

**BỘ KHOA HỌC VÀ CÔNG NGHỆ  
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
KHOA CÔNG NGHỆ THÔNG TIN 1**



**BÁO CÁO  
THỰC TẬP TỐT NGHIỆP**

**ĐƠN VỊ THỰC TẬP: CÔNG TY CỔ PHẦN VCCORP**

Địa chỉ: Toà nhà Center Building - Hapulico Complex, Số 01, phố Nguyễn Huy  
Tường, phường Thanh Xuân, thành phố Hà Nội.

Cán bộ hướng dẫn tại công ty/đơn vị: Nguyễn Trọng Hùng  
Giảng viên phối hợp của học viện: TS. Nguyễn Thị Mai Trang

Sinh viên thực hiện: Vũ Thị Mai Anh

Mã số sinh viên: B21DCCN167

Lớp: D21HTTT5

Niên khóa: 2021 - 2026

Ngành: Công nghệ thông tin

Hà Nội, ngày 15 tháng 8 năm 2025

## THÔNG TIN CHUNG

Họ và tên sinh viên : Vũ Thị Mai Anh                      Mã sinh viên: B21DCCN167  
Lớp : D21HTTT5                      Khóa: 2021 – 2026  
Chuyên ngành : Hệ thống thông tin  
SĐT : 0896 037 950  
Email : vumaianh1604@gmail.com  
Tên dự án - project : Triển khai ứng dụng To-do List trên nền tảng Render theo mô hình 5 – 4 – 3 Cloud Computing NIST  
Người hướng dẫn : Nguyễn Trọng Hùng  
Chức vụ (vị trí) : Trưởng nhóm phát triển dịch vụ IaaS  
SĐT : 0989867645  
Email : hungnguyentrong@vccorp.vn  
Cơ quan công tác : Công Ty cổ phần Vccorp ( Khối BizflyCloud )

**Cán bộ hướng dẫn**  
(Ký tên và ghi rõ họ tên)

*Hà Nội , ngày 15 tháng 8 năm 2025*  
**Sinh viên thực tập**  
(Ký tên và ghi rõ họ tên)

**Giảng viên phối hợp**  
(Ký tên và ghi rõ họ tên)

## THÔNG TIN VỀ HỌC PHẦN THỰC TẬP TỐT NGHIỆP

### ❖ Thông tin chung

- Số tín chỉ: 6
- Loại học phần: bắt buộc
- Giờ học thực tập tại doanh nghiệp: 270 tiết
- Thuộc thành phần học tập: chuyên ngành

### ❖ Mô tả vắn tắt học phần

Học phần Thực tập tốt nghiệp là học phần bắt buộc nằm trong khối kiến thức chuyên ngành của chương trình đào tạo ngành Công nghệ thông tin, được tổ chức vào học kỳ 9. Để tham gia học tốt được học phần này, người học cần nắm vững các nội dung lý thuyết và thực hành đã tích lũy từ các học phần trước đó để có thể áp dụng vào môi trường làm việc thực tế tại các tổ chức/doanh nghiệp. Danh sách các công ty và vị trí thực tập sẽ được duy trì để cung cấp các cơ hội thực tập cho sinh viên. Quy trình tuyển chọn toàn diện sẽ được thực hiện thông qua phỏng vấn trực tiếp và lựa chọn tại công ty. Sinh viên cũng có thể đề xuất các đợt thực tập, nhưng những đề xuất này sẽ cần phải được thiết lập giữa Khoa CNTT và các công ty, đơn vị, trước khi các cơ hội thực tập được cung cấp cho sinh viên.

Mục tiêu của học phần Thực tập tốt nghiệp là để giúp sinh viên CNTT phát triển các kỹ năng cần thiết để làm việc chuyên nghiệp ở môi trường doanh nghiệp, các cơ quan nhà nước và các đơn vị có liên quan. Trong quá trình thực tập, sinh viên sẽ làm việc trực tiếp tại các đơn vị tiếp nhận thực tập dưới sự hướng dẫn của giảng viên và cán bộ hướng dẫn tại đơn vị. Sinh viên có cơ hội áp dụng kiến thức đã học ở nhà trường vào các tình huống chuyên nghiệp thực tế. Ngoài công việc thực tập đã được phân công tại đơn vị, sinh viên có nhiệm vụ học hỏi kinh nghiệm của cán bộ hướng dẫn, tham khảo tài liệu kỹ thuật chuyên ngành và hoàn thành báo cáo kết quả thực tập. Kết quả thực tập của sinh viên sẽ được ghi nhận dựa trên giá trị kinh nghiệm học tập được đánh giá theo thể hiện về mặt học thuật của sinh viên.

