

**ĐẠI HỌC ĐÀ NẴNG**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL4: DỰ ÁN HỆ ĐIỀU HÀNH  
& MẠNG MÁY TÍNH**

**Đề tài : Xây dựng website hỗ trợ điểm danh bằng nhận  
diện khuôn mặt sử dụng mô hình Client – Server  
triển khai trên Raspberry Pi 4**

**SINH VIÊN THỰC HIỆN:**

TRẦN ĐỨC TRÍ	LỚP: 21TCLC_KHDL	NHÓM: 21N11B
PHẠM NGUYỄN ANH PHÁT	LỚP: 21TCLC_KHDL	NHÓM: 21N11B
LÊ ANH TUẤN	LỚP: 21TCLC_KHDL	NHÓM: 21N11B

**GIẢNG VIÊN HƯỚNG DẪN: TS. ĐẶNG THIỀN BÌNH**

**Đà Nẵng 12/2023**

### **LỜI CẢM ƠN**

*Lời đầu tiên, chúng em xin gửi lời cảm ơn chân thành đến quý thầy cô khoa Công nghệ thông tin, trường Đại học Bách khoa – Đại học Đà Nẵng đã tạo điều kiện và giúp đỡ chúng em trong quá trình thực hiện đồ án hệ điều hành và mạng máy tính. Đặc biệt, chúng em muốn bày tỏ lòng biết ơn sâu sắc đến thầy Đặng Thiên Bình, người đã không ngừng định hướng, đồng hành và hướng dẫn chúng em từng bước.*

*Trong quá trình thực hiện, chắc chắn không tránh khỏi những thiếu sót và vẫn còn những khía cạnh cần được cải thiện. Chúng em mong nhận được sự góp ý chân thành từ phía quý thầy cô và các bạn để sản phẩm của chúng em ngày càng hoàn thiện hơn. Chúng em xin chân thành cảm ơn.*

## MỤC LỤC

<b>DANH SÁCH HÌNH VẼ .....</b>	<b>4</b>
<b>DANH SÁCH CÁC TỪ VIẾT TẮT .....</b>	<b>9</b>
<b>GIỚI THIỆU ĐỀ TÀI .....</b>	<b>10</b>
<b>CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....</b>	<b>11</b>
1.1.    Mô hình TCP/IP:.....	11
1.1.1.    Tầng liên kết.....	11
1.1.2.    Tầng mạng (Internet).....	11
1.1.3.    Tầng giao vận .....	11
1.1.4.    Tầng ứng dụng.....	11
1.2.    Mô hình Client-Server: .....	12
1.3.    Các giao thức mạng : .....	13
1.3.1.    IP .....	13
1.3.2.    SMTP.....	13
1.3.3.    HTTP/HTTPS:.....	15
1.3.4.    SSH.....	16
1.3.5.    DNS .....	17
1.4.    Dịch vụ DDNS:.....	17
1.5.    Chứng chỉ SSL:.....	18
1.6.    Cân bằng tải (NLB): .....	21
1.7.    Connection Pool:.....	22
1.8.    WebSocket: .....	23
1.9.    Nhận diện khuôn mặt:.....	26
1.9.1.    Model VGG16:.....	26
1.9.2.    Thuật toán Kmean: .....	26
<b>CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....</b>	<b>30</b>
2.1.    Yêu cầu hệ thống: .....	30
2.1.1.    Yêu cầu phần cứng: .....	30
2.1.2.    Yêu cầu phần mềm: .....	30
2.2.    Thiết kế hệ thống: .....	31

---

2.2.1.	Thiết kế lớp (Model) và cơ sở dữ liệu (Database): .....	31
2.2.2.	Phân tích các chức năng của hệ thống:.....	34
<b>CHƯƠNG 3:</b>	<b>TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ .....</b>	<b>49</b>
3.1.	Triển khai:.....	49
3.1.1.	Triển khai dự án: .....	49
3.1.2.	Triển khai hạ tầng:.....	56
3.1.3.	Triển khai môi trường và mã nguồn.....	64
3.2.	Kết quả chạy chương trình.....	80
3.2.1.	Quản lý người dùng:.....	80
3.2.2.	Quản lý lịch học/dạy: .....	90
3.2.3.	Quản lý điểm danh: .....	93
3.2.4.	Trò chuyện:.....	99
	<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>102</b>

## DANH SÁCH HÌNH VẼ

<b>Hình 1.</b>	<b>Mô hình TCP/IP .....</b>	<b>12</b>
<b>Hình 2.</b>	<b>Mô hình Client/Server.....</b>	<b>12</b>
<b>Hình 3.</b>	<b>Cấu trúc gói tin IP .....</b>	<b>13</b>
<b>Hình 4.</b>	<b>Cơ chế hoạt động SMTP .....</b>	<b>14</b>
<b>Hình 5.</b>	<b>Cơ chế hoạt động HTTP .....</b>	<b>15</b>
<b>Hình 6.</b>	<b>Cơ chế hoạt động HTTPS .....</b>	<b>15</b>
<b>Hình 7.</b>	<b>Cơ chế hoạt động SSH.....</b>	<b>16</b>
<b>Hình 8.</b>	<b>Cơ chế hoạt động của DNS .....</b>	<b>17</b>
<b>Hình 9.</b>	<b>Cơ chế hoạt động dịch vụ DDNS - 1 .....</b>	<b>17</b>
<b>Hình 10.</b>	<b>Cơ chế hoạt động của dịch vụ DDNS - 2 .....</b>	<b>18</b>
<b>Hình 11.</b>	<b>Cơ chế hoạt động chứng chỉ SSL: .....</b>	<b>20</b>
<b>Hình 12.</b>	<b>Cơ chế hoạt động của cân bằng tải .....</b>	<b>21</b>
<b>Hình 13.</b>	<b>Connection Pool - 1.....</b>	<b>22</b>
<b>Hình 14.</b>	<b>Connection Pool - 2.....</b>	<b>22</b>
<b>Hình 15.</b>	<b>Cơ chế hoạt động của WebSocket .....</b>	<b>23</b>
<b>Hình 16.</b>	<b>Kiến trúc Channel Django .....</b>	<b>25</b>
<b>Hình 17.</b>	<b>Mạng tích chập VGG-16 .....</b>	<b>26</b>
<b>Hình 18.</b>	<b>Phân cụm K cluster của K-mean .....</b>	<b>27</b>
<b>Hình 19.</b>	<b>Thuật toán K-mean - 1 .....</b>	<b>27</b>
<b>Hình 20.</b>	<b>Thuật toán K-mean - 2 .....</b>	<b>28</b>
<b>Hình 21.</b>	<b>Thuật toán K-mean - 3 .....</b>	<b>28</b>
<b>Hình 22.</b>	<b>Thuật toán K-mean - 4 .....</b>	<b>28</b>
<b>Hình 23.</b>	<b>Thuật toán K-mean - 5 .....</b>	<b>29</b>
<b>Hình 24.</b>	<b>Sơ đồ class.....</b>	<b>31</b>
<b>Hình 25.</b>	<b>Sơ đồ database .....</b>	<b>32</b>
<b>Hình 26.</b>	<b>Danh sách API - 1 .....</b>	<b>33</b>
<b>Hình 27.</b>	<b>Danh sách API - 2 .....</b>	<b>33</b>
<b>Hình 28.</b>	<b>Danh sách API - 3 .....</b>	<b>33</b>
<b>Hình 29.</b>	<b>Sơ đồ usecase – Tài khoản và người dùng.....</b>	<b>34</b>

---

<b>Hình 30.</b>	<b>Postman – Đăng nhập .....</b>	<b>35</b>
<b>Hình 31.</b>	<b>Sơ đồ tuần tự - Đăng nhập.....</b>	<b>36</b>
<b>Hình 32.</b>	<b>Postman – Kích hoạt tài khoản .....</b>	<b>37</b>
<b>Hình 33.</b>	<b>Postman – Lấy thông tin người dùng .....</b>	<b>38</b>
<b>Hình 34.</b>	<b>Sơ đồ tuần tự - Lấy thông tin người dùng.....</b>	<b>38</b>
<b>Hình 35.</b>	<b>Postman – Tìm kiếm người dùng .....</b>	<b>39</b>
<b>Hình 36.</b>	<b>Sơ đồ usecase – Khóa học .....</b>	<b>40</b>
<b>Hình 37.</b>	<b>Postman – Danh sách khóa học.....</b>	<b>41</b>
<b>Hình 38.</b>	<b>Sơ đồ tuần tự - Xem lịch học hôm nay.....</b>	<b>41</b>
<b>Hình 39.</b>	<b>Sơ đồ usecase – Điểm danh .....</b>	<b>42</b>
<b>Hình 40.</b>	<b>Postman – Nhận diện khuôn mặt .....</b>	<b>43</b>
<b>Hình 41.</b>	<b>Sơ đồ tuần tự - Điểm danh.....</b>	<b>44</b>
<b>Hình 42.</b>	<b>Postman – Danh sách điểm danh .....</b>	<b>45</b>
<b>Hình 43.</b>	<b>Sơ đồ tuần tự - Lịch sử điểm danh.....</b>	<b>46</b>
<b>Hình 44.</b>	<b>Sơ đồ tuần tự - Lịch sử điểm danh.....</b>	<b>46</b>
<b>Hình 45.</b>	<b>Sơ đồ usecase – Trò chuyện và Trạng thái trực tuyến .....</b>	<b>47</b>
<b>Hình 46.</b>	<b>Trello – Dashboard.....</b>	<b>50</b>
<b>Hình 47.</b>	<b>Trello – Các mục của một chu kỳ.....</b>	<b>50</b>
<b>Hình 48.</b>	<b>Trello – Task .....</b>	<b>51</b>
<b>Hình 49.</b>	<b>Trang github của dự án.....</b>	<b>52</b>
<b>Hình 50.</b>	<b>Postman – Thư mục.....</b>	<b>53</b>
<b>Hình 51.</b>	<b>Postman – API trong một thư mục .....</b>	<b>54</b>
<b>Hình 52.</b>	<b>Postman - Thiết lập môi trường kiểm thử ở loca .....</b>	<b>54</b>
<b>Hình 53.</b>	<b>Đăng nhập vào trang quản trị .....</b>	<b>56</b>
<b>Hình 54.</b>	<b>Giao diện của trang quản trị Home Gateway - 1.....</b>	<b>57</b>
<b>Hình 55.</b>	<b>Giao diện của trang quản trị Home Gateway - 1.....</b>	<b>57</b>
<b>Hình 56.</b>	<b>Gán IP nội bộ LAN 192.168. Raspberry dựa vào MAC .....</b>	<b>58</b>
<b>Hình 57.</b>	<b>ARP tại HomeGateway với IP và MAC của Raspberry .....</b>	<b>58</b>
<b>Hình 58.</b>	<b>NAT port bằng Virtual Server .....</b>	<b>59</b>
<b>Hình 59.</b>	<b>Cấu hình NAT PORT - 1 .....</b>	<b>60</b>

---

<b>Hình 60.</b>	<b>Cấu hình NAT PORT - 2 .....</b>	<b>60</b>
<b>Hình 61.</b>	<b>Cài đặt dịch vụ DDNS sử dụng nhà cung cấp No-IP: .....</b>	<b>61</b>
<b>Hình 62.</b>	<b>Giao diện đăng nhập của No-IP .....</b>	<b>61</b>
<b>Hình 63.</b>	<b>Trang chủ của No-IP .....</b>	<b>62</b>
<b>Hình 64.</b>	<b>Đăng ký Hostname tại No-IP .....</b>	<b>62</b>
<b>Hình 65.</b>	<b>Cài đặt phần mềm cập nhật IP ở Raspberry .....</b>	<b>63</b>
<b>Hình 66.</b>	<b>Kiểm tra các cổng của Raspberry - 1.....</b>	<b>63</b>
<b>Hình 67.</b>	<b>Kiểm tra các cổng của Raspberry - 2.....</b>	<b>63</b>
<b>Hình 68.</b>	<b>IP mới được ánh xạ DDNS sau khi bật tắt modem .....</b>	<b>64</b>
<b>Hình 69.</b>	<b>Cài đặt SSL Let's Encrypt với Certbot .....</b>	<b>64</b>
<b>Hình 70.</b>	<b>Kiểm tra chứng chỉ SSL - 1 .....</b>	<b>65</b>
<b>Hình 71.</b>	<b>Kiểm tra chứng chỉ SSL.....</b>	<b>66</b>
<b>Hình 72.</b>	<b>Cấu hình Database Connection Pooling .....</b>	<b>66</b>
<b>Hình 73.</b>	<b>Cấu hình cân bằng tải .....</b>	<b>67</b>
<b>Hình 74.</b>	<b>File cấu hình nginx.conf - 1.....</b>	<b>68</b>
<b>Hình 75.</b>	<b>File cấu hình nginx.conf - 2.....</b>	<b>68</b>
<b>Hình 76.</b>	<b>File cấu hình dutchecker.conf - 1 .....</b>	<b>69</b>
<b>Hình 77.</b>	<b>File cấu hình dutchecker.conf - 2 .....</b>	<b>70</b>
<b>Hình 78.</b>	<b>File cấu hình dutchecker.conf - 3 .....</b>	<b>70</b>
<b>Hình 79.</b>	<b>File cấu hình dutchecker.conf - 4 .....</b>	<b>71</b>
<b>Hình 80.</b>	<b>File cấu hình dutchecker.conf - 5 .....</b>	<b>71</b>
<b>Hình 81.</b>	<b>Raspberry được cấp nguồn để hoạt động liên tục .....</b>	<b>75</b>
<b>Hình 82.</b>	<b>Giao diện của Raspberry sau khi cài đặt mã nguồn .....</b>	<b>76</b>
<b>Hình 83.</b>	<b>Khởi động server BackEnd thứ nhất.....</b>	<b>77</b>
<b>Hình 84.</b>	<b>Khởi động server BackEnd thứ hai .....</b>	<b>77</b>
<b>Hình 85.</b>	<b>Khởi động server WebSocket.....</b>	<b>78</b>
<b>Hình 86.</b>	<b>Khởi động server FrontEnd .....</b>	<b>78</b>
<b>Hình 87.</b>	<b>Khởi động server AI .....</b>	<b>79</b>
<b>Hình 88.</b>	<b>Kiểm tra trạng thái hoạt động của Nginx .....</b>	<b>79</b>
<b>Hình 89.</b>	<b>Giao diện đăng nhập admin.....</b>	<b>80</b>

---

<b>Hình 90.</b>	<b>Giao diện quản lý các model.....</b>	<b>80</b>
<b>Hình 91.</b>	<b>Giao diện quản lý người dùng .....</b>	<b>81</b>
<b>Hình 92.</b>	<b>Nhập thông tin .....</b>	<b>81</b>
<b>Hình 93.</b>	<b>Tạo tài khoản và gửi mail kích hoạt tài khoản .....</b>	<b>82</b>
<b>Hình 94.</b>	<b>Tài khoản được tạo nhưng chưa kích hoạt .....</b>	<b>82</b>
<b>Hình 95.</b>	<b>User Kiểm tra mail và truy cập link .....</b>	<b>83</b>
<b>Hình 96.</b>	<b>Giao diện sao khi kích vào link kích hoạt tài khoản .....</b>	<b>83</b>
<b>Hình 97.</b>	<b>Cập nhật mật khẩu và up ảnh để nhận diện khuôn mặt .....</b>	<b>84</b>
<b>Hình 98.</b>	<b>Cập nhật mật khẩu và kích hoạt tài khoản thành công.....</b>	<b>84</b>
<b>Hình 99.</b>	<b>Cập nhật đặc trưng khuôn mặt thành công.....</b>	<b>85</b>
<b>Hình 100.</b>	<b>Giao diện đăng nhập User .....</b>	<b>85</b>
<b>Hình 101.</b>	<b>Giao diện trang cá nhân User.....</b>	<b>86</b>
<b>Hình 102.</b>	<b>Nhập lại mật khẩu .....</b>	<b>86</b>
<b>Hình 103.</b>	<b>Nhập mã ID khi quên mật khẩu.....</b>	<b>87</b>
<b>Hình 104.</b>	<b>Mail cập nhật mật khẩu đã quên được gửi .....</b>	<b>87</b>
<b>Hình 105.</b>	<b>Vào Mail quên mật khẩu và truy cập đường link .....</b>	<b>88</b>
<b>Hình 106.</b>	<b>Cập nhật mật khẩu thành công.....</b>	<b>88</b>
<b>Hình 107.</b>	<b>Tìm kiếm User trên thanh search .....</b>	<b>89</b>
<b>Hình 108.</b>	<b>Giao diện trang cá nhân người cần tìm .....</b>	<b>89</b>
<b>Hình 109.</b>	<b>Lịch học hôm nay của sinh viên .....</b>	<b>90</b>
<b>Hình 110.</b>	<b>Lịch dạy của giáo viên sau khi đăng nhập .....</b>	<b>90</b>
<b>Hình 111.</b>	<b>xem lịch dạy hoặc học dưới vai trò admin .....</b>	<b>91</b>
<b>Hình 112.</b>	<b>Lịch học của sinh viên tại trang cá nhân.....</b>	<b>91</b>
<b>Hình 113.</b>	<b>Lịch dạy của giáo viên tại trang cá nhân.....</b>	<b>92</b>
<b>Hình 114.</b>	<b>Xem lịch dạy hoặc học của user bất kỳ dưới vai trò admin .</b>	<b>92</b>
<b>Hình 115.</b>	<b>Xem trạng thái điểm danh hiện tại .....</b>	<b>93</b>
<b>Hình 116.</b>	<b>Cho phép mở vị trí.....</b>	<b>93</b>
<b>Hình 117.</b>	<b>Kiểm tra vị trí thành công .....</b>	<b>94</b>
<b>Hình 118.</b>	<b>Cho phép mở camera .....</b>	<b>94</b>
<b>Hình 119.</b>	<b>Chụp ảnh khuôn mặt để điểm danh .....</b>	<b>95</b>

---

<b>Hình 120.</b>	<b>Kiểm tra khuôn mặt thành công .....</b>	<b>95</b>
<b>Hình 121.</b>	<b>Điểm danh thành công .....</b>	<b>96</b>
<b>Hình 122.</b>	<b>Cập nhật trạng thái đã điểm danh ở máy sinh viên .....</b>	<b>96</b>
<b>Hình 123.</b>	<b>Cập nhật trạng thái điểm danh ở máy giáo viên .....</b>	<b>97</b>
<b>Hình 124.</b>	<b>Lịch sử điểm danh vừa được cập nhật .....</b>	<b>97</b>
<b>Hình 125.</b>	<b>Lịch sử điểm danh xem tại máy giáo viên .....</b>	<b>98</b>
<b>Hình 126.</b>	<b>Lịch sử điểm danh xem tại máy admin .....</b>	<b>98</b>
<b>Hình 127.</b>	<b>Đăng nhập với vai trò giáo viên.....</b>	<b>99</b>
<b>Hình 128.</b>	<b>Trạng thái trực tuyến của giáo viên tại trang sinh viên.....</b>	<b>99</b>
<b>Hình 129.</b>	<b>Sinh viên tạo phòng chat với giáo viên .....</b>	<b>100</b>
<b>Hình 130.</b>	<b>Các phòng chat hiện tại của User.....</b>	<b>100</b>
<b>Hình 131.</b>	<b>Giao diện phòng chat từ phía sinh viên .....</b>	<b>101</b>
<b>Hình 132.</b>	<b>Giao diện từ phía giáo viên .....</b>	<b>101</b>

## DANH SÁCH CÁC TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
TCP/IP	Transmission Control Protocol/Internet Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
SSH	Secure Socket Shell
SMTP	Simple Mail Transfer Protocol
HTTP	HyperText Transfer Protocol
WS	WebSocket
DDNS	Dynamic Domain Name System
SSL	Secure Sockets Layer
NLB	Network Load Balancing

## GIỚI THIỆU ĐỀ TÀI

Trong bối cảnh hiện nay, quản lý và kiểm soát điểm danh trong các cơ sở giáo dục, doanh nghiệp và tổ chức ngày càng đặt ra yêu cầu cao về sự tự động hóa, nhanh chóng và tích hợp công nghệ tiên tiến. Hướng tới mục tiêu cải thiện quy trình điểm danh truyền thống, dự án của chúng em đã lựa chọn đề tài "**Xây dựng website hỗ trợ điểm danh thông qua nhận diện khuôn mặt, sử dụng mô hình Client – Server triển khai trên Raspberry Pi 4.**"

Đặc điểm độc đáo của dự án là sự kết hợp hài hòa giữa công nghệ nhận diện khuôn mặt và hệ thống Client – Server chạy trên Raspberry Pi 4 - một bo mạch nhỏ gọn, nhưng vô cùng mạnh mẽ. Qua đó, chúng em hướng đến mục tiêu tối ưu hóa hiệu suất và tiết kiệm năng lượng, đồng thời giảm bớt gánh nặng về thủ tục và thời gian cho cả sinh viên và nhân viên quản lý.

Hệ thống website đã được xây dựng với mục đích hỗ trợ quản lý điểm danh một cách thuận tiện và chính xác. Sự khả năng nhận diện khuôn mặt không chỉ loại bỏ các thủ tục phức tạp mà còn giảm thời gian cần thiết cho quá trình điểm danh. Điều này đồng thời giúp tiết kiệm chi phí và tăng tính tiện ích, đặc biệt so với các hệ thống trung tâm mạnh mẽ truyền thống.

Chúng em hy vọng rằng dự án này không chỉ đóng góp vào việc nâng cao chất lượng quản lý trong giáo dục đại học mà còn mang lại sự tiện lợi và hiệu quả cho cả sinh viên và nhân viên. Đồng thời, chúng em kỳ vọng rằng đề tài sẽ mở ra những triển vọng mới trong việc tích hợp công nghệ vào quản lý các cơ sở giáo dục và tổ chức, hướng tới tương lai hiện đại và thông minh.

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Mô hình TCP/IP:

Mô hình TCP/IP, hay còn được gọi là mô hình Internet, là một khung lý thuyết cơ bản cho việc thiết kế và triển khai mạng máy tính. Được tạo ra bởi DARPA (Defense Advanced Research Projects Agency) cho dự án ARPANET, mô hình này đã trở thành nền tảng cho Internet và nhiều mạng hiện đại. Mô hình TCP/IP chia quá trình truyền thông thành một số lớp, mỗi lớp đóng vai trò cụ thể trong quá trình này.

#### 1.1.1. Tầng liên kết

Đây là lớp thấp nhất trong mô hình, quản lý gửi và nhận dữ liệu giữa các thiết bị trực tiếp kết nối với nhau trên một mạng cục bộ. Nó thường bao gồm các giao thức như Ethernet và Wi-Fi.

#### 1.1.2. Tầng mạng (Internet)

Lớp này quản lý việc định tuyến dữ liệu từ nguồn đến đích trên mạng. Giao thức quan trọng ở đây là IP (Internet Protocol), chịu trách nhiệm đặt địa chỉ IP cho thiết bị và chuyển gói tin từ một nút đến nút khác.

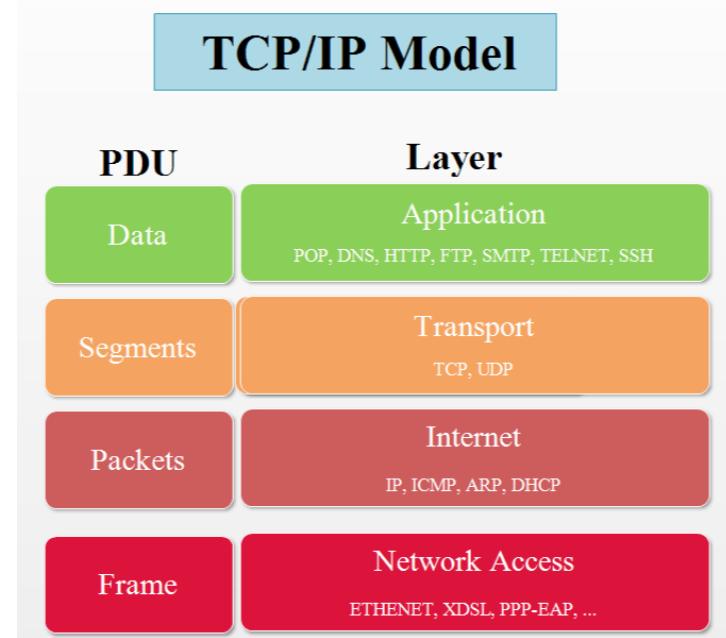
#### 1.1.3. Tầng giao vận

Lớp này quản lý kết nối giữa hai thiết bị truyền tin qua mạng. Giao thức chính ở đây bao gồm TCP (Transmission Control Protocol) và UDP (User Datagram Protocol), với TCP đảm bảo truyền tải tin cậy và UDP tập trung vào truyền tải nhanh chóng..

#### 1.1.4. Tầng ứng dụng

Lớp cuối cùng chịu trách nhiệm cho việc cung cấp các dịch vụ truy cập mạng cho người dùng. Giao thức ở đây bao gồm HTTP (Hypertext Transfer Protocol), FTP (File

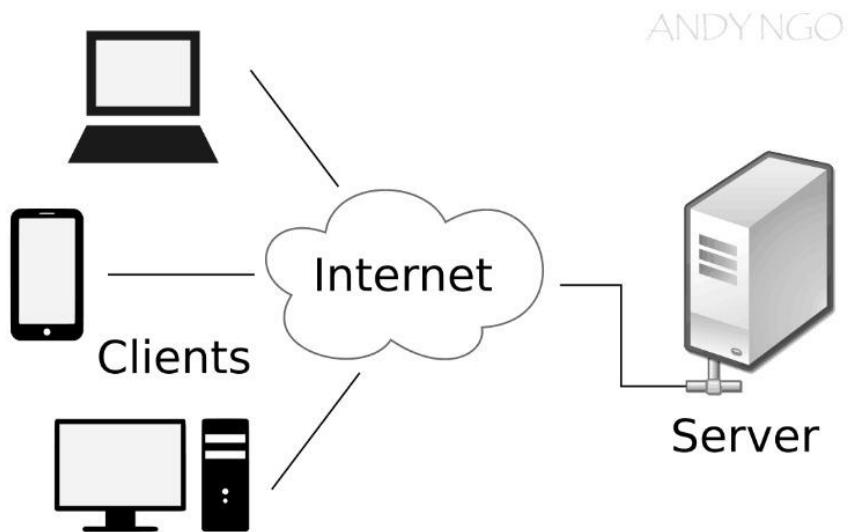
Transfer Protocol) và SMTP (Simple Mail Transfer Protocol).



Hình 1. Mô hình TCP/IP

## 1.2. Mô hình Client-Server:

**Mô hình Client Server** là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.



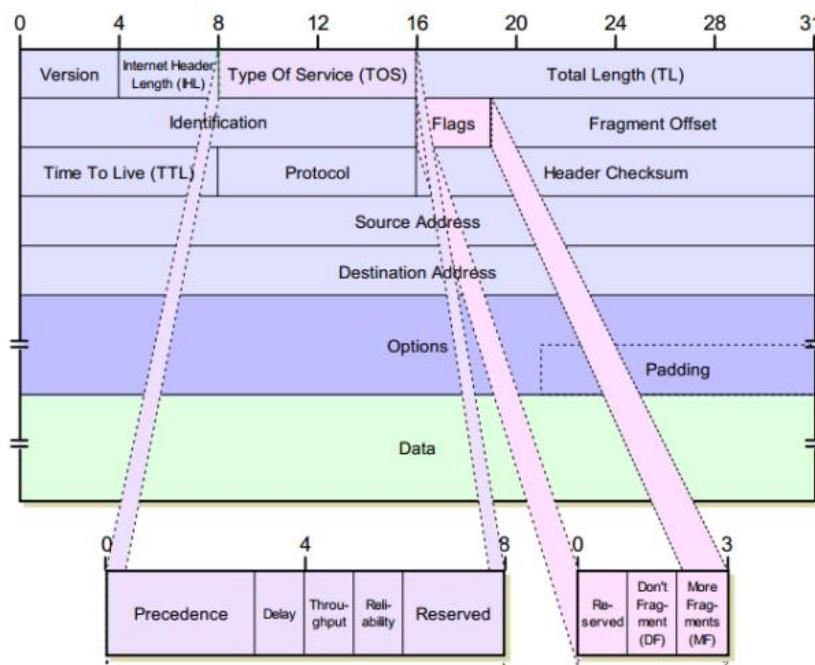
Hình 2. Mô hình Client/Server

**Client:** Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.

**Server:** Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn.

### 1.3. Các giao thức mạng :

#### 1.3.1. IP



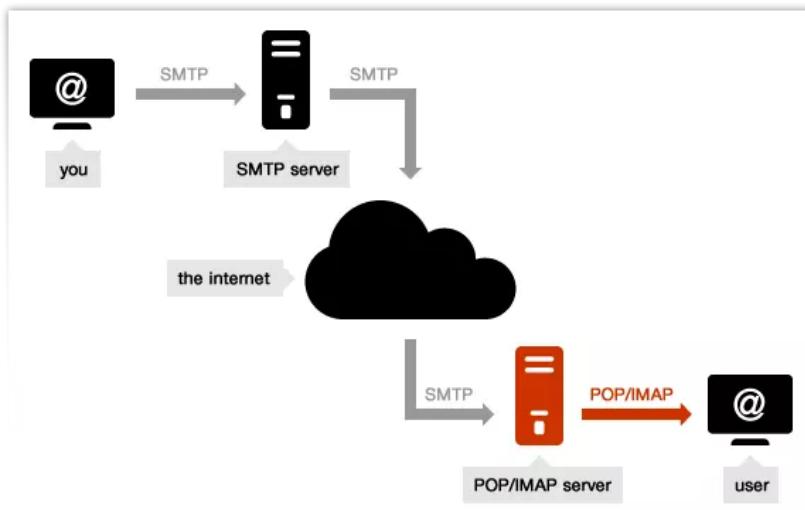
Hình 3.Cấu trúc gói tin IP

#### Cấu trúc gói tin

Internet Protocol là - Giao thức Internet (IP) là một giao thức hoặc bộ quy tắc để định tuyến và xử lý các gói dữ liệu để chúng có thể di chuyển qua các mạng máy tính và đến đích chính xác. Dữ liệu truyền qua Internet được chia thành các phần nhỏ hơn, được gọi là các gói. Thông tin IP được đính kèm vào mỗi gói và thông tin này giúp các bộ định tuyến gửi các gói đến đúng nơi. Mọi thiết bị hoặc miền kết nối với Internet đều được gán một địa chỉ IP - Internet Protocol và khi các gói được hướng đến địa chỉ IP - Internet Protocol được đính kèm với chúng, dữ liệu sẽ đến nơi cần thiết.

#### 1.3.2. SMTP

Simple Mail Transfer Protocol hay còn gọi là SMTP là hệ thống giao thức có nhiệm vụ nhận hay truyền tải dữ liệu trong email của người dùng. Hệ thống chỉ nhận và gửi thư điện tử email thông qua thiết bị có kết nối mạng Internet. Những thiết bị nhận và gửi email được gọi là máy chủ SMTP, mỗi máy chủ đều liên kết tới cổng mạng Internet 25 – cổng TCP

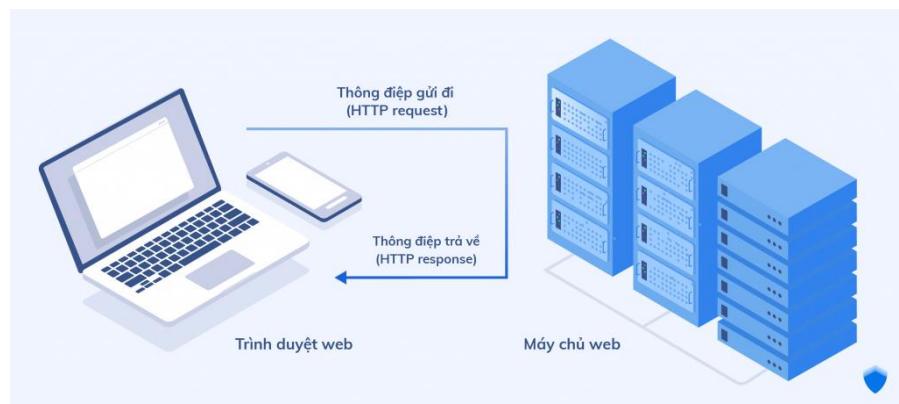


Hình 4. Cơ chế hoạt động SMTP

1. Ví dụ ứng dụng cả bạn sử dụng SendGrid SMTP, <https://smtp.sendgrid.net/>, dùng cổng 25 (theo mặc định) để gửi email.
2. Nếu bạn gửi email, ví dụ gửi đến một email có máy chủ là gmail chặng hạn (...@gmail.com). Đầu tiên Sendgrid sẽ liên hệ với phía Gmail để xác thực tài khoản người gửi là đang hoạt động. Sau đó chuyển tiếp thông tin thư, chặng hạn như địa chỉ người gửi, người nhận và nội dung tin nhắn bằng SMTP cho gmail.
3. SMTP server sendgrid (server gửi) gửi thông tin thư thu thập từ yêu cầu của bạn. Và sau đó lặp lại quá trình đàm thoại trong bước 2 với SMTP server gmail (server nhận).
4. SMTP server người nhận kiểm tra địa chỉ gửi, địa chỉ người nhận (để đảm bảo họ là người nhận hợp lệ) và nội dung thư. Nó kiểm tra domain gửi cho bất kỳ vấn đề DNS khả nghi. Chặng hạn như DKIM và chữ ký SPF không hợp lệ. Cho đến khi không có vấn đề, máy chủ thư của người nhận sẽ sử dụng giao thức POP3 hoặc IMAP để truy xuất email. Và cung cấp thông báo cho người nhận dự định của mình.

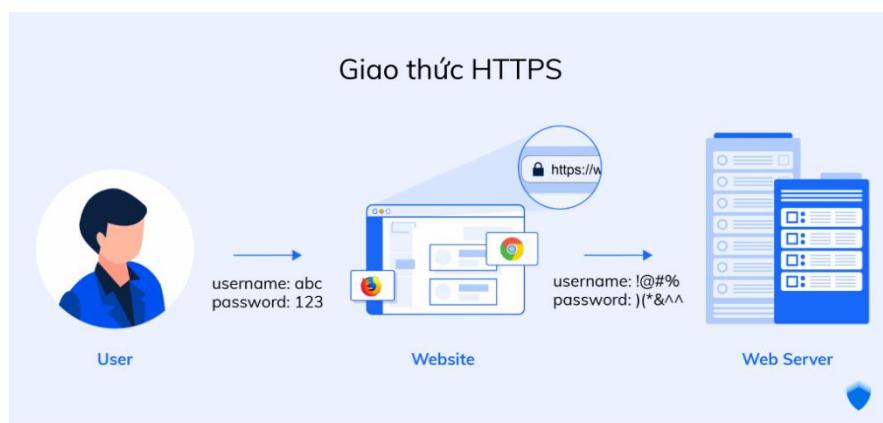
### 1.3.3. HTTP/HTTPS:

#### HTTP:



Hình 5. Cơ chế hoạt động HTTP

**HTTP (Hypertext Transfer Protocol)** là giao thức truyền tải siêu văn bản. Đây là giao thức tiêu chuẩn cho World Wide Web (www) để truyền tải dữ liệu dưới dạng văn bản, âm thanh, hình ảnh, video từ Web Server tới trình duyệt web của người dùng và ngược lại.



Hình 6. Cơ chế hoạt động HTTPS

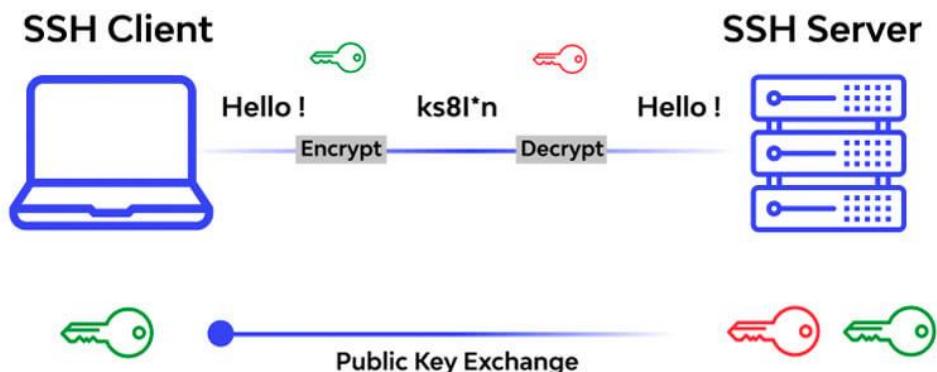
**HTTPS (Hypertext Transfer Protocol Secure)** là giao thức truyền tải siêu văn bản an toàn. Thực chất, đây chính là giao thức HTTP nhưng tích hợp thêm Chứng chỉ bảo mật SSL nhằm mã hóa các thông điệp giao tiếp để tăng tính bảo mật. Có thể hiểu, HTTPS là phiên bản HTTP an toàn, bảo mật hơn.

HTTPS hoạt động tương tự như HTTP, tuy nhiên được bổ sung thêm chứng chỉ **SSL (Secure Sockets Layer – tầng ố bảo mật)** hoặc **TLS (Transport Layer Security – bảo mật tầng truyền tải)**. Hiện tại, đây là các tiêu chuẩn bảo mật hàng đầu cho hàng triệu website trên toàn thế giới.

Cả SSL và TLS đều sử dụng hệ thống PKI (Public Key Infrastructure - hạ tầng khóa công khai) không đối xứng. Hệ thống này sử dụng hai “khóa” để mã hóa thông tin liên lạc, “khóa công khai” (public key) và “khóa riêng” (private key). Bất cứ thứ gì được mã

hóa bằng khóa công khai chỉ có thể được giải mã bởi khóa riêng và ngược lại. Các tiêu chuẩn này đảm bảo các nội dung sẽ được mã hóa trước khi truyền đi, và giải mã khi nhận. Điều này khiến hacker dù có chen ngang lấy được thông tin cũng không thể “hiểu” được thông tin đó.

### 1.3.4. SSH



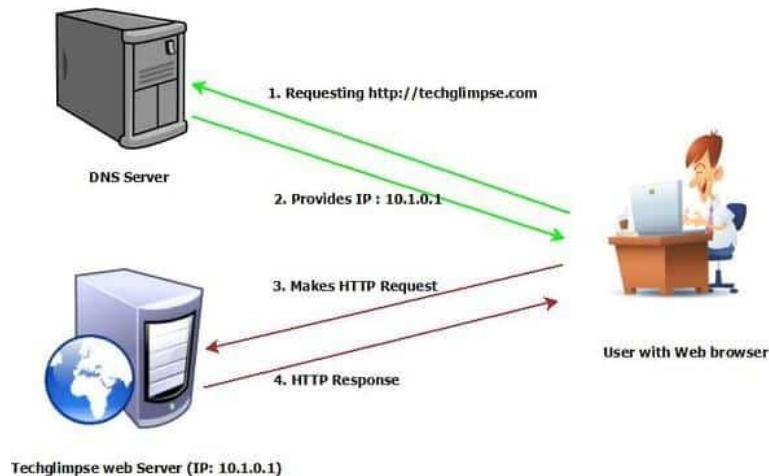
Hình 7. Cơ chế hoạt động SSH

**SSH** còn được gọi là Secure Shell, đây là một giao thức cho phép người dùng điều khiển từ xa, kiểm soát và chỉnh sửa Server thông qua Internet.

Giao thức SSH sử dụng mô hình Client-server kết nối với Session, bao gồm nhiều lớp và loại mã hóa khác nhau, cho phép chứng thực an toàn giữa hai thiết bị từ xa. Secure Shell vận hành trên TCP Port 22 mặc định, bắt đầu kết nối với Client bằng cách tạo ra TCP Handshake với Server. Hoạt động trên Server sau khi sử dụng Secure Shell đều được mã hóa, đảm bảo có thể kết nối Symmetric.

