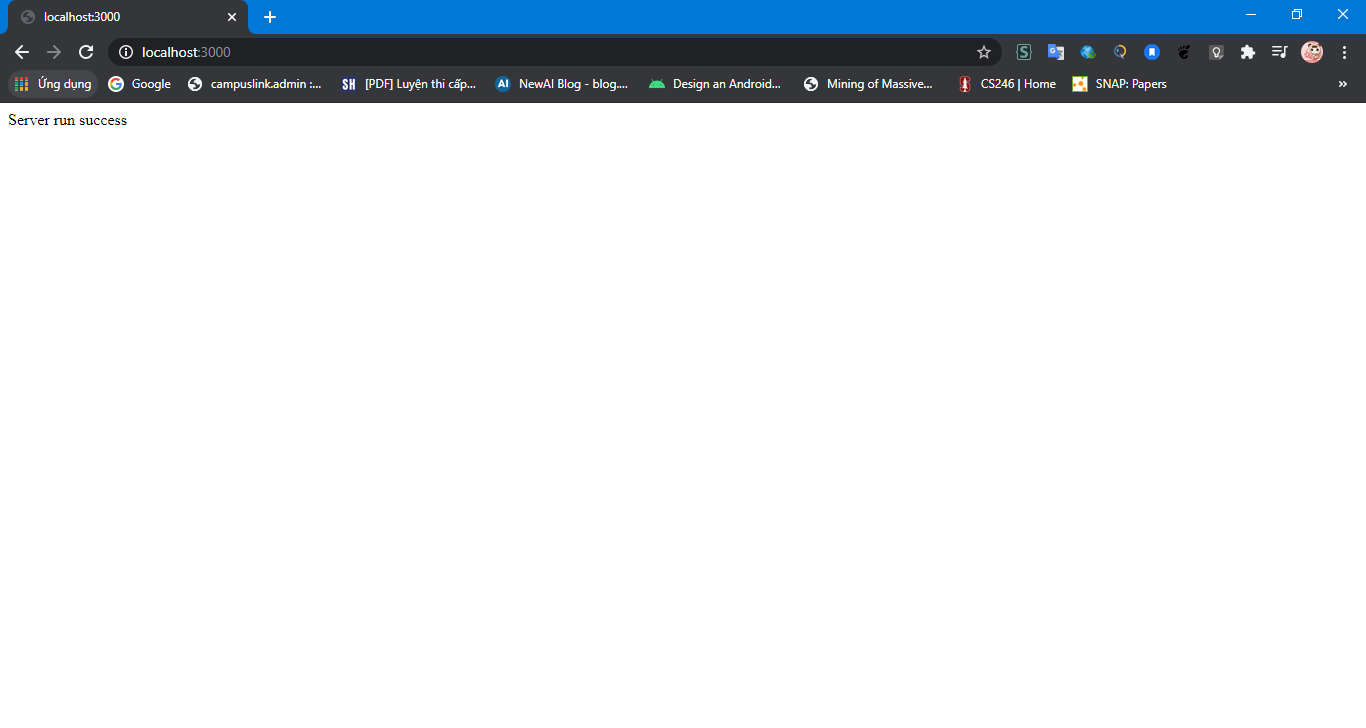
* 1. Phương pháp chạy backend

Các bước chạy backend của web.

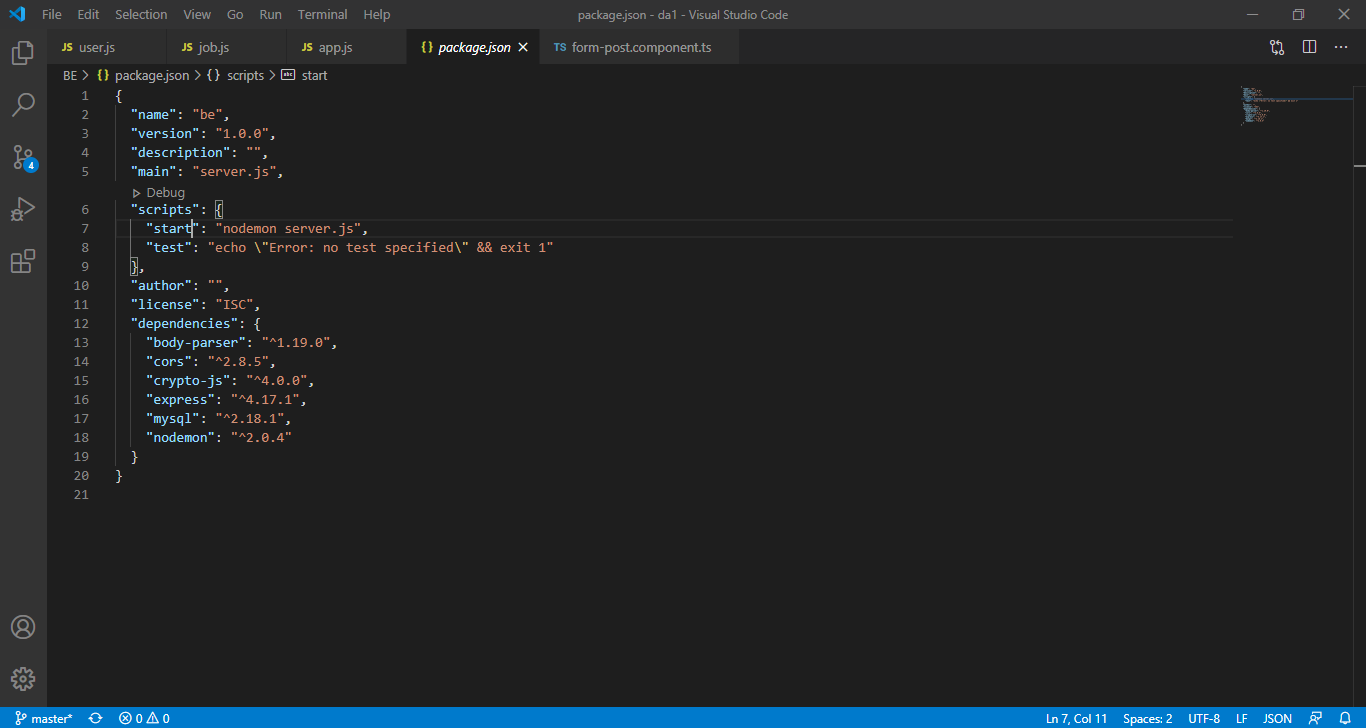
1. Khởi tạo chạy xampp
2. Dẫn vào thư mục BE
3. Chạy câu lệnh “npm start”
4. Chạy thử trên web site (<http://localhost:3000/>)



Hình 1.10: Console chạy backend



Hình 1.11: Chạy thử server



Hình 1.12: Package json hỗ trợ backend

* 1. Kết luận

Việc cấu trúc chạy web dạng API giúp linh hoạt trong các dạng dữ liệu trả về client, hỗ trợ tối đa mô hình MVC dễ dàng bảo trì, nâng cấp hệ thống cho từng module thích hợp

Chương 2 – THIẾT KẾ GIAO DIỆN (FRONTEND -ANGULAR)

2.1 Giới thiệu

Web tìm kiếm việc được viết dựa trên nền tảng Angular. Angular là nền tảng được sử dụng rộng rãi bởi các công ty và nhiều nhà phát triển khát như Google,… Việc viết dưới nền tảng Angular giúp web được phân chia các thư mục rõ ràng, các module nhỏ có thể tái sử dụng nhiều trong nhiều dự án không chỉ riêng một dự án. Việc tái sử dụng được các module, component giúp cho web viết nhanh hơn, hiệu quả, và dễ kiểm soát hơn với các web thông thường. Ngoài ra do hỗ trợ tính năng module lazy loading giúp cho web không trở nên quá tải khi phải load quá nhiều module.