### ❖ Chuẩn đầu ra của học phần – CLO, chuẩn đầu ra của chương trình đào tạo PLO và các chỉ báo PIs – chỉ thị đạt CLO

STT	Chuẩn đầu ra của học phần - CLO	PLO / PI
1	Xác định và trình bày một vấn đề thực tiễn trong lĩnh vực Công nghệ Thông tin để giải quyết trong quá trình thực tập và vận dụng các nguyên lý tính toán cùng kiến thức liên ngành để đề xuất giải pháp phù hợp	PLO1 / PI1.1-PI1.4
2	Giao tiếp hiệu quả các ý tưởng và giải pháp kỹ thuật bằng lời nói, văn bản và hình ảnh trong môi trường chuyên nghiệp	PLO2 / PI2.1-PI2.4
3	Vận dụng các nguyên tắc đạo đức, pháp lý và chuyên môn để đưa ra các quyết định phù hợp trong quá trình thực tập	PLO3 / PI3.1-PI3.4

<b>4</b>	Hợp tác hiệu quả với các thành viên trong nhóm để hoàn thành nhiệm vụ được giao và đóng góp vào mục tiêu chung của nhóm trong quá trình thực tập	PLO4 / PI4.1-PI4.4
<b>5</b>	Thực hiện một công đoạn phát triển phần mềm hoặc tham gia triển khai, vận hành một hệ thống thông tin trong khuôn khổ kỳ thực tập	PLO5 / PI5.1-PI5.4

❖ **Phương pháp đánh giá học phần theo các chuẩn đầu ra**

<b>CLOs</b>	<b>Phương pháp đánh giá</b>	<b>Tỷ trọng (%)</b>
	<b>Đánh giá bởi doanh nghiệp</b>	
<b>CLO1, CLO2, CLO3, CLO4, CLO5</b>	Phiếu đánh giá sinh viên thực tập tốt nghiệp tại đơn vị thực tập tốt nghiệp theo các tiêu chí – PI và mức điểm ở phần phụ lục – Đánh giá bởi người hướng dẫn thực tập tại doanh nghiệp (cuối kỳ)	44
<b>CLO5</b>	Kiểm tra thực hành code/cài đặt, triển khai hệ thống thông tin – Authentic assessment theo các tiêu chí – PI và mức điểm ở phần phụ lục – Đánh giá bởi người hướng dẫn thực tập tại doanh nghiệp (cuối kỳ)	12
	<b>Đánh giá bởi giảng viên</b>	
	Đánh giá nhật ký thực tập hàng tuần của sinh viên, có thể làm căn cứ chấm điểm chuyên cần và kỹ năng làm việc nhóm – Giảng viên hướng dẫn đánh giá (trong suốt quá trình, dùng để tính điểm trung bình chung GPA)	20
<b>CLO1, CLO2, CLO5</b>	Phiếu đánh giá sinh viên thực tập tốt nghiệp của giảng viên hướng dẫn theo các tiêu chí – PI và mức điểm ở phần phụ lục – Giảng viên hướng dẫn đánh giá (cuối kỳ)	24

## LỜI CẢM ƠN

Trước hết, em xin bày tỏ lòng biết ơn sâu sắc đến Ban Giám hiệu Học viện Công nghệ Bru chính Viễn thông, cùng toàn thể quý thầy cô giáo trong khoa công nghệ thông tin. Các thầy cô không chỉ truyền đạt kiến thức chuyên môn mà còn định hướng, khích lệ em trong việc áp dụng lý thuyết vào thực tiễn. Nhờ sự tạo điều kiện thuận lợi từ phía học viện, em đã có cơ hội tham gia kỳ thực tập tốt nghiệp, giúp em hoàn thiện kỹ năng và chuẩn bị hành trang cho sự nghiệp tương lai.

Em xin gửi lời tri ân đặc biệt đến TS. Nguyễn Thị Mai Trang, giảng viên hướng dẫn trực tiếp của em. Với sự tận tâm, nhiệt huyết và kinh nghiệm phong phú, cô đã dành thời gian quý báu để hướng dẫn em từng bước trong việc nghiên cứu, thực hiện và hoàn thiện báo cáo. Những góp ý sâu sắc, lời khuyên bổ ích cùng sự động viên kịp thời của cô đã giúp em vượt qua những khó khăn, khắc phục hạn chế và đạt được kết quả tốt hơn mong đợi. Em thực sự biết ơn vì đã được học hỏi từ một người thầy xuất sắc như cô.

Bên cạnh đó, em xin chân thành cảm ơn Ban Lãnh đạo Công ty Cổ phần VCCorp, đặc biệt là các anh chị trong khối Bizfly Cloud, nơi em đã có cơ hội thực tập và trải nghiệm môi trường làm việc chuyên nghiệp. Trong suốt thời gian thực tập, em đã nhận được sự hỗ trợ nhiệt tình từ các anh chị đồng nghiệp, những người đã chia sẻ kinh nghiệm thực tiễn, hướng dẫn em tiếp cận các công nghệ mới và áp dụng kiến thức vào các dự án thực tế. Nhờ đó, em không chỉ củng cố nền tảng chuyên môn mà còn học hỏi được tinh thần làm việc nhóm, trách nhiệm và sáng tạo trong lĩnh vực công nghệ thông tin. Sự đón tiếp ấm áp và môi trường làm việc năng động tại VCCorp đã để lại trong em những ấn tượng sâu sắc và bài học quý giá cho sự phát triển bản thân sau này.

Một lần nữa, em xin chân thành cảm ơn tất cả mọi người. Những đóng góp của quý vị đã góp phần quan trọng vào sự thành công của báo cáo này. Em hy vọng sẽ tiếp tục nhận được sự hướng dẫn và hỗ trợ trong tương lai.