Quá trình hoạt động của Secure Shell sẽ có 2 phần cơ bản: kết nối và xác thực. Sau khi đã kết nối được hai thiết bị từ xa thông qua dòng lệnh "`ssh server.example.org`", giao thức sẽ tiến hành so sánh thông tin đăng nhập của người dùng để xác thực đúng kết nối. Bản chất của quá trình xác thực Secure Shell chủ yếu là hỗ trợ giữa hệ thống Host với nhau. Hoạt động này hỗ trợ cho cả trình giả lập Terminal, ứng dụng giao thức hoặc truyền File.

### 1.3.5. DNS



Hình 8. Cơ chế hoạt động của DNS

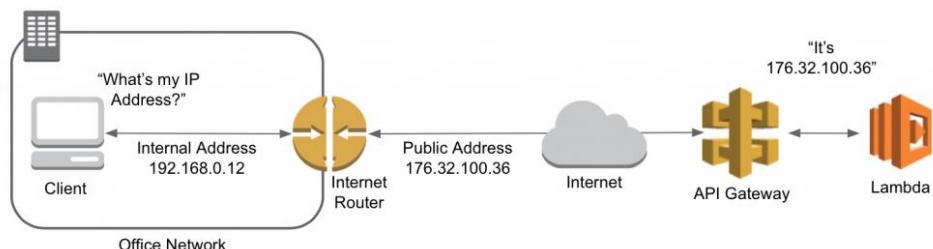
Hệ thống phân giải tên miền (DNS) về căn bản là một hệ thống giúp cho việc chuyển đổi các tên miền mà con người dễ ghi nhớ (dạng ký tự, ví dụ [www.example.com](http://www.example.com)) sang địa chỉ IP vật lý (dạng số, ví dụ **123.11.5.19**) tương ứng của tên miền đó. DNS giúp liên kết với các trang thiết bị mạng cho các mục đích định vị và địa chỉ hóa các thiết bị trên Internet.

### 1.4. Dịch vụ DDNS:

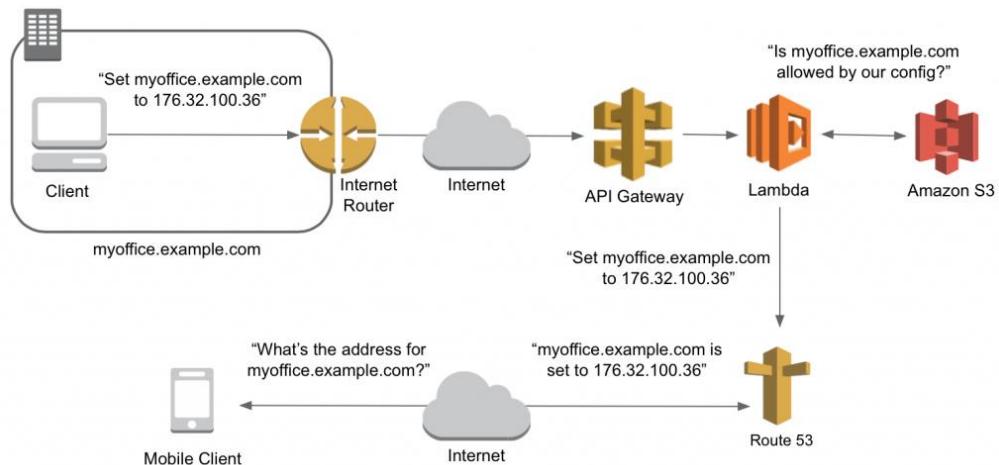
**DDNS** là viết tắt của "Dynamic Domain Name System" (Hệ thống Tên Miền Động). DDNS là một dịch vụ mà người dùng có thể sử dụng để định cấu hình một tên miền dễ nhớ cho một địa chỉ IP thay đổi động, thường được sử dụng trong các môi trường mà địa chỉ IP của thiết bị thường xuyên thay đổi.

Ở đây ta sử dụng nhà cung cấp dịch vụ DDNS No-IP để đơn giản hóa giải quyết vấn đề khi người dùng muốn truy cập các thiết bị như máy tính, máy chủ, hoặc camera IP từ xa thông qua Internet mà không phải lo lắng về việc theo dõi sự thay đổi của địa chỉ IP công cộng. Thay vì sử dụng địa chỉ IP, người dùng có thể sử dụng một tên miền dễ nhớ, và dịch vụ DDNS sẽ tự động cập nhật địa chỉ IP mới khi nó thay đổi.

Cơ chế hoạt động:



Hình 9. Cơ chế hoạt động dịch vụ DDNS - I



Hình 10. Cơ chế hoạt động của dịch vụ DDNS - 2

1. Bạn đăng ký tên miền với nhà cung cấp dịch vụ DNS động và định cấu hình cài đặt DNS
2. Bạn cung cấp cho nhà cung cấp địa chỉ IP ban đầu cho tên miền
3. Bạn cài đặt máy khách DNS động trên thiết bị hoặc phiên bản máy chủ với địa chỉ IP biến đổi

Máy khách DDNS liên tục theo dõi địa chỉ IP và phát hiện bất kỳ thay đổi nào xảy ra. Máy khách này gửi thông báo cập nhật bản ghi DNS đến nhà cung cấp DNS động để thông báo địa chỉ IP mới. Nhà cung cấp DNS động sửa đổi các bản ghi để chỉ đến địa chỉ IP mới.

Máy khách DNS động tiếp tục theo dõi địa chỉ IP để xem có thay đổi không. Bất cứ khi nào có một thay đổi mới, quá trình này sẽ lặp lại.

## 1.5. Chứng chỉ SSL:

Chứng chỉ SSL/TLS là đối tượng kỹ thuật số cho phép các hệ thống xác minh danh tính và sau đó thiết lập kết nối mạng được mã hóa với một hệ thống khác bằng giao thức Lớp cổng bảo mật/Bảo mật lớp truyền tải (SSL/TLS). Các chứng chỉ được sử dụng trong một hệ thống mật mã được gọi là cơ sở hạ tầng khóa công khai (PKI). PKI cung cấp phương thức để một bên thiết lập nhận dạng của một bên khác bằng cách sử dụng các chứng chỉ nếu cả hai đều tin cậy bên thứ ba - được gọi là cơ quan cấp chứng chỉ. Do đó, chứng chỉ SSL/TLS đóng vai trò là thẻ định danh kỹ thuật số để bảo mật hoạt động giao tiếp mạng, thiết lập danh tính của các trang web qua Internet cũng như tài nguyên trên các mạng riêng. Ở dự án này ta chọn tổ chức xác thực SSL Let's Encrypt để cài đặt lên máy chủ.

Nguyên tắc chính của chứng chỉ SSL:

### - Mã hóa

Mã hóa là quá trình xáo trộn thông điệp ban đầu để chỉ người nhận dự kiến mới có thể giải mã. Ví dụ: bạn thay đổi từ *cat* thành *ecv* bằng cách đưa tất cả chữ cái về phía trước hai vị trí theo như trong bảng chữ cái. Người nhận nắm được quy tắc (hoặc khóa) và đảo ngược từng chữ cái lùi về hai vị trí để đọc được đúng từ đó. Mã hóa SSL/TLS xây dựng dựa trên khái niệm này bằng cách mã hóa khóa công khai, sử dụng hai khóa khác nhau để mã hóa và giải mã một thông điệp. PKI cung cấp phương thức để một bên thiết lập nhận dạng của một bên khác bằng cách sử dụng các chứng chỉ nếu cả hai đều tin cậy bên thứ ba - được gọi là cơ quan cấp chứng chỉ. Cơ quan cấp chứng chỉ xác minh chứng chỉ và xác thực cả hai bên trước khi giao tiếp bắt đầu.

### - Khóa công khai

Trình duyệt và máy chủ web giao tiếp bằng cách mã hóa và giải mã thông tin thông qua các cặp khóa riêng và công khai. Khóa công khai là một khóa mã hóa mà máy chủ web cung cấp cho trình duyệt trong chứng chỉ SSL/TLS. Trình duyệt sử dụng khóa này để mã hóa thông tin trước khi gửi đến máy chủ web.

### - Khóa riêng

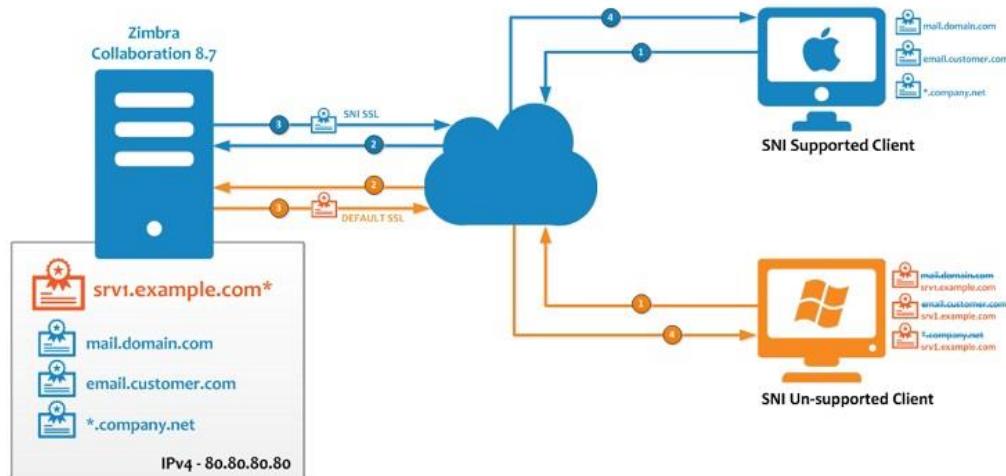
Chỉ máy chủ web mới có khóa riêng. Chỉ có thể dùng khóa công khai để giải mã một tệp dùng khóa riêng để mã hóa và ngược lại. Nếu khóa công khai chỉ có thể giải mã tệp đã được khóa riêng mã hóa, khả năng giải mã được tệp đó sẽ đảm bảo rằng bên gửi và bên nhận dự kiến là đúng người.

### - Xác thực

Máy chủ sẽ gửi khóa công khai trong chứng chỉ SSL/TLS đến trình duyệt. Trình duyệt này xác minh chứng chỉ từ một bên thứ ba đáng tin cậy. Do đó, trình duyệt này có thể xác minh rằng máy chủ web là đúng máy chủ web dự kiến.

### - Chữ ký số

Chữ ký số là số độc nhất cho từng chứng chỉ SSL/TLS. Bên nhận tạo và so sánh chữ ký số mới với chữ ký ban đầu để đảm bảo rằng các bên ở ngoài không can thiệp đến chứng chỉ khi chứng chỉ đó đi qua mạng.



*Hình 11. Cơ chế hoạt động chứng chỉ SSL:*

**- Bắt tay SSL/TLS:**

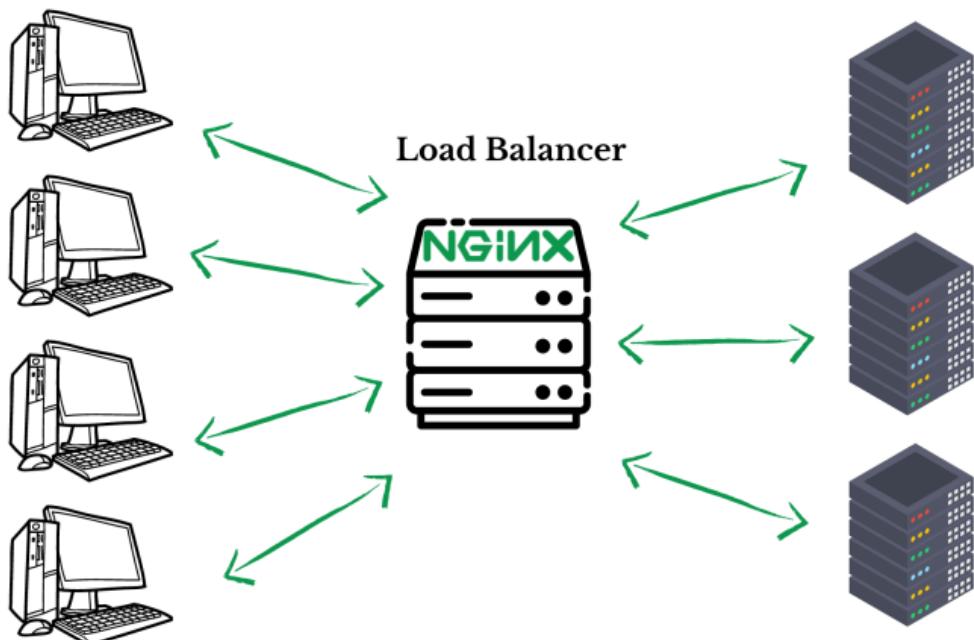
1. Trình duyệt mở một trang web SSL/TLS bảo mật và kết nối với máy chủ web.
2. Trình duyệt cố gắng xác minh tính xác thực của máy chủ web bằng cách yêu cầu thông tin nhận dạng.
3. Máy chủ web gửi chứng chỉ SSL/TLS có chứa một khóa công khai như một bài trả lời.
4. Trình duyệt xác minh chứng chỉ SSL/TLS, đảm bảo rằng chứng chỉ đó là hợp lệ và khớp với tên miền trang web. Một khi trình duyệt hài lòng với chứng chỉ SSL/TLS, nó sử dụng khóa công khai để mã hóa và gửi một tin nhắn có chứa một khóa phiên bí mật.
5. Máy chủ web sử dụng khóa riêng của nó để giải mã thư và truy xuất khóa phiên. Sau đó nó sử dụng khóa phiên để mã hóa và gửi một thông báo xác nhận đến trình duyệt.
6. Bây giờ, cả trình duyệt và máy chủ web chuyển sang sử dụng khóa phiên tương tự để trao đổi tin nhắn một cách an toàn.

**- Khóa phiên:**

Một khóa phiên duy trì giao tiếp được mã hóa giữa trình duyệt và máy chủ web sau khi xác thực SSL/TLS ban đầu được hoàn tất. Khóa phiên là một khóa mật mã cho mật mã đối xứng. Mật mã đối xứng sử dụng cùng một khóa cho cả mã hóa và giải mã. Mật mã bất đối xứng chiếm sức mạnh tính toán không lồ. Do đó, máy chủ web chuyển sang mật mã đối xứng đòi hỏi ít tính toán để duy trì kết nối SSL/TLS.

### 1.6. Cân bằng tải (NLB):

Cân bằng tải là phương thức phân phối lưu lượng truy cập mạng đều nhau trên một vùng tài nguyên hỗ trợ ứng dụng. Các ứng dụng hiện đại phải xử lý đồng thời hàng triệu người dùng và trả về chính xác văn bản, video, hình ảnh và dữ liệu khác cho từng người dùng một cách nhanh chóng và đáng tin cậy. Để xử lý lưu lượng truy cập cao như vậy, hầu hết các ứng dụng sở hữu nhiều máy chủ tài nguyên, trong đó dữ liệu được sao chép giữa các máy chủ với nhau. Bộ cân bằng tải là thiết bị nằm giữa người dùng và nhóm máy chủ, đồng thời đóng vai trò là bộ điều giải, đảm bảo rằng tất cả các máy chủ tài nguyên đều được sử dụng như nhau.



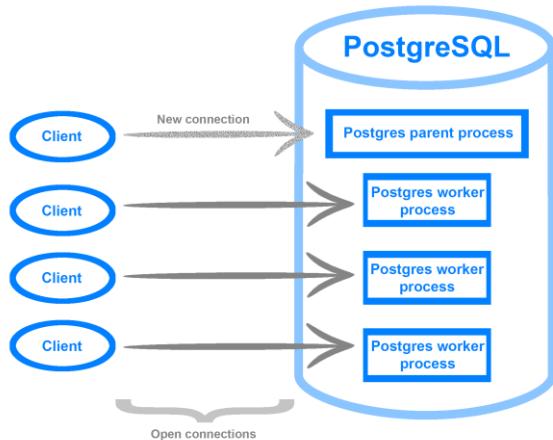
Hình 12. Cơ chế hoạt động của cân bằng tải

Các công ty thường để ứng dụng chạy trên nhiều máy chủ. Việc sắp xếp máy chủ như vậy được gọi là cụm máy chủ. Trước tiên, yêu cầu của người dùng gửi tới ứng dụng sẽ đi đến bộ cân bằng tải. Sau đó, bộ cân bằng tải định tuyến từng yêu cầu tới một máy chủ duy nhất, phù hợp nhất trong cụm máy chủ để xử lý yêu cầu.

Cân bằng tải giống như công việc của một quản lý nhà hàng. Hãy xem xét một nhà hàng có năm nhân viên phục vụ bàn. Nếu khách hàng được phép chọn nhân viên phục vụ bàn của họ, một hoặc hai nhân viên có thể bị quá tải công việc, trong khi đó, những người khác lại nhàn rỗi. Để tránh tình huống này, quản lý nhà hàng chỉ định khách hàng cho nhân viên phục vụ bàn cụ thể, phù hợp nhất để phục vụ khách hàng.

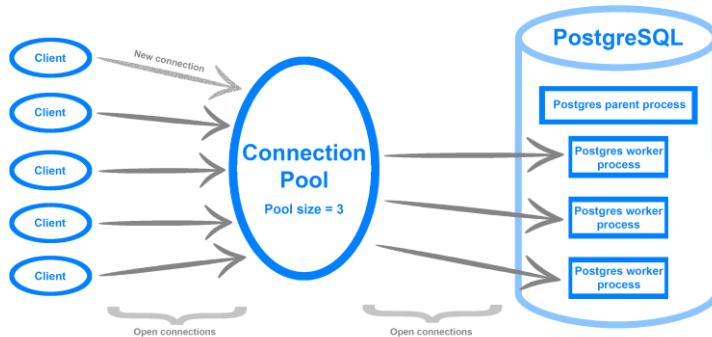
### 1.7. Connection Pool:

Connection pool (vùng kết nối) : là kỹ thuật cho phép tạo và duy trì 1 tập các kết nối dùng chung nhằm tăng hiệu suất cho các ứng dụng bằng cách sử dụng lại các kết nối khi có yêu cầu thay vì việc tạo kết nối mới.



**Without Connection Pool**

*Hình 13. Connection Pool - 1*



**With Connection Pool**

*Hình 14. Connection Pool - 2*

**Cơ chế hoạt động của nó như sau:** khi một connection (một kết nối) được tạo, nó sẽ được đưa vào pool và sử dụng lại cho các yêu cầu kết nối tiếp theo và chỉ bị đóng khi hết thời gian timeout.

Ví dụ, max pool size = 3 (số lượng tối đa connection trong pool là 3).

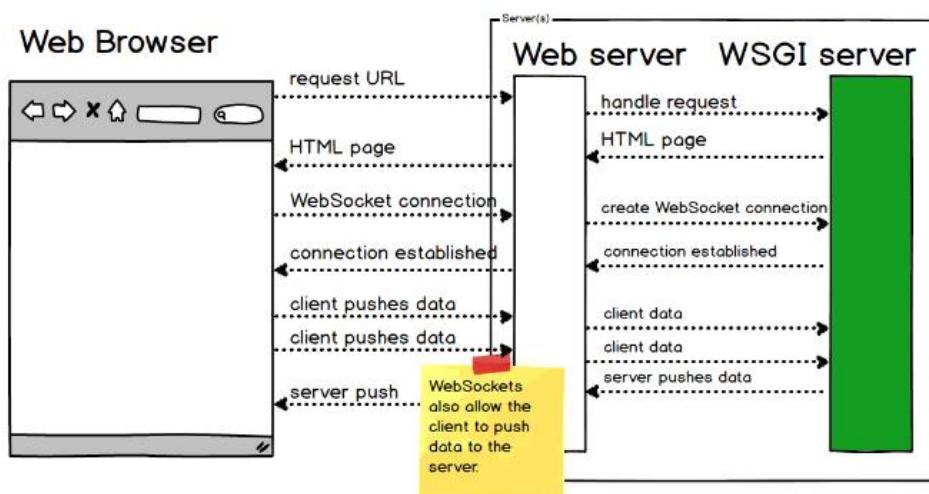
Bây giờ user kết nối tới database (truy vấn database), hệ thống sẽ kiểm tra trong connection pool có kết nối nào đang rảnh không?

- Trường hợp chưa có kết nối nào trong connection pool hoặc tất cả các kết nối đều bận (đang được sử dụng bởi user khác) và số lượng connection trong connection pool < 3 thì sẽ tạo một connection mới tới database để kết nối tới database đồng thời kết nối đó sẽ được đưa vào connection pool.
- Trường hợp tất cả các kết nối đang bận và số lượng connection trong connection pool = 3 thì người dùng phải đợi cho các user dùng xong để được dùng.

Sau khi một kết nối được tạo và sử dụng xong nó sẽ không đóng lại mà sẽ duy trì trong connection pool để dùng lại cho lần sau và chỉ thực sự bị đóng khi hết thời gian timeout.

## 1.8. WebSocket:

### WebSockets



Hình 15. Cơ chế hoạt động của WebSocket

#### - Khái niệm:

WebSocket là một giao thức truyền tải dữ liệu hai chiều (full-duplex) trên web, cho phép truyền tải dữ liệu trong thời gian thực giữa trình duyệt web và máy chủ. WebSocket cho phép một kết nối duy trì giữa máy khách và máy chủ, vì vậy các thông tin có thể được gửi đi và nhận lại một cách hiệu quả và nhanh chóng mà không cần phải thiết lập kết nối mới mỗi khi truyền tải thông tin.

#### - Các sự kiện chính trong WebSocket:

WebSocket cho phép các ứng dụng web tạo kết nối thời gian thực, có khả năng truyền dữ liệu hai chiều liên tục giữa máy chủ và trình duyệt. Các sự kiện chính của WebSocket bao gồm:

- Mở kết nối (Open): Khi một kết nối WebSocket được thiết lập giữa trình duyệt và máy chủ, sự kiện "open" được kích hoạt.

- Nhận dữ liệu (Message): Khi trình duyệt hoặc máy chủ nhận được dữ liệu mới, sự kiện "message" được kích hoạt.
- Gửi dữ liệu (Send): Khi trình duyệt hoặc máy chủ gửi dữ liệu, sự kiện "send" được kích hoạt.
- Đóng kết nối (Close): Khi kết nối WebSocket bị đóng, sự kiện "close" được kích hoạt.
- Lỗi (Error): Nếu có lỗi trong quá trình thiết lập hoặc sử dụng kết nối WebSocket, sự kiện "error" được kích hoạt.

Các sự kiện này đều cung cấp các thông tin cần thiết cho ứng dụng để xử lý các tình huống khác nhau khi sử dụng WebSocket.

So sánh WebSocket với Socket.io:

Phương diện	Websocket	Socket.io
<b>Giao thức/thư viện</b>	Thư viện được thiết kế để tăng cường chức năng của Websocket trên web (Cổng 80, 443)	Giao thức được thiết lập qua kết nối TCP
<b>Giao tiếp</b>	Cho phép giao tiếp dựa trên sự kiện giữa các trình duyệt, ứng dụng...	Tạo điều kiện cho giao tiếp duplex hoàn toàn trên TCP
<b>Dịch vụ hỗ trợ</b>	Hỗ trợ kết nối với proxy và cân bằng tải	Thiếu sự hỗ trợ vốn có của proxy và bộ cân bằng tải
<b>Broadcasting</b>	Hỗ trợ broadcasting tự động, có thể dễ dàng chia	Phải tự thiết lập tạo môi trường broadcasting
<b>Tùy chọn dự phòng</b>	Có khả năng chuyển đổi sang các phương thức kết nối thay thế như polling hay các kỹ thuật khác để vẫn giữ cho việc giao tiếp được thiết lập và duy trì khi Websocket không khả dụng	Không cung cấp tùy chọn dự phòng tích hợp nếu có vấn đề xảy ra

- **WebSocket với Django channel:**

- *Channels và consumers:*

**Kênh** (channel) là khái niệm cốt lõi của Channel Django. Kênh là một hàng đợi được sắp xếp có thời hạn sử dụng tin nhắn, hỗ trợ ngữ nghĩa gửi tin nhắn nhiều nhất một lần. Một kênh giống như một cái óng - nhà sản xuất gửi tin nhắn đến óng này từ một đầu và nó đến được người nghe ở đầu kia

**Consumer** là một thành phần quan trọng để xử lý các sự kiện thời gian thực, đặc biệt là trong ngữ cảnh của WebSockets. Mỗi kết nối WebSocket của client tương ứng với một consumer. Consumer trong Django Channels thường được triển khai dưới dạng lớp Python, và nó kế thừa từ các lớp cung cấp bởi Channels.

- *Channel Layer:*

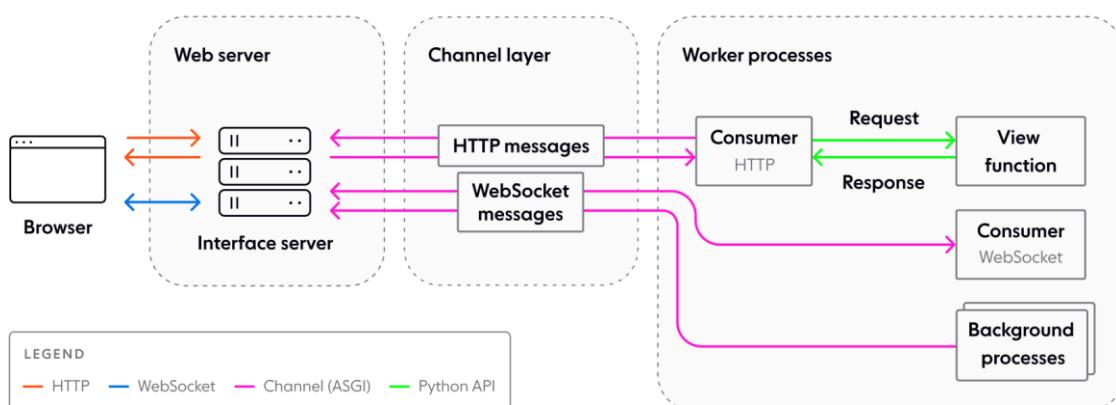
Các lớp kênh cho phép bạn giao tiếp giữa các phiên bản khác nhau của ứng dụng. Chúng là một phần hữu ích trong việc tạo ứng dụng thời gian thực phân tán nếu bạn không muốn phải chuyển tất cả tin nhắn hoặc sự kiện của mình qua cơ sở dữ liệu

- *Groups:*

Kênh Django cung cấp cơ chế phát sóng, rất hữu ích khi bạn muốn gửi cùng một tin nhắn tới nhiều người tiêu dùng. Ví dụ: hãy nghĩ đến các trường hợp sử dụng như phòng trò chuyện, nơi cần gửi tin nhắn cho tất cả người tham gia; hoặc đồng thời phát trực tuyến cập nhật điểm số trực tiếp cho N người dùng.

Cơ chế phát sóng tích hợp của Django Channel được gọi là **nhóm (Groups)**. Cơ chế này cho phép bạn thêm/xóa các kênh khỏi các nhóm được đặt tên và gửi tin nhắn đến các nhóm này (và ngầm định đến tất cả các kênh là một phần của một nhóm).

Kiến trúc Channel Django:



Hình 16. Kiến trúc Channel Django

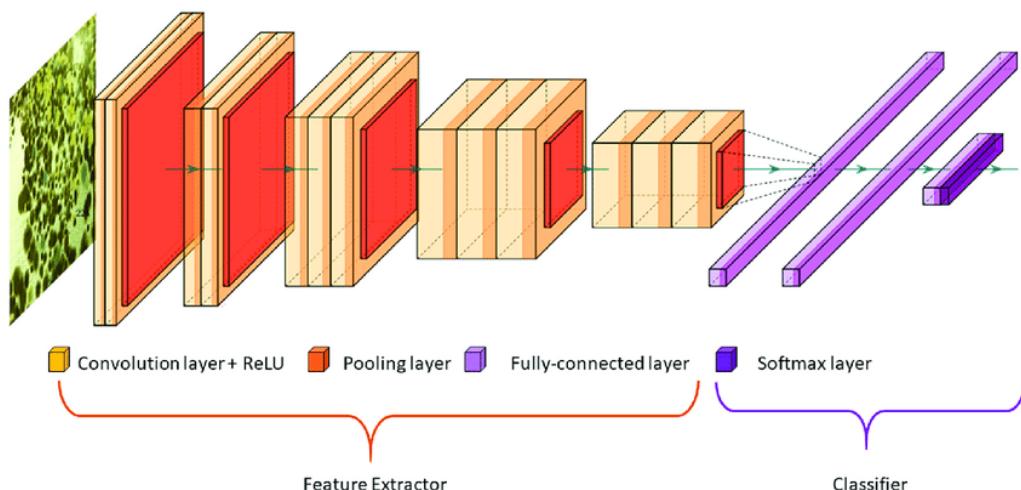
Nhà sản xuất (trong ví dụ của chúng em là khách hàng trình duyệt) gửi tin nhắn đến các channel để consumer nghe các channel đó nhận. Khi máy khách trình duyệt bắt đầu kết nối, nó sẽ kết nối với máy chủ giao diện (máy chủ ASGI, chẳng hạn như Daphne). Sau đó, kết nối được chuyển tiếp đến ứng dụng bộ định tuyến (một phần của Channel layer) để kiểm tra xem đó là kết nối HTTP hay kết nối WebSocket. Channel layer - sử dụng Redis làm cửa hàng hỗ trợ - sau đó gửi tin nhắn đến người consumer thích hợp (consumer WebSocket hoặc HTTP). Tất nhiên, các ứng dụng consumer có thể tuân theo cùng một quy trình (ngược lại) để gửi tin nhắn trở lại ứng dụng client trình duyệt.

### 1.9. Nhận diện khuôn mặt:

Trình phân tích khuôn mặt là phần mềm xác định hoặc nhận dạng danh tính của một người qua các điểm sinh trắc và khuôn mặt của họ. Công nghệ này hoạt động bằng cách xác định và đo lường các đặc điểm khuôn mặt trong hình ảnh. Công nghệ nhận dạng khuôn mặt có thể xác định khuôn mặt người trong hình ảnh hoặc video, xác định xem khuôn mặt xuất hiện trong hai hình ảnh có phải là cùng một người không hoặc tìm kiếm khuôn mặt trong một bộ sưu tập đồ sộ các hình ảnh hiện có.

#### 1.9.1. Model VGG16:

Trong các ứng dụng online cần sử dụng khuôn mặt để cấp quyền truy cập, cần phải có một model đáp ứng việc trích xuất đặc trưng có kết quả chính xác cao, nhanh trong một môi trường truy cập lớn, VGG-16 ra đời để đáp ứng được điều đó.



*Hình 17. Mạng tích chập VGG-16*

**VGG16** là mạng convolutional neural network được đề xuất bởi K. Simonyan and A. Zisserman, University of Oxford. Model sau khi train bởi mạng VGG16 đạt độ chính xác 92.7% top-5 test trong dữ liệu [ImageNet](#) gồm 14 triệu hình ảnh thuộc 1000 lớp khác nhau

#### 1.9.2. Thuật toán Kmean:

Với một lượng đầu vào request ảnh lớn nhận dạng khuôn mặt ở môi trường online, ta không thể dùng model để liên tục nhận diện hoặc nếu có một khuôn mặt mới ta không thể train lại model, sẽ gây đứng server hoặc kết quả trả về quá chậm, vì thế ta dùng K-mean để nhận diện khuôn mặt.



*Hình 18. Phân cụm K cluster của K-mean*

K-means là một thuật toán phân cụm đơn giản thuộc loại học không giám sát(tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm k-means là phân chia 1 bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k. Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng 1 cụm thì phải có cùng 1 số tính chất nhất định. Tức là giữa các điểm trong cùng 1 cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong 1 cụm đó sẽ là các điểm dữ liệu gần nhau.

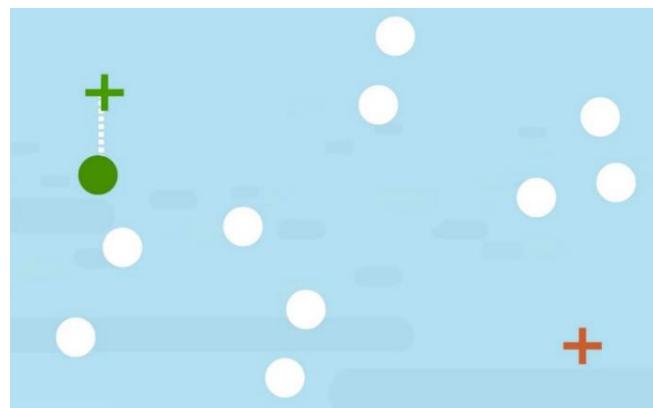
*Ý tưởng của thuật toán k-means:*

1. Khởi tạo K điểm dữ liệu trong bộ dữ liệu và tạm thời coi nó là tâm của các cụm dữ liệu của chúng ta.

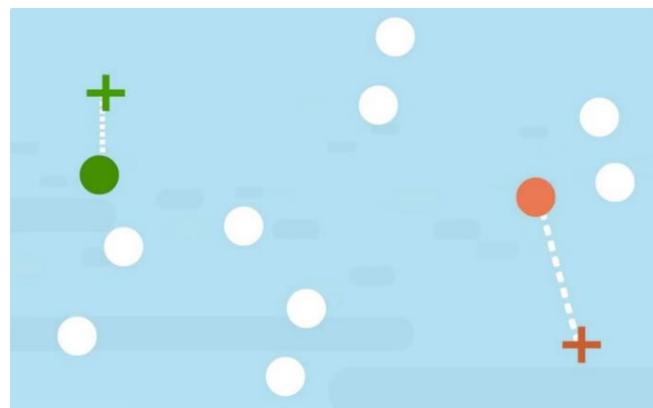


*Hình 19. Thuật toán K-mean - 1*

2. Với mỗi điểm dữ liệu trong bộ dữ liệu, tâm cụm của nó sẽ được xác định là 1 trong K tâm cụm gần nó nhất.

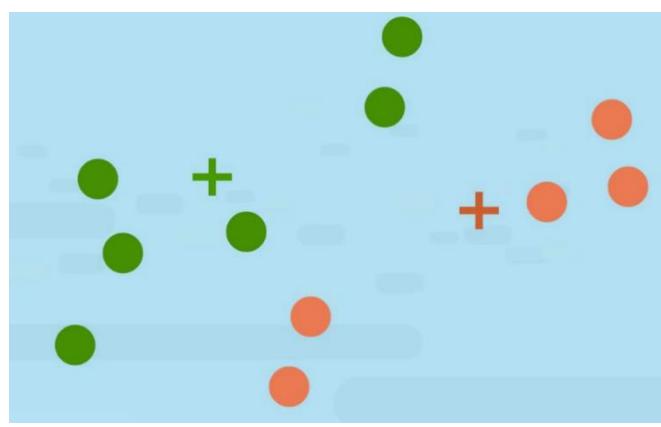


Hình 20. Thuật toán K-mean - 2



Hình 21. Thuật toán K-mean - 3

3. Sau khi tất cả các điểm dữ liệu đã có tâm, tính toán lại vị trí của tâm cụm để đảm bảo tâm của cụm nằm ở chính giữa cụm.



Hình 22. Thuật toán K-mean - 4



*Hình 23. Thuật toán K-mean - 5*

4. Bước 2 và bước 3 sẽ được lặp đi lặp lại cho tới khi vị trí của tâm cụm không thay đổi hoặc tâm của tất cả các điểm dữ liệu không thay đổi

## CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

### 2.1. Yêu cầu hệ thống:

#### 2.1.1. Yêu cầu phần cứng:

- Thiết bị: Raspberry Pi 4 Model B 8G

#### 2.1.2. Yêu cầu phần mềm:

a. Frontend

- Ngôn ngữ lập trình: Javascript
- Thư viện: Reactjs

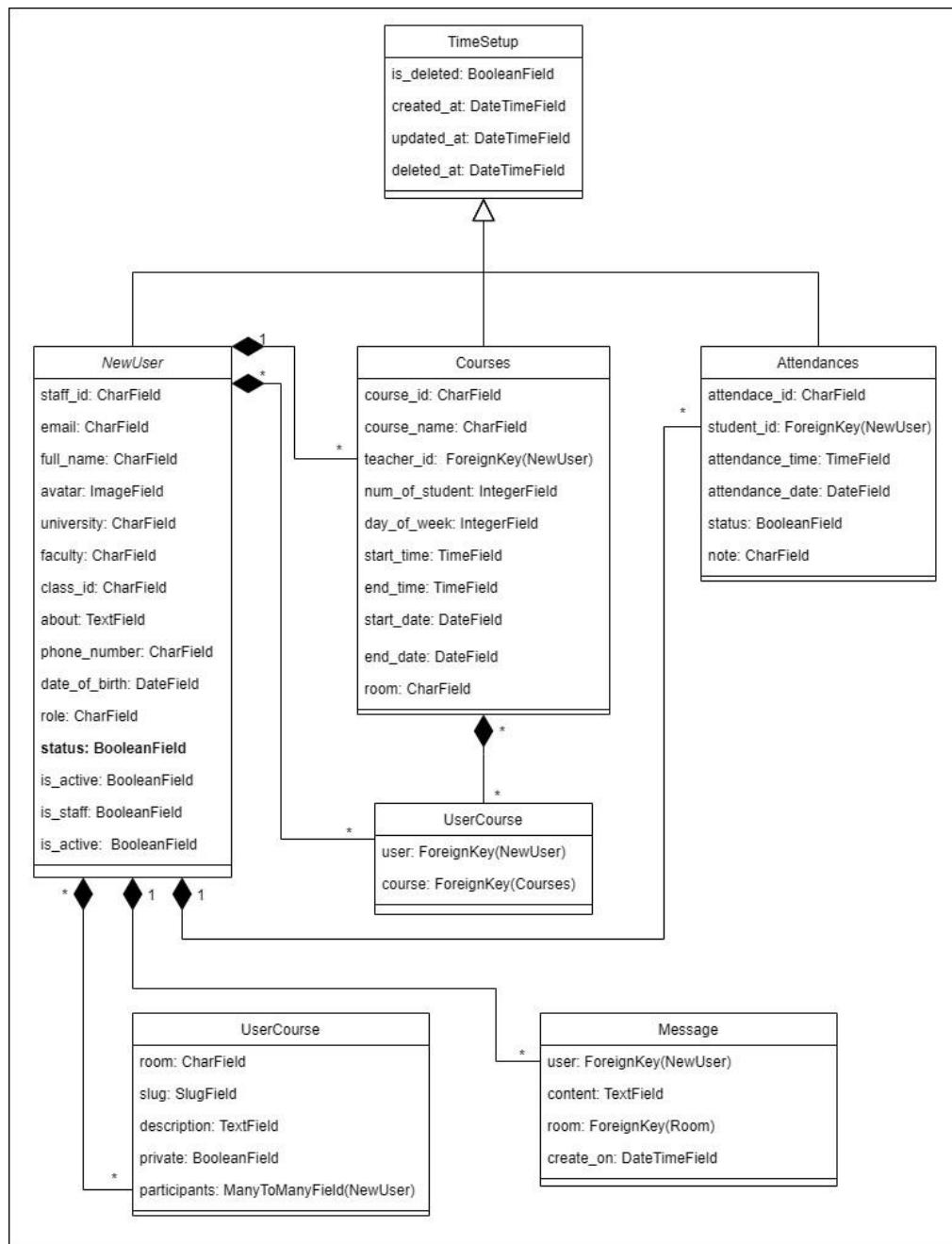
b. Backend:

- Ngôn ngữ lập trình: Python
- Framework: Flask, Django

## 2.2. Thiết kế hệ thống:

### 2.2.1. Thiết kế lớp (Model) và cơ sở dữ liệu (Database):

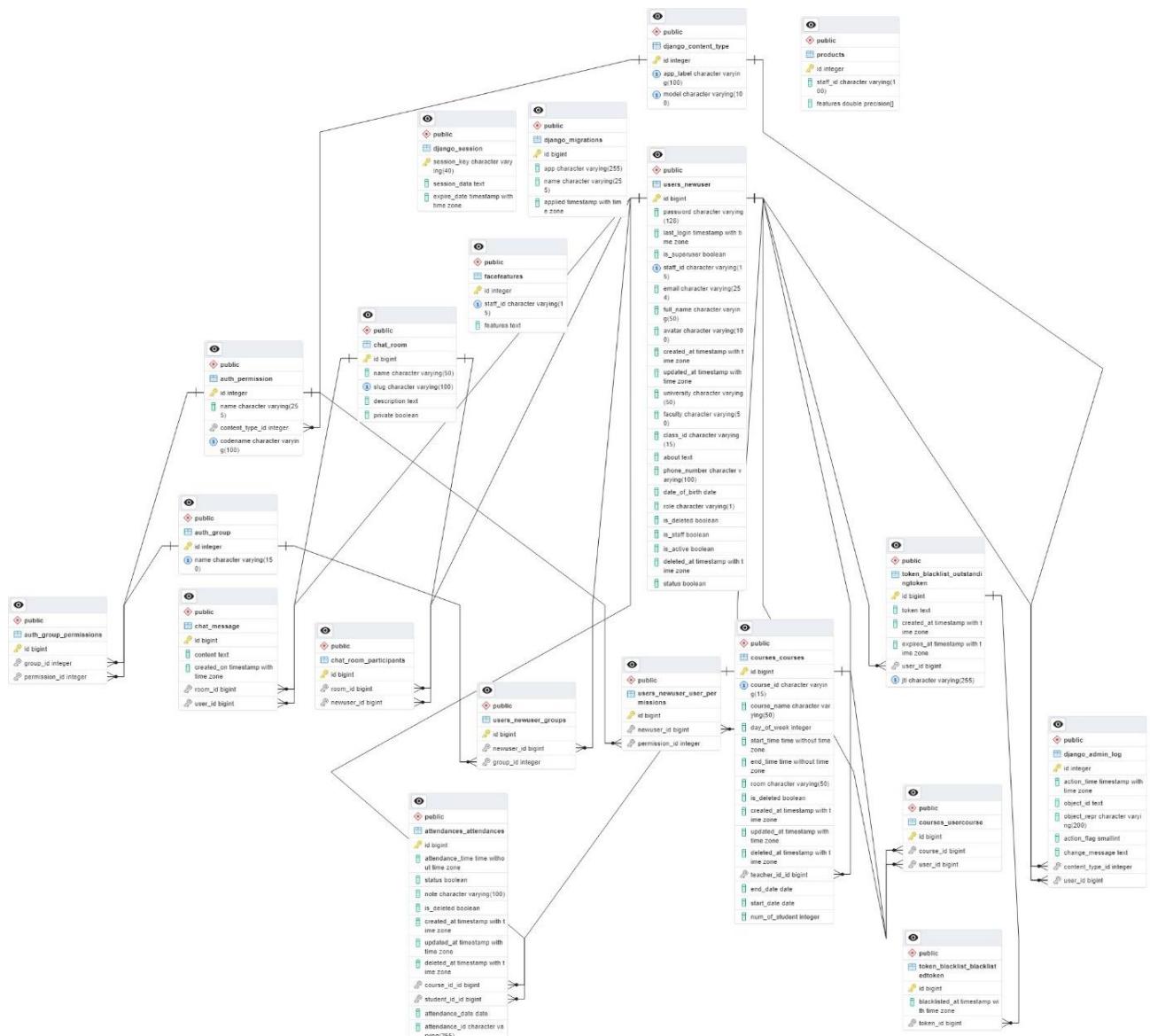
a. Sơ đồ lớp (Class Diagram):



Hình 24. Sơ đồ class

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

b. Sơ đồ cơ sở dữ liệu (Database Diagram):



**Hình 25.** Sơ đồ database

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

## c. Danh sách API:

The screenshot shows the Swagger UI for the PBL04\_DUT\_Student-Checkin-System API. The main title is "PBL04\_DUT\_Student-Checkin-System". Below it, there are links for "Terms of service", "Contact the developer", and "BSD License". A dropdown menu "Schemes" is set to "HTTP". On the right, there is a "Authorize" button. The main content area is titled "attendance" and lists four operations: "attendance\_list" (GET), "attendance\_delete" (DELETE), "attendance\_edit\_update" (PUT), and "attendance\_edit\_partial\_update" (PATCH). Each operation has a corresponding URL path below it.