- Web tìm kiếm việc làm gồm 6 module chính:

+ Module Auth: gồm các chức năng đăng nhập và tạo tài khoản

+ Module Home: gồm các chức năng hiển thị thông tin các công việc hiện có, tìm kiếm công việc.

+ Module Jobs: hiển thị danh sách công việc đầy đủ, tìm kiếm công việc và hiển thị chi tiết công việc

+ Module Post: hiển thị danh sách các công việc được đăng bởi user trong hệ thống, tạo và chỉnh sửa các bài đăng công việc.

+ Component user: hiển thị danh sách user trong hệ thống dưới quyền admin.

+ Module Shared: bao gồm các component dùng chung cho hệ thống, các api dùng chung.

2.2 Các công cụ sử dụng

Web tìm kiếm việc làm sử dụng chính nền tảng Angular. Ngoài ra còn sử dụng bootstrap, jquery hỗ trợ trong việc viết giao diện, hiệu ứng và tham khảo một số template từ trên các website như github, design,…

2.3 Cấu trúc thư mục

Do hỗ trợ từ phía Angular, là web được render chính từ một file html, nên web hỗ trợ nhanh trong việc chuyển đổi trang mà không cần load lại quá nhiều. Cấu trúc thư mục của angular sẽ được phân bố như sau.

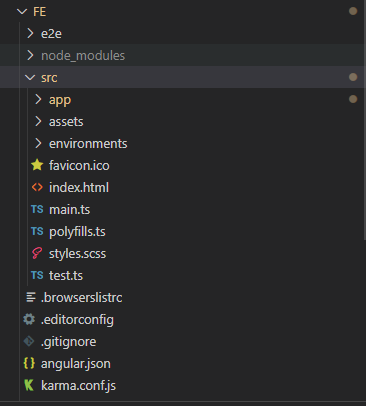
Thư mục code sẽ được để vào thư mục “src”. Trong đó sẽ bao gồm:

+ Thư mục app: nơi chứa các thư mục chính, module chính sẽ code trong hệ thống.

+ Thư mục assets: nơi chứa các tài nguyên, hình ảnh, style, trong hệ thống

+ Thư mục environments: nơi chứa các tài nguyên dùng chung config server api trong khi chạy thử và build ra sản phẩm.

+ Ngoài ra còn một số file để ngoài như : index.html, styles.scss, main.ts, favicon.ico hỗ trợ chính trong việc load trang web



Hình 2.13: Cây thư mục chính của project

2.4 Danh sách module và chức năng hệ thống

Cấu trúc module gồm các thư mục và file chính sau:

+ File name.module: hiển thị đầy đủ các thông tin, đường dẫn trong module với các components và screen trong module

+ File name-routing.module: hiển thị đường dẫn cụ thể đối với module đến các screen, components cụ thể

+ Thư mục screens: thể hiển các screen cụ thể trong module

+ Thư mục components: thể hiện các components được tách riêng sử dụng trong module đó

+ Thư mục service: thể hiển service cung cấp dữ liêu cho hệ thống

- Module Auth:

|  |  |
| --- | --- |
| Screens | Login screen, Register screen |
| Components |  |
| Services | Các dịch vụ cung cấp hỗ trợ đăng nhập, tạo tài khoản |
| Auth-routing.module | Cung cấp đường dẫn đến các trang con là login, register |
| Auth.module | Nơi thể hiện các module được sử dụng trong module auth, routing, components |

Bảng 2.13: Bảng module Auth

- Module home:

|  |  |
| --- | --- |
| Screens | Home screen |
| Components | Item1 thể hiện đối tượng hiển thị jobs trên giao diện chính |
| Services | Các dịch vụ cung cấp việc lấy các jobs trên hệ thống |
| Home-routing.module | Cung cấp đường dẫn đến các trang con là home |
| Home.module | Nơi thể hiện các module được sử dụng trong module home, routing, components |

Bảng 2.14: Bảng module Home

- Module jobs:

|  |  |
| --- | --- |
| Screens | Hiển thị thông tin 1 công việc, danh sách công việc |
| Components | Row-item hiển thị 1 công việc dạng dòng trong danh sách công việc |
| Services | Các dịch vụ cung cấp lấy tất cả công việc và lấy công việc cụ thể |
| Jobs-routing.module | Cung cấp đường dẫn đến các trang con là danh sách công việc và thông tin công việc cụ thể |
| Jobs.module | Nơi thể hiện các module được sử dụng trong module jobs, routing, components |

Bảng 2.14: Bảng module Jobs

- Bảng module post

|  |  |
| --- | --- |
| Screens | Hiển thị trang tạo công việc, chỉnh sửa công việc, và danh sách công việc đã đăng đối với user hệ thống |
| Components | Form-post form để điền thông tin đăng công việc |
| Services | Cung cấp các dịch vụ tạo, chỉnh sửa, lấy tất cả công việc với các quyền users |
| Post-routing.module | Cung cấp đường dẫn đến các trang con là danh sách công việc đã đăng, trang điền thông tin và chỉnh sửa công việc |
| Post.module | Nơi thể hiện các module được sử dụng trong module Post, routing, components |

Bảng 2.15: Bảng module post

- Bảng module shared

|  |  |
| --- | --- |
| Helpers | url.ts hỗ trợ tạo đường dẫn liên kết để gửi request |
| Components | Footer, Header, Pagination sử dụng chung trong hệ thống |
| Services | Cung cấp api service để truy xuất các request với phương thức phù hợp |
| Shared.module | Nơi thể hiện các module được sử dụng trong module Shared, components |