*Sinh viên*  
**Vũ Thị Mai Anh**

## MỤC LỤC

<b>THÔNG TIN CHUNG</b> .....	<b>i</b>
<b>THÔNG TIN VỀ HỌC PHẦN THỰC TẬP TỐT NGHIỆP</b> .....	<b>ii</b>
<b>LỜI CẢM ƠN</b> .....	<b>iv</b>
<b>MỤC LỤC</b> .....	<b>v</b>
<b>PHẦN 1 THÔNG TIN SINH VIÊN, CÔNG TY / ĐƠN VỊ THỰC TẬP, CÁN BỘ HƯỚNG DẪN, GIẢNG VIÊN, TRƯỜNG VÀ KHOA</b> .....	<b>1</b>
<b>PHẦN 2 KẾ HOẠCH THỰC TẬP</b> .....	<b>3</b>
<b>PHẦN 3 GIỚI THIỆU ĐƠN VỊ THỰC TẬP</b> .....	<b>10</b>
3.1 Thông tin doanh nghiệp.....	10
3.2 Lịch sử hình thành và phát triển của doanh nghiệp.....	12
3.3 Các lĩnh vực hoạt động, dịch vụ, hoặc sản phẩm của doanh nghiệp.....	13
3.4 Những đóng góp của doanh nghiệp trong lĩnh vực chuyên môn .....	17
<b>PHẦN 4 NHẬT KÝ THỰC TẬP</b> .....	<b>20</b>
Tuần 1 (từ ngày 2 tháng 7 năm 2025 đến ngày 6 tháng 7 năm 2025) .....	20
Tuần 2 (từ ngày 7 tháng 7 năm 2025 đến ngày 13 tháng 7 năm 2025) .....	21
Tuần 3 (từ ngày 14 tháng 7 năm 2025 đến ngày 20 tháng 7 năm 2025) .....	22
Tuần 4 (từ ngày 21 tháng 7 năm 2025 đến ngày 27 tháng 7 năm 2025) .....	23
Tuần 5 (từ ngày 28 tháng 7 năm 2025 đến ngày 3 tháng 8 năm 2025) .....	24
Tuần 6 (từ ngày 4 tháng 8 năm 2025 đến ngày 10 tháng 8 năm 2025) .....	25
Tuần 7 (từ ngày 11 tháng 8 năm 2025 đến ngày 15 tháng 8 năm 2025) .....	26
<b>PHẦN 5 BÁO CÁO PROJECT THỰC TẬP</b> .....	<b>27</b>
5.1 Tổng quan dự án.....	27
5.2 Cơ sở lý thuyết.....	28
5.3 Phân tích và thiết kế hệ thống .....	35
5.4 Triển khai.....	39
5.5 Kiểm thử và đánh giá .....	46
5.6 Kết luận và hướng phát triển.....	53
<b>Tài liệu tham khảo:</b> .....	<b>55</b>

## **PHẦN 1 THÔNG TIN SINH VIÊN, CÔNG TY / ĐƠN VỊ THỰC TẬP, CÁN BỘ HƯỚNG DẪN, GIẢNG VIÊN, TRƯỜNG VÀ KHOA**

### **Thông tin sinh viên**

Họ và tên sinh viên: Vũ Thị Mai Anh

Mã số sinh viên: B21DCCN167

Ngành học: Công nghệ thông tin

Sinh viên năm thứ: 4

Địa chỉ tạm trú trong thời gian thực tập: Số 160, ngõ 28 phố Đại Linh, phường Trung Văn, quận Nam Từ Liêm, thành phố Hà Nội

Số điện thoại: 0896037950

E-mail: vumaianh1604@gmail.com

### **Thông tin công ty / đơn vị thực tập**

Công ty / Đơn vị thực tập: Công ty Cổ phần VCCORP

Địa chỉ: Toà nhà Center Building - Hapulico Complex, Số 01, phố Nguyễn Huy Tưởng, phường Thanh Xuân, thành phố Hà Nội.

Số điện thoại: 024 7309 5555

Số fax: 024 3974 3413

Cán bộ hướng dẫn tại đơn vị thực tập: Nguyễn Trọng Hùng

Chức vụ (vị trí): Trưởng nhóm phát triển dịch vụ IaaS

Số điện thoại: 0989867645

E-mail: hungnguyentrong@vccorp.vn

Giảng viên phối hợp hướng dẫn: TS. Nguyễn Thị Mai Trang

Số điện thoại: 0865755525

E-mail: trangntm@ptit.edu.vn

Thời gian thực tập: 7 tuần

Bắt đầu: ngày 2 tháng 7 năm 2025

Kết thúc: ngày 15 tháng 8 năm 2025

## Thông tin Học viện Công nghệ Bưu chính Viễn thông

Địa chỉ: Km10 Nguyễn Trãi, Hà Đông, Hà Nội

Số điện thoại: 024 3756 2186

E-mail: [ctsv@ptit.edu.vn](mailto:ctsv@ptit.edu.vn)

Website: [www.ptit.edu.vn](http://www.ptit.edu.vn)

**Triết lý giáo dục:** Tri thức – Sáng tạo – Đạo đức – Trách nhiệm.

**Tầm nhìn:** Đến năm 2030, Học viện Công nghệ Bưu chính Viễn thông là trường đại học hàng đầu Việt Nam về quy mô, chất lượng đào tạo, nghiên cứu khoa học; là hình mẫu tiên phong về chuyển đổi số trong giáo dục đại học, trở thành trường đại học hàng đầu của khu vực, nằm trong nhóm trường đại học hàng đầu châu Á, nhóm 5 trường đại học hàng đầu Đông Nam Á về công nghệ số.