Hình 26. Danh sách API - 1

This screenshot shows a more comprehensive view of the API endpoints across several categories. The categories listed are "auth", "chat", "COURSE", and "user". Each category contains multiple operations, each with a method (POST, GET, PUT, PATCH, DELETE) and a URL path. To the right of each URL path, there is a small dropdown menu showing the operation name and a lock icon. The "auth" category includes "auth\_login\_create", "auth\_logout\_blacklist\_create", and "auth\_refresh\_create". The "chat" category includes "chat\_list", "chat\_create\_create", and "chat\_read". The "COURSE" category includes "course\_list", "course\_create", "course\_students\_read", "course\_read", "course\_update", "course\_partial\_update", and "course\_delete". The "user" category includes "user\_list", "user\_create", "user\_confirm-active-account\_create", "user\_confirm-reset-password\_create", "user\_send-reset-password\_create", "user\_update-all-user\_create", "user\_upload-txt\_create", "user\_read", "user\_update", "user\_partial\_update", and "user\_delete".

Hình 27. Danh sách API - 2

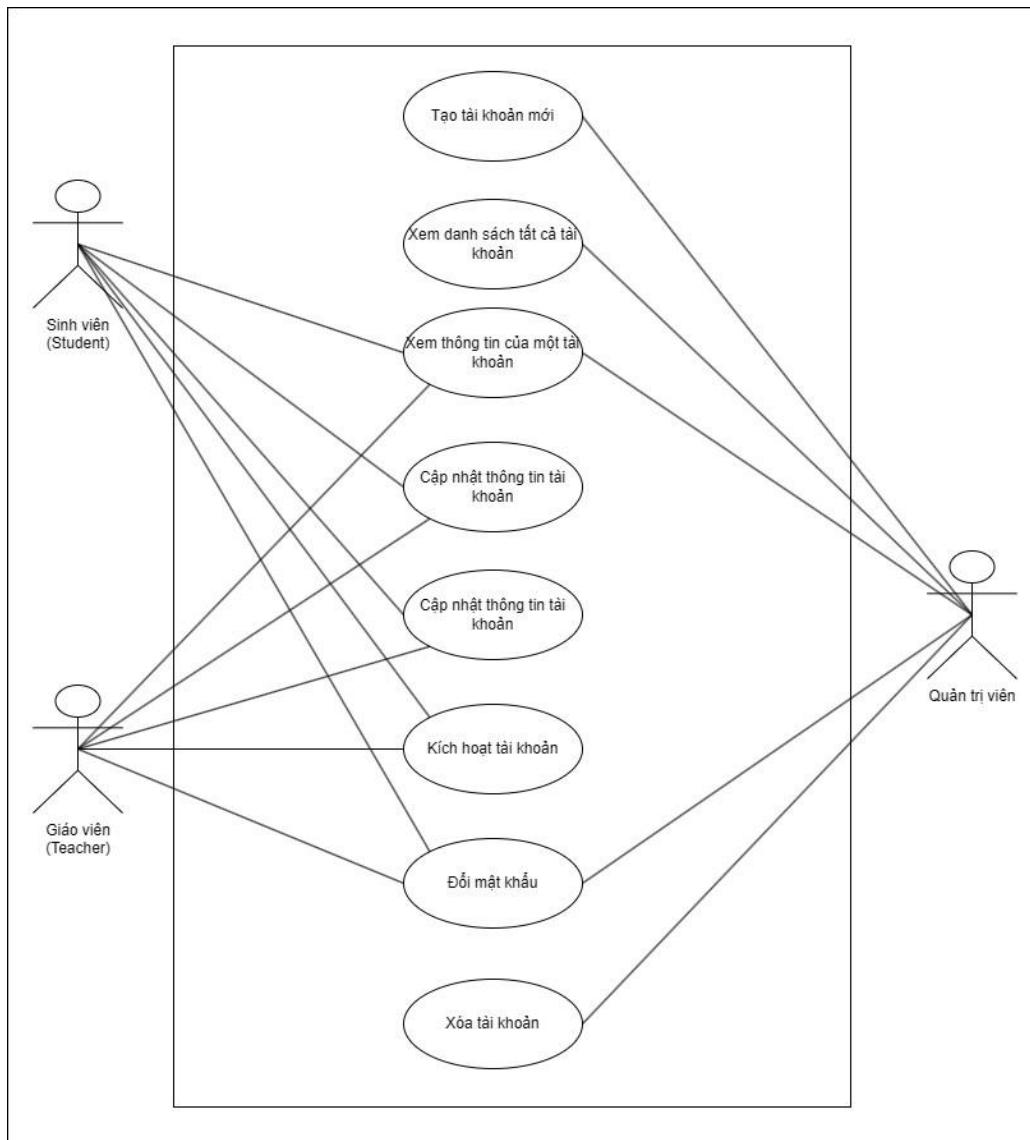
This screenshot focuses on the "user" category of the API. It lists several operations related to user management: "user\_list", "user\_create", "user\_confirm-active-account\_create", "user\_confirm-reset-password\_create", "user\_send-reset-password\_create", "user\_update-all-user\_create", "user\_upload-txt\_create", "user\_read", "user\_update", "user\_partial\_update", and "user\_delete". Each operation is shown with its corresponding URL path and a small dropdown menu indicating the specific operation name and security level.

Hình 28. Danh sách API - 3

### 2.2.2. Phân tích các chức năng của hệ thống:

#### a. Tài khoản và người dùng:

Các chức năng liên qua đến tài khoản và thông tin cá nhân của người dùng.

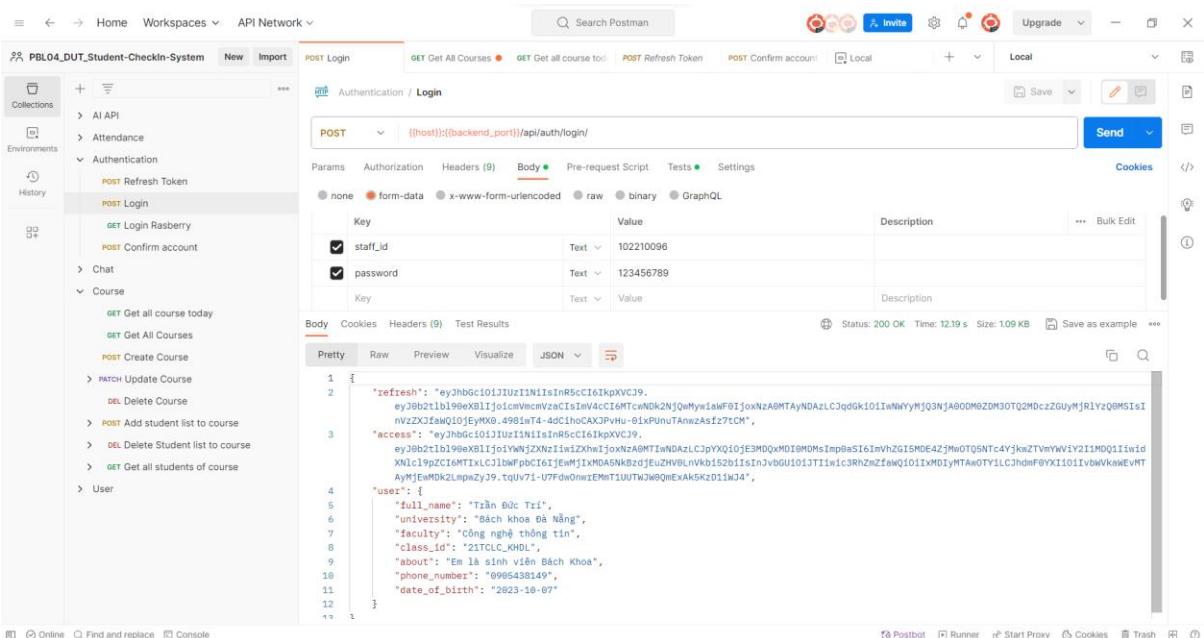


Hình 29. Sơ đồ usecase – Tài khoản và người dùng

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

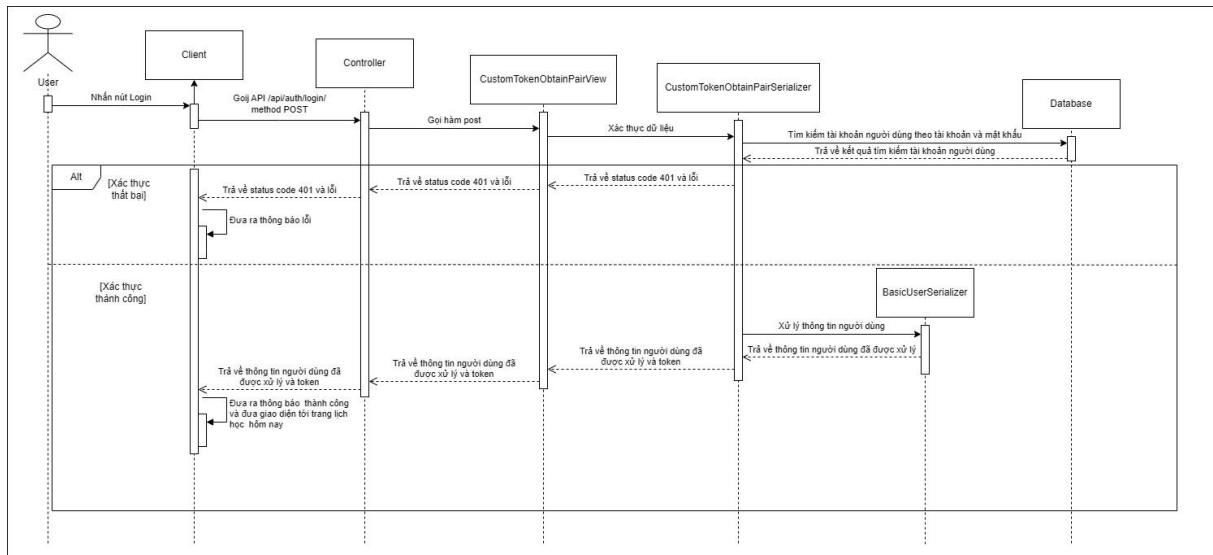
## i. Đăng nhập:

- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Thực hiện đăng nhập để thực hiện các chức năng khác và xem thông tin của bản thân.
- Mô tả API:
  - API: /api/auth/login/
  - Method: GET
  - Đầu vào:
    - staff\_id: là mã số sinh viên, mã giáo viên hoặc mã quản trị viên, và được sử dụng như là tài khoản đăng nhập vào website
    - password: là mật khẩu để đăng nhập vào tài khoản.
  - Đầu ra:
    - refresh: là refresh token, được sử dụng để làm mới token.
    - access: là access token, được dùng để xác thực người dùng.
    - user: thông tin người dùng trả về



Hình 30. Postman – Đăng nhập

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

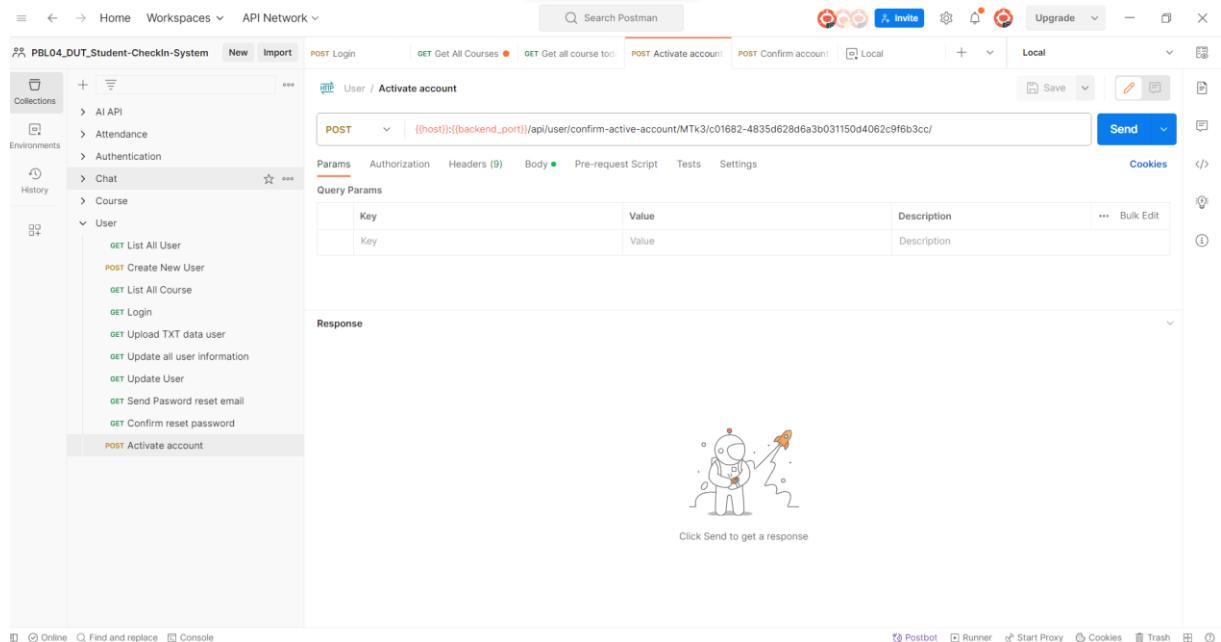


Hình 31. Sơ đồ tuần tự - Đăng nhập

### ii. Kích hoạt tài khoản:

- Người dùng: Sinh viên
- Mô tả: Kích hoạt tài khoản đăng nhập và tải lên ít nhất 10 ảnh chụp khuôn mặt của người dùng.
- Mô tả API:
  - API: /api/user/confirm-active-account/<str:uidb64>/<str:token>/
  - Method: POST
  - Đầu vào:
    - uidb64: id của tài khoản sinh viên
    - token: được sử dụng để xác thực tài khoản

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



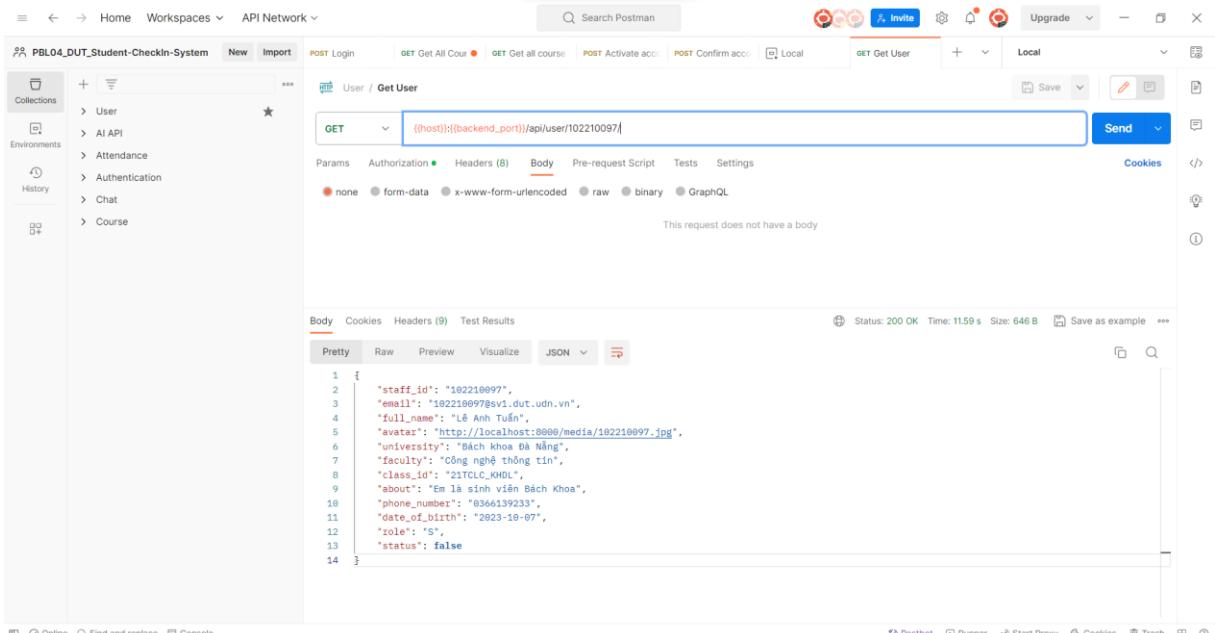
**Hình 32.** Postman – Kích hoạt tài khoản

### iii. Trang cá nhân:

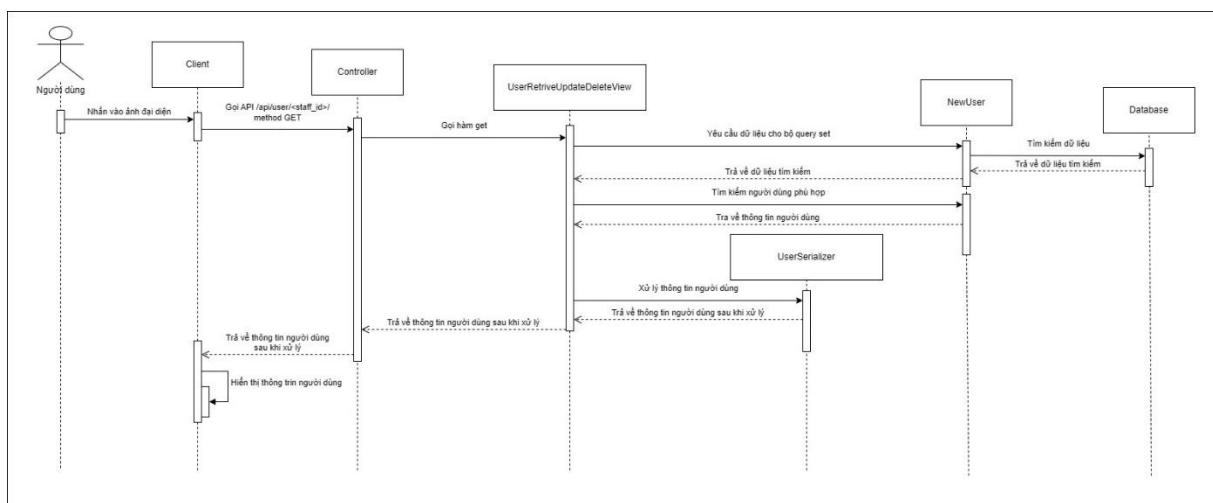
- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Xem thông tin cá nhân và lịch học của một người dùng.
- Mô tả API:

- API: /api/user/<str:staff\_id>/
- Method: GET
- Đầu vào:
  - staff\_id: Mã số sinh viên, mã giáo viên hoặc mã quản trị viên.
- Output: thông tin cá nhân của người dùng đó.

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 33. Postman – Lấy thông tin người dùng



Hình 34. Sơ đồ tuần tự - Lấy thông tin người dùng

#### iv. Tìm kiếm người dùng:

- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Thực hiện tìm kiếm tài khoản người dùng theo họ và tên của người dùng.

#### - Mô tả API:

- API: /api/user/?full\_name=<full\_name>
- Method: GET
- Đầu vào:

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

- full\_name: là từ khóa tìm kiếm, truyền vào họ tên hoặc một phần của họ tên người dùng cần tìm kiếm
- Đầu ra: Danh sách người dùng có tên phù hợp với từ khóa tìm kiếm.

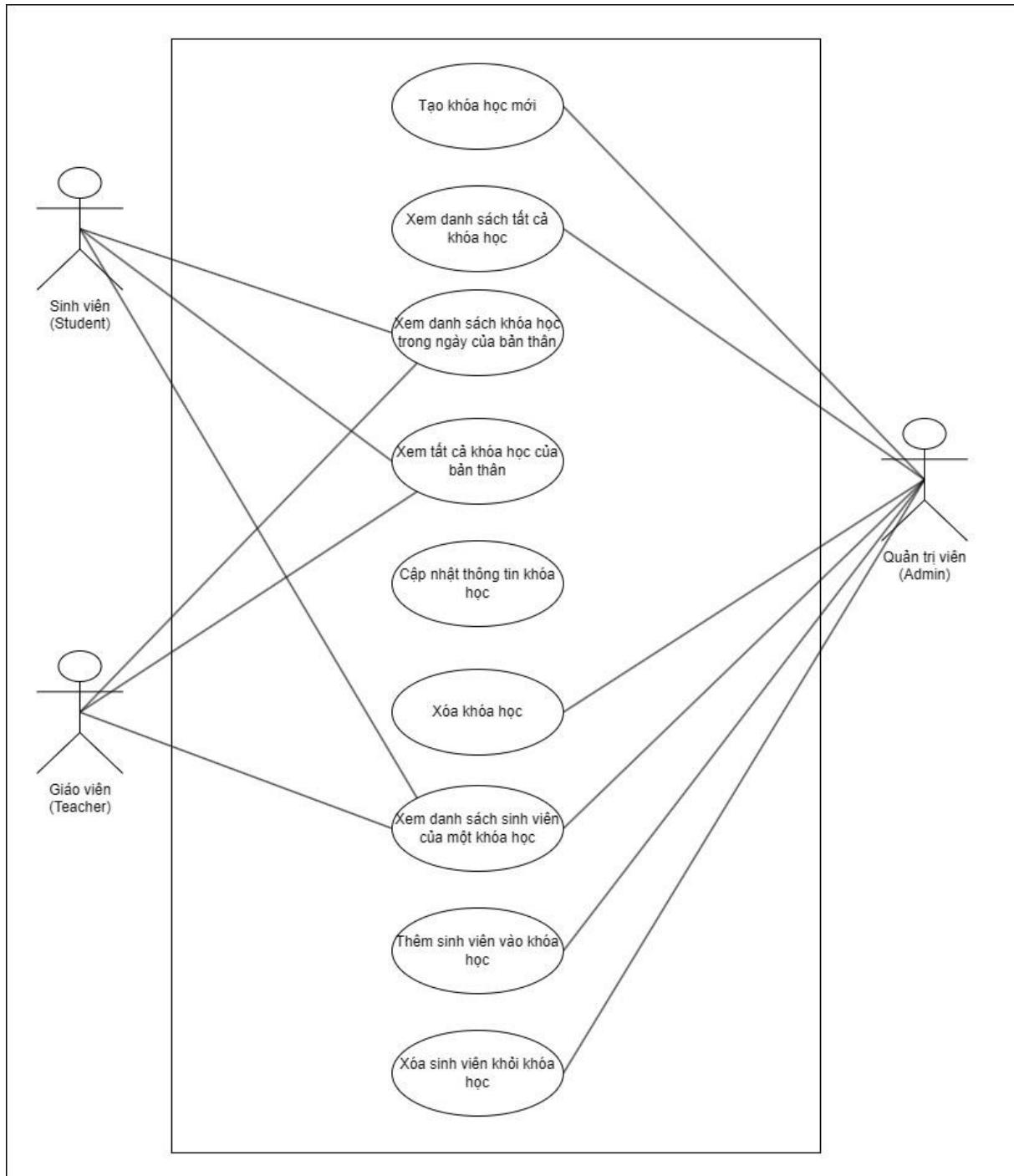
The screenshot shows the Postman interface with the following details:

- Collection:** PBL04\_DUT\_Student-CheckIn-System
- Request:** GET /user / List All User
- Params:** full\_name: Lê Anh
- Body:** (Pretty) JSON response showing a user object with fields like staff\_id, email, full\_name, etc.
- Status:** 200 OK

Hình 35. Postman – Tìm kiếm người dùng

b. Khóa học:

Các chức năng liên quan đến hiển thị lịch học của người dùng



**Hình 36.** Sơ đồ usecase – Khóa học

i. Xem danh sách các khóa học:

- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Hiển thị danh sách tất cả khóa học.

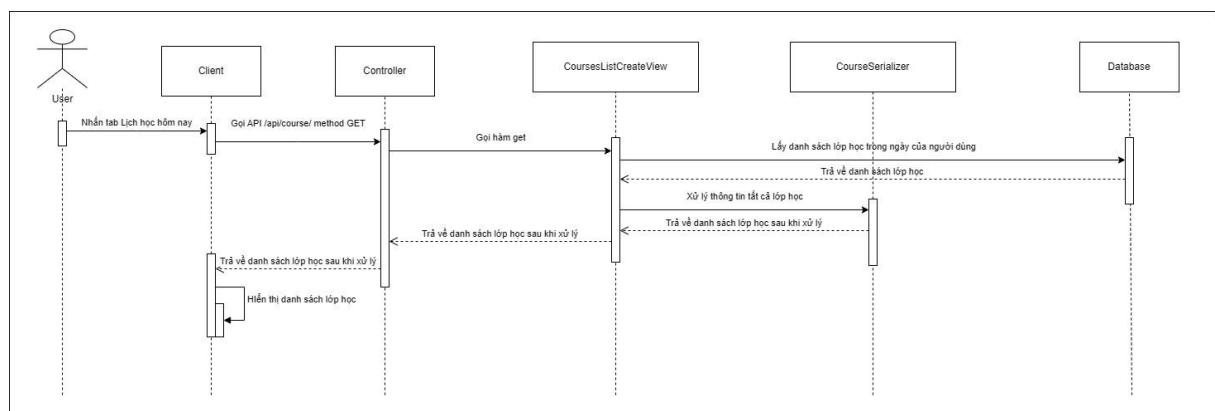
# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

## - Mô tả API:

- API: /api/course/
- Method: GET
- Đầu vào:
  - day\_of\_week (có thể có hoặc không): thứ ngày trong tuần (chủ nhật là 0, thứ hai là 1,...)
  - course\_id (có thể có hoặc không): mã khóa học.
  - staff\_id (có thể có hoặc không): mã số sinh viên, mã giáo viên hoặc mã quản trị viên.
- Đầu ra: Danh sách khóa học.

The screenshot shows the Postman interface with a collection named "PBL04\_DUT\_Student-Checkin-System". A GET request is made to `({host}):({backend_port})/api/course/`. The parameters are set to `day_of_week: 1`, `course_id: CNWEB`, and `staff_id: 102210096`. The response body is a JSON array containing one element, representing a course record with fields like course\_id, course\_name, num\_of\_student, day\_of\_week, start\_time, end\_time, start\_date, end\_date, room, teacher, and full\_name.

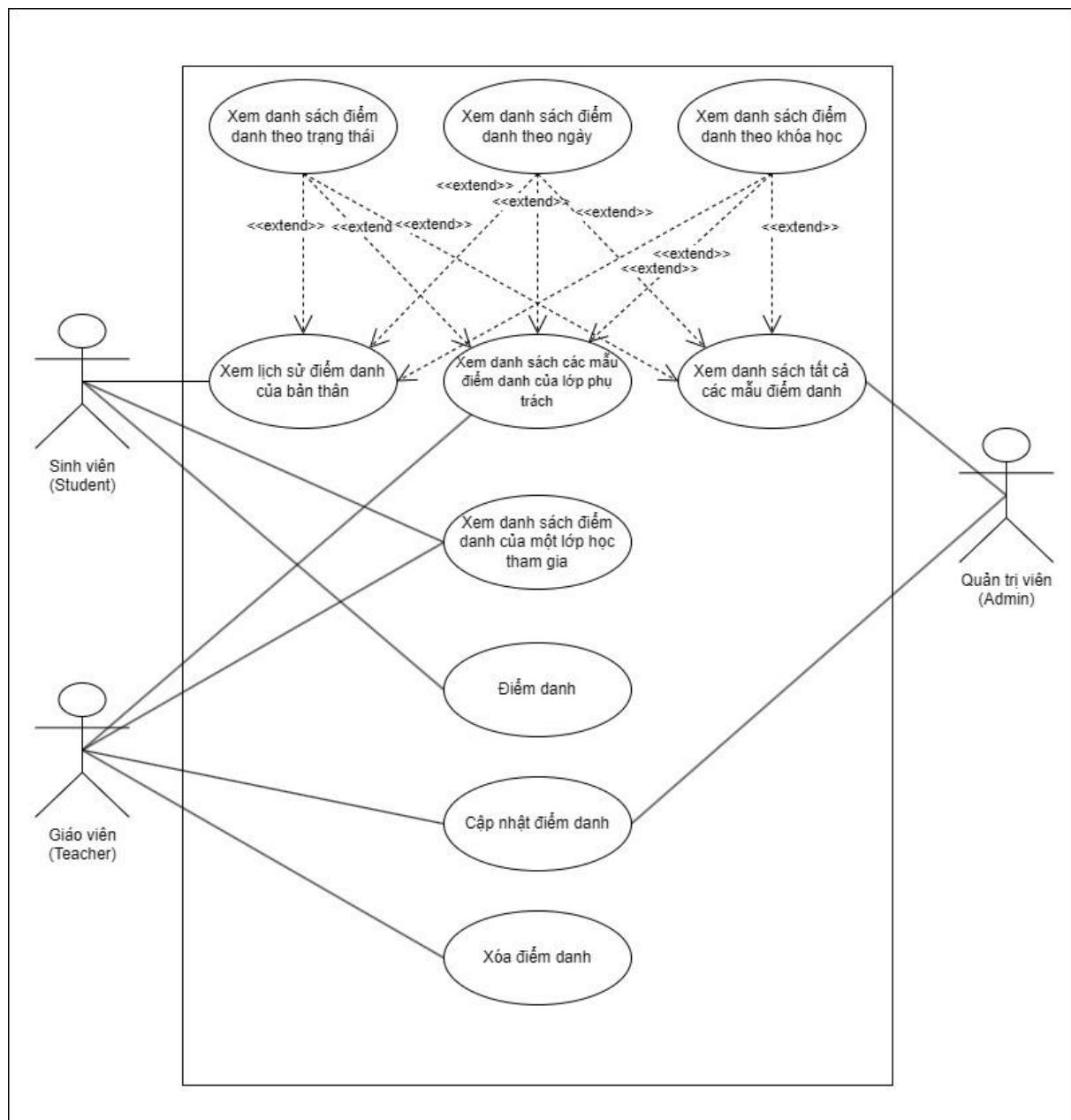
Hình 37. Postman – Danh sách khóa học



Hình 38. Sơ đồ tuần tự - Xem lịch học hôm nay

c. Điểm danh:

Các chức năng thực hiện điểm danh và lịch sử điểm danh

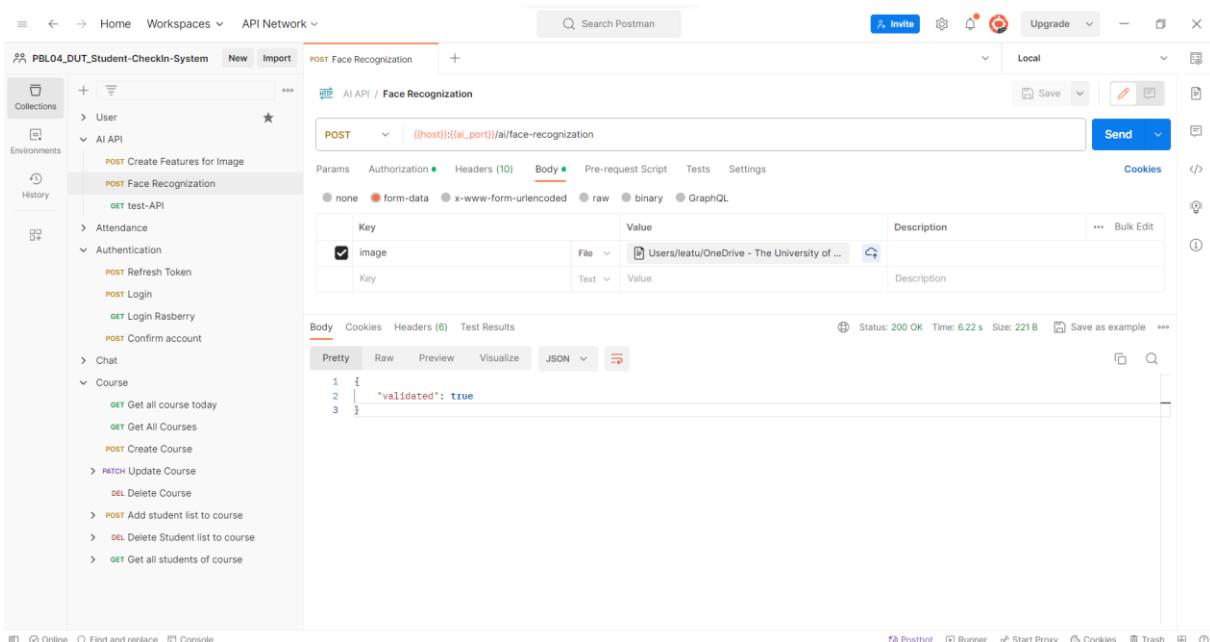


Hình 39. Sơ đồ usecase – Điểm danh

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

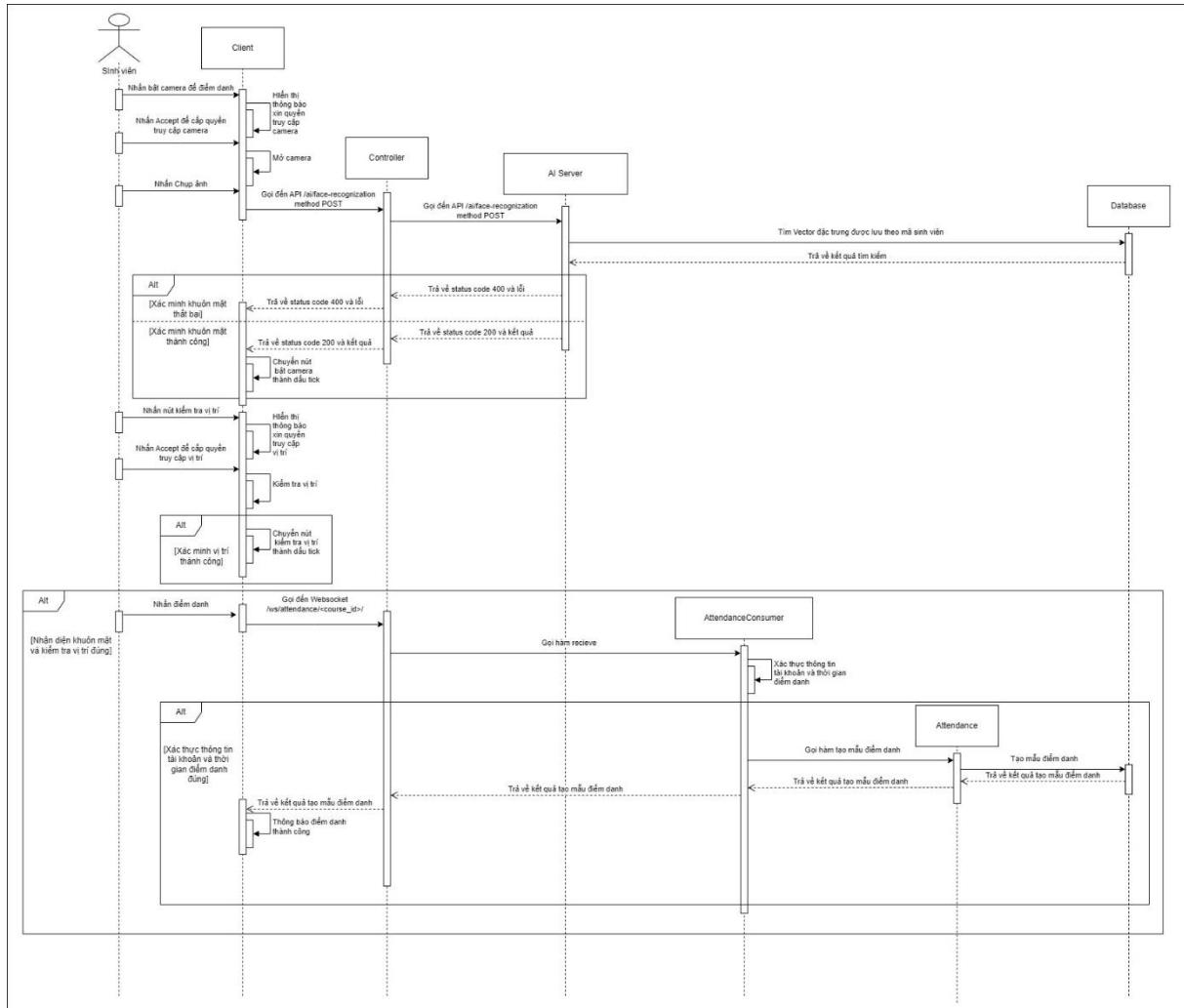
## i. Thực hiện điểm danh:

- Người dùng: Sinh viên
- Mô tả: Thực hiện điểm danh tại lớp học bằng nhận diện khuôn mặt và định vị vị trí của sinh viên điểm danh trong thời gian học của một khóa học đó trong ngày.
- Mô tả API nhận diện khuôn mặt:
  - Đầu vào:
    - image: là ảnh mà người dùng chụp khuôn mặt của mình bằng thiết bị.
  - Output: trả về kết quả đây có phải là chủ sở hữu của tài khoản đã đăng nhập không



Hình 40. Postman – Nhận diện khuôn mặt

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



## ii. Xem lịch sử điểm danh:

- Người dùng: Sinh viên, giáo viên, quản trị viên.