Bảng 2.16: Bảng module shared

- Components user: Hiển thị danh sách user với quyền admin

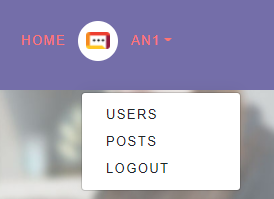
2.5 Hình ảnh về trang web



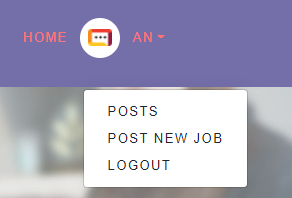
Hình 2.14: Hình ảnh header với khách



Hình 2.15: Hình ảnh header với user trong hệ thống



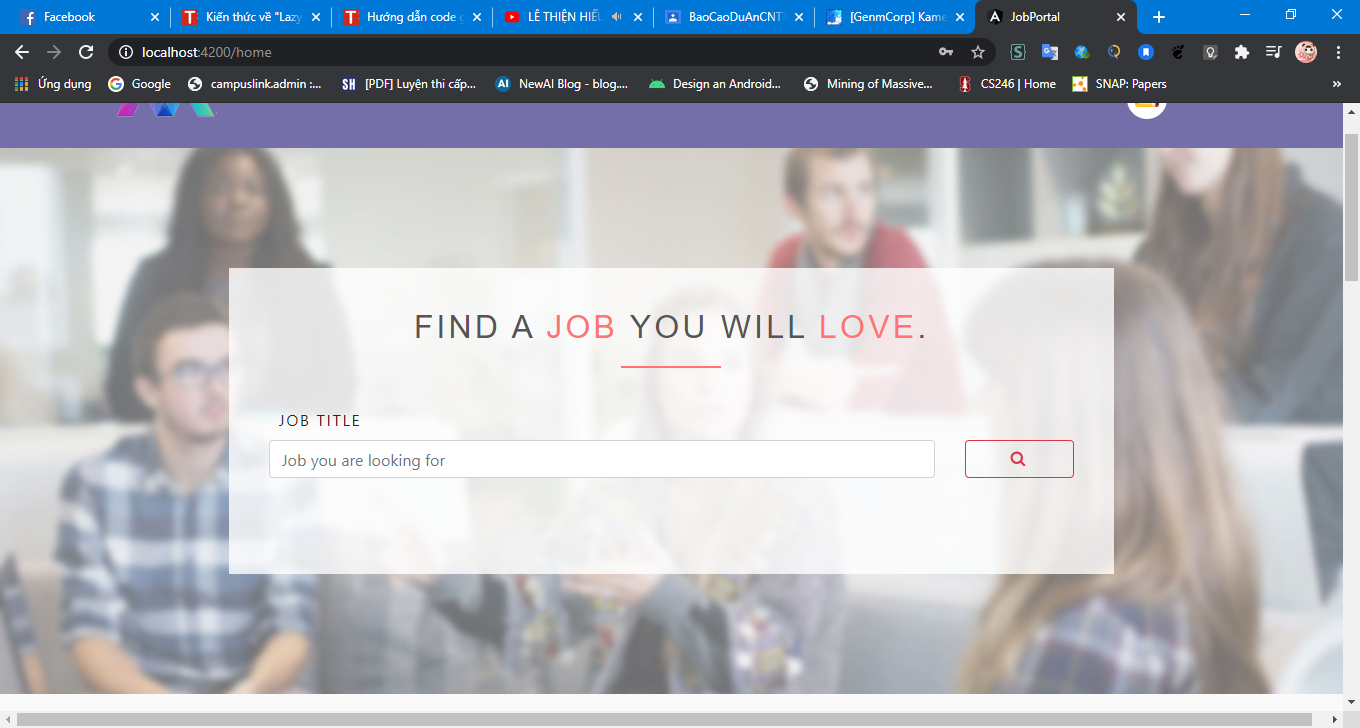
Hình 2.16: Hình ảnh menu với quyền admin