**Sứ mạng:** Sáng tạo và chuyển giao tri thức cho xã hội thông qua việc gắn kết các hoạt động đào tạo nguồn nhân lực chất lượng cao, nghiên cứu khoa học và chuyển giao công nghệ trong lĩnh vực thông tin, truyền thông và công nghệ số, góp phần xây dựng đất nước Việt Nam hùng cường.

**Giá trị cốt lõi:** Tiên phong – Sáng tạo; Chất lượng – Hiệu quả; Uy tín – Trách nhiệm; Tận tụy – Nghĩa tình.

## Thông tin Khoa Công nghệ Thông tin 1

Địa chỉ: Tầng 9 Nhà A2, Học viện Công nghệ Bưu chính Viễn thông, Km 10 Nguyễn Trãi, Hà Đông, Hà Nội

Số điện thoại: 024 3854 5604

E-mail: [phuongnd@ptit.edu.vn](mailto:phuongnd@ptit.edu.vn)

Website: [www.it.ptit.edu.vn](http://www.it.ptit.edu.vn)

### Giới thiệu

Khoa Công nghệ thông tin 1 là đơn vị đào tạo trực thuộc Học viện Công nghệ Bưu chính Viễn thông. Khoa có chức năng đào tạo và nghiên cứu khoa học thuộc lĩnh vực Công nghệ thông tin (Công nghệ thông tin, Công nghệ phần mềm, Hệ thống thông tin, Khoa học máy tính, Kỹ thuật máy tính).



## PHẦN 2 KẾ HOẠCH THỰC TẬP

Tuần 1	Nội dung công việc	Cách tiếp cận	Kế hoạch	Kết quả dự đoán
	<ul style="list-style-type: none"> <li>▪ Làm quen môi trường thực tập, quy định và quy trình làm việc.</li> <li>▪ Ôn tập và bổ sung kiến thức cơ bản về hệ điều hành Linux (Ubuntu Server).</li> <li>▪ Cài đặt và cấu hình cơ bản Ubuntu Server trên máy ảo.</li> <li>▪ Làm quen các lệnh quản trị cơ bản: quản lý user, phân quyền, quản lý file &amp; thư mục.</li> <li>▪ Tìm hiểu sơ đồ mạng nội bộ của đơn vị thực tập.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Trao đổi với người hướng dẫn để nắm rõ nội quy.</li> <li>▪ Tham khảo tài liệu nội bộ và giáo trình Ubuntu Server.</li> <li>▪ Thực hành trực tiếp trên máy ảo hoặc server test.</li> <li>▪ Quan sát hệ thống mạng và ghi chú lại.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Làm quen môi trường &amp; quy định.</li> <li>▪ Ngày 3–4: Cài Ubuntu Server và làm quen các lệnh cơ bản.</li> <li>▪ Ngày 5: Khảo sát sơ đồ mạng nội bộ và mô hình server đang sử dụng.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Nắm được môi trường và quy trình làm việc.</li> <li>▪ Thành thạo các lệnh Linux cơ bản.</li> <li>▪ Hiểu sơ đồ mạng nội bộ và các thành phần chính.</li> </ul>

Ngày 2 tháng 7 năm 2025  
 Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

Tuần 2	Nội dung công việc	Cách tiếp cận	Kế hoạch	Kết quả dự đoán
	<ul style="list-style-type: none"> <li>▪ Tìm hiểu về mô hình OSI, TCP/IP và các giao thức mạng cơ bản (IP, TCP, UDP, HTTP, HTTPS, DNS...).</li> <li>▪ Thực hành cấu hình IP tĩnh/động trên Ubuntu Server.</li> <li>▪ Kiểm tra kết nối mạng bằng ping, traceroute, netstat.</li> <li>▪ Tìm hiểu và cấu hình SSH để quản trị từ xa.</li> <li>▪ Tìm hiểu về tường lửa (ufw) trên Ubuntu.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Nghiên cứu tài liệu networking cơ bản.</li> <li>▪ Sử dụng lab ảo hoặc mạng test để thực hành cấu hình.</li> <li>▪ Trao đổi với người hướng dẫn để hiểu tình huống thực tế.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Học lý thuyết OSI, TCP/IP và giao thức cơ bản.</li> <li>▪ Ngày 3: Cấu hình IP tĩnh/động và kiểm tra kết nối.</li> <li>▪ Ngày 4: Cấu hình SSH, thử kết nối từ máy khác.</li> <li>▪ Ngày 5: Cấu hình và kiểm tra tường lửa ufw.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Hiểu mô hình mạng và giao thức cơ bản.</li> <li>▪ Cấu hình được IP và quản trị server qua SSH.</li> <li>▪ Biết bật/tắt và cấu hình tường lửa cơ bản.</li> </ul>