- Mô tả:

- Sinh viên có thể xem lịch sử điểm danh của bản thân.
- Giáo viên có thể xem lịch sử điểm danh của các lớp học mà giáo viên đó dạy.
- Quản trị viên có thể xem toàn bộ lịch sử điểm danh.

- Mô tả API:

- API: /api/attendance

- Method: GET

- Đầu vào:

- course\_id (có thể có hoặc không): mã khóa học
- attendance\_date (có thể có hoặc không): ngày điểm danh
- staff\_id (có thể có hoặc không): mã người dùng

- Đầu ra: Danh sách lịch sử điểm danh.

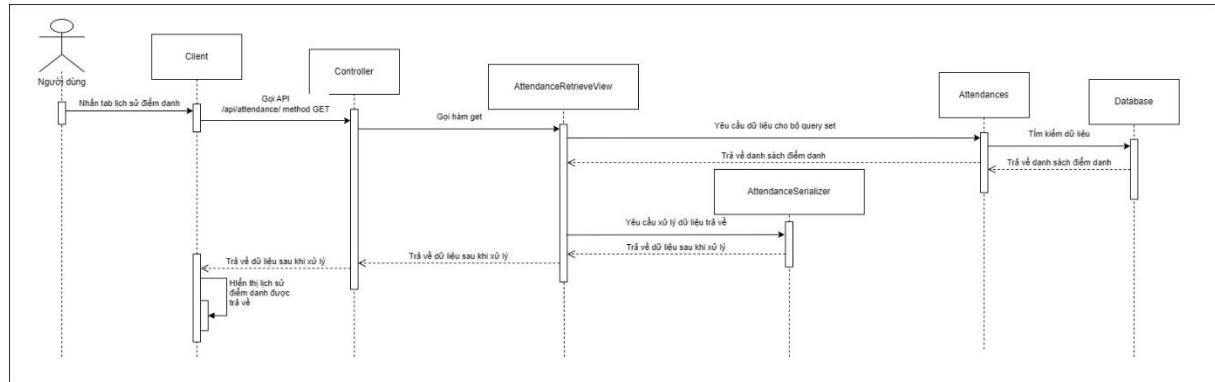
The screenshot shows the Postman application interface. On the left, the 'Collections' sidebar is visible, showing the 'Attendance' collection is currently selected. In the main workspace, a 'GET List all attendances' request is displayed under the 'Attendance' section. The 'Params' tab shows query parameters: course\_id (tc123), attendance\_date (2023-11-01), and staff\_id (102210097). The 'Body' tab shows a JSON response with a count of 5 results. Each result contains an attendance\_id, student (with staff\_id 102210097, email 102210097@sv1.dut.udn.vn, full\_name Lê Anh Tuấn, avatar media/102210097.jpg, and university Bách Khoa Đà Nẵng), and a timestamp (2023-11-01T10:00:00Z).

```

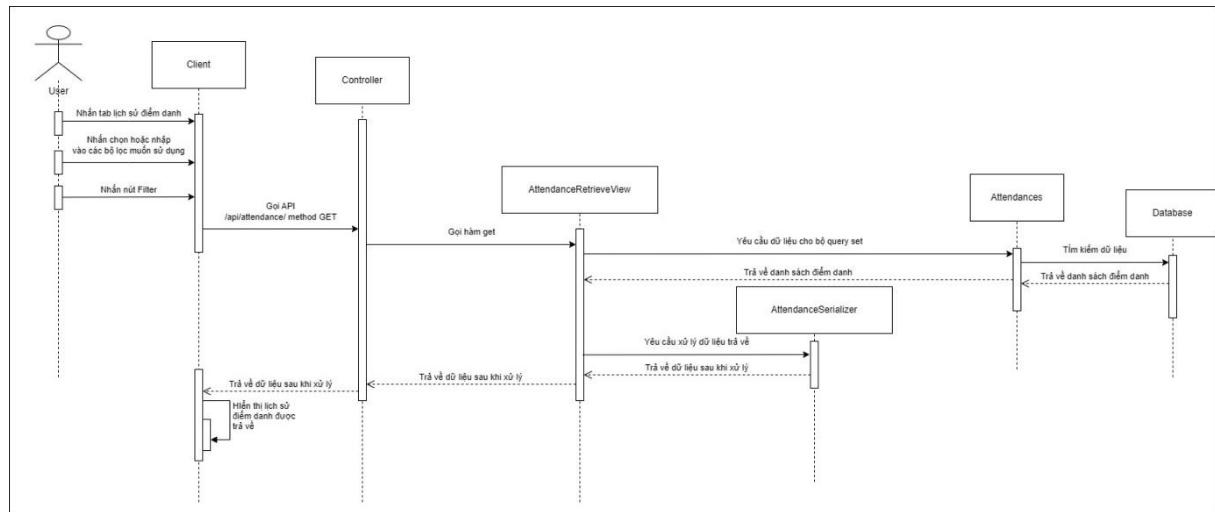
1 {
2   "count": 5,
3   "next": null,
4   "previous": null,
5   "results": [
6     {
7       "attendance_id": "i22-5-2023-12-30",
8       "student": {
9         "staff_id": "102210097",
10        "email": "102210097@sv1.dut.udn.vn",
11        "full_name": "Lê Anh Tuấn",
12        "avatar": "/media/102210097.jpg",
13        "university": "Bách Khoa Đà Nẵng",
14      }
15    }
16  ]
17}
  
```

**Hình 42.** Postman – Danh sách điểm danh

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



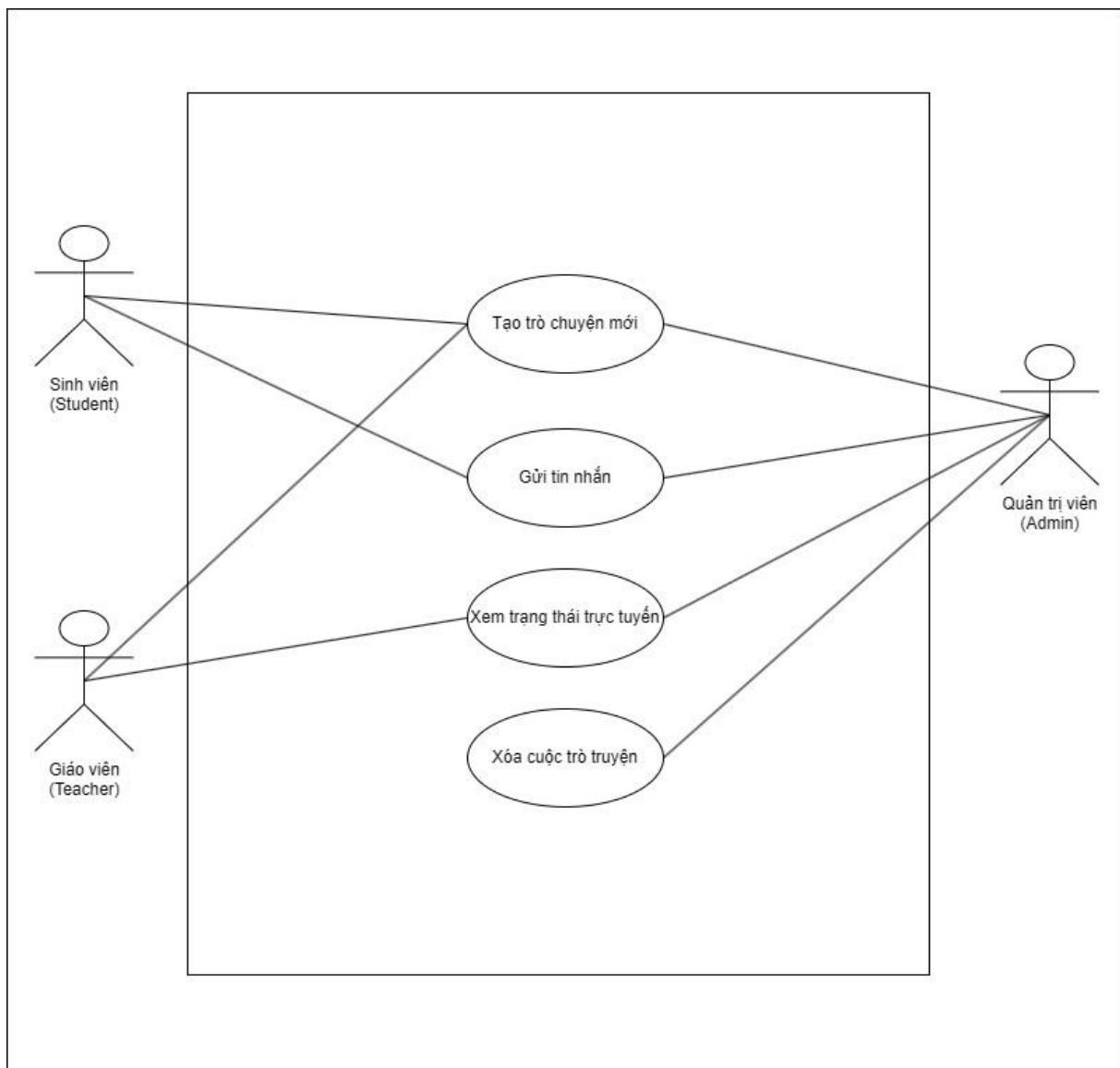
Hình 43. Sơ đồ tuần tự - Lịch sử điểm danh



Hình 44. Sơ đồ tuần tự - Lịch sử điểm danh

d. Trò chuyện:

Bao gồm chức năng trò chuyện và xem trạng thái trực tuyến



**Hình 45.** Sơ đồ usecase – Trò chuyện và Trạng thái trực tuyến

i. Chức năng trò chuyện:

- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Gửi tin nhắn với người dùng khác.
- Mô tả API:
  - Đầu vào:
    - slug: mã phòng chat
  - Output: thông tin phòng chat

ii. Chức năng xem trạng thái trực tuyến của giáo viên:

- Người dùng: Sinh viên, giáo viên, quản trị viên.
- Mô tả: Xem trạng thái trực tuyến của giáo viên.

## CHƯƠNG 3: TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1. Triển khai:

#### 3.1.1. *Triển khai dự án:*

a. Phân công nhiệm vụ:

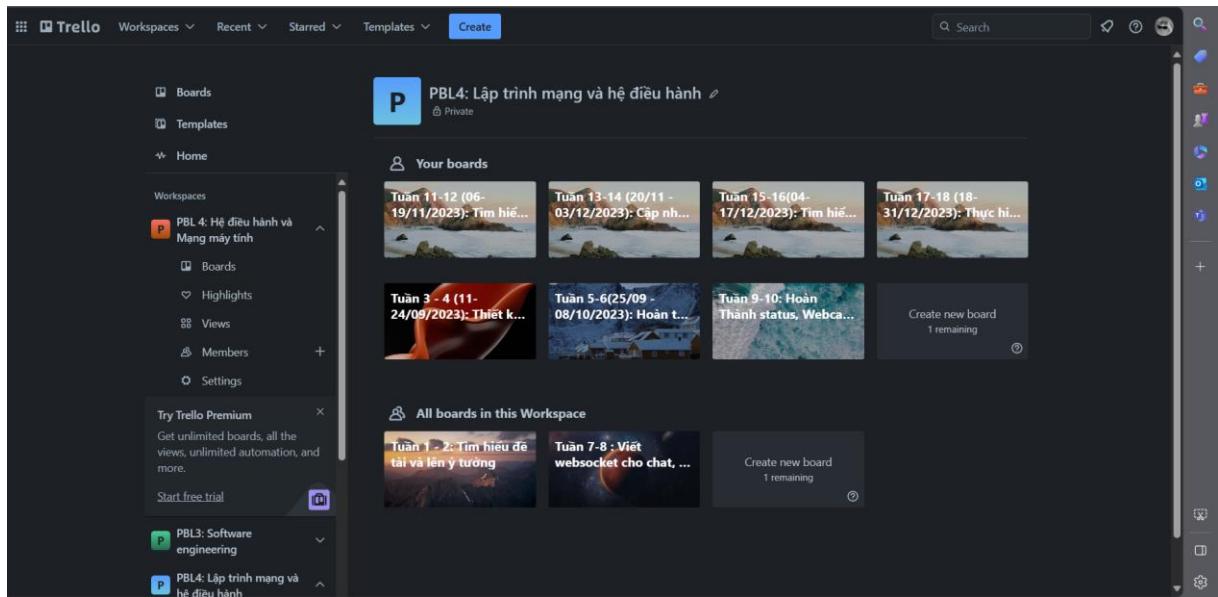
Nhiệm vụ	Người đảm nhận
Tìm kiếm, lựa chọn và phân tích đề tài	Trần Đức Trí, Lê Anh Tuấn, Phạm Nguyễn Anh Phát
Quản lý Git	Trần Đức Trí
Quản lý Trello và Postman	Lê Anh Tuấn
Thiết kế cơ sở dữ liệu, ứng dụng.	Trần Đức Trí, Lê Anh Tuấn, Phạm Nguyễn Anh Phát
Tạo API cho khóa học, điểm danh, thông tin cá nhân	Trần Đức Trí, Lê Anh Tuấn, Phạm Nguyễn Anh Phát
Tạo API cho tài khoản.	Trần Đức Trí
Tạo WebSocket cho trạng thái trực tuyến	Phạm Nguyễn Anh Phát
Tạo WebSocket cho trò chuyện	Trần Đức Trí
Kiểm thử và sửa chữa API	Lê Anh Tuấn
Viết giao diện người dùng	Trần Đức Trí, Phạm Nguyễn Anh Phát
Thiết lập phần cứng	Trần Đức Trí
Chức năng nhận diện khuôn mặt	Trần Đức Trí, Lê Anh Tuấn
Sửa chữa và cập nhật API	Trần Đức Trí, Lê Anh Tuấn
Viết báo cáo	Trần Đức Trí, Lê Anh Tuấn, Phạm Nguyễn Anh Phát

b. Ứng dụng hỗ trợ:

i. Trello:

- Link: <https://trello.com/w/pbl4lpchinmngvahdiuhanh/home>
- Sử dụng ứng dụng Trello làm ứng dụng quản lý công việc.
- Khi có một nhiệm vụ mới sẽ tạo 1 task trên Trello để theo dõi nó.
- Chu kỳ công việc là 2 tuần.

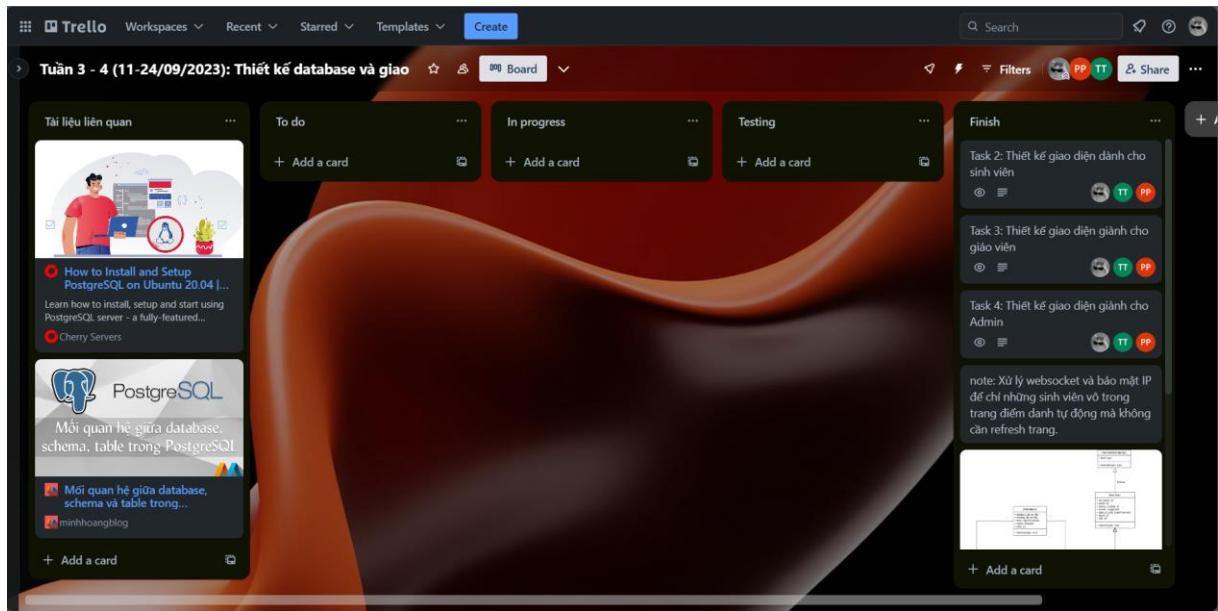
## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 46. Trello – Dashboard

- Mỗi chu kỳ chia ra 5 mục:

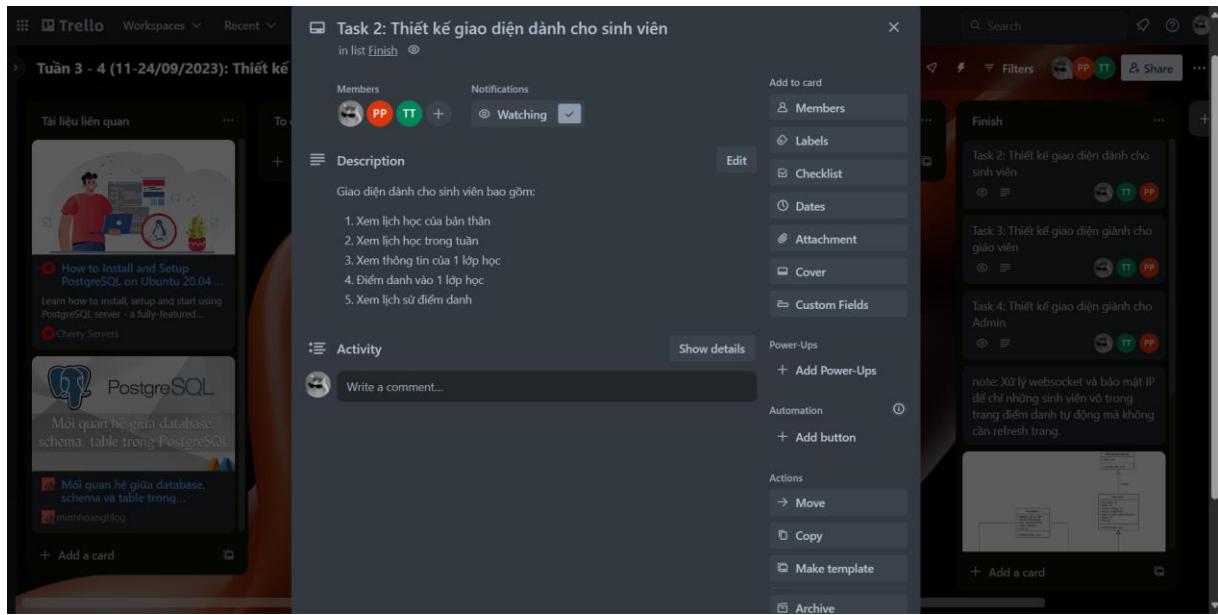
- Tài liệu liên quan
- Danh sách công việc còn lại (To do).
- Danh sách công việc đang làm (In progress).
- Danh sách công việc cần kiểm thử (Testing).
- Danh sách công việc đã hoàn thành (Finish).



Hình 47. Trello – Các mục của một chu kỳ

- Các nhiệm vụ (task) sẽ được mô tả ở phần description và gán cho người liên quan.

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 48. Trello – Task

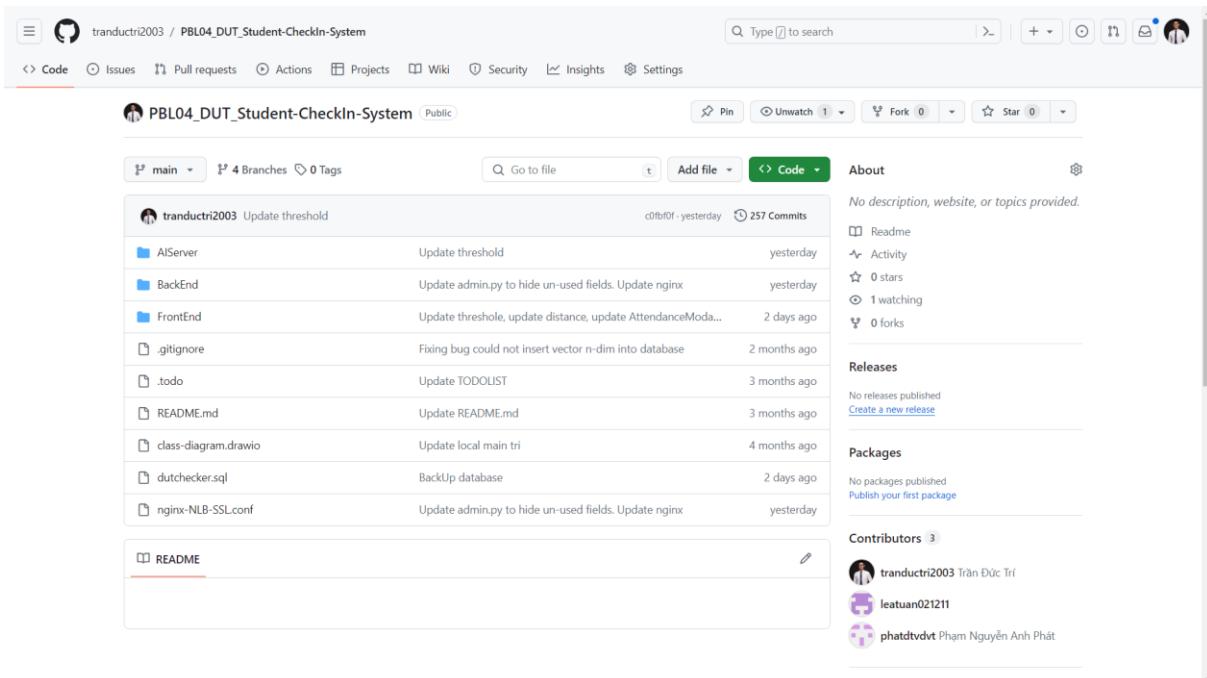
## ii. Github:

- Link:

- Sử dụng Github để lưu trữ code, cập nhật code mới khi có thay đổi, tránh bất đồng về source code.

- Tạo ra 4 nhánh:

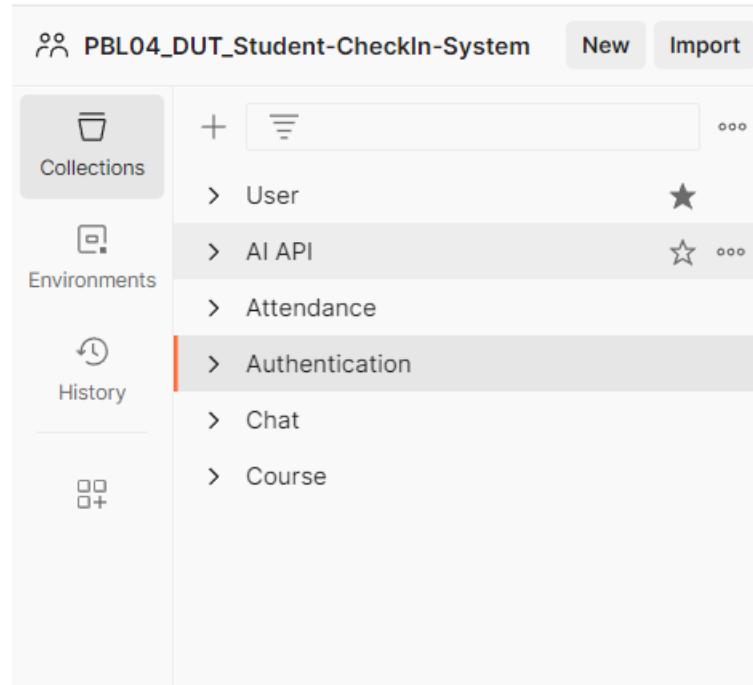
- main (nhánh chính): dùng để cập nhật code mới sau khi được kiểm thử.
- tri, tuan, phat-dev: 3 nhánh này sẽ được sử dụng cho việc viết code mới và kiểm thử code mới tránh xuất hiện lỗi ở nhánh chính.



Hình 49. Trang github của dự án

### iii. Postman:

- Postman được sử dụng để kiểm thử các API.
- Chia ra thành nhiều thư mục dành cho từng chức năng riêng.



*Hình 50. Postman – Thư mục*

PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

- Mỗi thư mục chứa các API tương ứng với chức năng đó.

PBL04\_DUT\_Student-CheckIn-System

New Import

Collections

Environments

History

Course

GET Get all course today

GET Get All Courses

POST Create Course

PATCH Update Course

DEL Delete Course

POST Add student list to course

DEL Delete Student list to course

GET Get all students of course

**Hình 51.** Postman – API trong một thư mục

The screenshot shows the Postman application interface. The left sidebar has tabs for Collections, Environments, and History, with Environments selected. The main area shows a table of local variables under the 'Local' tab. The table has columns for Variable, Type, Initial value, and Current value. Variables listed are host, backend\_port, ai\_port, access\_token, day\_of\_week, and refresh\_token. Each variable has a checkbox next to it. The 'access\_token' and 'refresh\_token' rows show long encoded strings as their values. A button 'Add new variable' is at the bottom. A tooltip at the bottom left says 'Use variables to reuse values and protect sensitive data'. The top right has buttons for Invite, Settings, Help, Upgrade, and a maximize/minimize window control.

Variable	Type	Initial value	Current value
host	default	localhost	localhost
backend_port	default	8000	8000
ai_port	default	5000	5000
access_token	default	eyJhbGciOiJIUzI1NiIsInR5cCl6kpXV... eyJhbGciOiJIUzI1NiIsInR5cCl6kpXVCj9.eyJ0b2tlb90eXBII...	eyJhbGciOiJIUzI1NiIsInR5cCl6kpXV... eyJhbGciOiJIUzI1NiIsInR5cCl6kpXVCj9.eyJ0b2tlb90eXBII...
day_of_week	default	0	0
refresh_token	default		eyJhbGciOiJIUzI1NiIsInR5cCl6kpXV... eyJhbGciOiJIUzI1NiIsInR5cCl6kpXVCj9.eyJ0b2tlb90eXBII...
Add new variable			

**Hình 52** Postman - Thiết lập môi trường kiểm thử ở loca

iv. Quá trình triển khai:

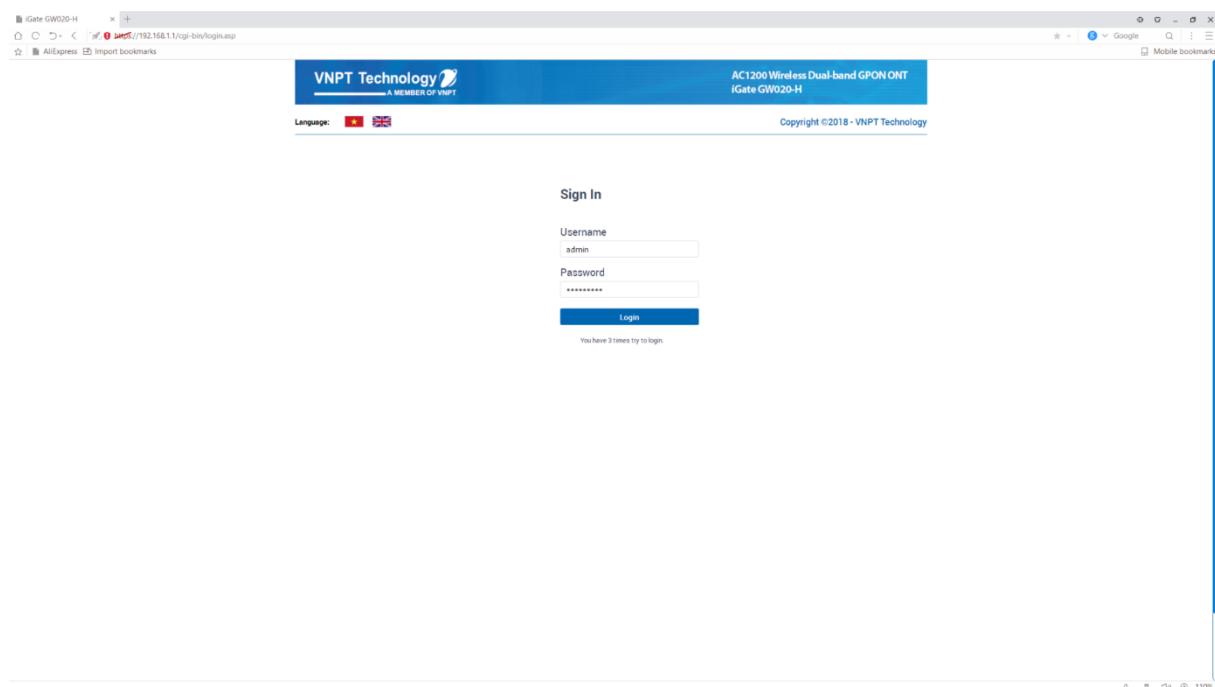
Tuần	Nội dung công việc	Sản phẩm
<b>Tuần 1 - 2 (02/09/2023 - 10/09/2023)</b>	Tìm hiểu ý tưởng	Đề tài dự án và phân tích đề tài
<b>Tuần 3 – 4 (11/09/2023 - 24/09/2023)</b>	Thiết kế database và giao diện	Các bảng cở sở dữ liệu chính và giao diện cơ bản.
<b>Tuần 5 - 6 (25/09/2023 - 08/10/2023)</b>	Xây dựng API Backend, tìm hiểu về React và Websocket	API cơ bản cho tài khoản, người dùng, khóa học, điểm danh; nắm bắt kiến thức về React và Websocket
<b>Tuần 7 – 8 (09/10/2023 - 22/10/2023)</b>	Xây dựng Websocket cho chat và trạng thái trực tuyến	Websocket cho chat và trạng thái trực tuyến
<b>Tuần 9 – 10 (23/10/2023 – 05/11/2023)</b>	Hoàn thành Websocket status, webcam, và API điểm danh	Cập nhật API điểm danh, status, kết nối webcam
<b>Tuần 11 – 12 (06/11/2023 – 19/11/2023)</b>	Tìm hiểu về AI và nhận diện khuôn mặt.	Lựa chọn Flask làm Server AI và nắm bắt kiến thức về các mô hình nhận diện, ứng dụng DDNS
<b>Tuần 13 – 14 (20/11/2023 - 03/12/2023)</b>	Cập nhật AI và tạo giao diện cho thông tin cá nhân của người dùng	Cập nhật mô hình nhận diện khuôn mặt và giao diện
<b>Tuần 15 – 16 (04/12/2023 )</b>	Tìm hiểu cân bằng tải, thực hiện chỉnh sửa và hoàn thiện cho Frontend và Backend	Cân bằng tải, hoàn thiện trang sinh viên và các API
<b>Tuần 17-18 (18/12/2023 - 31/12/2023)</b>	Thực hiện code giao diện cho giáo viên và Admin, Kiểm lại tất cả các tính năng và thực hiện viết báo cáo	Giao diện cơ bản cho trang giáo viên và admin; báo cáo môn học.

### 3.1.2. Triển khai hạ tầng:

#### NAT Port trên Modem iGate GW020-H của nhà mạng VNPT:

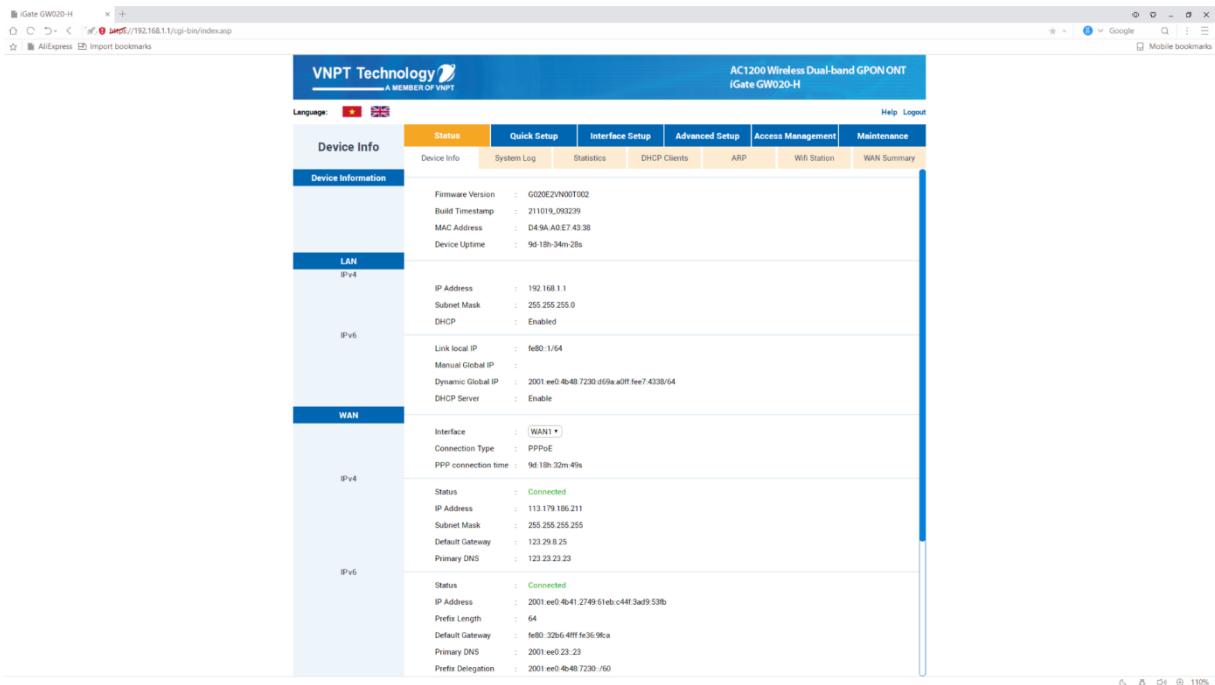
Ở đề tài này, nhóm chúng em dự định sẽ triển khai Web lên không gian mạng thông qua Raspberry Pi 4 trong đó Raspberry Pi 4 sẽ được đặt tại nhà của một thành viên trong nhóm và được cấp nguồn để hoạt động 24/7. Trên thực tế, xuyên suốt quá trình hoàn thiện đồ án, chỉ yêu cầu triển khai Web lên môi trường Internet vào giai đoạn cuối cùng để có thể bảo vệ. Tuy nhiên, để thuận tiện trong quá trình phát triển, nhóm em đã sử dụng Raspberry như một server online để truy vấn database thay vì mỗi thành viên phải chạy localhost với database tại chỉ riêng máy tính cá nhân của họ.

Để giải quyết bài toán trên, điều kiện tiên quyết là phải NAT cổng tại Modem của nhà Trí để cho phép người dùng bên ngoài có thể truy cập vào mạng bên trong nhà Trí bằng cách sử dụng port đó từ bên ngoài thông qua bộ định tuyến (đã mở NAT – Network Address Translation). Lúc này, nhóm đã liên hệ đến nhân viên nhà mạng VNPT để xin mật khẩu trang Home Gateway. Lại nói về trang Home Gateway, hiểu một cách đơn giản, đây chính là địa chỉ Default Gateway, nơi cho phép người dùng truy cập trực tiếp vào hệ thống quản trị của thiết bị, Router hoặc các loại wifi.

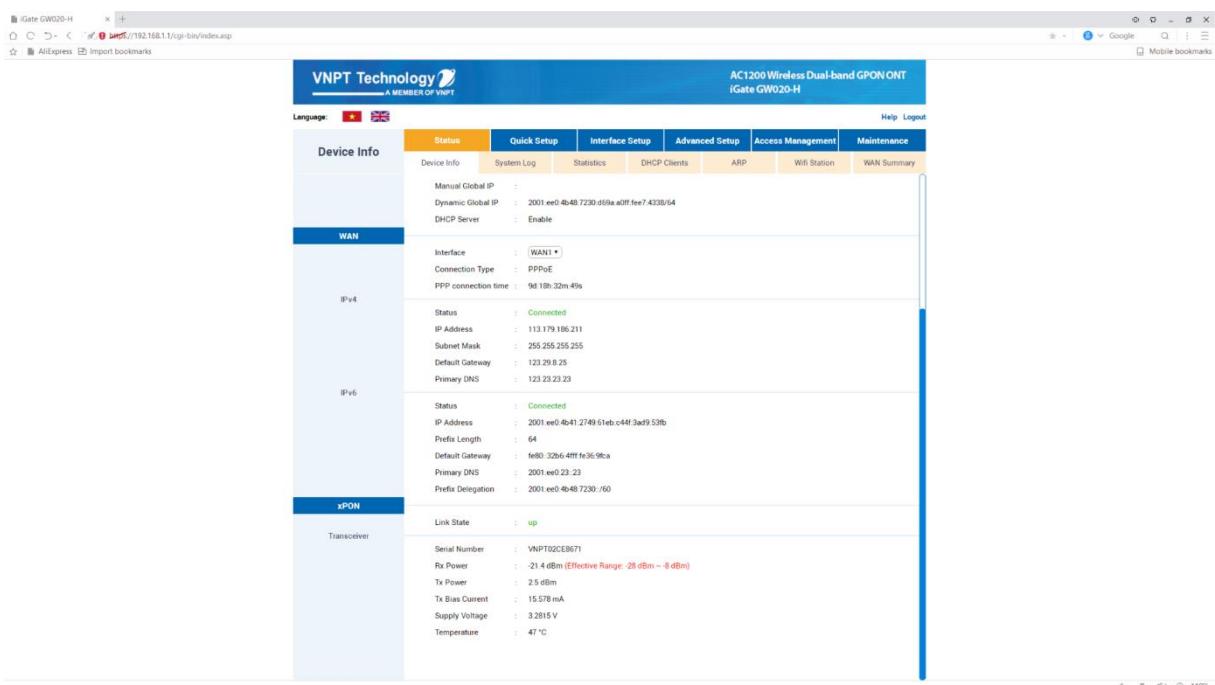


Hình 53. Đăng nhập vào trang quản trị

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 54. Giao diện của trang quản trị Home Gateway - 1



Hình 55. Giao diện của trang quản trị Home Gateway - 1

Đến đây, chúng ta cần xác định được địa chỉ IP trong LAN của Raspberry để có thể tiến hành mở Port trên router cho địa chỉ IP này. Vì trong nội mạng LAN có thể có rất nhiều thiết bị và kết nối ra bên ngoài với cùng địa chỉ IP WAN nên ta sẽ cần mở 1 Port duy nhất cho một thiết bị được xác định bởi 1 địa chỉ IP LAN.

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Ở đây chúng ta đã gán địa chỉ IP nội bộ 192.168.1.200 dựa vào địa chỉ MAC của Raspberry D8:3A:DD:2A:10:01. Địa chỉ IP với Octet 4 có giá trị 200 đảm bảo khi Router gán DHCP cho các thiết bị kết nối với mạng nhà Trí cho dù có nhiều thiết bị đến bao nhiêu cũng sẽ không ảnh hưởng đến IP nội bộ của Raspberry.