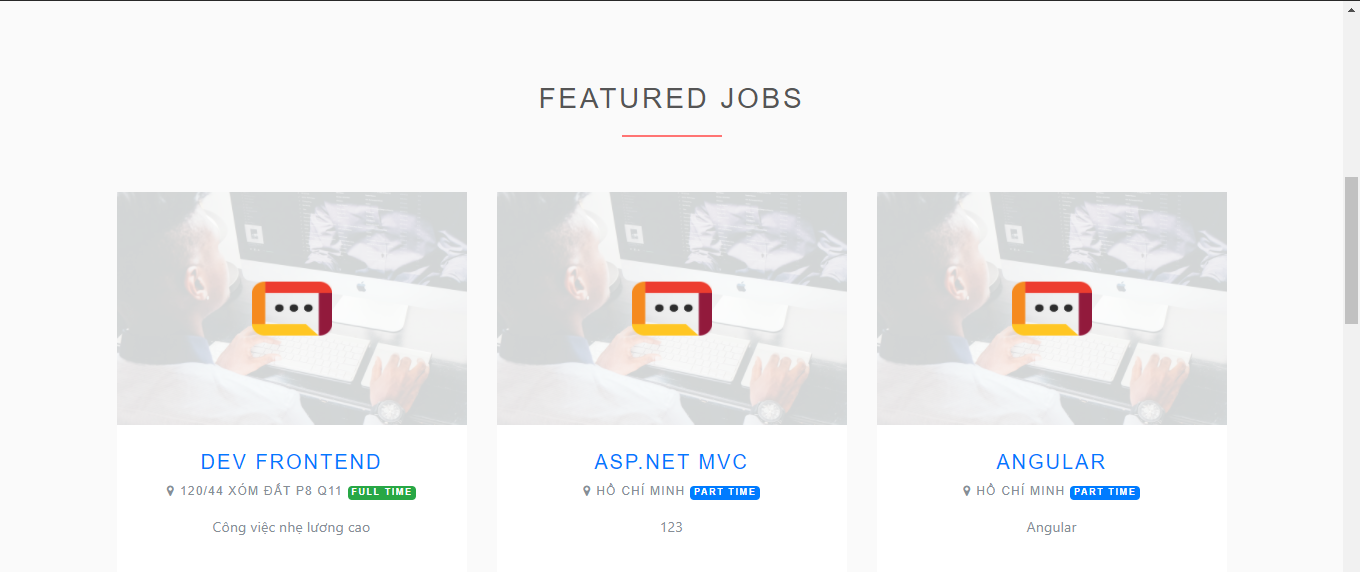
Hình 2.17: Hình ảnh menu với quyền user



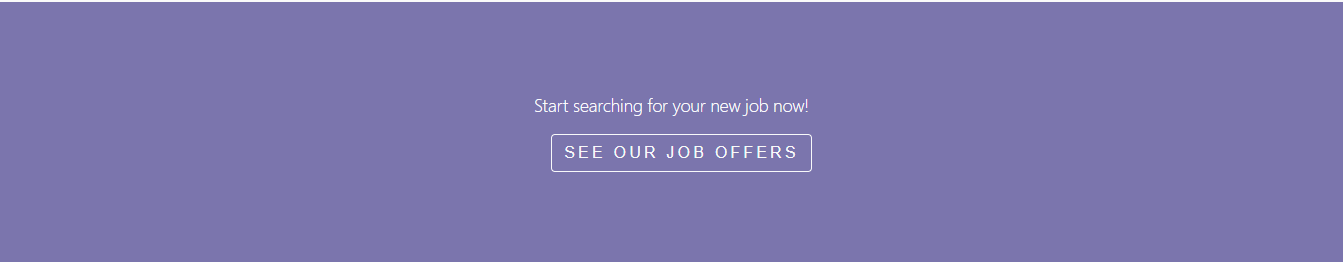
Hình 2.18: Hình ảnh footer trang web



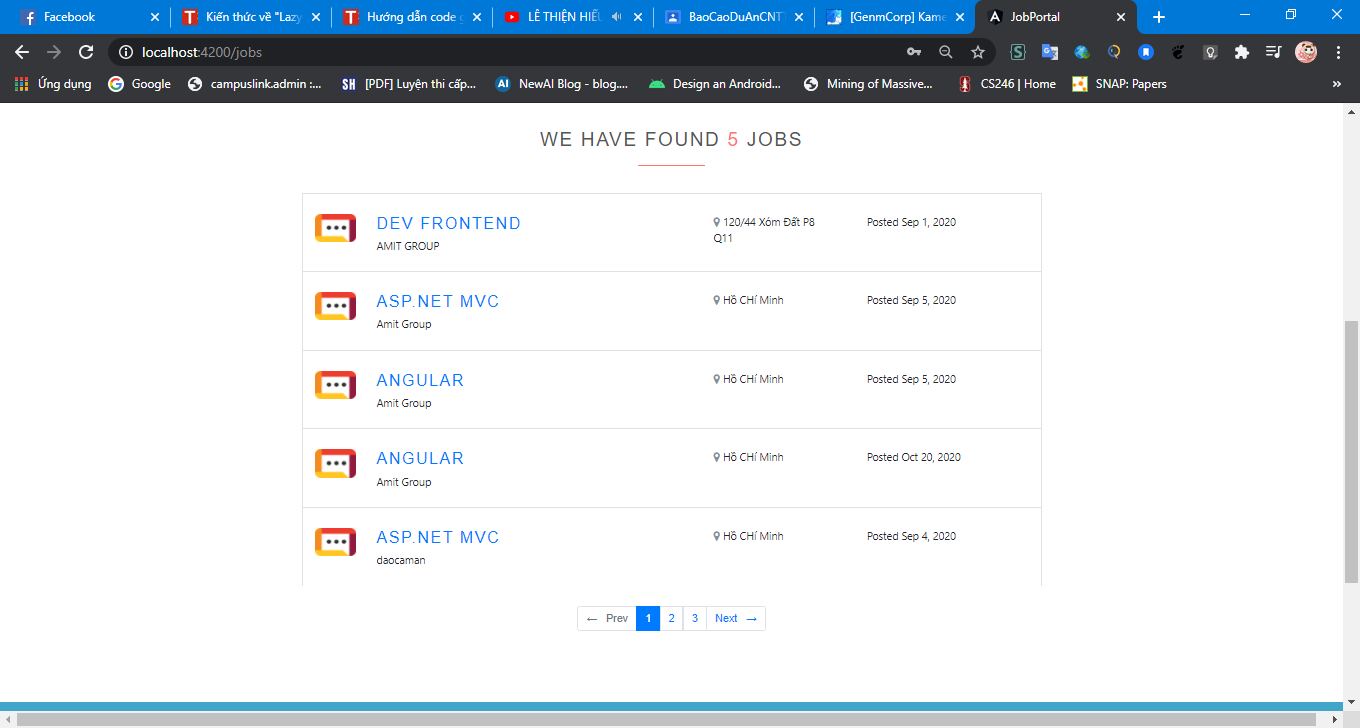
Hình 2.19: Search bar trang home



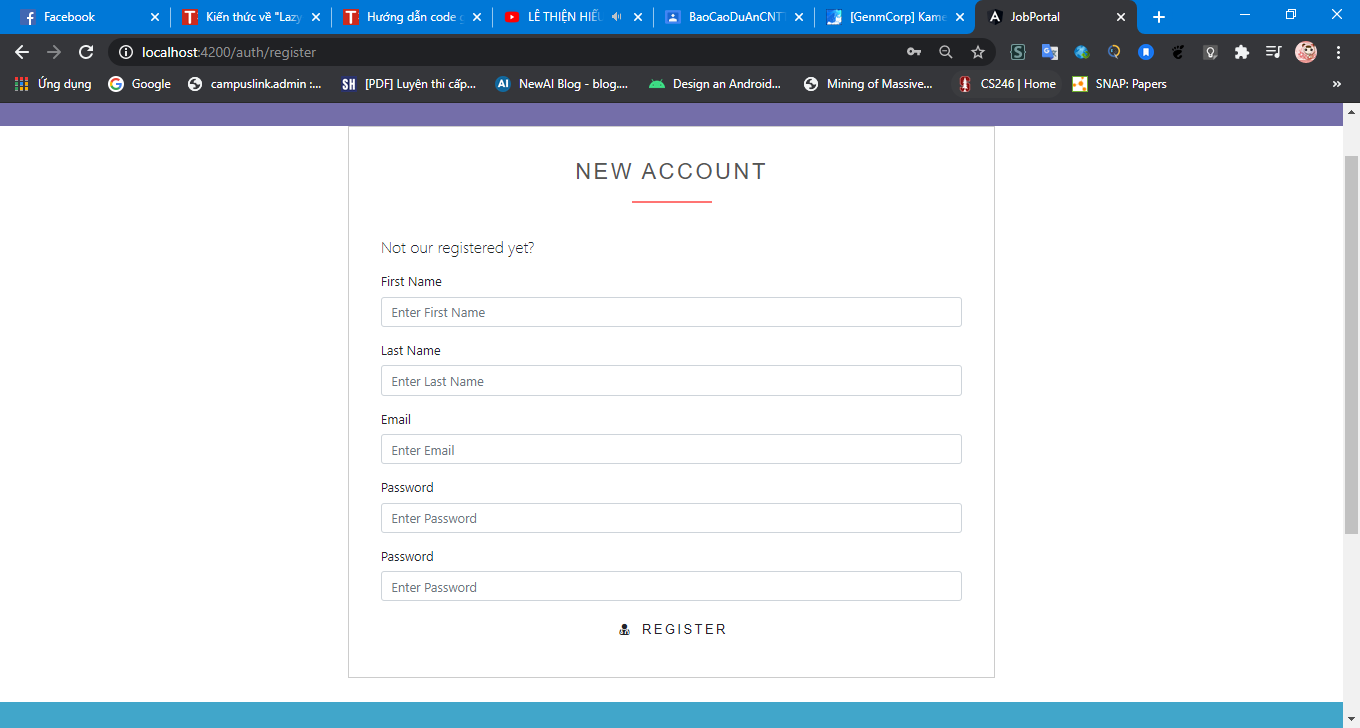
Hình 2.19: Danh sách công việc trên trang home