Ngày 7 tháng 7 năm 2025  
Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

Tuần 3	Nội dung công việc	Cách tiếp cận	Kế hoạch	Kết quả dự đoán
	<ul style="list-style-type: none"> <li>▪ Tìm hiểu và thực hành quản lý dịch vụ trên Ubuntu (systemctl, journalctl).</li> <li>▪ Cài đặt và cấu hình dịch vụ web (Apache hoặc Nginx).</li> <li>▪ Tạo và cấu hình Virtual Host.</li> <li>▪ Cài đặt và cấu hình MySQL/MariaDB.</li> <li>▪ Kết nối dịch vụ web với cơ sở dữ liệu.</li> <li>▪ Kiểm tra truy cập từ máy khác trong mạng.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Học qua tài liệu chính thức Apache/Nginx và MySQL.</li> <li>▪ Thực hành trực tiếp trên server test.</li> <li>▪ Kiểm tra kết quả bằng truy cập từ client khác.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Quản lý dịch vụ và cài đặt web server.</li> <li>▪ Ngày 3: Tạo Virtual Host.</li> <li>▪ Ngày 4: Cài và cấu hình MySQL.</li> <li>▪ Ngày 5: Kết nối web server với MySQL và test từ máy khác.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Cài và cấu hình thành công web server và database.</li> <li>▪ Hiểu cách kết nối dịch vụ với nhau.</li> <li>▪ Truy cập website nội bộ thành công.</li> </ul>

Ngày 14 tháng 7 năm 2025  
Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

<b>Tuần 4</b>	<b>Nội dung công việc</b>	<b>Cách tiếp cận</b>	<b>Kế hoạch</b>	<b>Kết quả dự đoán</b>
	<ul style="list-style-type: none"> <li>▪ Tìm hiểu và thực hành triển khai dịch vụ DNS nội bộ.</li> <li>▪ Cấu hình dịch vụ DHCP cấp phát IP tự động.</li> <li>▪ Quản lý và giám sát log hệ thống.</li> <li>▪ Tìm hiểu và sử dụng công cụ giám sát mạng (Nagios/Zabbix).</li> </ul>	<ul style="list-style-type: none"> <li>▪ Nghiên cứu tài liệu BIND9 và DHCP.</li> <li>▪ Thực hành cấu hình trên server test.</li> <li>▪ Cài đặt bản Zabbix/Nagios bản nhỏ để test.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Cấu hình DNS nội bộ.</li> <li>▪ Ngày 3: Cấu hình DHCP.</li> <li>▪ Ngày 4: Giám sát log và xử lý lỗi cơ bản.</li> <li>▪ Ngày 5: Thử cài và chạy thử công cụ giám sát.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Hiểu và triển khai được DNS/DHCP cơ bản.</li> <li>▪ Biết giám sát log hệ thống.</li> <li>▪ Nắm được nguyên lý công cụ giám sát mạng.</li> </ul>

Ngày 21 tháng 7 năm 2025  
 Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

<b>Tuần 5</b>	<b>Nội dung công việc</b>	<b>Cách tiếp cận</b>	<b>Kế hoạch</b>	<b>Kết quả dự đoán</b>
	<ul style="list-style-type: none"> <li>▪ Tìm hiểu và thực hành cấu hình VLAN, Trunking (mô phỏng bằng GNS3/Packet Tracer hoặc thiết bị thật).</li> <li>▪ Tìm hiểu và cấu hình định tuyến tĩnh và động (RIP/OSPF).</li> <li>▪ Thử nghiệm kết nối giữa các VLAN.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Học qua tài liệu Cisco Networking.</li> <li>▪ Dùng phần mềm mô phỏng hoặc switch/router thật.</li> <li>▪ Thực hành và kiểm tra bằng ping, traceroute.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Cấu hình VLAN trên switch.</li> <li>▪ Ngày 3: Cấu hình Trunking.</li> <li>▪ Ngày 4: Cấu hình định tuyến tĩnh.</li> <li>▪ Ngày 5: Cấu hình OSPF hoặc RIP và test kết nối.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Hiểu và cấu hình VLAN, Trunking.</li> <li>▪ Nắm nguyên lý định tuyến.</li> <li>▪ Kết nối thành công các VLAN khác nhau.</li> </ul>

Ngày 28 tháng 7 năm 2025  
 Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

Tuần 6	Nội dung công việc	Cách tiếp cận	Kế hoạch	Kết quả dự đoán
	<ul style="list-style-type: none"> <li>▪ Tìm hiểu về Docker và containerization.</li> <li>▪ Cài đặt Docker trên Ubuntu Server.</li> <li>▪ Tìm hiểu Image, Container, Volume, Network trong Docker.</li> <li>▪ Thực hành build image từ Dockerfile.</li> <li>▪ Triển khai ứng dụng mẫu bằng Docker Compose.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Đọc tài liệu chính thức Docker.</li> <li>▪ Thực hành trực tiếp trên server.</li> <li>▪ Test ứng dụng bằng trình duyệt hoặc curl.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–2: Cài Docker và tìm hiểu khái niệm.</li> <li>▪ Ngày 3: Thực hành build image.</li> <li>▪ Ngày 4–5: Triển khai ứng dụng mẫu (Nginx, MySQL, WordPress).</li> </ul>	<ul style="list-style-type: none"> <li>▪ Cài đặt thành công Docker.</li> <li>▪ Hiểu các khái niệm cơ bản.</li> <li>▪ Triển khai ứng dụng mẫu hoạt động tốt.</li> </ul>