#	HostName	IP	MAC Address	Expire Time
1	N/A	192.168.1.3	14:40:67:71:23:0C	0days 22:48:49
2	DUCTNTRAN	192.168.1.6	5C:61:99:20:9F:B3	0days 23:51:26
3	raspberrypi	192.168.1.200	D8:3A:DD:2A:10:01	0days 22:38:43
4	RedmiB-Redmi	192.168.1.10	7C:D6:61:BF:43:F7	0days 21:28:30
5	N/A	192.168.1.11	E6:35:A7:15:ED:F5	0days 23:49:7
6	N/A	192.168.1.12	E6:79:3C:58:3D:40	0days 13:16:48
7	N/A	192.168.1.9	CC:6E:A4:4B:7D:4E	0days 19:18:12
8	N/A	192.168.1.8	BE:39:50:8F:C7:1D	0days 20:24:7
9	N/A	192.168.1.4	36:35:41:1C:C5:71	0days 22:8:14
10	N/A	192.168.1.7	FE:0E:22:D2:20:7A	0days 22:21:29

Hình 56. Gán IP nội bộ LAN 192.168. Raspberry dựa vào MAC

Sau khi thực hiện gán địa chỉ IP nội bộ từ địa chỉ MAC cho Raspberry nêu trên, khi di chuyển sang Tab ARP, ta sẽ thấy có thêm một trường với IP Address 192.168.1.200 và địa chỉ MAC D8:3A:DD:2A:10:01 được thêm vào.

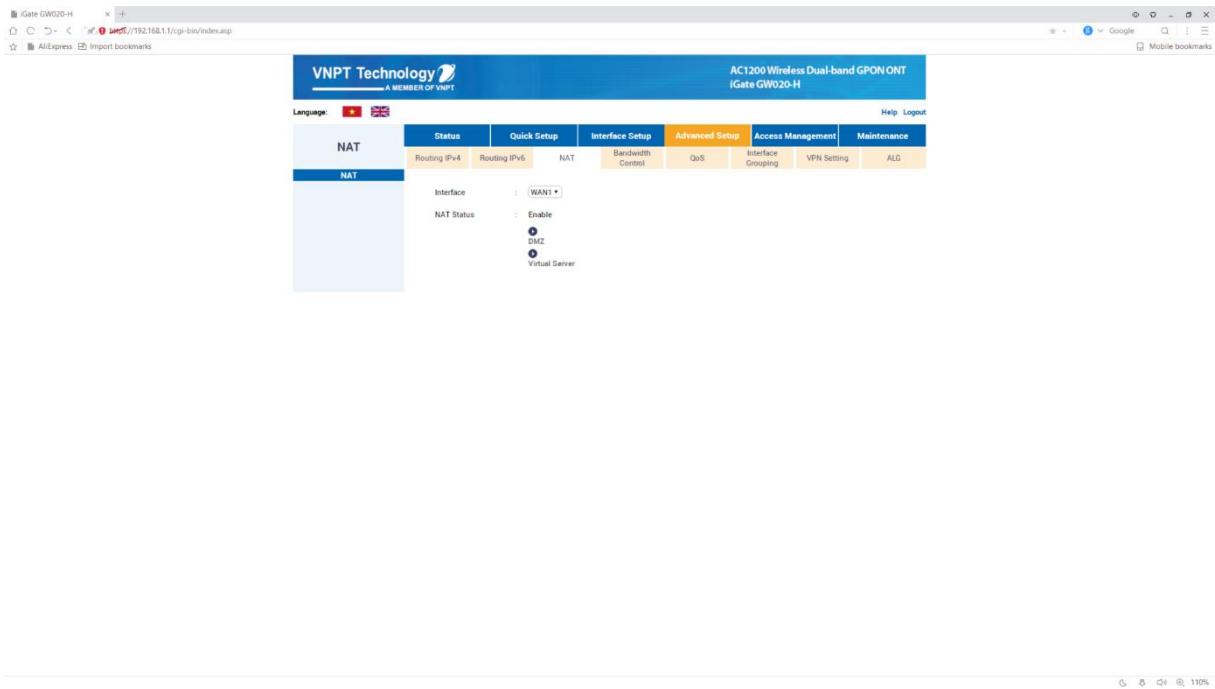
IP address	Flags	HW Address	Device
192.168.1.200	STALE	08:3a:dd:2a:10:01	br0
192.168.1.6	REACHABLE	5c:61:99:20:9f:b3	br0
192.168.1.11	REACHABLE	e6:35:a7:15:ed:f5	br0

IP address	Flags	HW Address	Device
fe80::da3a:ddff%eth0	STALE	08:3a:dd:2a:10:01	br0
fe80::7ed5:61ff%eth0	STALE	7c:d6:61:bf:43:f7	br0
2001:ee0:4b48:7230:dc2d:b1bd:a080:403	REACHABLE	e6:35:a7:15:ed:f5	br0
2001:ee0:4b48:7230:1498:9b5b:8e04:8156	STALE	fe:0e:22:d2:20:7a	br0
fe80::259c:76e%br0	STALE	5c:61:99:20:9f:b3	br0
2001:ee0:4b48:7230:bc48:103:3bb9:7a1b	STALE	e6:79:3c:5b:3d:40	br0
2001:ee0:4b48:7230:da3a:ddff%br0	STALE	d8:3a:dd:2a:10:01	br0

Hình 57. ARP tại HomeGateway với IP và MAC của Raspberry

Sau khi tiến hành gán IP nội bộ LAN tĩnh cho Raspberry, ta sẽ chọn “Advanced Setup” và chọn “NAT” để tiến hành mở Port bằng Virtual Server.



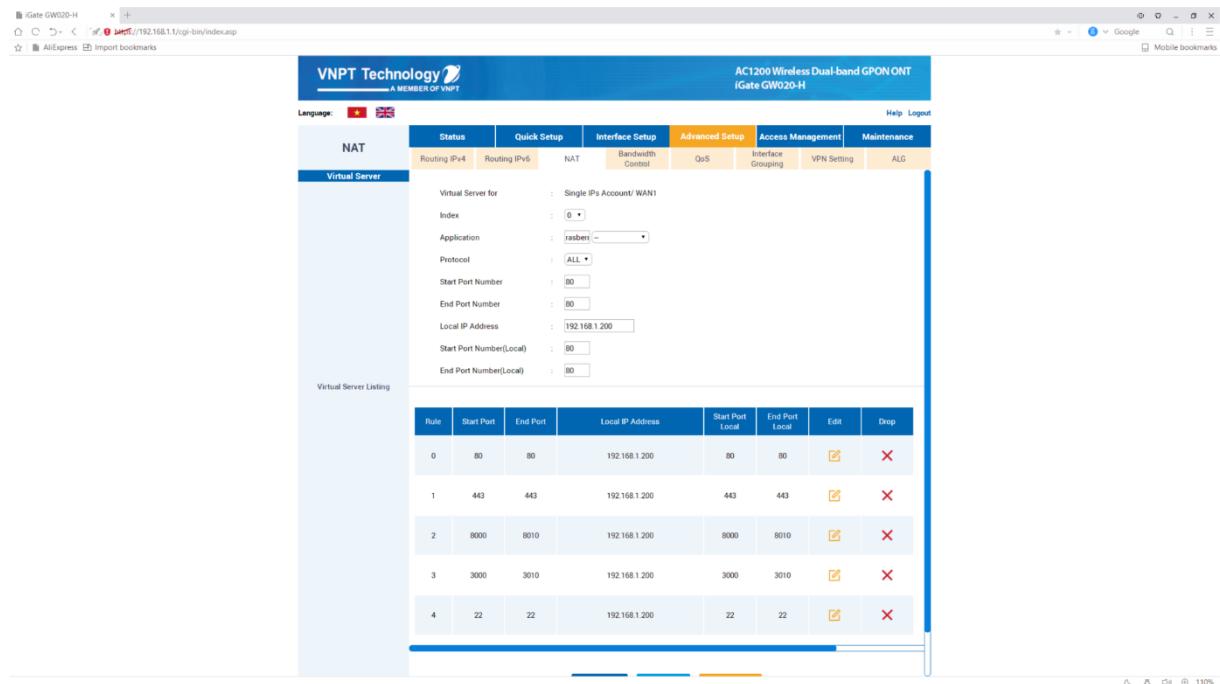
Hình 58. NAT port bằng Virtual Server

Trong giao diện của Virtual Server, chúng ta lần lượt nhập:

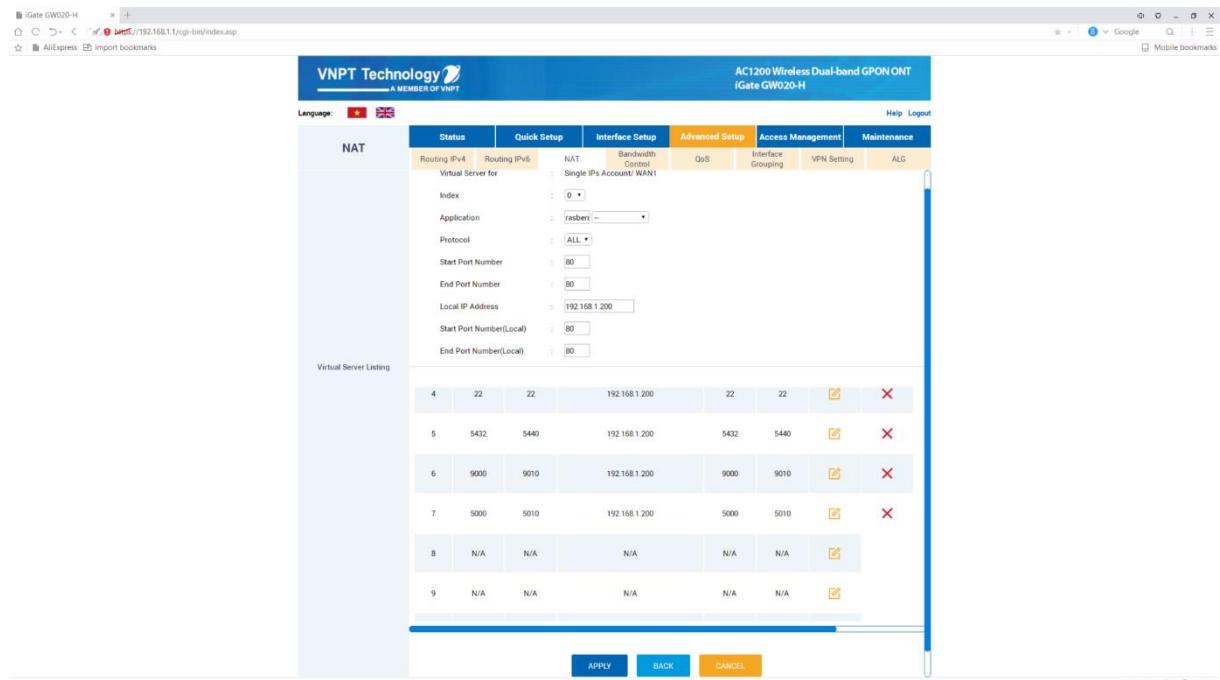
- Application : Chúng ta đặt tên cho dễ phân biệt giữa các thiết bị
- Protocol : Chọn giao thức cho phù hợp thường là all (hoặc chúng ta có thể TCP/UDP tùy thuộc vào mục đích sử dụng )
- Start Port Number : ý nghĩa của nó là mở các port trong khoảng từ
- End Port Number : Có thể truy cập đến Port này
- Local IP Address : Địa chỉ IP của thiết bị cần mở Port
- Start Port Number(Local) : Ý nghĩa của nó là mở các port trong khoảng từ ( Chú ý đây là port của thiết bị )
- End Port Number(Local) : Có thể truy cập đến Port này

Ở đây, em đã mở các cổng 80 (phục vụ giao thức HTTP), cổng 443 (phục vụ giao thức HTTPS), cổng 8000 (nhằm mục đích test cho Django), cổng 3000 (nhằm mục đích test cho ReactJS), cổng 5000 (nhằm mục đích test cho Flask), cổng 22 (nhằm để SSH đến Raspberry từ xa) và cổng 5432 (nhằm mục đích truy cập hệ quản trị cơ sở dữ liệu Postgresql từ xa).

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 59. Cấu hình NAT PORT - 1

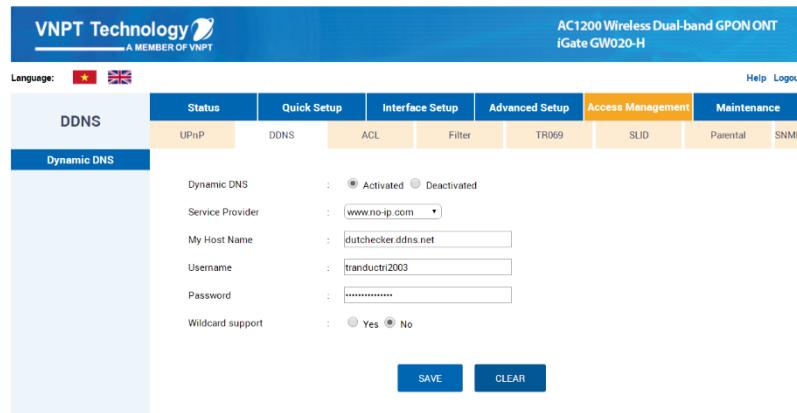


Hình 60. Cấu hình NAT PORT - 2

Cuối cùng, vì nếu chỉ cấu hình như trên, mỗi khi nhà mạng gặp sự cố và cần phải tắt mở lại router, IP WAN của nhà Trí sẽ ngay lập tức thay đổi. Do đó, nếu chúng ta chỉ sử dụng một tên miền bình thường mà chúng ta thuê từ nhà cung cấp tên miền (ví dụ: tenten), IP mà tên miền đó trỏ đến có thể không tồn tại. Để giải quyết vấn đề này, chúng

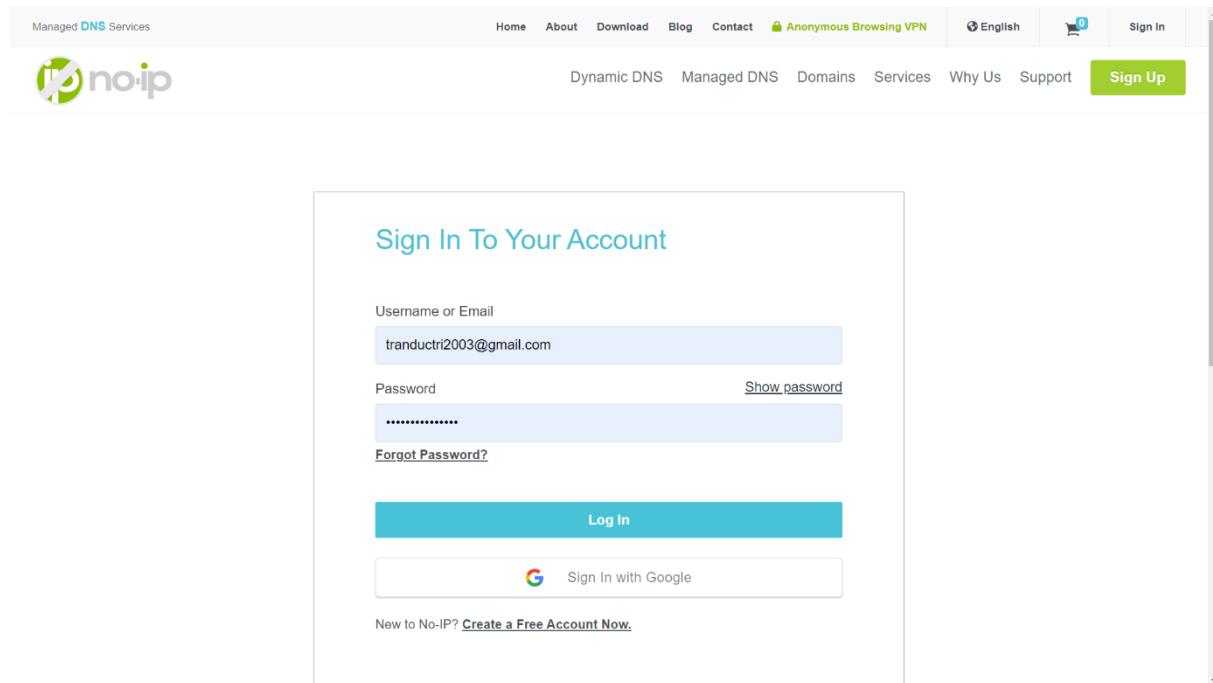
## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

ta cần cấu hình một dịch vụ tên miền động để đảm bảo rằng tên miền sẽ luôn trả về đúng IP WAN của nhà Trí.



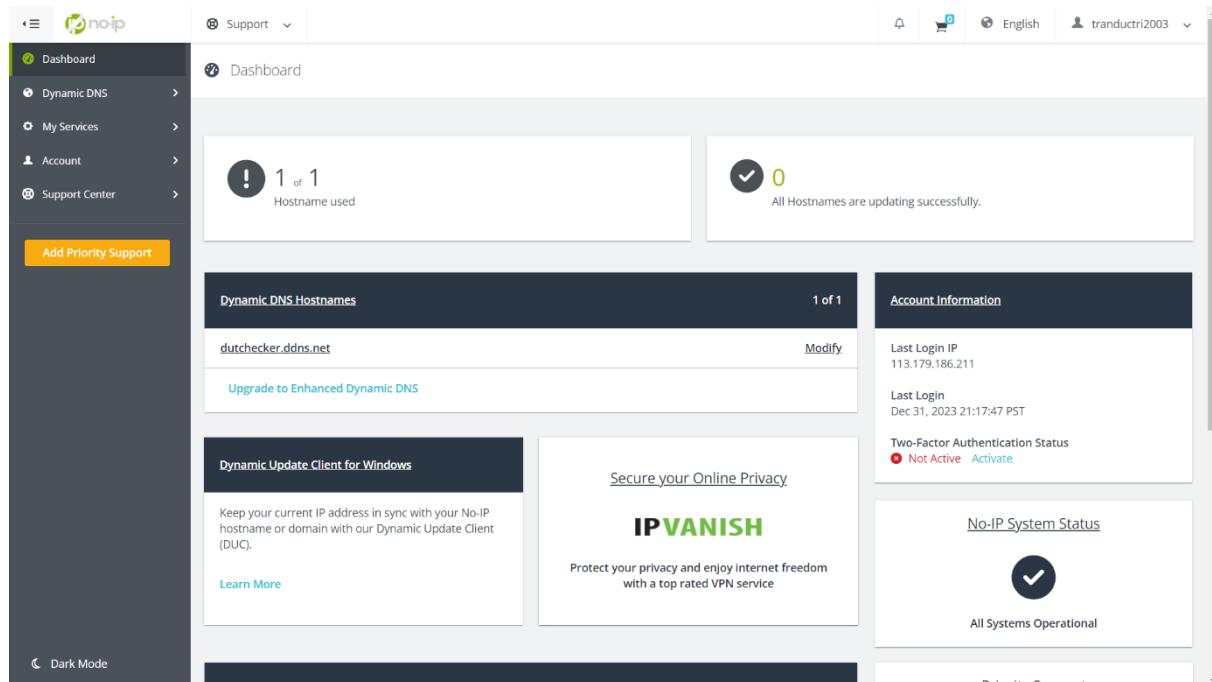
Hình 61. Cài đặt dịch vụ DDNS sử dụng nhà cung cấp No-IP:

Để sử dụng dịch vụ tên miền động DDNS, nhóm chúng em chọn nhà cung cấp dịch vụ No-IP bởi No-IP cung cấp tùy chọn sử dụng dịch vụ miễn phí cho người dùng.

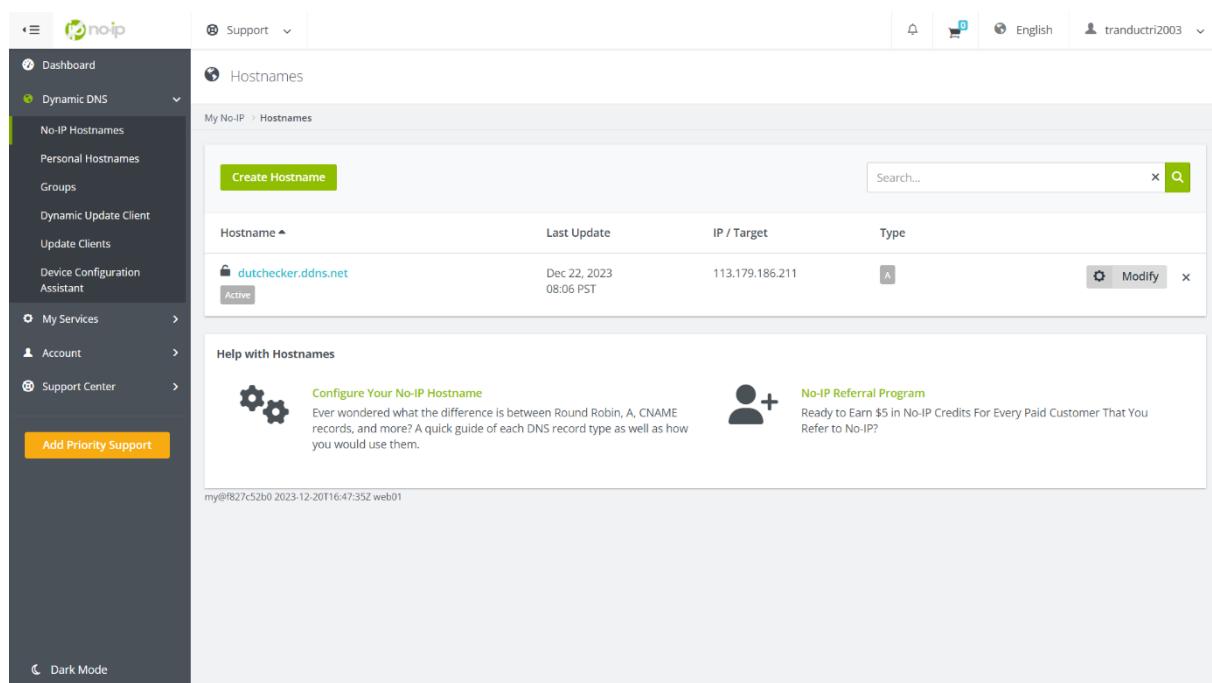


Hình 62. Giao diện đăng nhập của No-IP

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 63.Trang chủ của No-IP



Hình 64.Đăng ký Hostname tại No-IP

Bây giờ chúng ta cần cài trình cập nhật No-IP trên Raspberry để phần mềm có thể liên tục kiểm tra địa chỉ IP WAN và gửi thông báo cập nhật đến server của No-IP.

```
(env) tranductri2003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System/BackEnd$ sudo noip2 -S
[sudo] password for tranductri2003:
1 noip2 process active.

Process 1329, started as noip2, (version 2.1.9)
Using configuration from /usr/local/etc/no-ip2.conf
Last IP Address set 113.179.186.211
Account tranductri2003@gmail.com
configured for:
    host dutchecker.ddns.net
Updating every 5 minutes via /dev/wlan0 with NAT enabled.
```

Hình 65. Cài đặt phần mềm cập nhật IP ở Raspberry

Như được mô tả ở hình trên, hiện đang có một tiến trình đang chạy trên Raspberry Pi, thực hiện kiểm tra liên tục mỗi 5 phút và cập nhật địa chỉ IP kết nối tới server của No-IP. Do đó, trong trường hợp xảy ra sự cố khiến Router tắt hoặc mất kết nối, phần mềm sẽ tự động cập nhật giá trị IP mới và gửi lên server để tên miền dutchecker.ddns.net luôn ánh xạ đến địa chỉ IP mới nhất của Raspberry Pi.

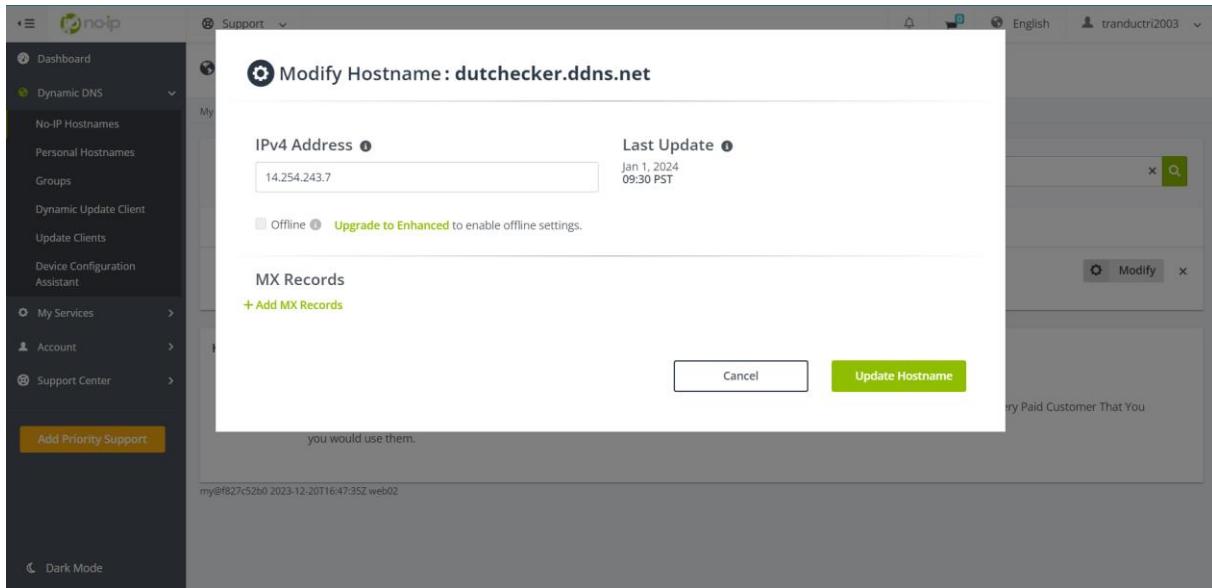
Cuối cùng, chúng ta có thể kiểm tra xem các cổng của Raspberry Pi hiện tại có thể được truy cập từ xa hay không.



Hình 66. Kiểm tra các cổng của Raspberry - 1



Hình 67. Kiểm tra các cổng của Raspberry - 2



Hình 68. IP mới được ánh xạ DDNS sau khi bật tắt modem

### 3.1.3. Triển khai môi trường và mã nguồn

#### a. Cấu hình SSL Let's Encrypt với Certbot

Việc điểm danh yêu cầu người dùng chia sẻ vị trí và hình ảnh trích xuất từ camera, nếu trang web chỉ sử dụng giao thức HTTP thông thường thì không thể mở camera và chia sẻ vị trí. Vì vậy chúng ta cần cài đặt chứng chỉ SSL Let's Encrypt để có thể chuyển đổi từ giao thức HTTP sang HTTPS, cho phép người dùng chia sẻ vị trí và camera đồng thời tăng tính bảo mật cho trang web.

```
tranductri2003@raspberrypi:~/Code/PBL4_DUT_Student-CheckIn-System$ sudo certbot --nginx -d dutcheck
ns.net
[sudo] password for tranductri2003:
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for dutchecker.ddns.net

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/dutchecker.ddns.net/fullchain.pem
Key is saved at:      /etc/letsencrypt/live/dutchecker.ddns.net/privkey.pem
This certificate expires on 2024-03-22.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

Deploying certificate
Successfully deployed certificate for dutchecker.ddns.net to /etc/nginx/sites-enabled/dutchecker.com
Congratulations! You have successfully enabled HTTPS on https://dutchecker.ddns.net

-----
If you like Certbot, please consider supporting our work by:
* Donating to ISRG / Let's Encrypt:  https://letsencrypt.org/donate
* Donating to EFF:                 https://eff.org/donate-le
```

Hình 69. Cài đặt SSL Let's Encrypt với Certbot

Server Hostname

dutchecker.ddns.net Check SSL

dutchecker.ddns.net resolves to 113.179.186.211

Server Type: nginx/1.22.0 (Ubuntu)

The certificate should be trusted by all major web browsers (all the correct intermediate certificates are installed).

The certificate will expire in 80 days. Remind me

The hostname (dutchecker.ddns.net) is correctly listed in the certificate.

**Server** Common name: dutchecker.ddns.net  
SANs: dutchecker.ddns.net  
Valid from December 23, 2023 to March 22, 2024  
Serial Number: 0426ed883497c33a38c5247b72bbcd8f8da  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: R3

**Chain** Common name: R3  
Organization: Let's Encrypt  
Location: US  
Valid from September 3, 2020 to September 15, 2025  
Serial Number: 912b084acf0c18a753f6d62e25a75f5a  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: ISRG Root X1

**Chain** Common name: ISRG Root X1  
Organization: Internet Security Research Group  
Location: US  
Valid from January 20, 2021 to September 30, 2024  
Serial Number: 4001772137d4e942b8ee76aa3c640ab7  
Signature Algorithm: sha256WithRSAEncryption  
Issuer: DST Root CA X3

Hình 70. Kiểm tra chứng chỉ SSL - I

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a web application interface. On the left, there's a sidebar titled "Lịch hôm nay" (Attendance today) and "Lịch sử điểm danh" (Attendance history). The main area displays a table of attendance records with columns: Subject, Attendance Date, Status, and Note. Most entries show a checkmark in the status column and the note "Đã diêm danh". On the right, a modal window titled "Trình xem Chứng chỉ dutchecker.ddns.net" (View Certificate) is open. It shows the certificate details for the domain dutchecker.ddns.net, issued by Let's Encrypt. The certificate is valid from December 16, 2023, to March 22, 2024. The certificate includes a SHA-256 fingerprint and a public key.

Hình 71. Kiểm tra chứng chỉ SSL

## b. Cấu hình Database Connection Pooling:

Để tăng hiệu suất chương trình và tối ưu hóa sử dụng kết nối khi truy vấn cơ sở dữ liệu, nhóm chúng em cấu hình cài đặt Database Connection Pooling với Django.

```
DATABASES = {
    "default": {
        "ENGINE": "dj_db_conn_pool.backends.postgresql",
        "NAME": os.getenv("DB_NAME", 'postgres'),
        "USER": os.getenv("DB_USER", 'postgres'),
        "HOST": os.getenv("DB_HOST", 'localhost'),
        "PASSWORD": os.getenv("DB_PASSWORD", 'postgres'),
        "PORT": os.getenv("DB_PORT", 5432),
        "POOL_OPTIONS": {
            'POOL_SIZE': 50,
            'MAX_OVERFLOW': 10,
            'RECYCLE': 120,
        }
    }
}
```

Hình 72. Cấu hình Database Connection Pooling

Với Postgresql, số lượng connection tối đa mặc định là 100 connections. Điều đó có nghĩa là trong cùng 1 thời điểm, có tối đa 100 kết nối truy vấn cơ sở dữ liệu.

Với cấu hình trên, database được thiết kế như sau:

- POOL\_SIZE (50): Cho phép Connection Pool có thể giữ đồng thời 50 kết nối. Khi ứng dụng cần thực hiện truy vấn đến cơ sở dữ liệu, nó sẽ mượn một trong những kết nối trong pool. Nếu số lượng kết nối mượn vượt quá giới hạn này, các truy vấn khác sẽ phải đợi.
- MAX\_OVERFLOW (10): Nếu có nhu cầu tăng cao và số lượng kết nối đang sử dụng vượt quá 50, pool có thể tạo thêm tối đa 10 kết nối nữa.
- RECYCLE (120): Nếu một kết nối đã tồn tại trong pool và thời gian nó tồn tại vượt quá 120 giây, kết nối đó sẽ được đóng và một kết nối mới sẽ được tạo ra thay thế.

Việc cấu hình Connection Pooling như trên không chỉ tối ưu hiệu suất mà còn giúp ứng dụng linh hoạt và chủ động đối với tình huống tăng cường nhu cầu, đồng thời giảm thiểu tác động đến tài nguyên hệ thống.

### c.Cấu hình cân bằng tải NLB với round-robin:

Nhằm tối ưu hiệu suất, đảm bảo sự ổn định và sẵn sàng của hệ thống, cũng như đáp ứng linh hoạt với số lượng người dùng lớn và giảm thiểu nguy cơ rủi ro, nhóm chúng em đã triển khai cấu hình cân bằng tải cho máy chủ.

```
upstream backend {
    server 127.0.0.1:8000;
    server 127.0.0.1:8001;
}
```

Hình 73.Cấu hình cân bằng tải

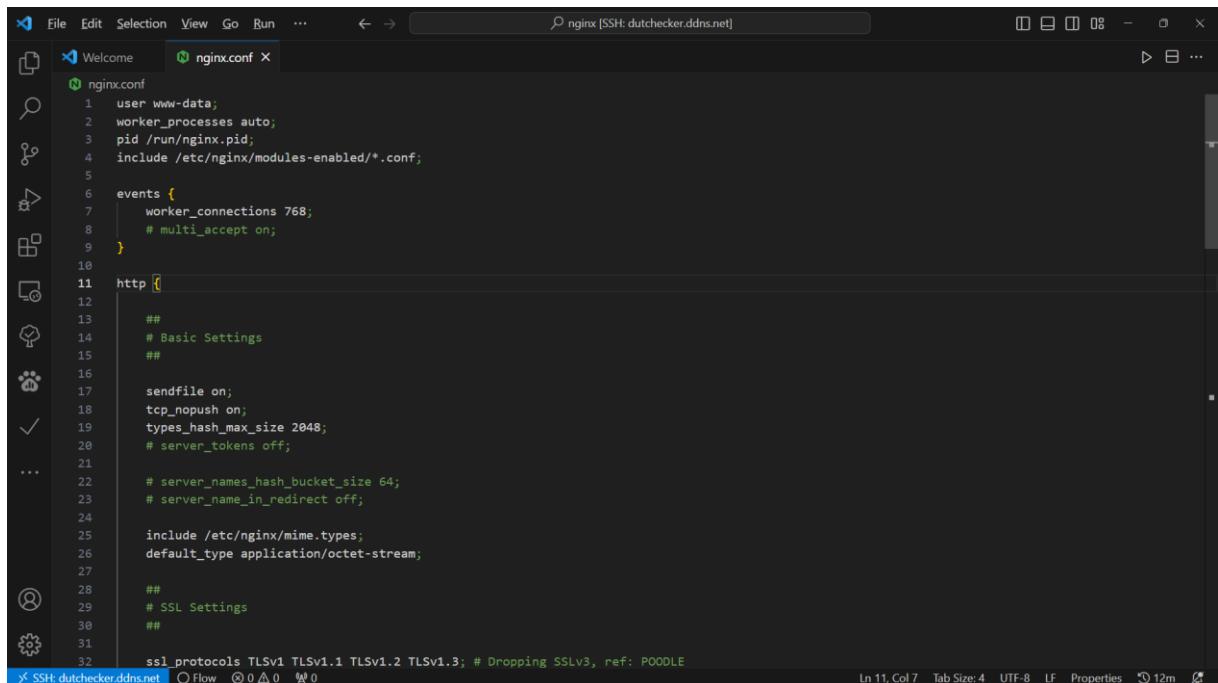
Với cấu hình như trên, nhóm chúng em đã định nghĩa một nhóm các máy chủ mà cân bằng tải sẽ phân phối các yêu cầu đến. Trong trường hợp này, có hai máy chủ được định nghĩa:

- server 127.0.0.1:8000;: Đây là máy chủ đầu tiên với địa chỉ IP là 127.0.0.1 (localhost) và cổng là 8000
- server 127.0.0.1:8001;: Máy chủ thứ hai có địa chỉ IP là 127.0.0.1 (localhost) và cổng là 8001

Như vậy, khi một yêu cầu đến, cân bằng tải sẽ phân phối yêu cầu này đến một trong các máy chủ trong nhóm, giúp chia đều công việc và tăng khả năng chịu tải của hệ thống. Nếu một trong các máy chủ gặp sự cố, cân bằng tải có thể tự động điều chỉnh để tránh gửi yêu cầu đến máy chủ đó.

## d. Cấu hình Nginx:

Để có thể ánh xạ (proxy pass) các URL vào các server cụ thể, nhóm chúng em đã cấu hình Nginx để có thể chuyển hướng các yêu cầu khác nhau đến các đích đến chính xác. Điều này giúp tạo ra một môi trường linh hoạt cho việc điều hướng traffic và cung cấp các dịch vụ web từ nhiều nguồn khác nhau.



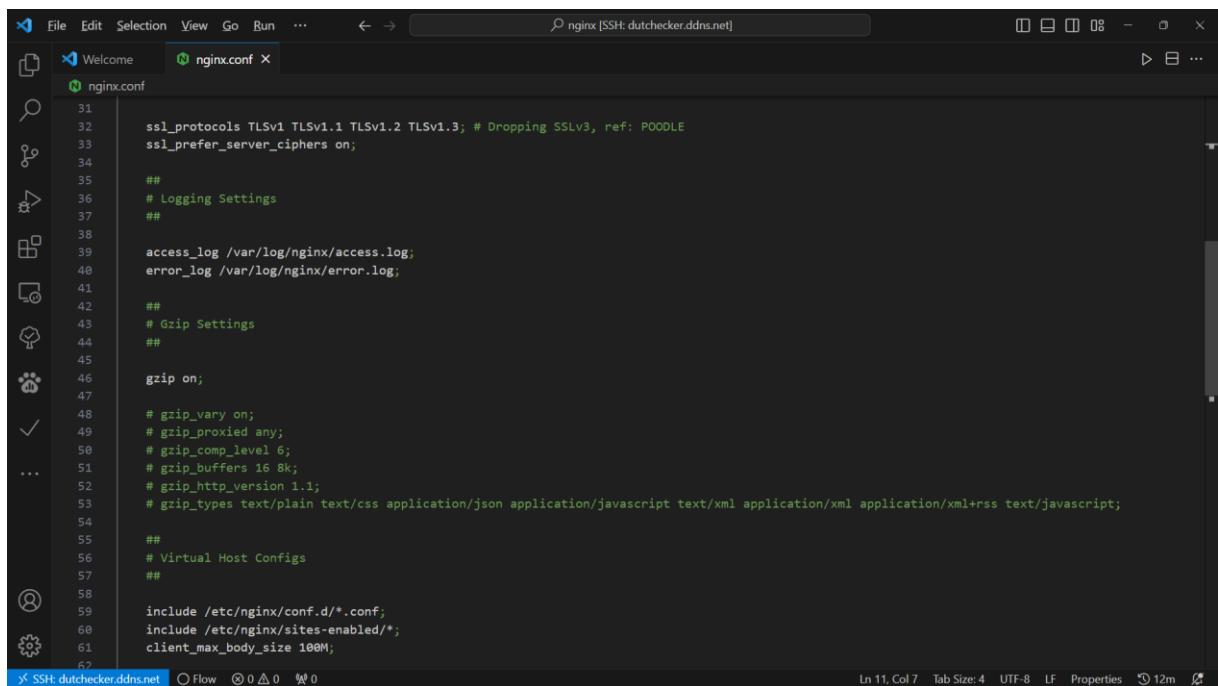
```

File Edit Selection View Go Run ... ← → ○ nginx [SSH: dutchecker.ddns.net]
Welcome nginx.conf x
nginx.conf
1 user www-data;
2 worker_processes auto;
3 pid /run/nginx.pid;
4 include /etc/nginx/modules-enabled/*.conf;
5
6 events {
7     worker_connections 768;
8     # multi_accept on;
9 }
10
11 http {
12
13     ##
14     # Basic Settings
15     ##
16
17     sendfile on;
18     tcp_nopush on;
19     types_hash_max_size 2048;
20     # server_tokens off;
21
22     # server_names_hash_bucket_size 64;
23     # server_name_in_redirect off;
24
25     include /etc/nginx/mime.types;
26     default_type application/octet-stream;
27
28     ##
29     # SSL Settings
30     ##
31
32     ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE

```

In 11, Col 7 Tab Size: 4 UTF-8 LF Properties ⌂ 12m ⌂

Hình 74. File cấu hình nginx.conf - 1



```

File Edit Selection View Go Run ... ← → ○ nginx [SSH: dutchecker.ddns.net]
Welcome nginx.conf x
nginx.conf
31
32     ssl_protocols TLSv1 TLSv1.1 TLSv1.2 TLSv1.3; # Dropping SSLv3, ref: POODLE
33     ssl_prefer_server_ciphers on;
34
35     ##
36     # Logging Settings
37     ##
38
39     access_log /var/log/nginx/access.log;
40     error_log /var/log/nginx/error.log;
41
42     ##
43     # Gzip Settings
44     ##
45
46     gzip on;
47
48     # gzip_vary on;
49     # gzip_proxied any;
50     # gzip_comp_level 6;
51     # gzip_buffers 16 8k;
52     # gzip_http_version 1.1;
53     # gzip_types text/plain text/css application/json application/javascript text/xml application/xml+rss text/javascript;
54
55     ##
56     # Virtual Host Configs
57     ##
58
59     include /etc/nginx/conf.d/*.conf;
60     include /etc/nginx/sites-enabled/*;
61     client_max_body_size 100M;
62

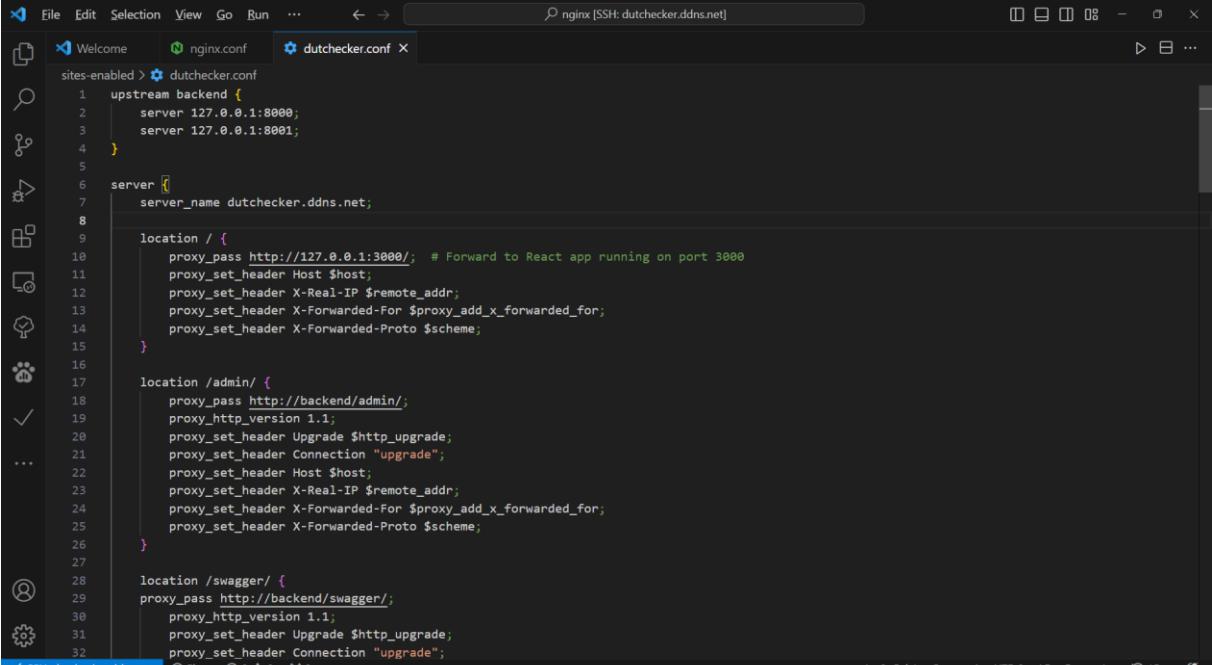
```

In 11, Col 7 Tab Size: 4 UTF-8 LF Properties ⌂ 12m ⌂

Hình 75. File cấu hình nginx.conf - 2

Câu hình trên chứa một dòng quan trọng có tác dụng đặt giới hạn dung lượng cho mỗi yêu cầu từ client là `client_max_body_size 100M;`. Điều này có nghĩa là Nginx sẽ từ chối mọi yêu cầu có kích thước body lớn hơn 100 megabytes. Nếu không có câu hình này, mặc định Nginx có thể từ chối những yêu cầu lớn để bảo vệ máy chủ trước các cuộc tấn công hoặc lạm dụng.