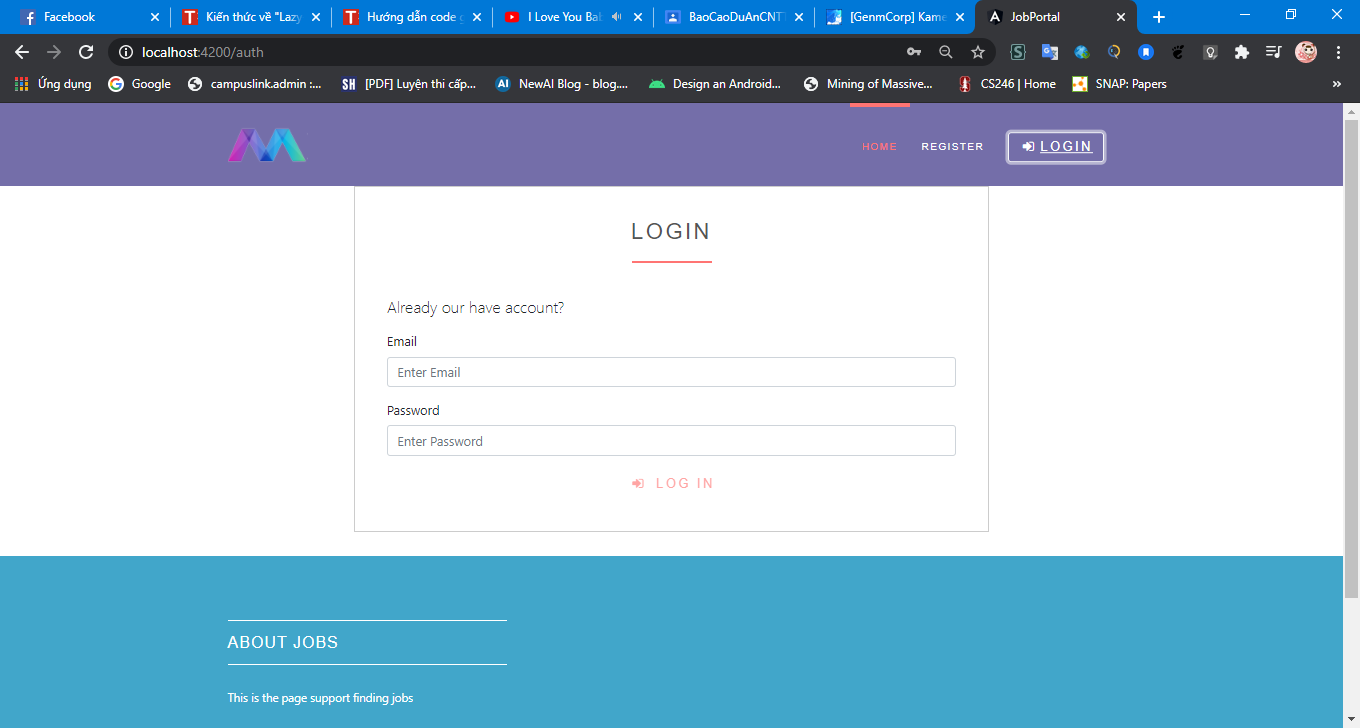
Hình 2.20: Hình ảnh trên trang home



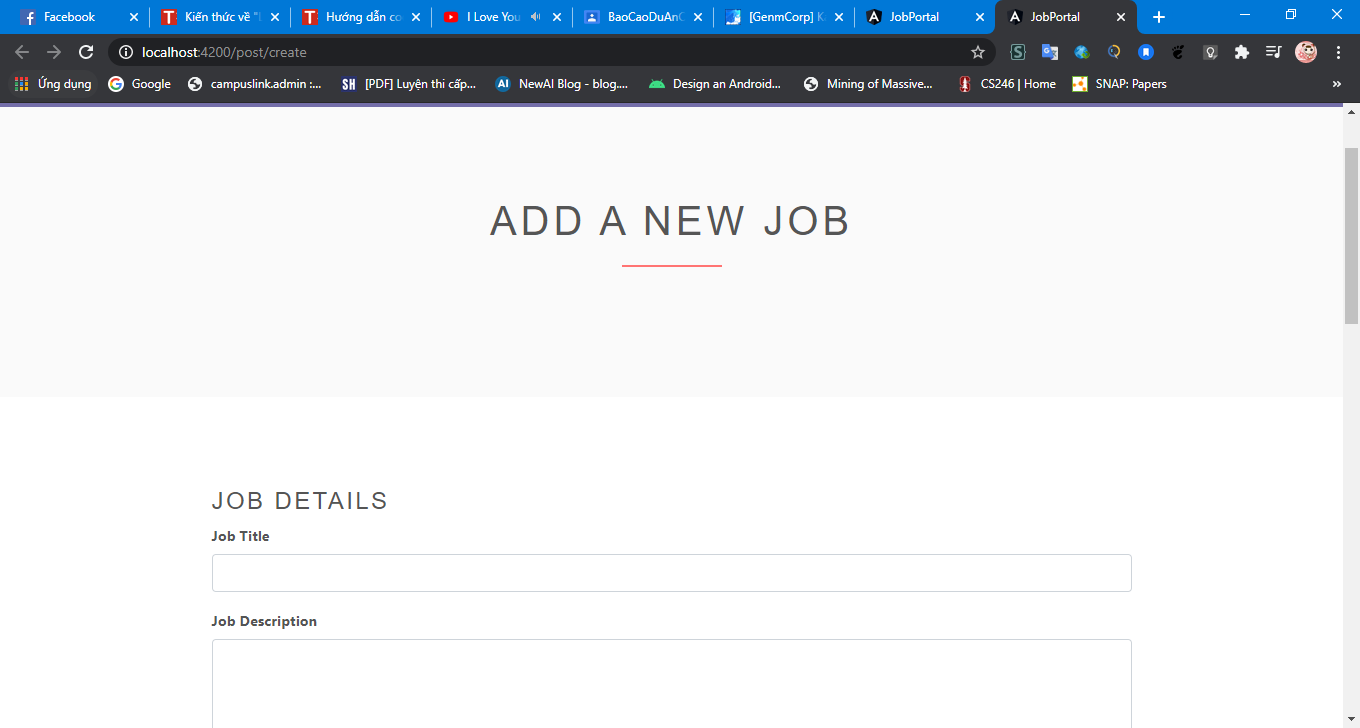
Hình 2.21: Trang danh sách công việc cho khách



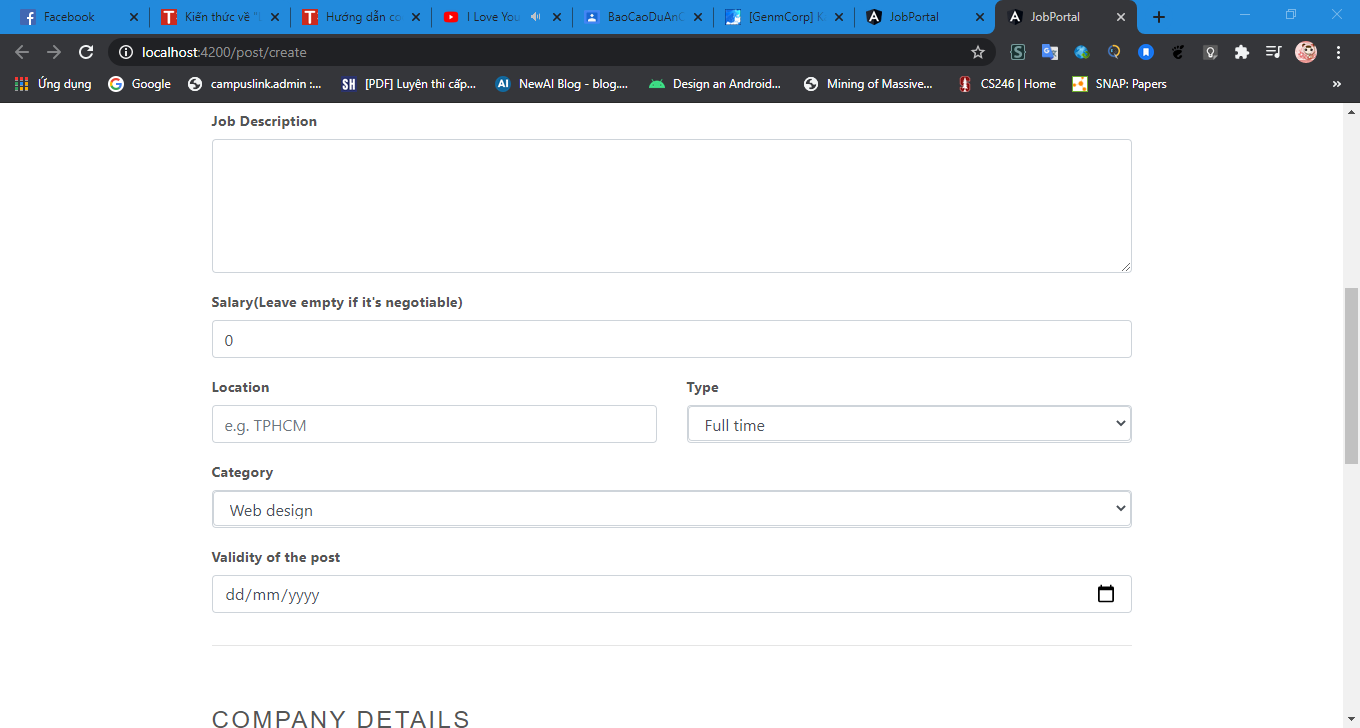
Hình 2.22: Form tạo đăng nhập trang đăng ký tài khoản