Ngày 4 tháng 8 năm 2025  
Cán bộ hướng dẫn  
(Ký tên và họ tên)

Sinh viên thực tập  
(Ký tên và họ tên)

<b>Tuần 7</b>	<b>Nội dung công việc</b>	<b>Cách tiếp cận</b>	<b>Kế hoạch</b>	<b>Kết quả dự đoán</b>
	<ul style="list-style-type: none"> <li>▪ Ôn tập và củng cố toàn bộ kiến thức thực tập.</li> <li>▪ Viết báo cáo tổng kết.</li> <li>▪ Thực hiện demo lại một số cấu hình quan trọng.</li> <li>▪ Bàn giao tài khoản/mật khẩu và môi trường test.</li> <li>▪ Hoàn thiện tài liệu hướng dẫn sử dụng/cấu hình.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Xem lại toàn bộ ghi chú thực tập.</li> <li>▪ Thực hành lại các bài lab chính.</li> <li>▪ Tham khảo ý kiến người hướng dẫn để bổ sung báo cáo.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Ngày 1–3: Ôn tập và thực hành lại.</li> <li>▪ Ngày 4–5: Hoàn thiện báo cáo, bàn giao và kết thúc thực tập.</li> </ul>	<ul style="list-style-type: none"> <li>▪ Hoàn thiện báo cáo.</li> <li>▪ Củng cố kiến thức.Đủ khả năng tự triển khai các dịch vụ đã học.</li> </ul>

Ngày 11 tháng 8 năm 2025  
Cán bộ hướng dẫn  
*(Ký tên và họ tên)*

Sinh viên thực tập  
*(Ký tên và họ tên)*

## **PHẦN 3    GIỚI THIỆU ĐƠN VỊ THỰC TẬP**

### **3.1    Thông tin doanh nghiệp**

Công ty Cổ phần VCCorp (VCCorp Corporation) là một trong những doanh nghiệp tiên phong và hàng đầu tại Việt Nam trong lĩnh vực công nghệ thông tin và nội dung số. Được thành lập vào năm 2006, sau 19 năm phát triển, VCCorp đã xây dựng và ứng dụng thành công hạ tầng cloud computing có khả năng lưu trữ và tính toán lượng dữ liệu lớn (Big Data, Data mining) hỗ trợ cho nền tảng công nghệ quảng cáo trực tuyến Admicro, phục vụ trên 50 triệu người dùng (chiếm trên 90% tổng số người dùng Internet tại Việt Nam) với hơn 200 website uy tín, gần 30 website trong số đó thuộc sở hữu độc quyền, phục vụ hàng tỷ lượt xem mỗi tháng. Ngoài ra, VCCorp nằm trong top 3 công ty phát hành game trên nền tảng di động với thương hiệu Sohagame và mới đây hệ sinh thái chuyển đổi số Bizfly đang được VCCorp phát triển và triển khai mạnh mẽ. Đồng thời VCCorp cũng là đơn vị xây dựng Lotus - mạng xã hội do người Việt Nam làm chủ.

Để có được hàng loạt sản phẩm công nghệ mang tính đột phá và ưu việt ngày hôm nay, VCCorp đã xây dựng một đội ngũ nhân sự hùng hậu với trên 2000 nhân viên làm việc và hoạt động chuyên nghiệp tại Hà Nội, thành phố Hồ Chí Minh và các văn phòng đại diện, chi nhánh khác tại 06 thành phố phát triển nhất Việt Nam. Với tinh thần “Innovation-NonStop – Sáng tạo cho cộng đồng”, VCCorp là mái nhà chung dành cho những người trẻ yêu công nghệ, đam mê sáng tạo.

Khối Bizfly Cloud là bộ phận công nghệ cốt lõi của VCCorp, chuyên cung cấp các giải pháp hạ tầng công nghệ thông tin dựa trên nền tảng điện toán đám mây. Bizfly Cloud tập trung vào việc hỗ trợ doanh nghiệp chuyển đổi số, cung cấp các dịch vụ như máy chủ ảo (Cloud Server), CDN, Kubernetes Engine và nhiều giải pháp khác, đáp ứng nhu cầu của hơn 4.000 doanh nghiệp và 100.000 khách hàng cá nhân trên toàn quốc.





Hình 3.1: VCCorp (Nguồn: <https://vccorp.vn/>)

### 3.2 Lịch sử hình thành và phát triển của doanh nghiệp

Công ty Cổ phần VCCorp (VCCorp Corporation) được thành lập vào năm 2006, đánh dấu bước khởi đầu cho hành trình trở thành một trong những doanh nghiệp tiên phong và dẫn đầu tại Việt Nam trong lĩnh vực công nghệ thông tin và nội dung số. Với sứ mệnh phát triển hệ sinh thái công nghệ và truyền thông, VCCorp đã không ngừng mở rộng quy mô và khẳng định vị thế của mình trong hơn 19 năm qua. Khối Bizfly Cloud, một bộ phận quan trọng của VCCorp, cũng có sự phát triển song hành, đóng vai trò cốt lõi trong việc cung cấp các giải pháp hạ tầng công nghệ thông tin dựa trên điện toán đám mây.