Nếu không có dòng câu hình giới hạn dung lượng nêu trên, khi các API gửi ảnh khuôn mặt để trích xuất đặc trưng có kích thước lớn hơn 100 megabytes, Nginx sẽ trả về lỗi 413 ("Request Entity Too Large"), từ chối yêu cầu và không chuyển tiếp nó đến server ứng dụng. Việc này giúp bảo vệ máy chủ tránh quá tải và tiềm ẩn rủi ro từ các yêu cầu có dung lượng lớn không mong muốn.



```
File Edit Selection View Go Run ... nginx [SSH: dutchecker.ddns.net]
Welcome nginx.conf dutchecker.conf
sites-enabled > dutchecker.conf
1   upstream backend {
2     server 127.0.0.1:8000;
3     server 127.0.0.1:8001;
4   }
5
6   server {
7     server_name dutchecker.ddns.net;
8
9     location / {
10       proxy_pass http://127.0.0.1:3000/; # Forward to React app running on port 3000
11       proxy_set_header Host $host;
12       proxy_set_header X-Real-IP $remote_addr;
13       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
14       proxy_set_header X-Forwarded-Proto $scheme;
15     }
16
17     location /admin/ {
18       proxy_pass http://backend/admin/;
19       proxy_http_version 1.1;
20       proxy_set_header Upgrade $http_upgrade;
21       proxy_set_header Connection "upgrade";
22       proxy_set_header Host $host;
23       proxy_set_header X-Real-IP $remote_addr;
24       proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
25       proxy_set_header X-Forwarded-Proto $scheme;
26     }
27
28     location /swagger/ {
29       proxy_pass http://backend/swagger/;
30       proxy_http_version 1.1;
31       proxy_set_header Upgrade $http_upgrade;
32       proxy_set_header Connection "upgrade";
33     }
34 }
```

Hình 76. File cấu hình dutchecker.conf - 1

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a terminal window titled "nginx [SSH: dutchecker.ddns.net]". The current file is "dutchecker.conf". The code in the editor is as follows:

```
31 proxy_set_header Upgrade $http_upgrade;
32 proxy_set_header Connection "Upgrade";
33 proxy_set_header Host $host;
34 proxy_set_header X-Real-IP $remote_addr;
35 proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
36 proxy_set_header X-Forwarded-Proto $scheme;
37
38
39 location /swagger<format> {
40     proxy_pass http://backend/swagger<format>;
41     proxy_http_version 1.1;
42     proxy_set_header Upgrade $http_upgrade;
43     proxy_set_header Connection "upgrade";
44     proxy_set_header Host $host;
45     proxy_set_header X-Real-IP $remote_addr;
46     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
47     proxy_set_header X-Forwarded-Proto $scheme;
48 }
49
50 location /redoc {
51     proxy_pass http://backend/redoc/;
52     proxy_http_version 1.1;
53     proxy_set_header Upgrade $http_upgrade;
54     proxy_set_header Connection "upgrade";
55     proxy_set_header Host $host;
56     proxy_set_header X-Real-IP $remote_addr;
57     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
58     proxy_set_header X-Forwarded-Proto $scheme;
59 }
60
61 location /media {
62     proxy_pass http://backend/media/;
63 }
```

At the bottom of the terminal window, it says "SSH: dutchecker.ddns.net". The status bar at the bottom right shows "Ln 32, Col 47" and "Properties 15m".

Hình 77. File cấu hình dutchecker.conf - 2

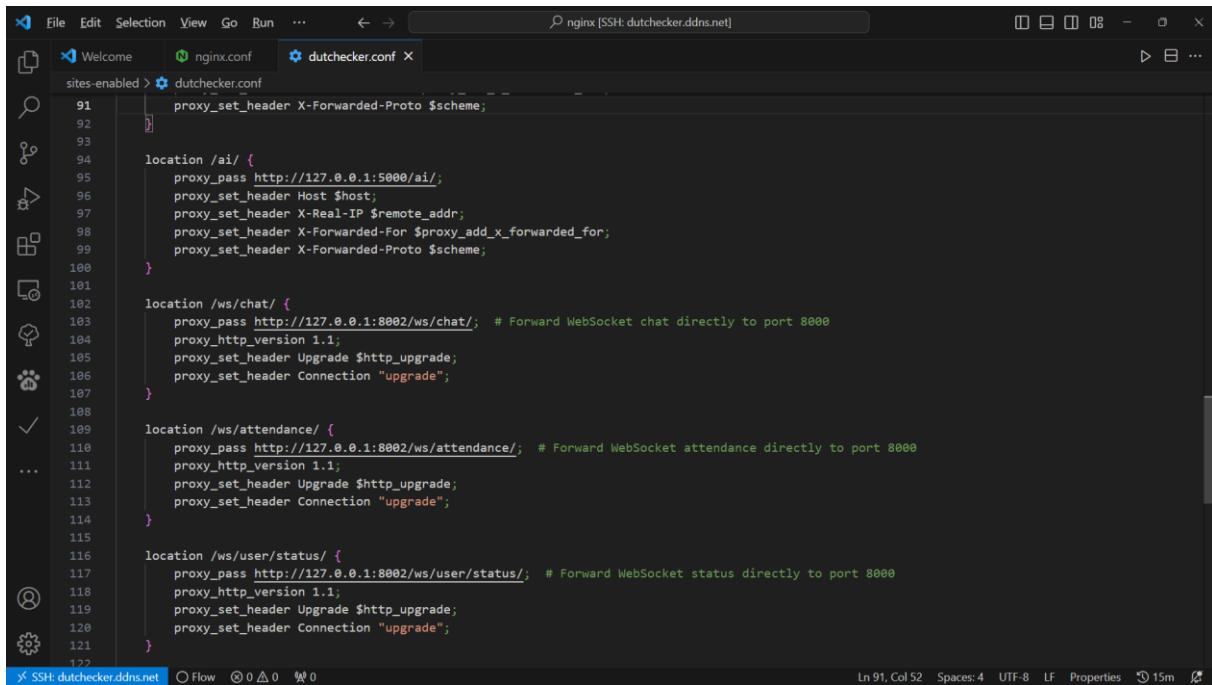
The screenshot shows a terminal window titled "nginx [SSH: dutchecker.ddns.net]". The current file is "dutchecker.conf". The code in the editor is as follows:

```
61 location /media {
62     proxy_pass http://backend/media/;
63     proxy_http_version 1.1;
64     proxy_set_header Upgrade $http_upgrade;
65     proxy_set_header Connection "upgrade";
66     proxy_set_header Host $host;
67     proxy_set_header X-Real-IP $remote_addr;
68     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
69     proxy_set_header X-Forwarded-Proto $scheme;
70 }
71
72 location /django-static/ {
73     proxy_pass http://backend/django-static/;
74     proxy_http_version 1.1;
75     proxy_set_header Upgrade $http_upgrade;
76     proxy_set_header Connection "upgrade";
77     proxy_set_header Host $host;
78     proxy_set_header X-Real-IP $remote_addr;
79     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
80     proxy_set_header X-Forwarded-Proto $scheme;
81 }
82
83 location /api/ {
84     proxy_pass http://backend/api/;
85     proxy_http_version 1.1;
86     proxy_set_header Upgrade $http_upgrade;
87     proxy_set_header Connection "upgrade";
88     proxy_set_header Host $host;
89     proxy_set_header X-Real-IP $remote_addr;
90     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
91     proxy_set_header X-Forwarded-Proto $scheme;
92 }
```

At the bottom of the terminal window, it says "SSH: dutchecker.ddns.net". The status bar at the bottom right shows "Ln 91, Col 52" and "Properties 15m".

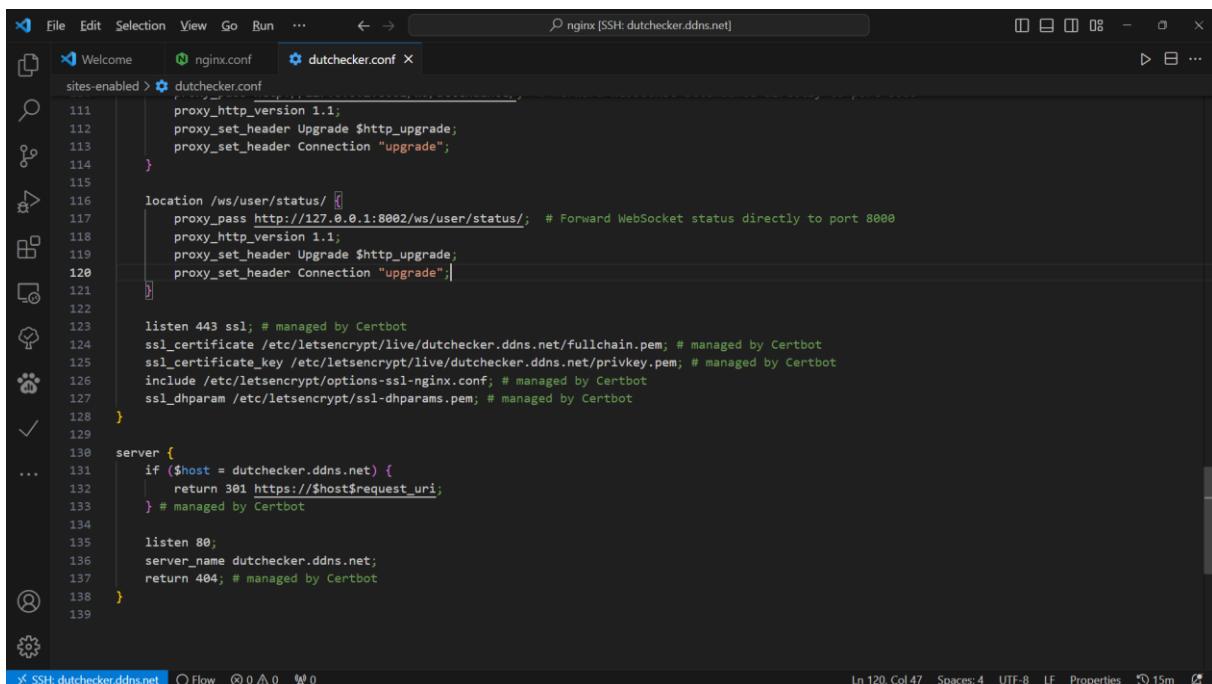
Hình 78. File cấu hình dutchecker.conf - 3

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



```
nginx [SSH: dutchecker.ddns.net]
File Edit Selection View Go Run ... ln 91, Col 52 Spaces: 4 UTF-8 LF Properties ⌂ 15m ⌂
Welcome nginx.conf dutchecker.conf
sites-enabled > dutchecker.conf
91     proxy_set_header X-Forwarded-Proto $scheme;
92 }
93
94 location /ai/ {
95     proxy_pass http://127.0.0.1:5000/ai/;
96     proxy_set_header Host $host;
97     proxy_set_header X-Real-IP $remote_addr;
98     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
99     proxy_set_header X-Forwarded-Proto $scheme;
100 }
101
102 location /ws/chat/ {
103     proxy_pass http://127.0.0.1:8002/ws/chat/; # Forward WebSocket chat directly to port 8000
104     proxy_http_version 1.1;
105     proxy_set_header Upgrade $http_upgrade;
106     proxy_set_header Connection "upgrade";
107 }
108
109 location /ws/attendance/ {
110     proxy_pass http://127.0.0.1:8002/ws/attendance/; # Forward WebSocket attendance directly to port 8000
111     proxy_http_version 1.1;
112     proxy_set_header Upgrade $http_upgrade;
113     proxy_set_header Connection "upgrade";
114 }
115
116 location /ws/user/status/ {
117     proxy_pass http://127.0.0.1:8002/ws/user/status/; # Forward WebSocket status directly to port 8000
118     proxy_http_version 1.1;
119     proxy_set_header Upgrade $http_upgrade;
120     proxy_set_header Connection "upgrade";
121 }
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
```

Hình 79. File cấu hình dutchecker.conf - 4



```
nginx [SSH: dutchecker.ddns.net]
File Edit Selection View Go Run ... ln 120, Col 47 Spaces: 4 UTF-8 LF Properties ⌂ 15m ⌂
Welcome nginx.conf dutchecker.conf
sites-enabled > dutchecker.conf
111     proxy_http_version 1.1;
112     proxy_set_header Upgrade $http_upgrade;
113     proxy_set_header Connection "upgrade";
114 }
115
116 location /ws/user/status/ {
117     proxy_pass http://127.0.0.1:8002/ws/user/status/; # Forward WebSocket status directly to port 8000
118     proxy_http_version 1.1;
119     proxy_set_header Upgrade $http_upgrade;
120     proxy_set_header Connection "upgrade";
121 }
122
123 listen 443 ssl; # managed by Certbot
124 ssl_certificate /etc/letsencrypt/live/dutchecker.ddns.net/fullchain.pem; # managed by Certbot
125 ssl_certificate_key /etc/letsencrypt/live/dutchecker.ddns.net/privkey.pem; # managed by Certbot
126 include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
127 ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot
128 }
129
130 server {
131     if ($host = dutchecker.ddns.net) {
132         return 301 https://$host$request_uri;
133     } # managed by Certbot
134
135     listen 80;
136     server_name dutchecker.ddns.net;
137     return 404; # managed by Certbot
138 }
139
```

Hình 80. File cấu hình dutchecker.conf - 5

Với cấu hình như trên, các URL đầu vào sẽ được ánh xạ đến các server đích đến khác nhau như sau:

- URL chính (React app):

**URL: http://dutchecker.ddns.net/**

Chuyển hướng đến: http://127.0.0.1:3000 (React app)

Proxy pass: proxy\_pass http://127.0.0.1:3000/;

- Backend Django - /admin/:

**URL: http://dutchecker.ddns.net/admin/**

Chuyển hướng đến: http://backend/admin/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";

Proxy pass: proxy\_pass http://backend/admin/;

- Backend Django - /swagger/:

**URL: http://dutchecker.ddns.net/swagger/**

Chuyển hướng đến: http://backend/swagger/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";

Proxy pass: proxy\_pass http://backend/swagger/;

- Backend Django - /swagger<format>/:

**URL: http://dutchecker.ddns.net/swagger<format>/**

Chuyển hướng đến: http://backend/swagger<format>/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";

Proxy pass: proxy\_pass http://backend/swagger<format>/;

- Backend Django - /redoc/:

**URL: http://dutchecker.ddns.net/redoc/**

Chuyển hướng đến: http://backend/redoc/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";

Proxy pass: proxy\_pass http://backend/redoc/;

- Backend Django - /media/:

**URL: http://dutchecker.ddns.net/media/**

Chuyển hướng đến: http://backend/media/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";  
Proxy pass: proxy\_pass http://backend/media/;

**- Backend Django - /django-static/:**

**URL: http://dutchecker.ddns.net/django-static/**

Chuyển hướng đến: http://backend/django-static/  
Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";  
Proxy pass: proxy\_pass http://backend/django-static/;

**- Backend Django - /api/:**

**URL: http://dutchecker.ddns.net/api/**

Chuyển hướng đến: http://backend/api/  
Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";  
Proxy pass: proxy\_pass http://backend/api/;

**- Backend Flask AI - /ai/:**

**URL: http://dutchecker.ddns.net/ai/**

Chuyển hướng đến: http://127.0.0.1:5000/ai/  
Proxy pass: proxy\_pass http://127.0.0.1:5000/ai/;

**- WebSocket - /ws/chat/:**

**URL: ws://dutchecker.ddns.net/ws/chat/**

Chuyển hướng đến: http://127.0.0.1:8002/ws/chat/  
Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";  
Proxy pass: proxy\_pass http://127.0.0.1:8002/ws/chat/;

**- WebSocket - /ws/attendance/:**

**URL: ws://dutchecker.ddns.net/ws/attendance/**

Chuyển hướng đến: http://127.0.0.1:8002/ws/attendance/  
Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";  
Proxy pass: proxy\_pass http://127.0.0.1:8002/ws/attendance/;

- **WebSocket - /ws/user/status/:**

**URL: ws://dutchecker.ddns.net/ws/user/status/**

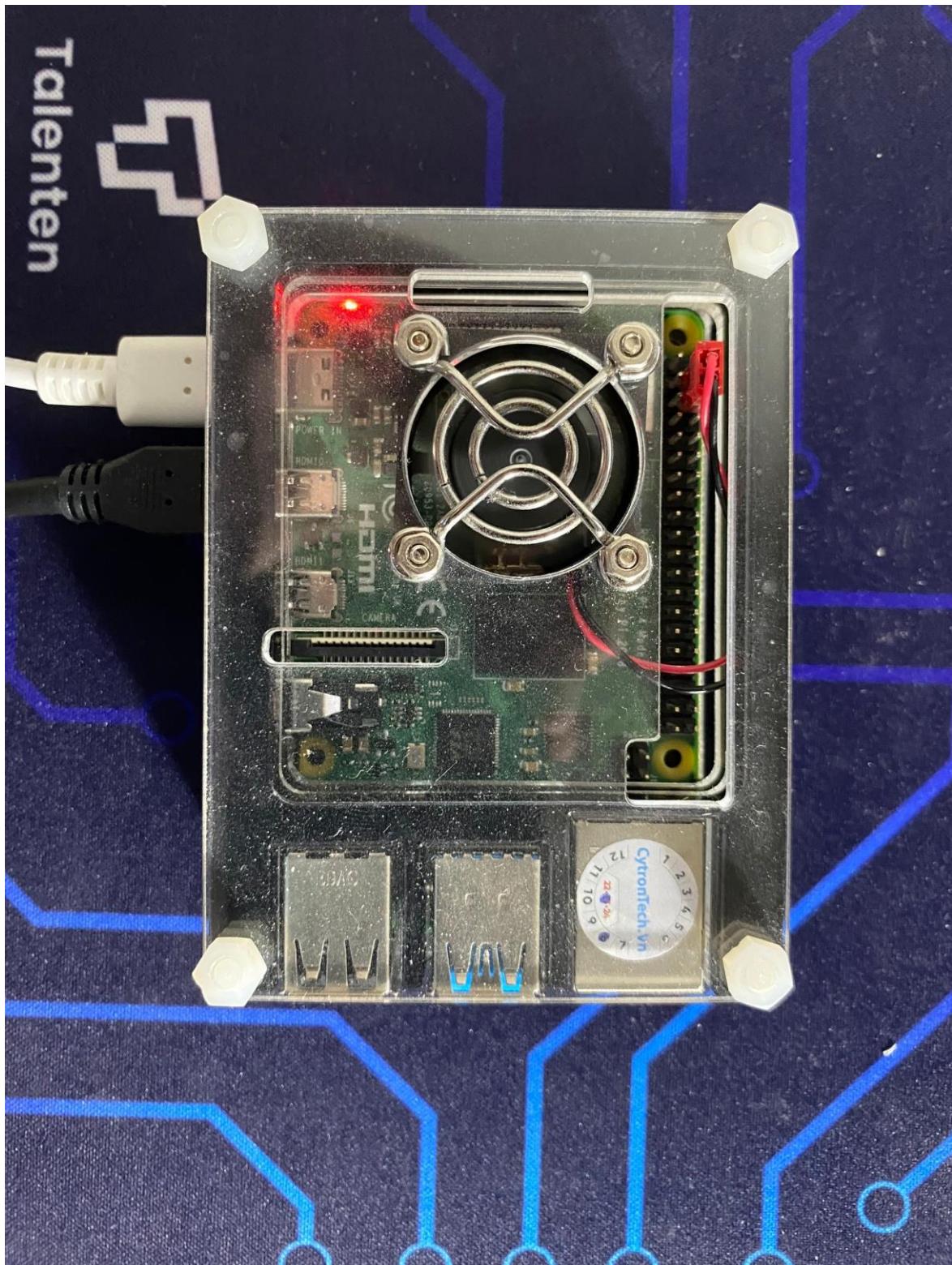
Chuyển hướng đến: http://127.0.0.1:8002/ws/user/status/

Cấu hình cho WebSocket: proxy\_http\_version 1.1;, proxy\_set\_header Upgrade \$http\_upgrade;, proxy\_set\_header Connection "upgrade";

Proxy pass: proxy\_pass <http://127.0.0.1:8002/ws/user/status/>;

e. Thiết lập môi trường Web Server:

Sau khi cài đặt hệ điều hành trên Raspberry bằng phần mềm Raspberry Pi Imager, ta sẽ thiết lập kết nối SSH đến Raspberry để thuận tiện cho việc cài đặt và triển khai mã nguồn.



**Hình 81.**Raspberry được cấp nguồn để hoạt động liên tục

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

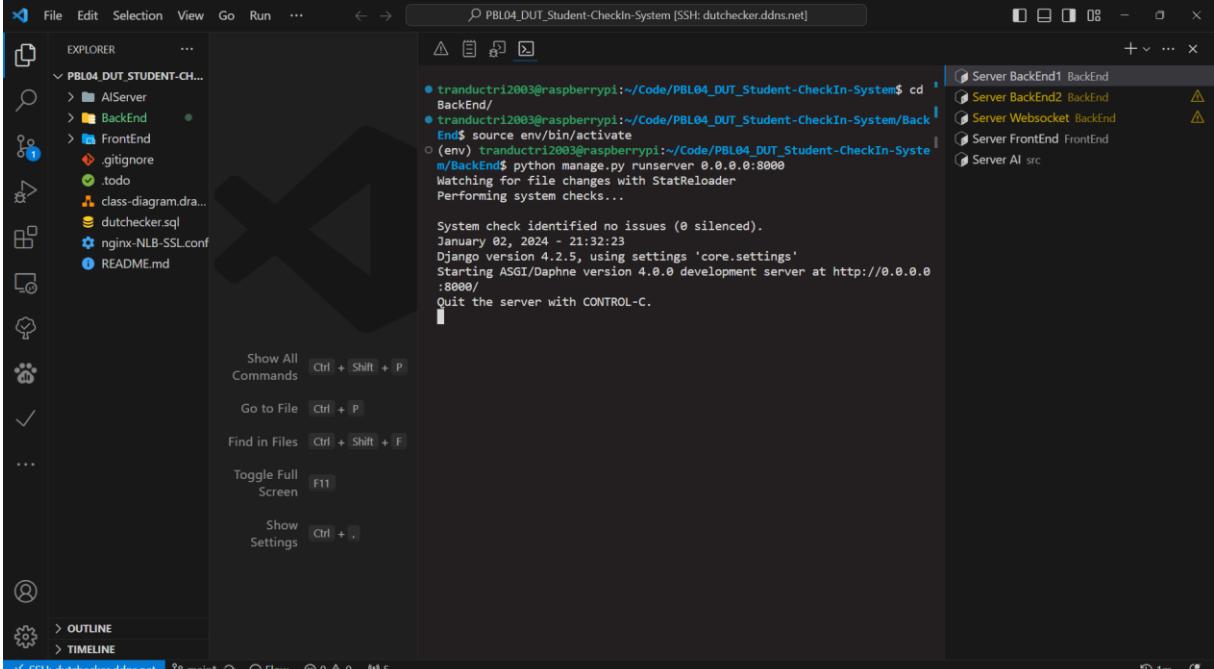
Sau khi việc kết nối từ xa thông qua SSH với Raspberry được hoàn thành, chúng ta sẽ tiến hành cài đặt mã nguồn của dự án, các gói thư viện cùng các file môi trường thông qua lệnh: [https://github.com/tranductri2003/PBL04\\_DUT\\_Student-CheckIn-System.git](https://github.com/tranductri2003/PBL04_DUT_Student-CheckIn-System.git)

```
tranductri2003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ neofetch
tranductri2003@raspberrypi:~/
OS: Ubuntu 23.04 aarch64
Host: Raspberry Pi 4 Model B Rev 1.5
Kernel: 6.2.0-1018-raspi
Uptime: 10 days, 21 hours, 56 mins
Packages: 1318 (dpkg), 4 (snap)
Shell: bash 5.2.15
Terminal: bash
CPU: BCM2835 (4) @ 1.80GHz
Memory: 549MiB / 7869MiB
```

Hình 82. Giao diện của Raspberry sau khi cài đặt mã nguồn

Các URL gọi đến tên miền dutchecker.ddns.net sẽ được Nginx chuyển hướng đến các server đang chạy localhost của Raspberry. Trong đó, lần lượt chia ra 2 server BackEnd chạy mã nguồn Django để xử lý các API thực hiện truy vấn cơ sở dữ liệu được liên kết với nhau trong cụm Cluster của cân bằng tải, 1 server AI chạy mã nguồn Flask để xử lý các yêu cầu trích xuất đặc trưng ảnh nhằm mục đích tạo dữ liệu điểm ảnh khuôn mặt cũng như xác thực khuôn mặt, 1 server WebSocket chạy mã nguồn Django để xử lý các thao tác liên quan đến việc kết nối WebSocket (chat, điểm danh, kiểm tra trạng thái trực tuyến) và 1 server FrontEnd chạy mã nguồn ReactJS để tạo ra giao diện tương tác người dùng cho trang web.

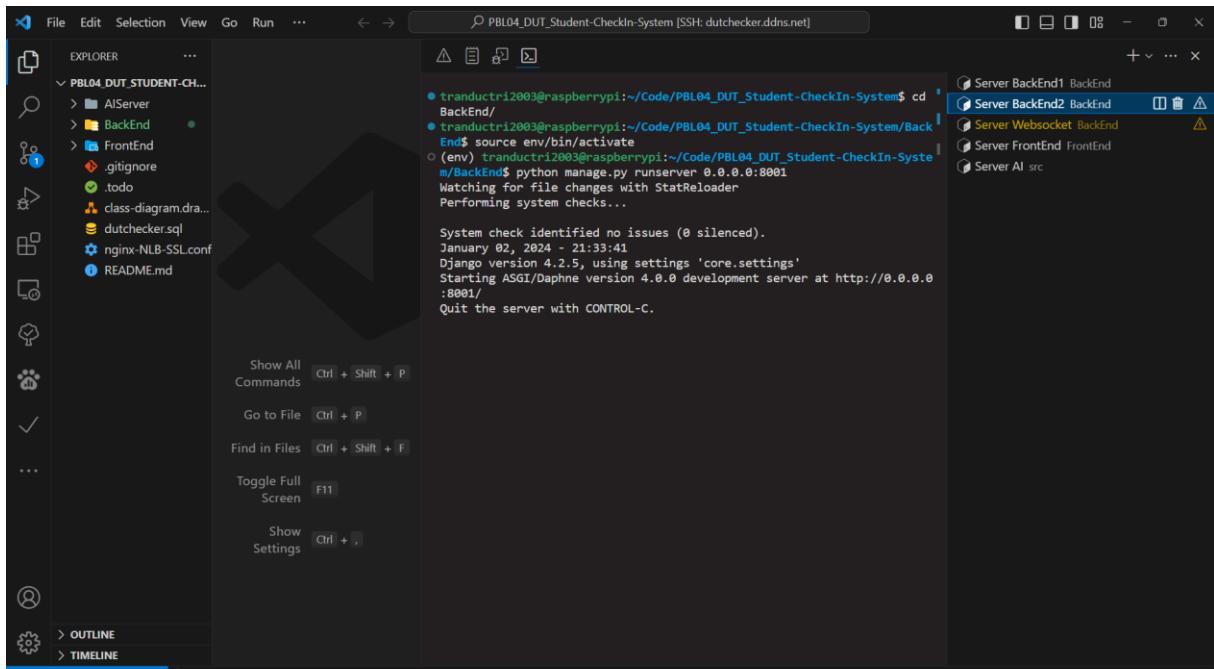
## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



The screenshot shows a terminal window in VS Code connected via SSH to a Raspberry Pi at `dutchecker.ddns.net`. The terminal output shows the user navigating to the `Backend` directory and running the command `python manage.py runserver 0.0.0.0:8000`. The server starts successfully, listening on port 8000.

```
tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ cd Backend/
tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System/Backend$ source env/bin/activate
(env) tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System/Backend$ python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
January 02, 2024 - 21:32:23
Django version 4.2.5, using settings 'core.settings'
Starting ASGI/Daphne version 4.0.0 development server at http://0.0.0.0:8000/
Quit the server with CONTROL-C.
```

Hình 83. Khởi động server BackEnd thứ nhất



The screenshot shows a terminal window in VS Code connected via SSH to a Raspberry Pi at `dutchecker.ddns.net`. The terminal output shows the user navigating to the `Backend` directory and running the command `python manage.py runserver 0.0.0.0:8001`. The server starts successfully, listening on port 8001.

```
tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ cd Backend/
tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System/Backend$ source env/bin/activate
(env) tranductri12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System/Backend$ python manage.py runserver 0.0.0.0:8001
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
January 02, 2024 - 21:33:41
Django version 4.2.5, using settings 'core.settings'
Starting ASGI/Daphne version 4.0.0 development server at http://0.0.0.0:8001/
Quit the server with CONTROL-C.
```

Hình 84. Khởi động server BackEnd thứ hai

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a terminal window titled "PBL4\_DUT\_Student-CheckIn-System [SSH: dutchecker.ddns.net]". The command "python manage.py runserver 0.0.0.0:8002" is being run, and the output shows the server starting up on port 8002.

```
tranductr12003@raspberrypi:~/Code/PBL4_DUT_Student-CheckIn-System/Backend$ source env/bin/activate
(env) tranductr12003@raspberrypi:~/Code/PBL4_DUT_Student-CheckIn-System/Backend$ python manage.py runserver 0.0.0.0:8002
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
January 02, 2024 - 21:38:04
Django version 4.2.5, using settings 'core.settings'
Starting ASGI/Daphne version 4.0.0 development server at http://0.0.0.0:8002/
Quit the server with CONTROL-C.
```

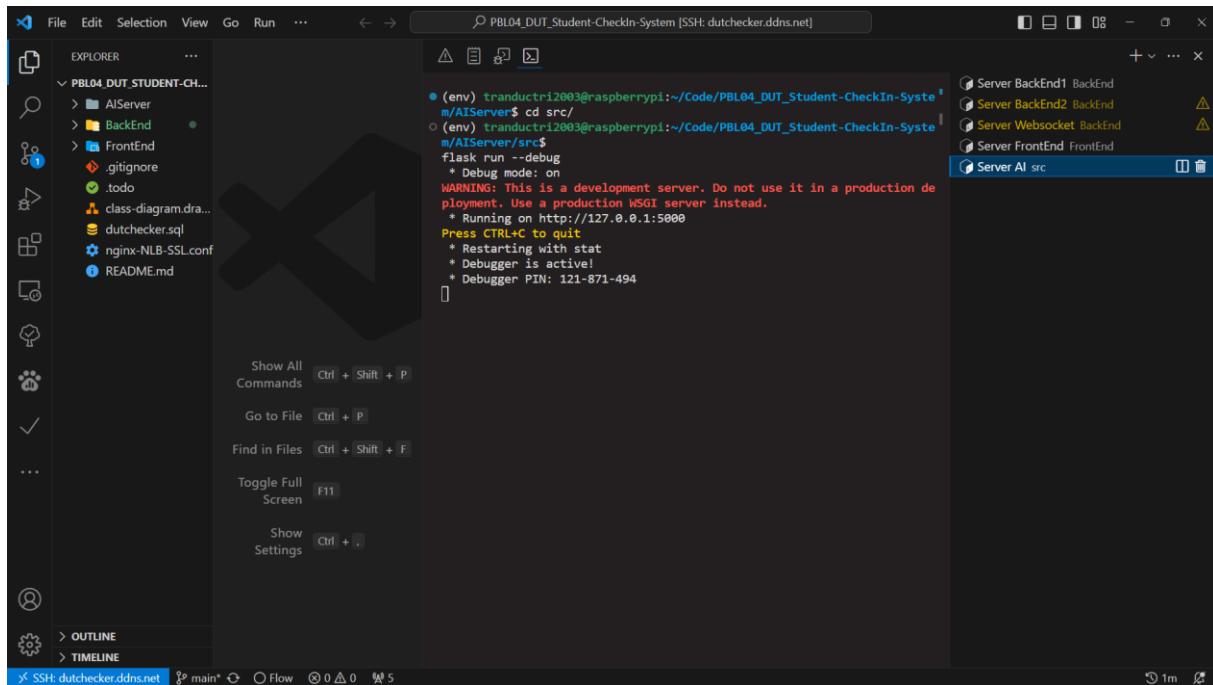
Hình 85.Khởi động server Websocket

The screenshot shows a terminal window titled "PBL4\_DUT\_Student-CheckIn-System [SSH: dutchecker.ddns.net]". The command "npm run build" is being run, and the output shows the build process completed successfully.

```
Compiled successfully!
You can now view core in the browser.
Local:          http://localhost:3000
On Your Network: http://192.168.1.200:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
webpack compiled successfully
```

Hình 86.Khởi động server FrontEnd

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

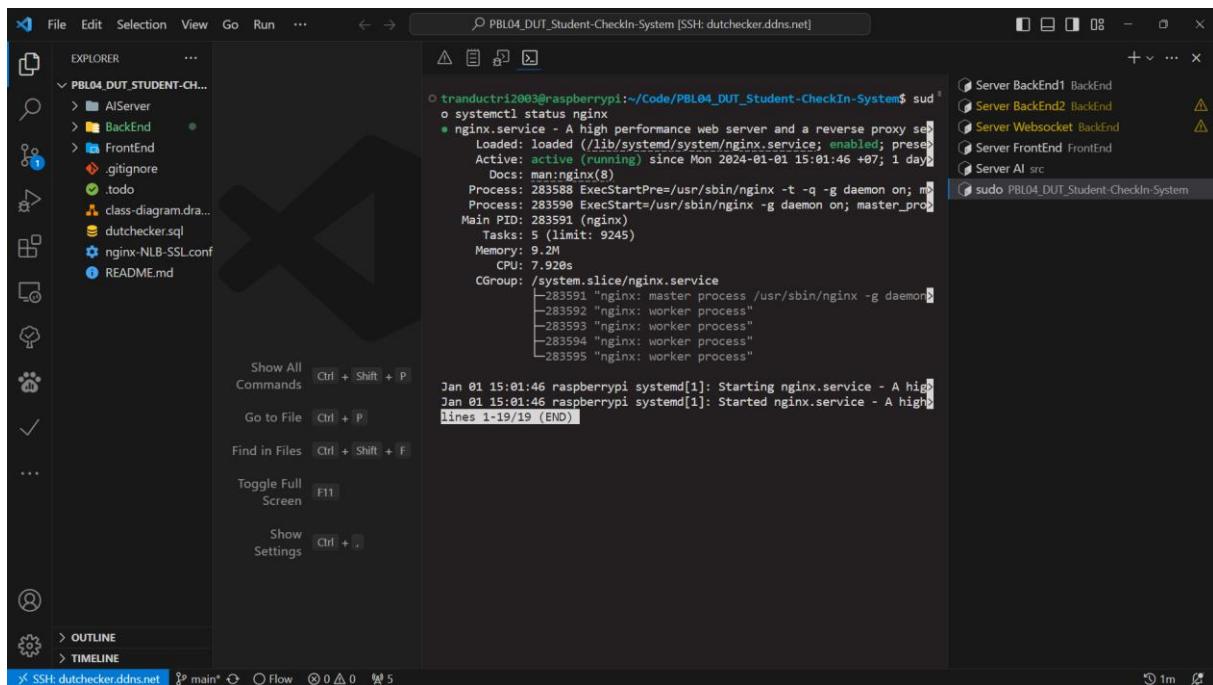


The screenshot shows the Visual Studio Code interface connected via SSH to a Raspberry Pi at dutchecker.ddns.net. The Explorer sidebar shows the project structure for 'PBL4\_DUT\_STUDENT-CH...'. The terminal window displays the command:

```
(env) tranductr12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ cd src/  
(env) tranductr12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ flask run -debug  
 * Debug mode: on  
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.  
 * Running on http://127.0.0.1:5000  
Press CTRL+C to quit  
 * Restarting with stat  
 * Debugger is active!  
 * Debugger PIN: 121-871-494
```

On the right side, there is a sidebar with icons for various servers: Server BackEnd1, Server BackEnd2, Server WebSocket, Server FrontEnd, and Server AI.