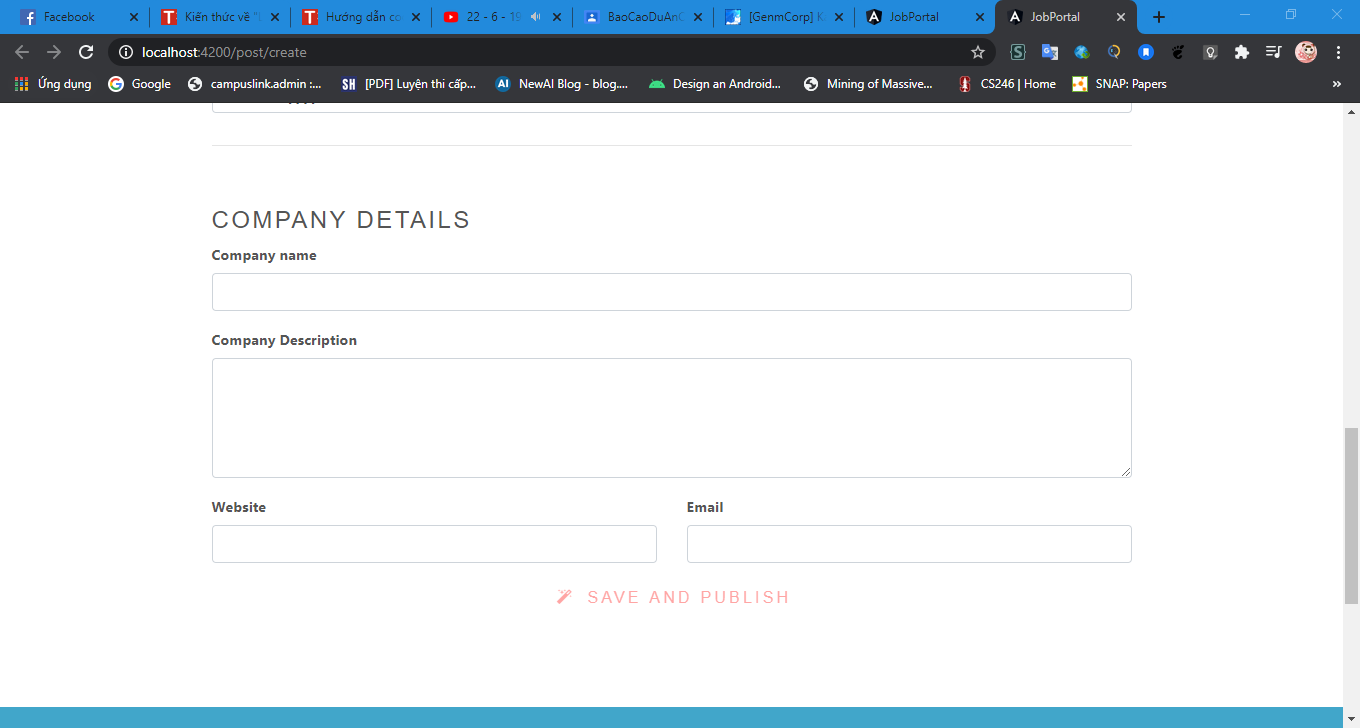
Hình 2.23: Form đăng nhập trang login



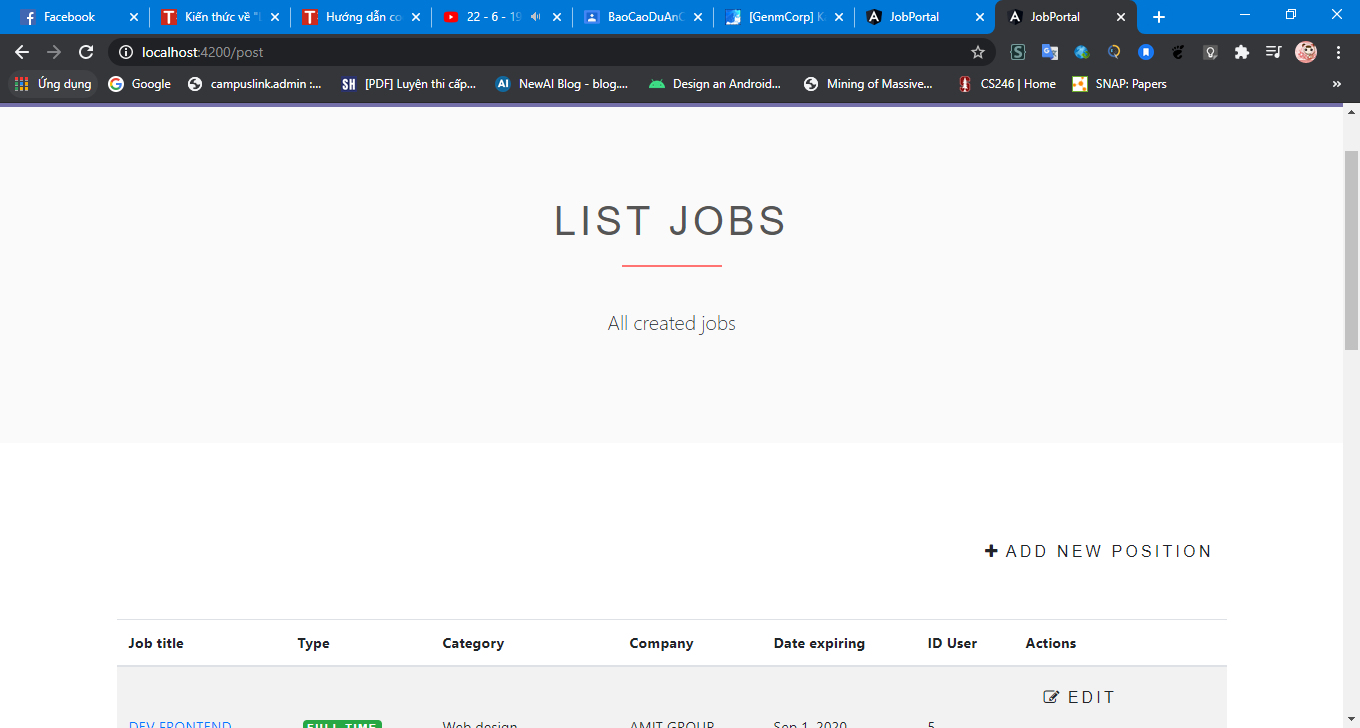
Hình 2.24: Form tạo công việc



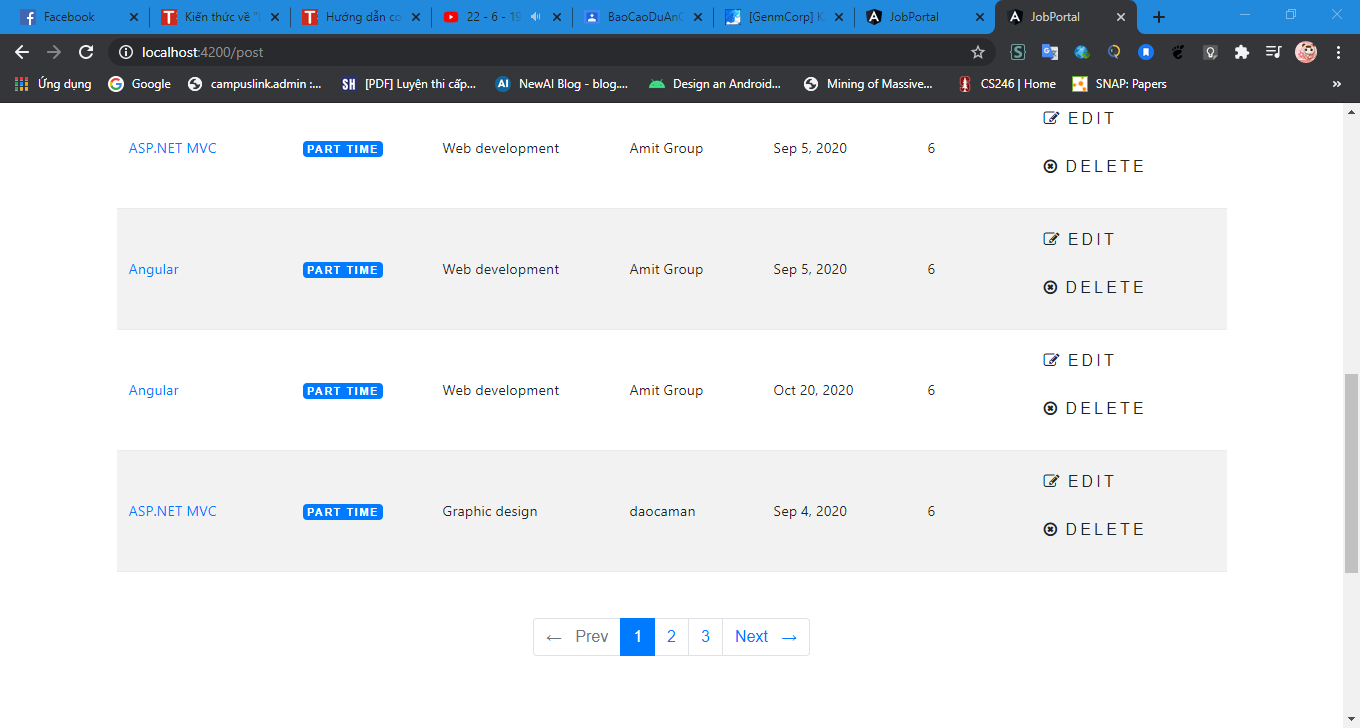
Hình 2.25: Form tạo công việc



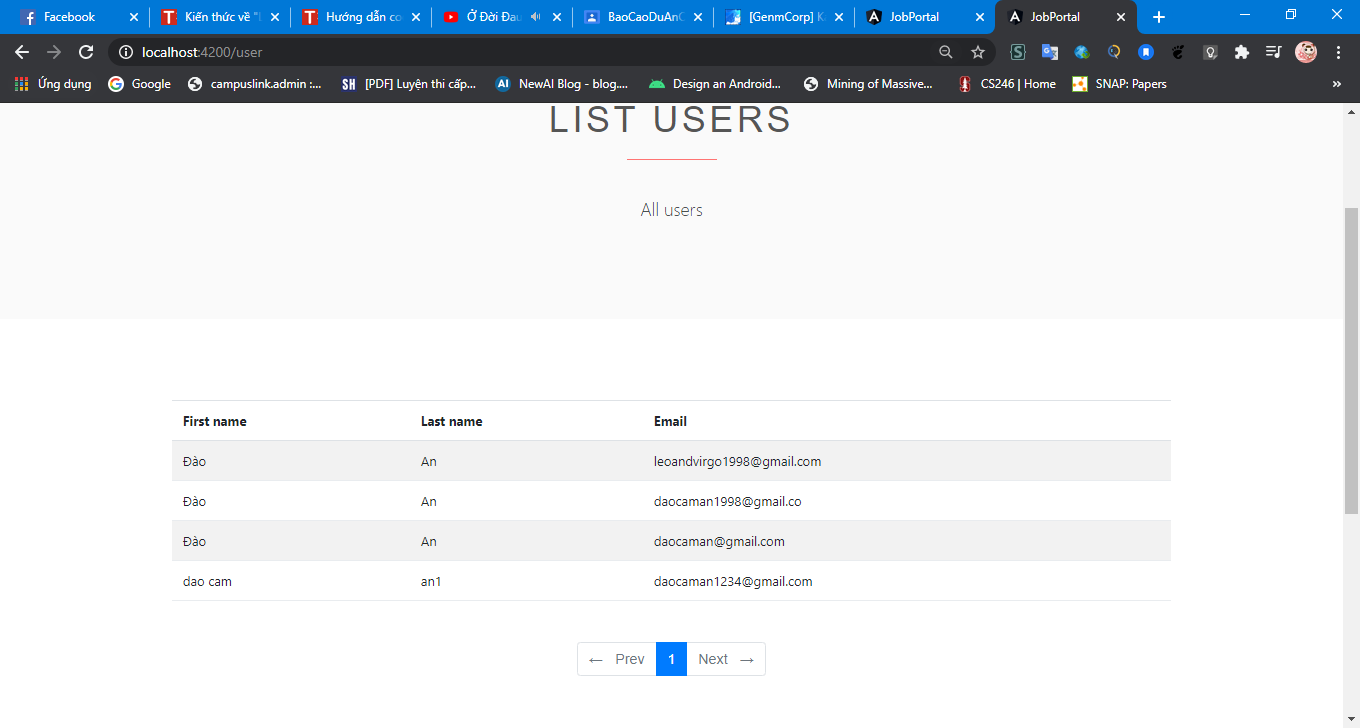
Hình 2.26: Form tạo công việc



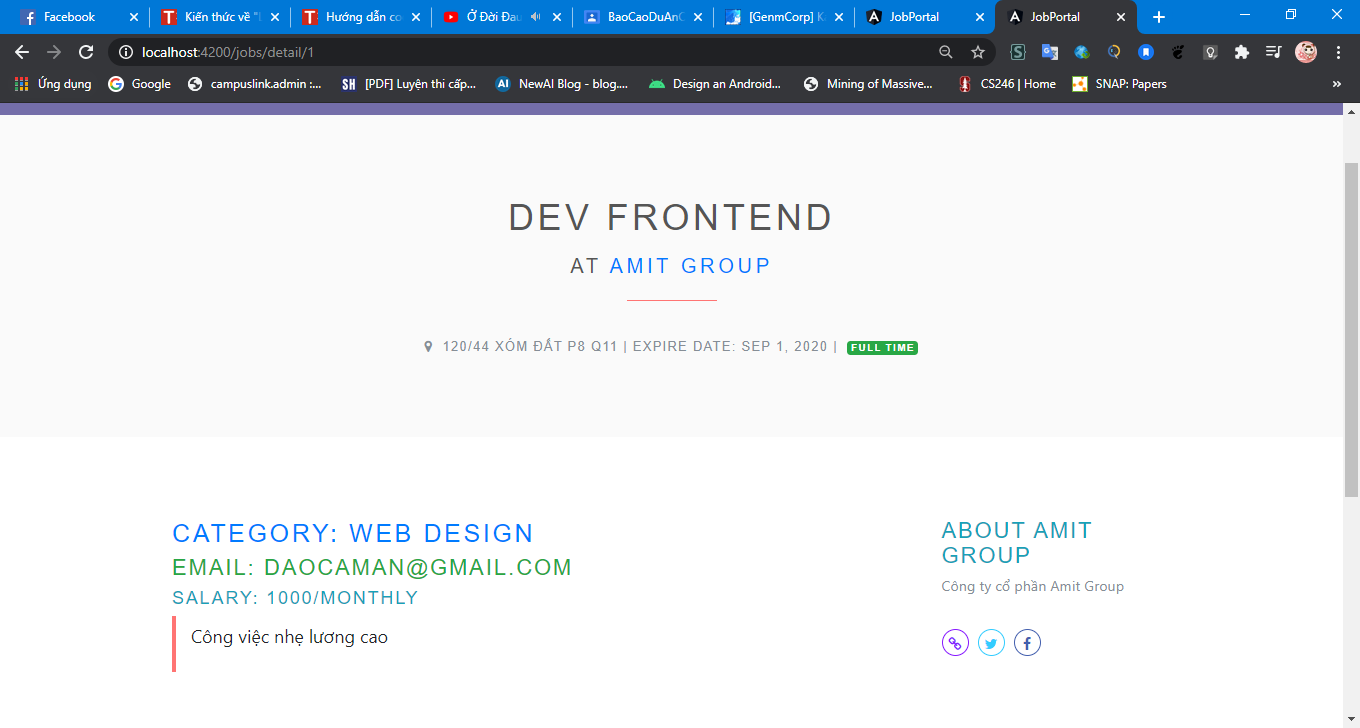
Hình 2.27: Danh sách công việc quyền user



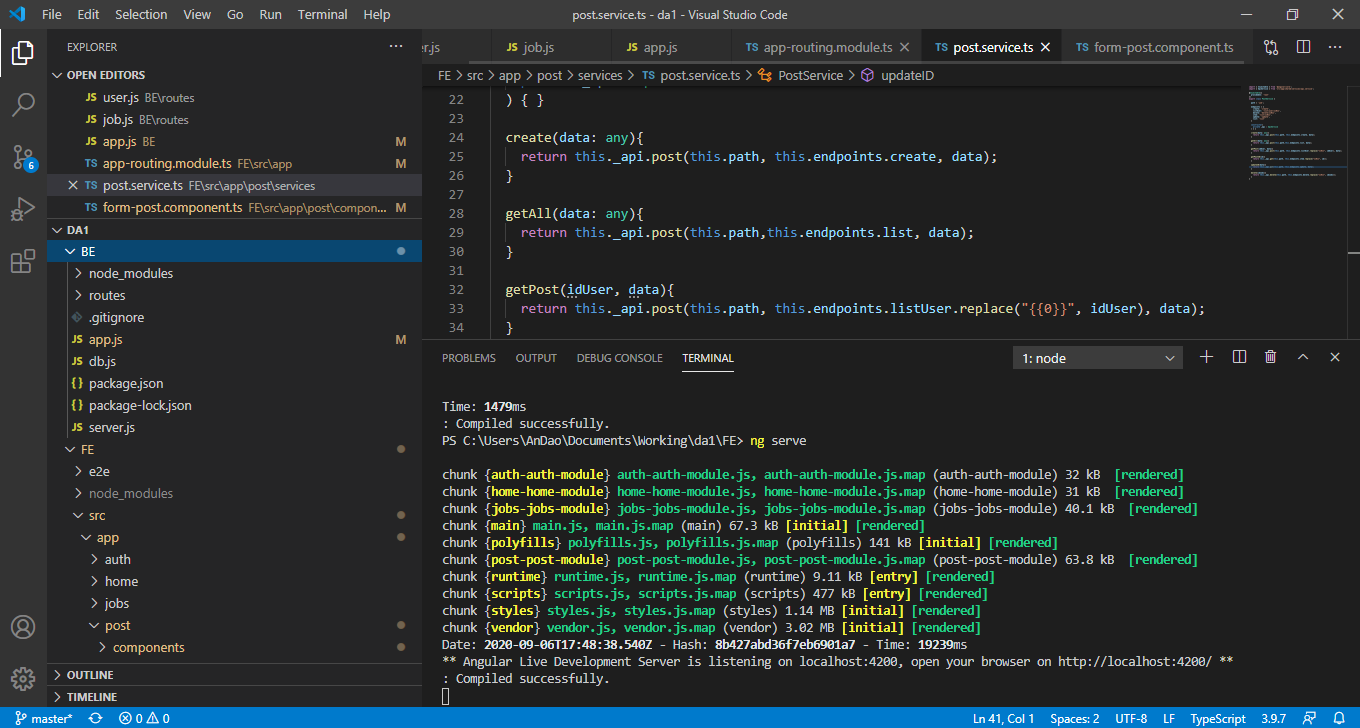
Hình 2.28: Danh sách công việc quyền user



Hình 2.29: Danh sách user với quyền admin

\

Hình 2.30: Trang hiển thị công việc cụ thể