#### 3.2.1 Giai đoạn khởi đầu và xây dựng nền tảng (2006 - 2012)

- **Năm 2006:** VCCorp chính thức ra đời với mục tiêu tiên phong trong lĩnh vực công nghệ và truyền thông tại Việt Nam. Tiền thân của Bizfly Cloud là phòng Công nghệ Hệ thống (Platform), tập trung nghiên cứu và phát triển các giải pháp hạ tầng nội bộ để hỗ trợ các dự án công nghệ của công ty.
- **Năm 2012:** VCCorp hoàn thiện hệ thống Private Cloud, đảm bảo hạ tầng ổn định cho hơn 200 trang tin trực tuyến thuộc hệ sinh thái của công ty. Đây là cột mốc quan trọng, đánh dấu bước đầu trong việc xây dựng năng lực công nghệ đám mây của Bizfly Cloud.

#### 3.2.2 Giai đoạn mở rộng và phát triển dịch vụ (2014 - 2018)

- **Năm 2014:** Bizfly Cloud chính thức ra mắt hệ thống Public Cloud, cung cấp dịch vụ Cloud Server và Content Delivery Network (CDN) cho các cá nhân và doanh nghiệp. Đồng thời, công ty triển khai giải pháp chống tấn công DDoS, nâng cao khả năng bảo mật cho các website thuộc hệ sinh thái VCCorp và khách hàng bên ngoài.
- **Năm 2015:** Bizfly Cloud phát triển nền tảng live streaming, được ứng dụng thành công trong các sự kiện lớn như ra mắt VinFast tại Paris Motor Show 2018, khẳng định năng lực trong lĩnh vực truyền thông trực tuyến.
- **Năm 2018:** Bizfly Cloud mở rộng đội ngũ với hơn 150 nhân sự, trong đó 80% là chuyên gia kỹ thuật cao cấp. Công ty tiếp tục ra mắt các dịch vụ mới như Auto Scaling, Business Email, Load Balancer và Simple Storage, đáp ứng nhu cầu đa dạng của thị trường.

### ***3.2.3 Giai đoạn khẳng định vị thế và chuyển đổi số (2020 - 2022)***

- **Năm 2020:** Bizfly Cloud được Bộ Thông tin và Truyền thông công nhận là một trong bốn nền tảng điện toán đám mây “Make in Vietnam” đáp ứng đầy đủ 153 tiêu chí kỹ thuật phục vụ Chính phủ điện tử. Công ty hoàn thiện hệ thống Multi Datacenter tại Hà Nội và Hồ Chí Minh, đảm bảo hoạt động liên tục 24/7. Các dịch vụ như Bizfly Cloud Kubernetes Engine và Cloud VoD cũng được triển khai, hỗ trợ mạnh mẽ cho các doanh nghiệp trong hành trình chuyển đổi số.
- **Năm 2022:** Bizfly Cloud tiếp tục phát triển các giải pháp tiên tiến như API Gateway, Cloud Database, Kafka as a Service và Cloud VoD, phục vụ các ngành tài chính, ngân hàng, y tế và giáo dục. Sự phát triển này đã giúp công ty hỗ trợ hơn 4.000 doanh nghiệp và 100.000 khách hàng cá nhân trên toàn quốc.

### ***3.2.4 Hiện tại và định hướng tương lai***

Tính đến năm 2025, VCCorp và Bizfly Cloud tiếp tục khẳng định vị thế dẫn đầu trong lĩnh vực công nghệ thông tin tại Việt Nam. Với đội ngũ hơn 2.500 nhân sự, trong đó Bizfly Cloud sở hữu hơn 200 chuyên gia kỹ thuật, công ty không chỉ cung cấp các giải pháp hạ tầng công nghệ mà còn đóng góp vào việc đào tạo nhân lực chất lượng cao. Trong tương lai, VCCorp đặt mục tiêu trở thành nhà cung cấp dịch vụ điện toán đám mây hàng đầu khu vực, sánh ngang với các tiêu chuẩn quốc tế, đồng thời tiếp tục thúc đẩy chuyển đổi số cho doanh nghiệp và cộng đồng.