Hình 87. Khởi động server AI



The screenshot shows the Visual Studio Code interface connected via SSH to a Raspberry Pi at dutchecker.ddns.net. The Explorer sidebar shows the project structure for 'PBL4\_DUT\_STUDENT-CH...'. The terminal window displays the command:

```
tranductr12003@raspberrypi:~/Code/PBL04_DUT_Student-CheckIn-System$ sudo systemctl status nginx  
● nginx.service - A high performance web server and a reverse proxy server  
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)  
   Active: active (running) since Mon 2024-01-01 15:01:46 +07; 1 day ago  
     Docs: man:nginx(8)  
   Process: 283588 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process  
   Process: 283590 ExecStart=/usr/sbin/nginx -g daemon on; master_process  
 Main PID: 283591 (nginx)  
    Tasks: 5 (limit: 9245)  
      Memory: 9.2M  
        CPU: 7.920s  
       CGroup: /system.slice/nginx.service  
           ├─283591 "nginx: master process /usr/sbin/nginx -g daemon on; master_<...>  
           ├─283592 "nginx: worker process"  
           ├─283593 "nginx: worker process"  
           ├─283594 "nginx: worker process"  
           └─283595 "nginx: worker process"  
  
Jan 01 15:01:46 raspberrypi systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...  
Jan 01 15:01:46 raspberrypi systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
```

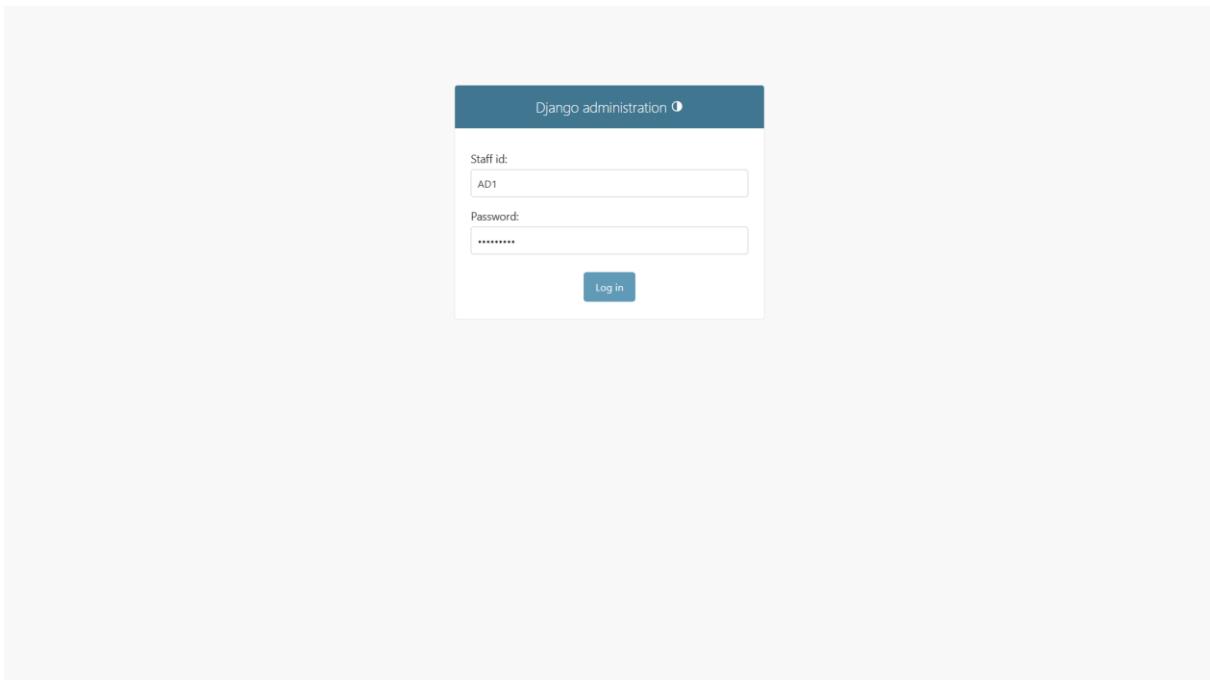
On the right side, there is a sidebar with icons for various servers: Server BackEnd1, Server BackEnd2, Server WebSocket, Server FrontEnd, and Server AI.

Hình 88. Kiểm tra trạng thái hoạt động của Nginx

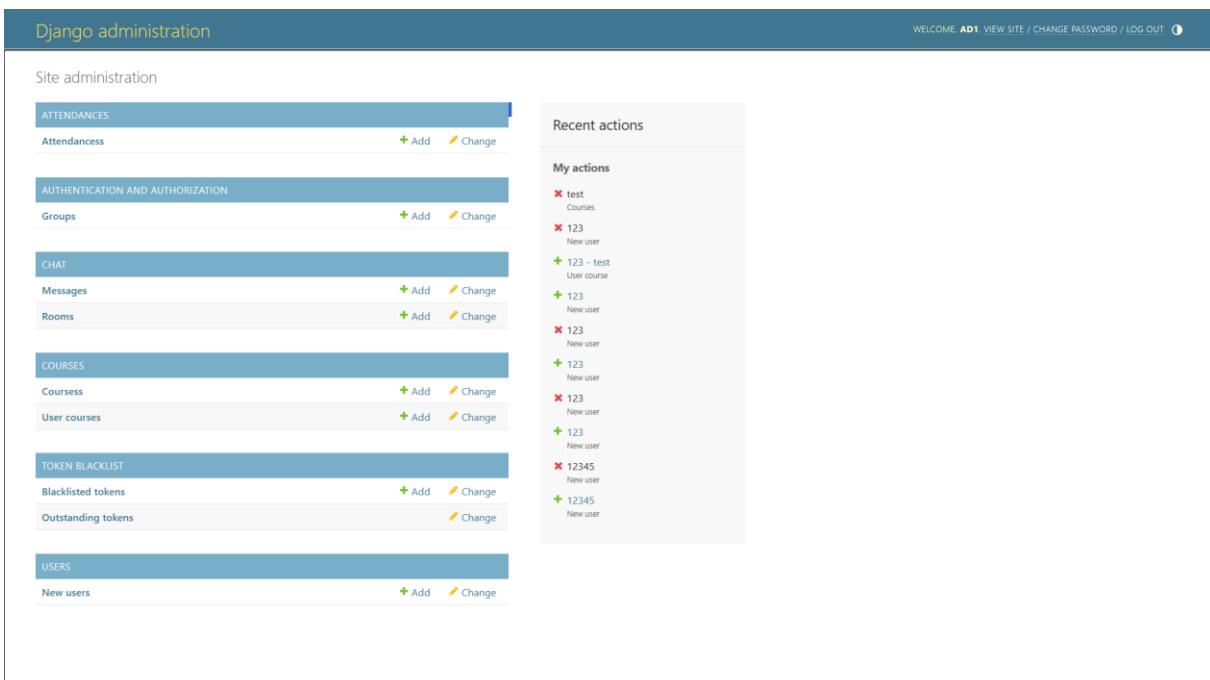
## 3.2. Kết quả chạy chương trình

### 3.2.1. Quản lý người dùng:

- Tạo tài khoản:



Hình 89. Giao diện đăng nhập admin



Hình 90. Giao diện quản lý các model

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows the Django administration interface for the 'Users' section, specifically the 'New users' list view. The left sidebar contains navigation links for ATTENDANCES, AUTHENTICATION AND AUTHORIZATION, CHAT, COURSES, TOKEN BLACKLIST, and USERS. Under USERS, 'New users' is selected. The main content area displays a table of 50 users, each with a checkbox, staff ID, full name, role, and status. A search bar and filter options are at the top of the list table.

Action:	EMAIL ADDRESS	STAFF ID	FULL NAME	ROLE	IS ACTIVE	IS STAFF	STATUS
<input type="checkbox"/>	102210336@sv1.dut.udn.vn	102210336	Lê Hữu Minh Vũ	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210335@sv1.dut.udn.vn	102210335	Lê Quốc Vinh	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210334@sv1.dut.udn.vn	102210334	Lê Văn Tư	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210332@sv1.dut.udn.vn	102210332	Nguyễn Thúy Trinh	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210331@sv1.dut.udn.vn	102210331	Vũ Hoàng Tin	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210330@sv1.dut.udn.vn	102210330	Hoàng Gia Tin	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210328@sv1.dut.udn.vn	102210328	Lê Đức Thành	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210326@sv1.dut.udn.vn	102210326	Nguyễn Phạm Phúc Tân	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210325@sv1.dut.udn.vn	102210325	Nguyễn Khắc Nhân Tâm	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210324@sv1.dut.udn.vn	102210324	Trịnh Thái Quân	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210323@sv1.dut.udn.vn	102210323	Hồ Duy Phúc	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210322@sv1.dut.udn.vn	102210322	Trương Nguyễn Hoàng Phát	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210321@sv1.dut.udn.vn	102210321	Võ Trọng Nhán	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210320@sv1.dut.udn.vn	102210320	Đỗ Huỳnh Trung Nhán	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210319@sv1.dut.udn.vn	102210319	Lê Văn Công Luận	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210318@sv1.dut.udn.vn	102210318	Trần Hoàng Thế Linh	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="checkbox"/>	102210316@sv1.dut.udn.vn	102210316	Châu Hoàng Huy	Student	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

Hình 91. Giao diện quản lý người dùng

The screenshot shows the Django administration interface for the 'Users' section, specifically the 'Add new user' form. The left sidebar contains navigation links for ATTENDANCES, AUTHENTICATION AND AUTHORIZATION, CHAT, COURSES, TOKEN BLACKLIST, and USERS. Under USERS, 'New users' is selected. The main content area displays a form with fields for Staff id, Email address, Full name, Phone number, Avatar, University, Faculty, Class id, Role, and checkboxes for Is active and Is staff. Buttons at the bottom include 'SAVE', 'Save and add another', and 'Save and continue editing'.

Hình 92. Nhập thông tin

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows the Django Admin interface for 'New users'. A success message at the top states: 'The new user "Phạm Nguyễn Anh Phát" was added successfully. You may edit it again below.' The user details are listed: Staff id: 102210999, Email address: phathanhpham2003@gmail.com, Full name: Phạm Nguyễn Anh Phát, Avatar: 102210902\_AVQa02.jpg, University: Đại học Bách Khoa, Faculty: Công nghệ thông tin. Under 'Permissions', the role is set to 'Student'. In the 'Groups' section, 'Available groups' is empty, while 'Chosen groups' contains a single item.

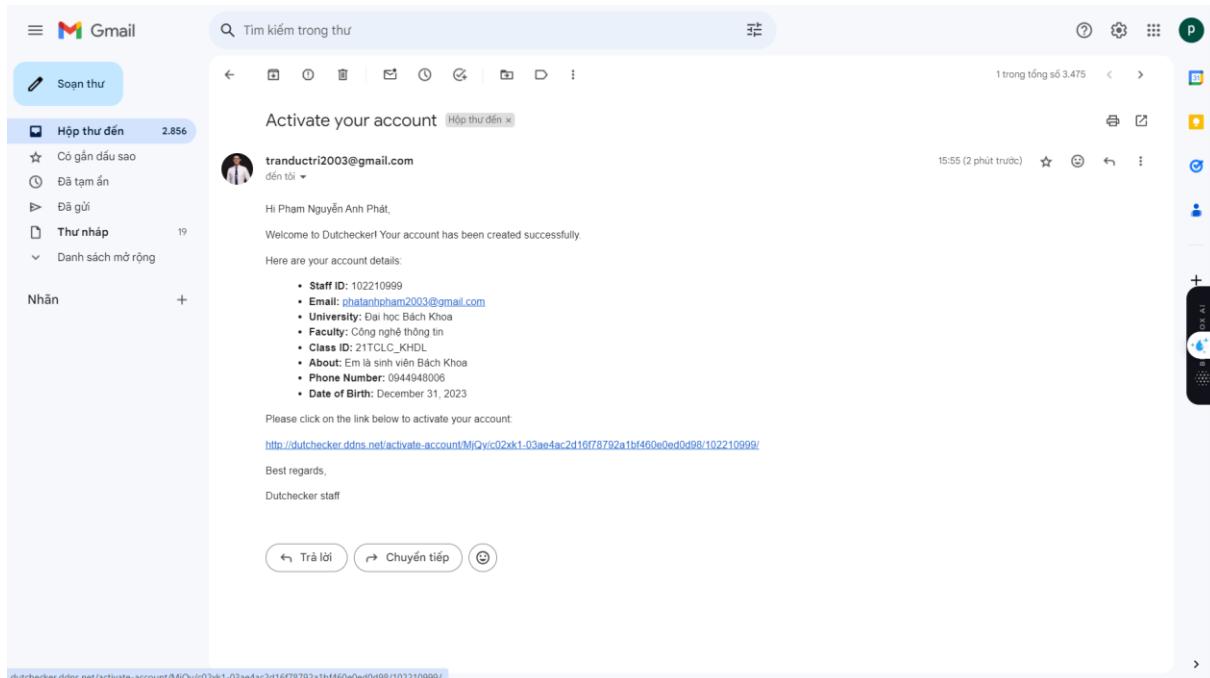
**Hình 93.**Tạo tài khoản và gửi mail kích hoạt tài khoản

The screenshot shows the Django Admin interface for 'New users'. A search bar at the top is empty. On the right, there's a 'FILTER' sidebar with 'By email address' and a list of email addresses. The main table lists 51 users, all of whom have 'IS ACTIVE' status set to red (inactive). The columns include: EMAIL ADDRESS, STAFF ID, FULL NAME, ROLE, IS ACTIVE, IS STAFF, and STATUS. The first user listed is 'phathanhpham2003@gmail.com' with Staff ID 102210999 and Full Name 'Phạm Nguyễn Anh Phát'.

**Hình 94.**Tài khoản được tạo nhưng chưa kích hoạt

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

## b. Kích hoạt tài khoản

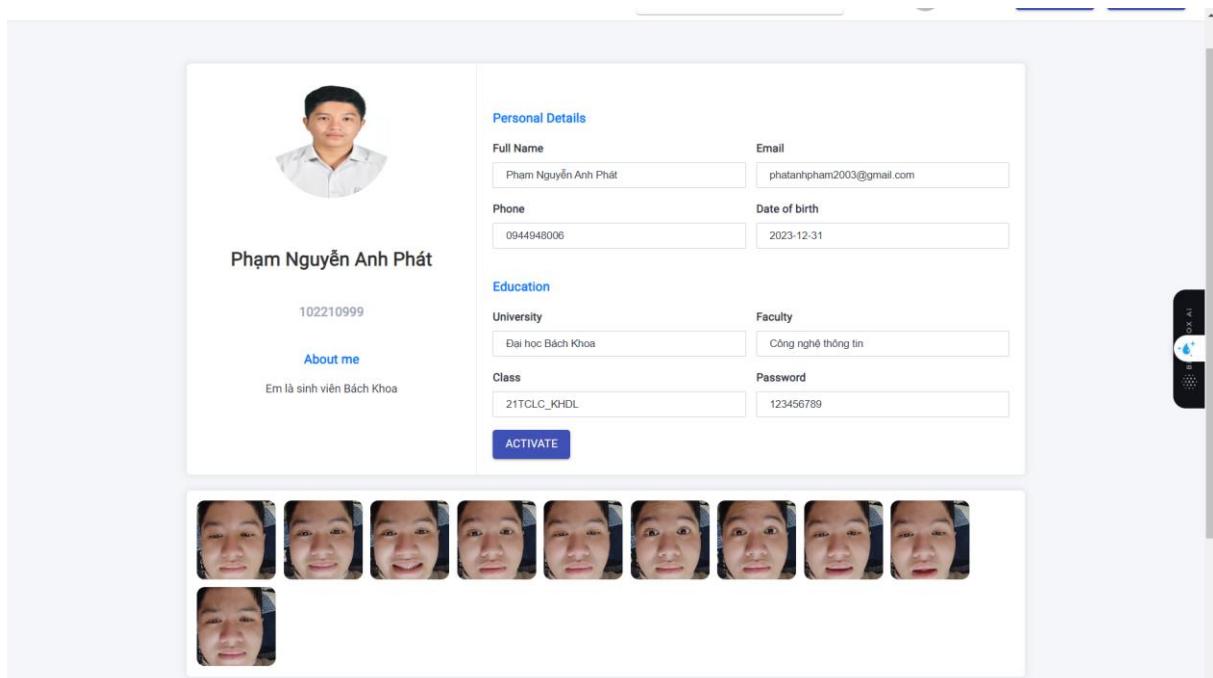


Hình 95. User Kiểm tra mail và truy cập link

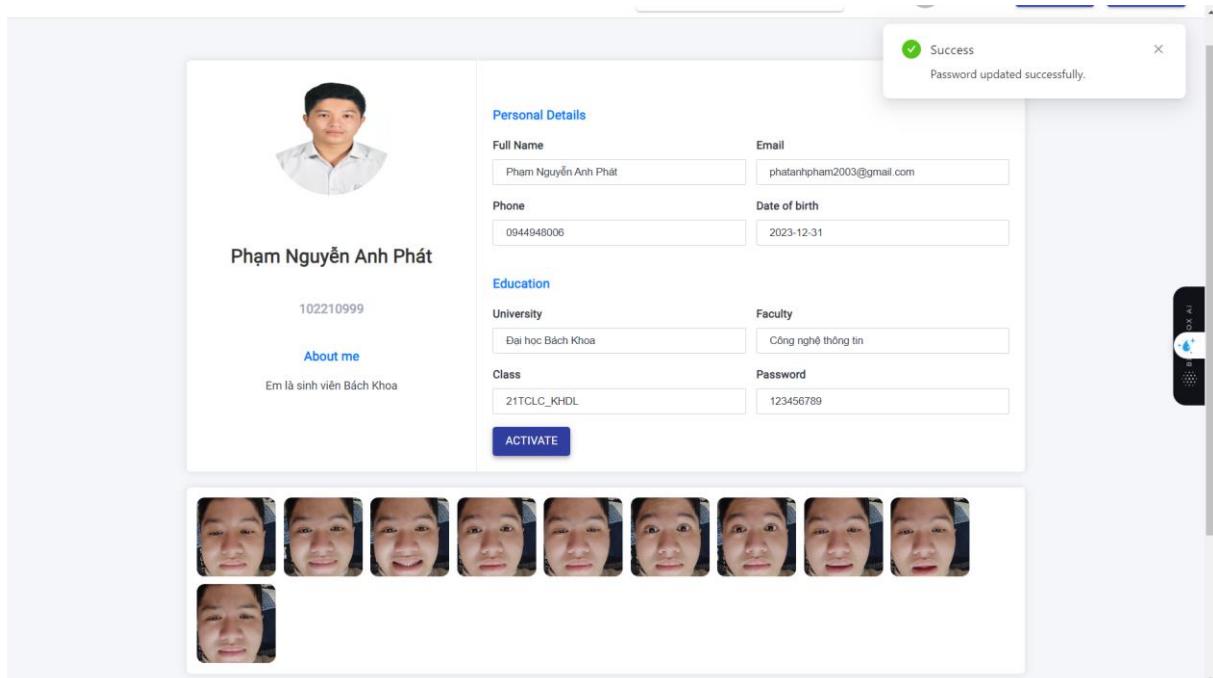
The screenshot shows the Dutchecker user profile page. On the left, there is a sidebar with tabs for "Lịch hôm nay", "Lịch sử điểm danh", and "Chat". The main area displays a user profile picture and personal information. The user's name is "Phạm Nguyễn Anh Phát" and their ID is "102210999". Below this, there is an "About me" section stating "Em là sinh viên Bách Khoa". To the right, there is a form titled "Personal Details" with fields for "Full Name" (filled with "Phạm Nguyễn Anh Phát") and "Email" (filled with "phatanhpham2003@gmail.com"). There is also a "Phone" field (filled with "0944948006") and a "Date of birth" field (filled with "2023-12-31"). Below this, there is an "Education" section with "University" (filled with "Đại học Bách Khoa") and "Faculty" (filled with "Công nghệ thông tin"). There is also a "Class" field (filled with "21TCLC\_KHDL") and a "Password" field (containing placeholder text "Mật khẩu mới mà bạn muốn thay đổi"). A blue "ACTIVATE" button is at the bottom of the form. At the bottom left of the main area, there is a "UPLOAD IMAGE" button.

Hình 96. Giao diện sao khi kích vào link kích hoạt tài khoản

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

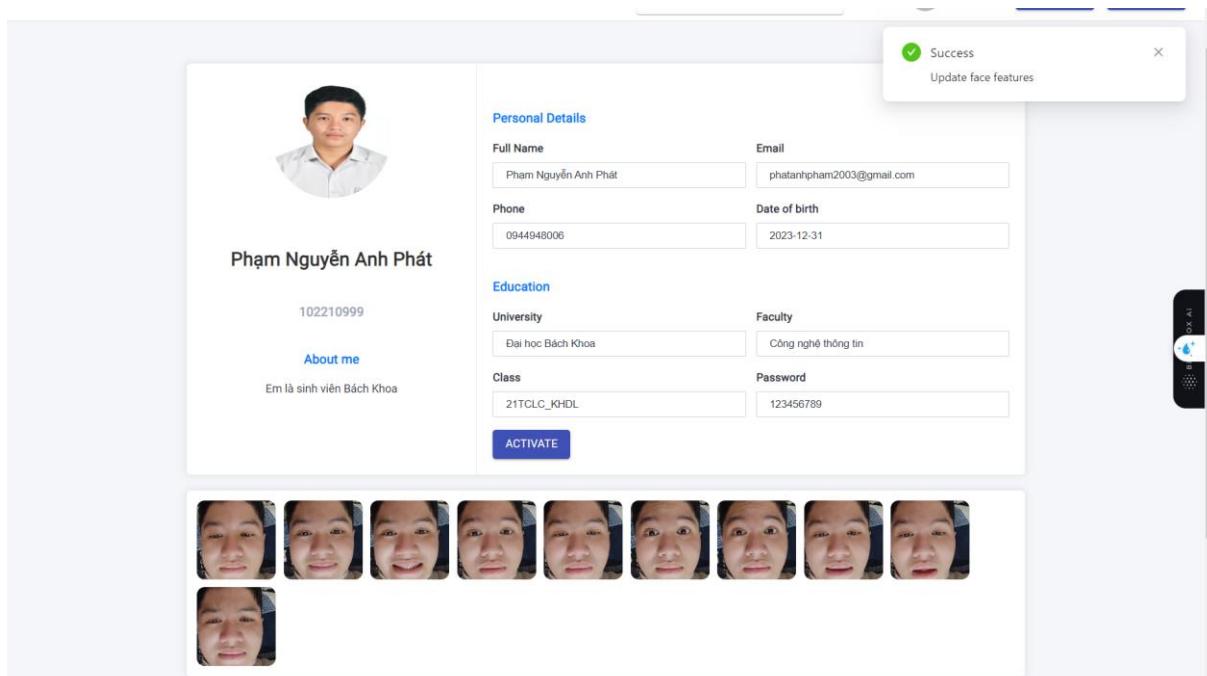


Hình 97. Cập nhật mật khẩu và up ảnh để nhận diện khuôn mặt



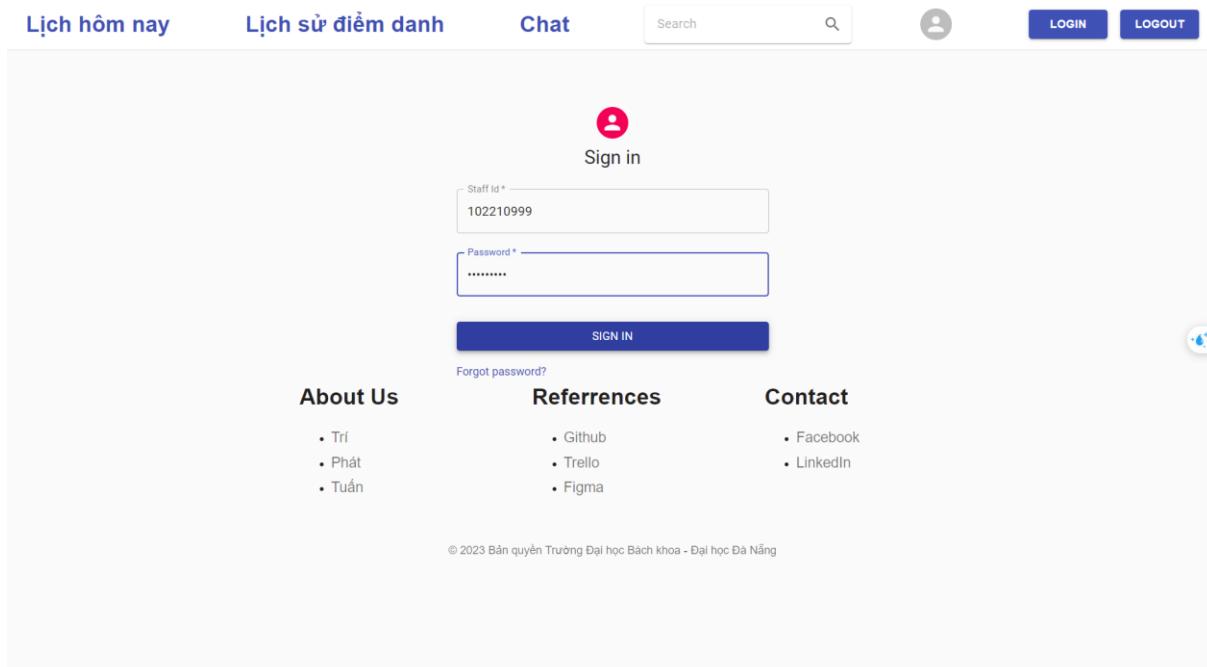
Hình 98. Cập nhật mật khẩu và kích hoạt tài khoản thành công

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



**Hình 99.**Cập nhật đặc trưng khuôn mặt thành công

### c. Đăng nhập:



**Hình 100.**Giao diện đăng nhập User

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a user profile page. At the top, there are navigation links: 'Lịch hôm nay', 'Lịch sử điểm danh', and 'Chat'. A search bar and user account information ('102210999') are also at the top. The main content area includes:

- Personal Details:** Fields for Full Name (Phạm Nguyễn Anh Phát), Email (phatanhpham2003@gmail.com), Phone (0944948006), and Date of birth (2023-12-31).
- Education:** Fields for University (Đại học Bách Khoa) and Faculty (Công nghệ thông tin). There is also a 'Class' field (21TCLC\_KHDL) and a 'Update Password' button.
- About me:** A short bio: 'Em là sinh viên Bách Khoa'.
- Table:** A table showing a course schedule:

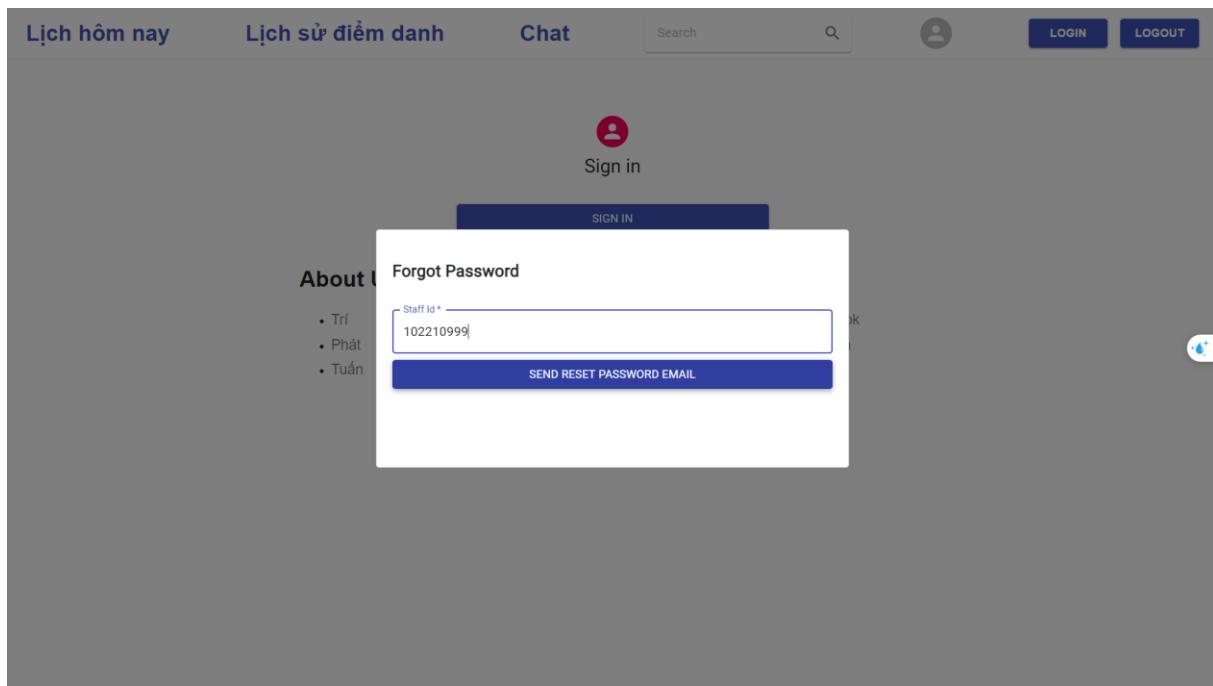
STT	Mã lớp học phần	Tên lớp học phần	Giảng viên	Thời khóa biểu
1	1023610.2310.21	Điện toán đám mây	Nguyễn Thế Xuân Ly	Chủ nhật 16:00:00 - 18:00:00 - E112

Hình 101. Giao diện trang cá nhân User

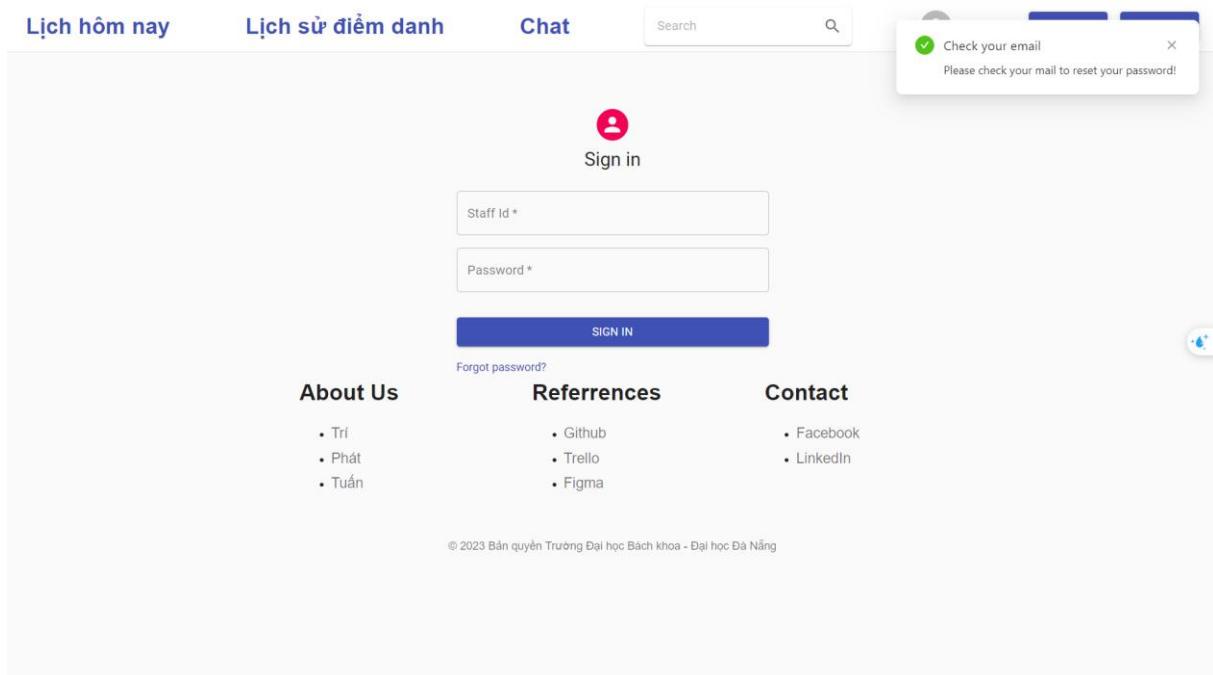
The screenshot shows a modal dialog box titled 'Update Password' overlaid on the user profile page. The dialog contains fields for 'Old Password' (\*\*\*\*\*), 'New Password' (\*\*\*\*\*), and 'Confirm New Password' (\*\*\*\*\*). A blue 'UPDATE' button is at the bottom. In the background, the user profile page is visible with the same layout as in Figure 101.

Hình 102. Nhập lại mật khẩu

## d. Quên mật khẩu

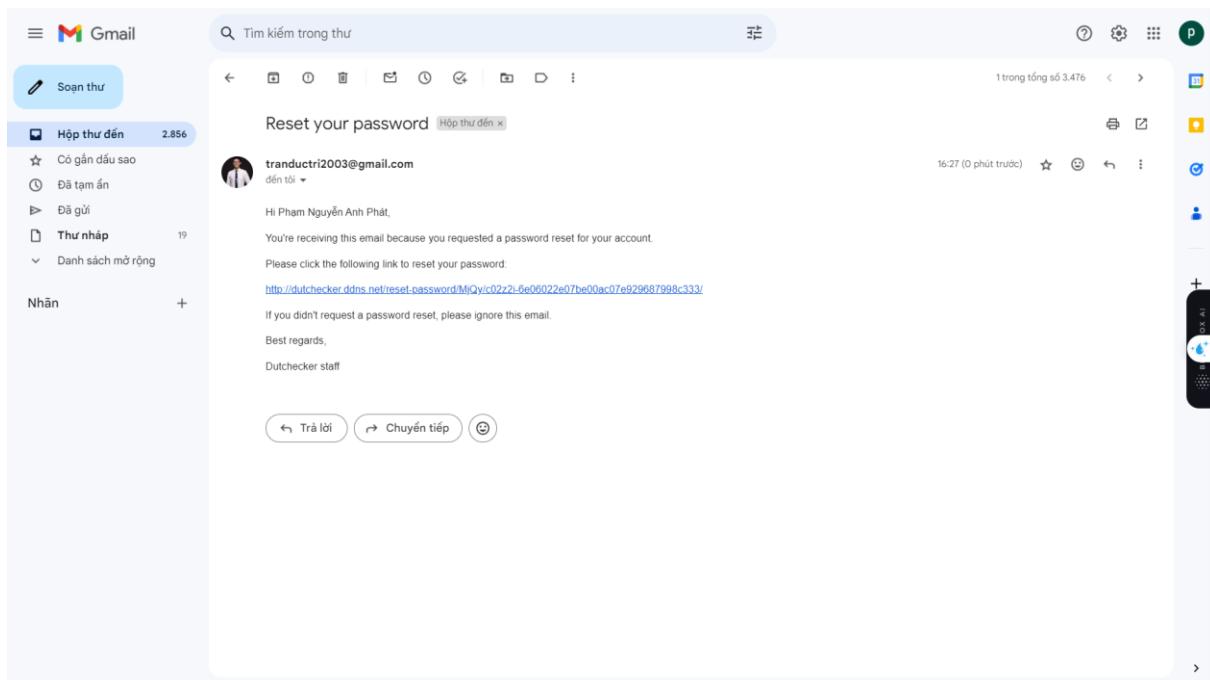


Hình 103. Nhập mã ID khi quên mật khẩu

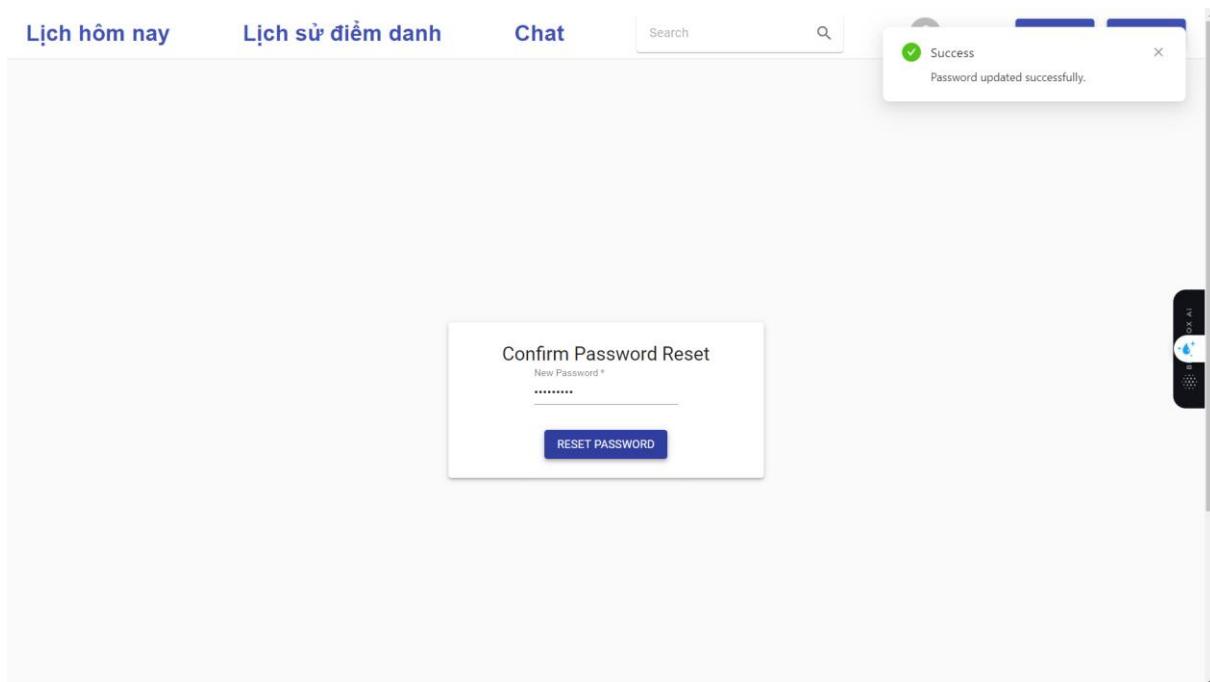


Hình 104. Mail cập nhật mật khẩu đã quên được gửi

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 105. Vào Mail quên mật khẩu và truy cập đường link



Hình 106. Cập nhật mật khẩu thành công

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

e. Tìm kiếm người dùng:

Lịch hôm nay    Lịch sử điểm danh    Chat

Linh 102210999 LOGIN LOGOUT

STT	Staff id	Họ và tên	Email	SDT	Lớp	Nhắn tin
1	<a href="#">102210299</a>	Trần Thị Huỳnh Linh	102210299@sv1.dut.udn.vn	0896454218	21TCLC_KHDL	<a href="#">Nhắn tin</a>
2	<a href="#">102210318</a>	Trần Hoàng Thé Linh	102210318@sv1.dut.udn.vn	0911756630	21TCLC_KHDL2	<a href="#">Nhắn tin</a>

**About Us**

- Trí
- Phát
- Tuấn

**References**

- Github
- Trello
- Figma

**Contact**

- Facebook
- LinkedIn

© 2023 Bản quyền Trường Đại học Bách khoa - Đại học Đà Nẵng

*Hình 107. Tìm kiếm User trên thanh search*

Lịch hôm nay    Lịch sử điểm danh    Chat

Linh 102210999 LOGIN LOGOUT

Trần Thị Huỳnh Linh  
102210299

**About me**  
Em là sinh viên Bách Khoa

**Personal Details**

Full Name	Email
Trần Thị Huỳnh Linh	102210299@sv1.dut.udn.vn
Phone	Date of birth
0896454218	2023-10-07

**Education**

University	Faculty
Bách Khoa Đà Nẵng	Công nghệ thông tin
Class	
21TCLC_KHDL	

STT	Mã lớp học phần	Tên lớp học phần	Giảng viên	Thời khóa biểu
1	1020353.2310.2	Lập trình mạng	Mai Văn Hà	Thứ năm 09:00:00 - 23:00:00 - H103
2	1021523.2310.21	Công nghệ Web	Mai Văn Hà	Thứ ba 09:00:00 - 23:38:00 - E401
3	1022470.2310.21	Xử lý tín hiệu số	Ninh Khánh Duy	Thứ ba 00:00:00 - 09:00:00 - E207

*Hình 108. Giao diện trang cá nhân người cần tìm*

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

### 3.2.2. Quản lý lịch học/dạy:

#### a. Lịch học trong ngày

The screenshot shows a web-based application for managing schedules. At the top, there are navigation links: 'Lịch hôm nay' (Today's Schedule), 'Lịch sử điểm danh' (Attendance History), and 'Chat'. On the right, there is a search bar, a user profile icon with the number '102210999', and 'LOGIN' and 'LOGOUT' buttons. Below the header, a title 'Class Today' is displayed with a pencil icon. A table lists the student's current class: 'STT' (1), 'Mã lớp học phần' (1023610.2310.21), 'Tên lớp học phần' (Điện toán đám mây), 'Giảng viên' (Nguyễn Thế Xuân Ly), and 'Thời khóa biểu' (Chủ nhật 16:00:00 - 18:00:00 - E112). Buttons for 'Diểm danh' (Check-in) and 'Nhân tin' (Share) are shown next to the last two columns. Below the table, there are three sections: 'About Us' (with members Trí, Phát, Tuấn), 'References' (with links to Github, Trello, Figma), and 'Contact' (with links to Facebook and LinkedIn). A copyright notice at the bottom states: '© 2023 Bản quyền Trường Đại học Bách khoa - Đại học Đà Nẵng'.