Hình 2.31: Cách chạy project

2.6 Kết luận

Việc viết web bằng Angular giúp em học hỏi được cách phân tách components nhỏ, sử dụng các lazy loading module để cải thiện tính năng, tốc độ của trang web. Bên cạnh đó việc tái sử dụng code được cải thiện rõ rệt và dễ quản lý với các dự án lớn và trung. Bên cạnh đó khi chia thành các module độc lập có thể tái sử dụng ở nhiều project khác nhau mà không quá khó khăn trong việc phải viết lại từ đâu. Bên cạnh đó việc viết bằng Angular cũng trở nên phức tạp khi các components được tách nhỏ và quản lý chặt chẽ khiến sự phát sinh các file trong project lớn. Nhưng với khả năng build optimize và hashing source giúp giảm thiểu tối đa tài nguyên bên cạnh đó còn mã hóa khiến cho việc bảo mật đạt được tốt nhất.

**TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. <https://viblo.asia/p/dung-mot-api-restful-don-gian-su-dung-express-va-mongodb-6J3Zg99R5mB>

**Tiếng Anh**

1. <https://github.com/expressjs/express>
2. <https://github.com/manjurulhoque/django-job-portal\>
3. <https://www.npmjs.com/package/mysql>
4. <https://www.npmjs.com/package/nodemon>
5. <https://angular.io/>