Hình 109. Lịch học hôm nay của sinh viên

This screenshot shows the same web application from the teacher's perspective. The top navigation and user interface are identical. The table below shows the teacher's class: 'STT' (1), 'Mã lớp học phần' (1023610.2310.21), 'Tên lớp học phần' (Điện toán đám mây), 'Giảng viên' (Nguyễn Thế Xuân Ly), and 'Thời khóa biểu' (Chủ nhật 16:00:00 - 18:00:00 - E112). A green 'Xem' (View) button is visible next to the last column. The 'About Us', 'References', and 'Contact' sections are also present. The copyright notice at the bottom is identical to the student version.

Hình 110. Lịch dạy của giáo viên sau khi đăng nhập

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a web-based application titled "Class Today". At the top, there are tabs for "Lịch hôm nay" (Schedule today), "Lịch sử điểm danh" (Attendance history), and "Chat". A search bar and a user profile for "AD1" are also at the top. Below the header is a table showing a single class entry:

STT	Mã lớp học phần	Tên lớp học phần	Giảng viên	Thời khóa biểu	Trạng thái điểm danh
1	1023610.2310.21	Điện toán đám mây	Nguyễn Thế Xuân Ly	Chủ nhật 16:00:00 - 18:00:00 - E112	Xem

Below the table, there are three navigation links: "About Us", "References", and "Contact". Under "About Us", there is a list of names: Trí, Phát, Tuấn. Under "References", there are links to GitHub, Trello, and Figma. Under "Contact", there are links to Facebook and LinkedIn.

At the bottom of the page, a copyright notice reads: © 2023 Bản quyền Trường Đại học Bách khoa - Đại học Đà Nẵng.

*Hình 111. xem lịch dạy hoặc học dưới vai trò admin*

b. Lịch học của một người:

The screenshot shows a student profile for "Trần Đức Trí" with ID "102210096". The profile includes sections for "Education" (University: Bách Khoa Đà Nẵng, Faculty: Công nghệ thông tin, Class: 21TCLC\_KHDL) and "About me" (Em là sinh viên Bách Khoa). There is a "Update Password" button.

Below the profile, there is a table showing the course schedule:

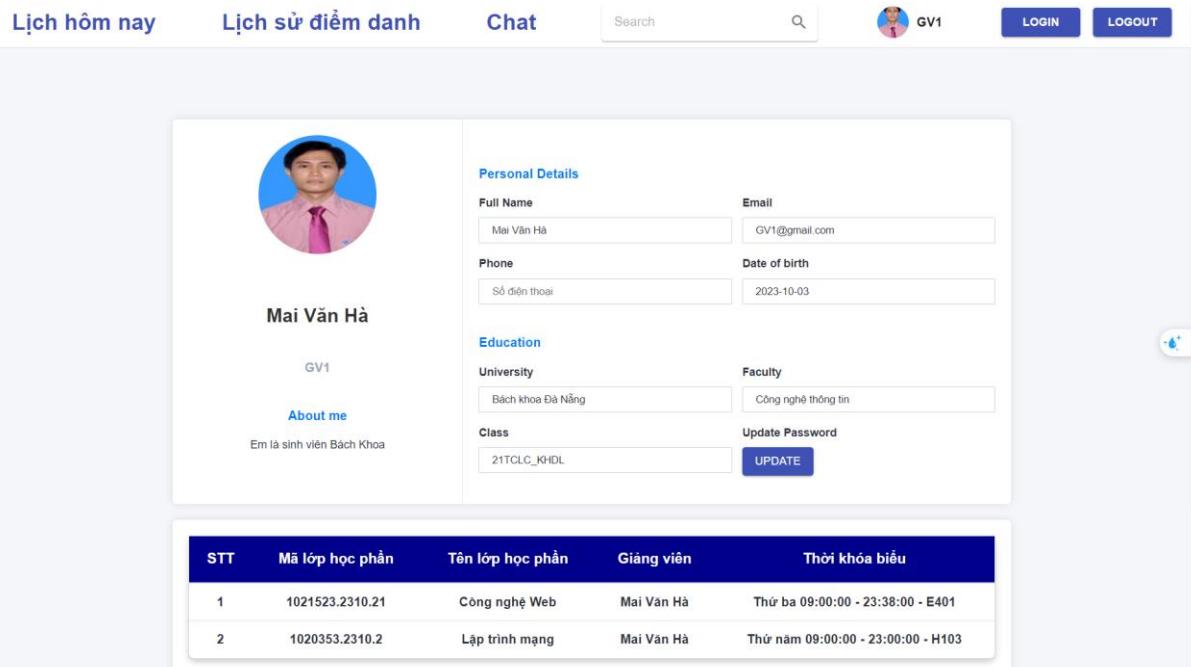
STT	Mã lớp học phần	Tên lớp học phần	Giảng viên	Thời khóa biểu
1	1020353.2310.2	Lập trình mạng	Mai Văn Hà	Thứ năm 09:00:00 - 23:00:00 - H103
2	1021523.2310.21	Công nghệ Web	Mai Văn Hà	Thứ ba 09:00:00 - 23:30:00 - E401
3	1022470.2310.21	Xử lý tín hiệu số	Ninh Khánh Duy	Thứ ba 00:00:00 - 09:00:00 - E207
4	1020113.2310.2	Chương trình dịch	Nguyễn Thị Minh Hy	Thứ năm 07:00:00 - 09:00:00 - H103
5	1023610.2310.21	Điện toán đám mây	Nguyễn Thế Xuân Ly	Chủ nhật 16:00:00 - 18:00:00 - E112
6	1023220.2310.21	Toán ứng dụng công nghệ thông tin	Nguyễn Văn Hiệu	Thứ sáu 00:00:00 - 23:59:00 - H304
7	1023830.2310.2	PBL 4: DA HĐH và mạng MT	Đặng Thiên Bình	Thứ bảy 00:00:01 - 13:00:00 - E204B

Below the table, there are three navigation links: "About Us", "References", and "Contact".

*Hình 112.Lịch học của sinh viên tại trang cá nhân*

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Lịch hôm nay    Lịch sử điểm danh    Chat    Search     GV1    LOGIN    LOGOUT



**Personal Details**

Full Name	Email
Mai Văn Hà	GV1@gmail.com
Phone	Date of birth
Số điện thoại	2023-10-03

**Education**

University	Faculty
Bách khoa Đà Nẵng	Công nghệ thông tin

**About me**

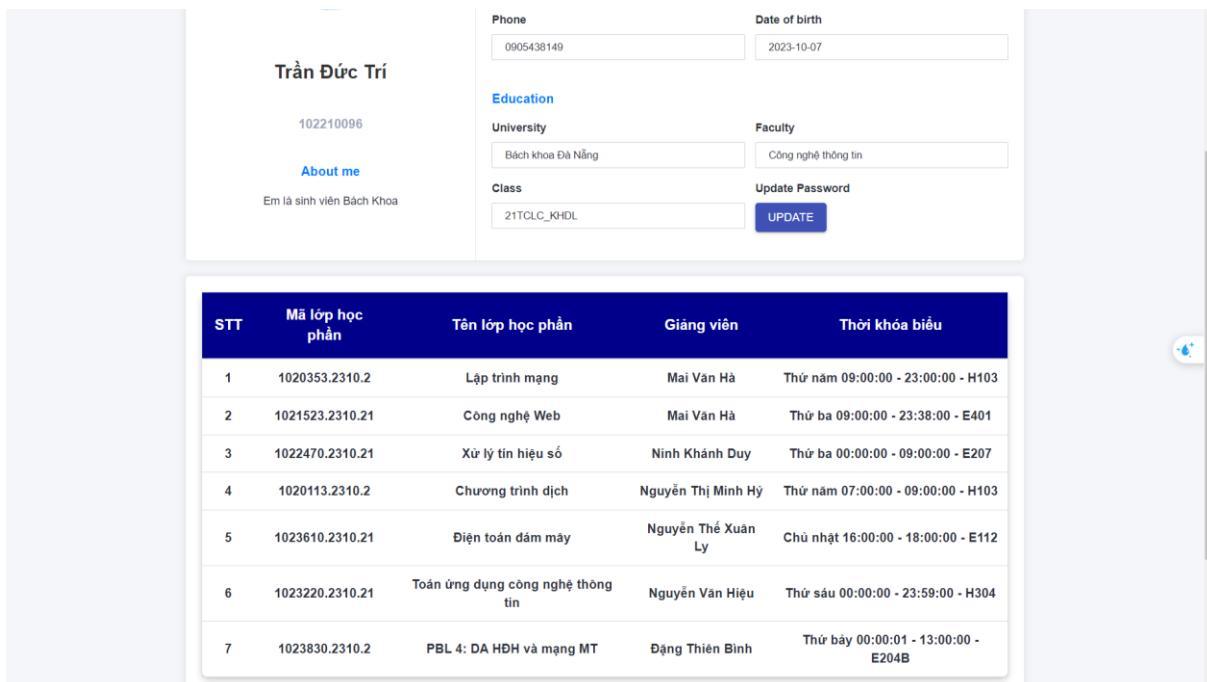
Em là sinh viên Bách Khoa

**Class** 21TCLC\_KHDL    **Update Password**    **UPDATE**

**STT**    **Mã lớp học phần**    **Tên lớp học phần**    **Giảng viên**    **Thời khóa biểu**

1	1021523.2310.21	Công nghệ Web	Mai Văn Hà	Thứ ba 09:00:00 - 23:38:00 - E401
2	1020353.2310.2	Lập trình mạng	Mai Văn Hà	Thứ năm 09:00:00 - 23:00:00 - H103

Hình 113. Lịch dạy của giáo viên tại trang cá nhân



**Trần Đức Trí**

102210096

**About me**

Em là sinh viên Bách Khoa

**Phone** 0905438149    **Date of birth** 2023-10-07

**Education**

University	Faculty
Bách khoa Đà Nẵng	Công nghệ thông tin

**Class** 21TCLC\_KHDL    **Update Password**    **UPDATE**

**STT**    **Mã lớp học phần**    **Tên lớp học phần**    **Giảng viên**    **Thời khóa biểu**

1	1020353.2310.2	Lập trình mạng	Mai Văn Hà	Thứ năm 09:00:00 - 23:00:00 - H103
2	1021523.2310.21	Công nghệ Web	Mai Văn Hà	Thứ ba 09:00:00 - 23:38:00 - E401
3	1022470.2310.21	Xử lý tín hiệu số	Ninh Khánh Duy	Thứ ba 00:00:00 - 09:00:00 - E207
4	1020113.2310.2	Chương trình dịch	Nguyễn Thị Minh Hy	Thứ năm 07:00:00 - 09:00:00 - H103
5	1023610.2310.21	Điện toán đám mây	Nguyễn Thế Xuân Ly	Chủ nhật 16:00:00 - 18:00:00 - E112
6	1023220.2310.21	Toán ứng dụng công nghệ thông tin	Nguyễn Văn Hiệu	Thứ sáu 00:00:00 - 23:59:00 - H304
7	1023830.2310.2	PBL 4: DA HĐH và mạng MT	Đặng Thiên Bình	Thứ bảy 00:00:01 - 13:00:00 - E204B

Hình 114. Xem lịch dạy hoặc học của user bất kỳ dưới vai trò admin

### 3.2.3. Quản lý điểm danh:

#### a. Điểm danh

The screenshot shows a web-based application for managing student attendance. At the top, there are tabs for 'Lịch hôm nay' (Today's schedule), 'Lịch sử điểm danh' (Attendance history), and 'Chat'. The user is logged in as student number 102210096. On the right, there is a sidebar with a message 'Xin giáo viên nghỉ' (Ask teacher to take a break) and a 'Nhắn tin' (Send message) button. The main content area displays a table of students with their attendance status. The columns are: STT (Student ID), Mã lớp học (Class ID), Avatar, Mã số sinh viên (Student ID), Họ và tên sinh viên (Student Name), Môn học (Subject), Trạng thái điểm danh (Attendance status), and Ghi chú (Remarks). The attendance status is indicated by a red 'X' or a green 'Điểm danh' (Present). Buttons at the bottom of the table include 'Bật camera điểm danh' (Turn on camera for attendance), 'Kiểm tra vị trí' (Check location), and 'Điểm danh' (Present). The table data is as follows:

STT	Mã lớp học	Avatar	Mã số sinh viên	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
1	1023610.231		102210089	Hồ Quốc Thiên Anh	Điện toán đám mây	X	-
			102210092	Phạm Gia Hùng	Điện toán đám mây	X	-
			102210095	Phan Minh Nhật	Điện toán đám mây	X	-
			102210096	Trần Đức Trí	Điện toán đám mây	X	-
			102210097	Lê Anh Tuấn	Điện toán đám mây	X	-
			102210289	Phan Chí Cương	Điện toán đám mây	X	-

Hình 115.Xem trạng thái điểm danh hiện tại

The screenshot shows the same web-based application for managing student attendance. A browser dialog box from 'dutchecker.ddns.net' is displayed, asking if the user wants to 'Know your location'. It has 'Allow' and 'Block' buttons. The main content area of the application shows the same table of student attendance as in Figure 115. The table data is identical to the previous screenshot.

Hình 116.Cho phép mở vị trí

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

Lịch hôm nay    Lịch sử điểm danh    Chat    Search

Avatar	Mã số sinh viên	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
	102210089	Hồ Quốc Thiên Anh	Điện toán đám mây	X	
	102210092	Phạm Gia Hùng	Điện toán đám mây	X	
	102210095	Phan Minh Nhật	Điện toán đám mây	X	
	102210096	Trần Đức Trí	Điện toán đám mây	X	
	102210097	Lê Anh Tuấn	Điện toán đám mây	X	
	102210289	Phan Chí Cường	Điện toán đám mây	X	

Bật camera điểm danh        Diểm danh

Nhắn tin

Current Location  
Latitude: 16.070847, Longitude: 108.154738

Distance Check  
The distance is smaller than 1km. Good.

25

Hình 117. Kiểm tra vị trí thành công

dutchecker.ddns.net wants to  
Use your camera

Allow    Block

Lịch sử điểm danh    Chat    Search

Avatar	Mã số	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
	102210089	Hồ Quốc Thiên Anh	Điện toán đám mây	X	
	102210092	Phạm Gia Hùng	Điện toán đám mây	X	
	102210095	Phan Minh Nhật	Điện toán đám mây	X	
	102210096	Trần Đức Trí	Điện toán đám mây	X	
	102210097	Lê Anh Tuấn	Điện toán đám mây	X	
	102210289	Phan Chí Cường	Điện toán đám mây	X	

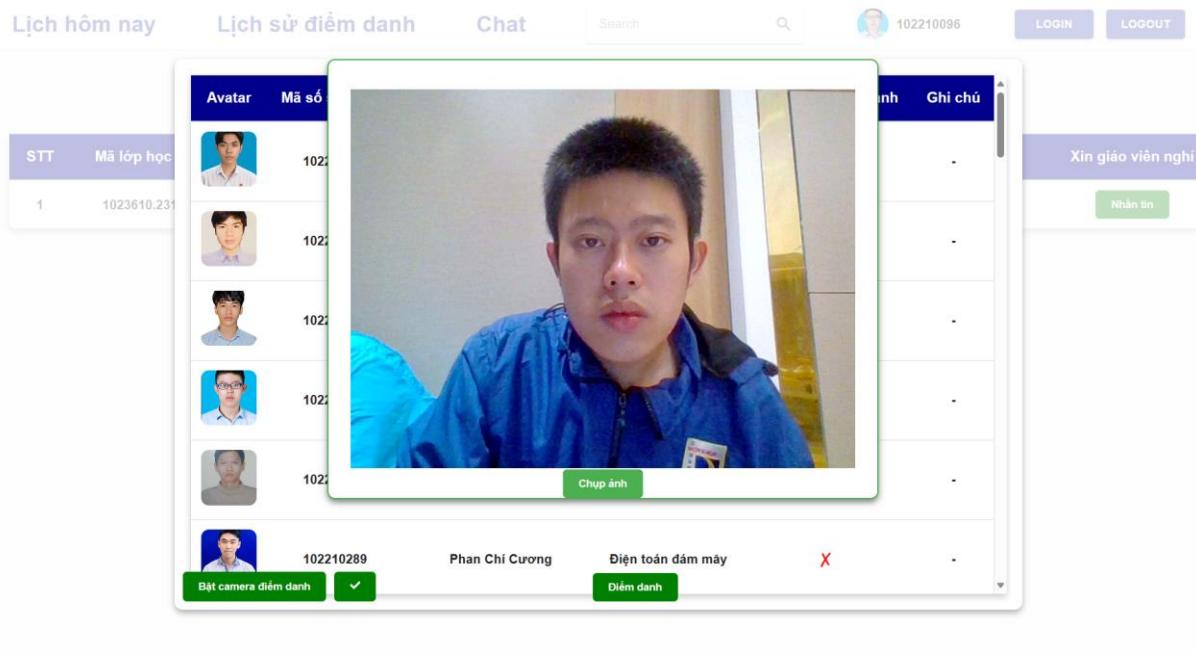
Bật camera điểm danh        Chụp ảnh

Xin giáo viên nghỉ

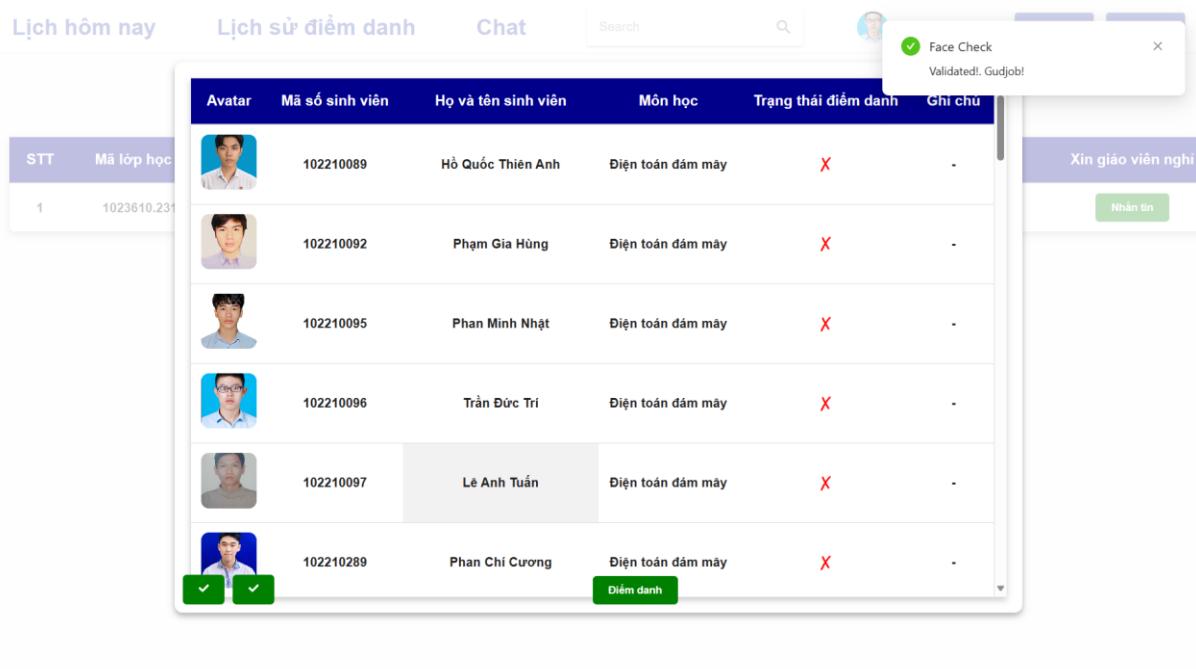
Nhắn tin

Hình 118. Cho phép mở camera

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



Hình 119. Chụp ảnh khuôn mặt để điểm danh



Hình 120. Kiểm tra khuôn mặt thành công

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows a user interface for managing student attendance. At the top, there are tabs for 'Lịch hôm nay' (Today's schedule), 'Lịch sử điểm danh' (Attendance history), and 'Chat'. A search bar and a notification bubble are also present. The main area displays a table of students with columns for 'Avatar', 'Mã số sinh viên' (Student ID), 'Họ và tên sinh viên' (Student Name), 'Môn học' (Subject), 'Trạng thái điểm danh' (Attendance status), and 'Ghi chú' (Notes). The 'Trạng thái' column contains red 'X' marks for most students, except for the first one which has a green checkmark. A message box at the top right says 'Trần Đức Trí đã điểm danh' and 'Congratulations!!!'. On the right side, there is a sidebar with a button labeled 'Nhắn tin' (Send message).

STT	Mã lớp học	Avatar	Mã số sinh viên	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
1	1023610.231		102210092	Phạm Gia Hùng	Điện toán đám mây	X	
			102210095	Phan Minh Nhật	Điện toán đám mây	X	
			102210096	Trần Đức Trí	Điện toán đám mây	✓	
			102210097	Lê Anh Tuấn	Điện toán đám mây	X	
			102210289	Phan Chí Cương	Điện toán đám mây	X	
			102210293	Nguyễn Văn Tiến Dũng	Điện toán đám mây	X	

Hình 121. Điểm danh thành công

This screenshot shows the same application interface after an update. The table now lists all six students with red 'X' marks in the 'Trạng thái' column, indicating they have not yet checked in. The message box at the top right still shows 'Trần Đức Trí đã điểm danh' and 'Congratulations!!!'. The sidebar on the right includes a 'Nhắn tin' button.

STT	Mã lớp học	Avatar	Mã số sinh viên	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
1	1023610.231		102210089	Hồ Quốc Thiên Anh	Điện toán đám mây	X	
			102210092	Phạm Gia Hùng	Điện toán đám mây	X	
			102210095	Phan Minh Nhật	Điện toán đám mây	X	
			102210096	Trần Đức Trí	Điện toán đám mây	✓	
			102210097	Lê Anh Tuấn	Điện toán đám mây	X	
			102210289	Phan Chí Cương	Điện toán đám mây	X	

Hình 122. Cập nhật trạng thái đã điểm danh ở máy sinh viên

## PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

STT	Mã lớp	Avatar	Mã số sinh viên	Họ và tên sinh viên	Môn học	Trạng thái điểm danh	Ghi chú
1	1023610		102210089	Hồ Quốc Thiên Anh	Điện toán đám mây	X	-
			102210092	Phạm Gia Hùng	Điện toán đám mây	X	-
			102210095	Phan Minh Nhật	Điện toán đám mây	X	-
			102210096	Trần Đức Trí	Điện toán đám mây	✓	-
			102210097	Lê Anh Tuấn	Điện toán đám mây	X	-
			102210289	Phan Chí Cường	Điện toán đám mây	X	-

Hình 123. Cập nhật trạng thái điểm danh ở máy giáo viên

### b. Lịch sử điểm danh

Subject	Student	Attendance Time	Attendance Date	Status	Note
Điện toán đám mây (1023610.2310.21)	Trần Đức Trí (102210096)	17:04:30	2023-12-31	✓	Đã điểm danh
PBL 4: DA HDH và mạng MT (1023830.2310.2)	Trần Đức Trí (102210096)	01:21:41	2023-12-30	✓	Đã điểm danh
Toán ứng dụng công nghệ thông tin (1023220.2310.21)	Trần Đức Trí (102210096)	01:32:29	2023-12-29	✓	Đã điểm danh
Toán ứng dụng công nghệ thông tin (1023220.2310.21)	Trần Đức Trí (102210096)	23:21:56	2023-11-10	✓	Đã điểm danh
PBL 4: DA HDH và mạng MT (1023830.2310.2)	Trần Đức Trí (102210096)	01:33:58	2023-11-04	✓	Đã điểm danh
Toán ứng dụng công nghệ thông tin (1023220.2310.21)	Trần Đức Trí (102210096)	23:53:22	2023-11-03	✓	Đã điểm danh
Công nghệ Web (1021523.2310.21)	Trần Đức Trí (102210096)	23:24:34	2023-10-31	✓	Đã điểm danh
Toán ứng dụng công nghệ thông tin (1023220.2310.21)	Trần Đức Trí (102210096)	18:30:46	2023-10-27	✓	Đã điểm danh
Lập trình mạng (1020353.2310.2)	Trần Đức Trí (102210096)	22:10:57	2023-10-26	✓	Đã điểm danh

Hình 124. Lịch sử điểm danh vừa được cập nhật

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

The screenshot shows the 'Attendance History' section of the application. At the top, there are navigation links: 'Lịch hôm nay', 'Lịch sử điểm danh', 'Chat', and a search bar. On the right, there is a user profile for 'GV3' and buttons for 'LOGIN' and 'LOGOUT'. Below the header, the title 'Attendance History' is displayed with a small icon. There are three dropdown filters: 'Select Subject', 'Select Status', and 'Select Date', followed by a 'FILTER' button. A table lists one attendance record:

Subject	Student	Attendance Time	Attendance Date	Status	Note
Điện toán đám mây (1023610.2310.21)	Trần Đức Trí (102210096)	17:04:30	2023-12-31	✓	Đã điểm danh

Below the table, there are three footer sections: 'About Us' (with members Trí, Phát, Tuấn), 'References' (Github, Trello, Figma), and 'Contact' (Facebook, LinkedIn). A copyright notice at the bottom states: © 2023 Bản quyền Trường Đại học Bách khoa - Đại học Đà Nẵng.

Hình 125. Lịch sử điểm danh xem tại máy giáo viên

The screenshot shows the 'Attendance History' section of the application. The layout is identical to the teacher's view, with navigation links, a search bar, and user profile for 'AD1'. The table shows the same attendance record as the teacher's view.

Subject	Student	Attendance Time	Attendance Date	Status	Note
Điện toán đám mây (1023610.2310.21)	Trần Đức Trí (102210096)	17:04:30	2023-12-31	✓	Đã điểm danh

The 'About Us', 'References', and 'Contact' sections are also present at the bottom.

Hình 126. Lịch sử điểm danh xem tại máy admin

### 3.2.4. Trò chuyện:

#### a. Trạng thái trực tuyến

The screenshot shows a web interface for managing classes. At the top, there are tabs for 'Lịch hôm nay' (Today's schedule), 'Lịch sử điểm danh' (Attendance history), and 'Chat'. A search bar is also present. A success message box is displayed: 'Sign in successfully' and 'Welcome Nguyễn Thé Xuân Ly!!!'. Below this, a section titled 'Class Today' lists a single class entry: STT 1, Mã lớp học phần 1023610.2310.21, Tên lớp học phần Điện toán đám mây, Giảng viên Nguyễn Thé Xuân Ly, Thời khóa biểu Chủ nhật 16:00:00 - 18:00:00 - E112, and a green 'Điểm danh' button. At the bottom, there are three footer sections: 'About Us' (Trí, Phát, Tuấn), 'References' (Github, Trello, Figma), and 'Contact' (Facebook, LinkedIn).

**Hình 127.** Đăng nhập với vai trò giáo viên

The screenshot shows a web interface for a chat hall. At the top, there are tabs for 'Lịch hôm nay' (Today's schedule), 'Lịch sử điểm danh' (Attendance history), and 'Chat'. A user profile is shown: 102210096, male, with 'LOGIN' and 'LOGOUT' buttons. Below this, a section titled 'Welcome to the Chat Hall' displays a list of users and their chat rooms. On the left, a sidebar shows six student profiles: Nguyễn Văn Hiệu, Ninh Khánh Duy, Nguyễn Thé Xuân Ly, Nguyễn Thị Minh Hỷ, Mai Văn Hà, and Đặng Thiên Bình. To the right, six chat room boxes are listed: 'Mai Văn Hà and Trần Đức Trí' (Room ID: 8), 'Lê Anh Tuấn and Trần Đức Trí' (Room ID: 10), 'Phạm Nguyễn Anh Phát and Trần Đức Trí' (Room ID: 15), 'Nguyễn Thé Xuân Ly and Trần Đức Trí' (Room ID: 16), 'Trần Đức Trí and Đặng Thiên Bình' (Room ID: 19), and 'Phòng Chat Chung' (Room ID: 5). At the bottom, there are three footer sections: 'About Us' (Trí, Phát), 'References' (Github, Trello), and 'Contact' (Facebook, LinkedIn).

**Hình 128.** Trạng thái trực tuyến của giáo viên tại trang sinh viên

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH

## b. Tạo phòng chat

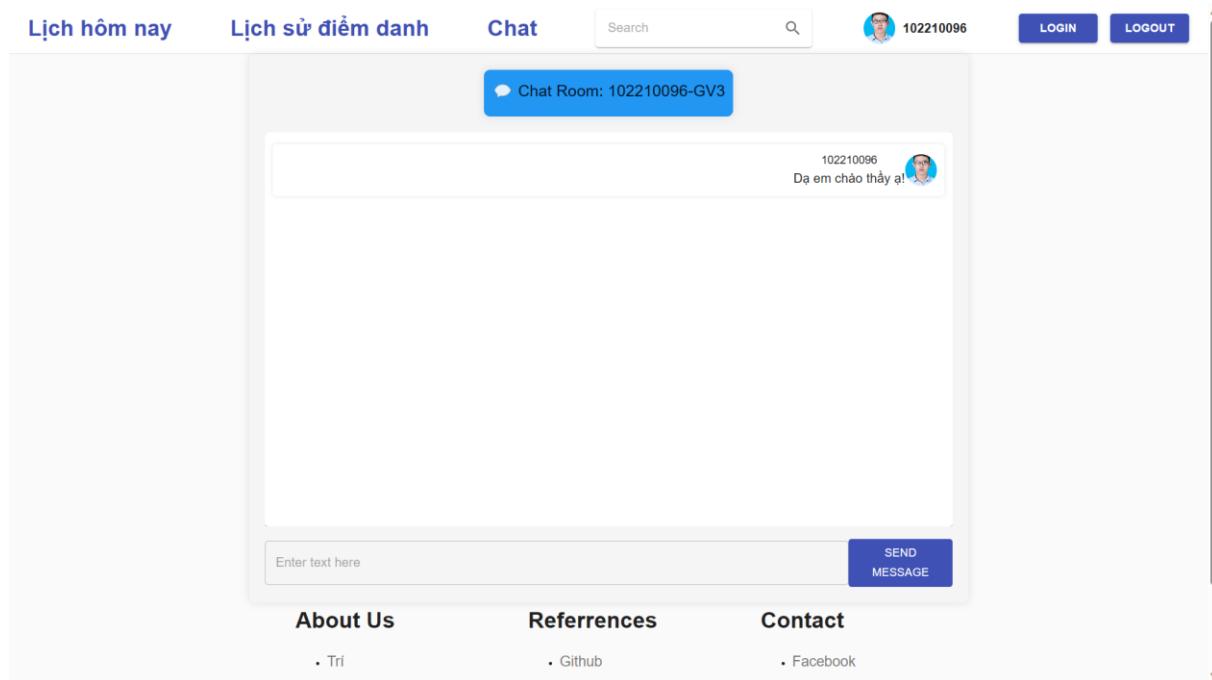
The screenshot shows a user interface for managing classes and creating rooms. At the top, there are tabs for 'Lịch hôm nay' (Today's schedule), 'Lịch sử điểm danh' (Attendance history), and 'Chat'. A search bar is also present. A success message 'Room Created' with the subtext 'Room created successfully!' is displayed. Below this, a section titled 'Class Today' lists a single class entry: '1 1023610.2310.21 Điện toán đám mây Nguyễn Thế Xuân Ly Chủ nhật 16:00:00 - 18:00:00 - E112'. Buttons for 'Điểm danh' (Check-in) and 'Nhận tin' (Receive notifications) are shown. At the bottom, there are sections for 'About Us', 'References', and 'Contact' with respective links.

Hình 129. Sinh viên tạo phòng chat với giáo viên

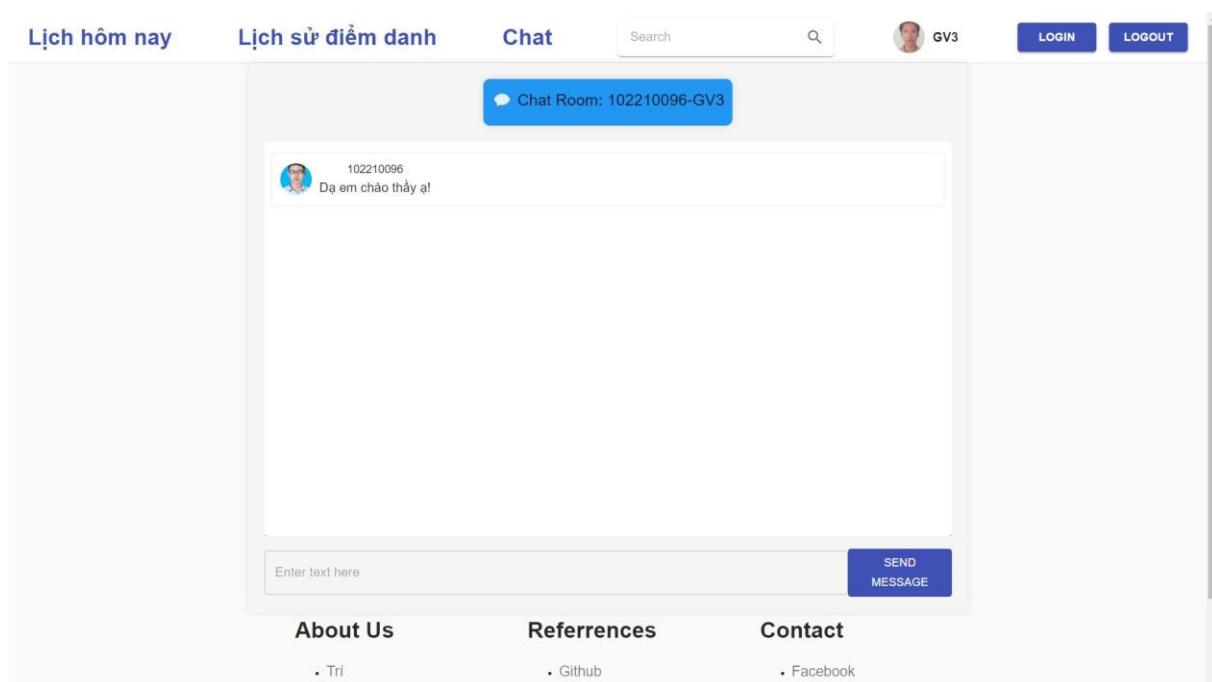
The screenshot shows the 'Welcome to the Chat Hall' section. It displays a list of available rooms for users to join. On the left, a sidebar shows user profiles: Nguyễn Văn Hiệu, Ninh Khánh Duy, Nguyễn Thị Minh Hỷ, Đặng Thiên Bình, Mai Văn Hà, and Nguyễn Thế Xuân Ly. In the center, several room cards are listed: 'Mai Văn Hà and Trần Đức Trí' (Room ID: 8), 'Lê Anh Tuấn and Trần Đức Trí' (Room ID: 10), 'Phạm Nguyễn Anh Phát and Trần Đức Trí' (Room ID: 15), 'Trần Đức Trí and Đặng Thiên Bình' (Room ID: 19), 'Nguyễn Thế Xuân Ly and Trần Đức Trí' (Room ID: 21), and 'Phòng Chat Chung' (Room ID: 5). At the bottom, there are sections for 'About Us', 'References', and 'Contact' with respective links.

Hình 130. Các phòng chat hiện tại của User

# PBL4: DỰ ÁN HỆ ĐIỀU HÀNH & MẠNG MÁY TÍNH



**Hình 131.** Giao diện phòng chat từ phía sinh viên



**Hình 132.** Giao diện từ phía giáo viên

## KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 1. Kết luận:

Trong quá trình phát triển dự án xây dựng website hỗ trợ điểm danh bằng nhận diện khuôn mặt, chúng tôi đã tích hợp **connection pool** và cân bằng tải nhằm cải thiện hiệu suất của hệ thống. Sự ứng dụng linh hoạt của connection pool giúp quản lý tối ưu các kết nối đến cơ sở dữ liệu, giảm thiểu độ trễ và tối ưu hóa tài nguyên, đồng thời cân bằng tải giúp phân phối công việc đều đặn trên các máy chủ, ngăn chặn hiện tượng quá tải và đảm bảo ổn định hệ thống.

Tuy nhiên, dù đã có những cải tiến về hiệu suất, nhận diện khuôn mặt vẫn đối diện với thách thức về độ chính xác. Qua quá trình thử nghiệm, chúng tôi nhận thấy rằng còn một số vấn đề cần được cải thiện để nâng cao khả năng nhận diện. Các nguyên nhân có thể bao gồm đa dạng về ánh sáng, góc độ chụp, hay đặc điểm đặc trưng của từng người dùng. Để giải quyết vấn đề này, chúng tôi đề xuất một số hướng phát triển tiếp theo.

### 2. Hướng phát triển:

**Tối ưu hóa Thuật toán Nhận diện:** Nghiên cứu và áp dụng các thuật toán nhận diện khuôn mặt tiên tiến hơn để cải thiện độ chính xác, có thể sử dụng các mô hình máy học mạnh mẽ hoặc mạng nơ-ron sâu (deep learning).

**Tích Hợp Giải Pháp Thiết Bị Nâng Cao:** Nếu khả năng xử lý trên Raspberry Pi 4 không đáp ứng đủ, xem xét tích hợp các thiết bị phần cứng mạnh mẽ hơn để tăng cường khả năng tính toán cho quá trình nhận diện.

**Cải Thiện Giao Diện Người Dùng:** Tối ưu hóa giao diện người dùng để làm cho quá trình sử dụng trở nên thân thiện hơn và thuận tiện hơn cho người dùng cuối.

**Phát Triển Ứng Dụng Di Động:** Tích hợp chức năng điểm danh thông qua ứng dụng di động, giúp sinh viên và nhân viên có thể thực hiện điểm danh mọi lúc, mọi nơi một cách thuận tiện. Điều này có thể bao gồm cả hỗ trợ cho các hệ điều hành di động phổ biến như iOS và Android.

**Bảo Mật và Tuân Thủ Quy định:** Tăng cường bảo mật hệ thống và đảm bảo tuân thủ quy định về quyền riêng tư và an ninh thông tin, đặc biệt là khi xử lý dữ liệu nhận diện khuôn mặt.

## TÀI LIỆU THAM KHẢO

- [1] Mai Văn Hà, *Bài giảng Lập Trình Mạng*, Công nghệ Web Khoa Công Nghệ Thông Tin – Trường Đại học Bách Khoa – Đại học Đà Nẵng.
- [2] Nguyễn Thé Xuân Ly, *Bài giảng Điện toán đám mây*, Khoa Công Nghệ Thông Tin – Trường Đại học Bách Khoa – Đại học Đà Nẵng.
- [3] paperswithcode.com, *Face Verification with model VGG-16*, 7/11/2023.
- [4] www.datacamp.com, Vector Databases, 5/12/2023.
- [5] azdigi.com, *Set HTTP to HTTPS auto-redirect*, 20/12/2023.
- [6] karammajdi7, Defeat the Traffic Monster: An Nginx Load Balancing Odyssey for Django Apps, 24/12/2023.

Link github dự án: [https://github.com/tranductri2003/PBL04\\_DUT\\_Student-CheckIn-System](https://github.com/tranductri2003/PBL04_DUT_Student-CheckIn-System)

Link trello dự án: <https://trello.com/w/pbl4lptrinhmngvahdiuhanh/home>