## **3.3 Các lĩnh vực hoạt động, dịch vụ, hoặc sản phẩm của doanh nghiệp**

Công ty Cổ phần VCCorp (VCCorp) là một trong những doanh nghiệp hàng đầu tại Việt Nam với hệ sinh thái đa dạng, bao gồm công nghệ thông tin, nội dung số, truyền thông trực tuyến, thương mại điện tử và phát hành game. Khối Bizfly Cloud, một bộ phận chiến lược của VCCorp, đóng vai trò quan trọng trong việc cung cấp các giải pháp hạ tầng công nghệ dựa trên nền tảng điện toán đám mây. Dưới đây là các lĩnh vực hoạt động, dịch vụ và sản phẩm nổi bật của doanh nghiệp:

### **3.3.1 Hạ tầng công nghệ và giải pháp chuyển đổi số (Bizfly Cloud)**

Bizfly Cloud là đơn vị chuyên cung cấp các dịch vụ điện toán đám mây tiên tiến, hỗ trợ doanh nghiệp trong hành trình chuyển đổi số. Các sản phẩm và dịch vụ tiêu biểu bao gồm:

- **Bizfly Cloud Server:** Dịch vụ máy chủ ảo với tính năng Auto Scaling, băng thông miễn phí và hiệu năng cao, phù hợp cho các ứng dụng xử lý dữ liệu lớn.
- **Bizfly Cloud CDN:** Giải pháp tăng tốc độ tải website, giảm hơn 90% băng thông tiêu thụ, đồng thời tích hợp chống tấn công DDoS để bảo vệ dữ liệu.
- **Bizfly Cloud Kubernetes Engine:** Nền tảng tự động triển khai và quản lý các Kubernetes cluster, là giải pháp tiên phong tại Việt Nam.
- **Bizfly Cloud Simple Storage:** Dịch vụ lưu trữ dữ liệu không giới hạn dung lượng, tối ưu chi phí cho doanh nghiệp.
- **Bizfly Cloud Business Email và Call Center:** Cung cấp email doanh nghiệp không giới hạn tài khoản và tổng đài ảo linh hoạt, hỗ trợ giao tiếp hiệu quả.
- **Các dịch vụ khác:** Bao gồm Cloud Database, Cloud VoD, API Gateway, Anti-DDoS và Web Application Firewall (WAF), đáp ứng nhu cầu bảo mật và quản lý hạ tầng.

### **3.3.2 Online Media**

VCCorp sở hữu mạng lưới hơn 200 website và nền tảng truyền thông, tiếp cận 90% người dùng Internet tại Việt Nam. Các sản phẩm tiêu biểu bao gồm:

- **Kenh14.vn:** Trang tin tức giải trí hàng đầu, thu hút lượng lớn độc giả trẻ.
- **Soha.vn:** Cổng thông tin tổng hợp với nội dung đa dạng, từ tin tức đến văn hóa.
- **CafeF.vn:** Trang chuyên về tài chính và kinh doanh, phục vụ cộng đồng doanh nhân.
- **Autopro.vn:** Nguồn thông tin chuyên sâu về ô tô và xe máy.
- **Afamily.vn:** Trang dành cho gia đình và phụ nữ, tập trung vào phong cách sống.
- **Gamek.vn:** Trang tin tức và đánh giá game, phục vụ cộng đồng game thủ.

- Ngoài ra, các chiến dịch truyền thông như Wechoice, Here We Go và We Do đã tạo dấu ấn mạnh mẽ trong cộng đồng.

### 3.3.3 *Quảng cáo trực tuyến (Admicro)*

Admicro là nền tảng quảng cáo trực tuyến hàng đầu Việt Nam, tận dụng hạ tầng cloud computing của Bizfly Cloud để xử lý hàng tỷ lượt xem mỗi tháng. Dịch vụ này hỗ trợ các chiến dịch quảng cáo số hóa, mang lại hiệu quả cao cho doanh nghiệp.

#1 Content network tại Việt Nam

#1 Ad network tại Việt Nam

Chiếm 38% thị phần

200 website uy tín

VCCorp đã xây dựng và ứng dụng thành công hạ tầng cloud computing có khả năng lưu trữ và tính toán lượng dữ liệu lớn (Big Data, Data mining) hỗ trợ cho nền tảng công nghệ quảng cáo trực tuyến Admicro. Dẫn đầu về mạng lưới content tại Việt Nam với hơn 200 website uy tín, gần 30 website trong số đó thuộc sở hữu độc quyền, phục vụ hàng tỷ lượt xem mỗi tháng.

Với độ phủ rộng rãi, Admicro cũng xây dựng một mạng lưới rộng rãi gồm các đối tác là tập đoàn truyền thông lớn ở trong nước và nước ngoài về lĩnh vực truyền thông và quảng cáo như WPP, groupM, Publicis, MMS, Starcom, Omnicom, Dentsu, Hakuhodo, Dat viet và nền tảng công nghệ truyền thông: Google, The Trade Desk, AppNexus, Geniee...

**Hệ thống website:**

- Mạng Tài chính - Kinh doanh (hạng A) 50 triệu lượt xem trang và 8,2 triệu lượt truy cập mỗi tháng: CafeF, VnEconomy, Diendan doanhnghiep, Autopro...
- Mạng Công nghệ và Ô tô 30 triệu lượt xem trang và 10 triệu lượt truy cập mỗi tháng: GenK, Autopro
- Mạng thông tin giải trí được yêu mến và truy cập thường xuyên với hơn 24 triệu độc giả trẻ từ độ tuổi trong số 13-25, bao gồm: Kenh14, Gamek và Genk
- Mạng Gia đình - Phụ nữ - Sức khỏe: 175 triệu lượt xem trang và 35 triệu lượt truy cập mỗi tháng: Sức khỏe đời sống, Giadinh, Afamily, Guu và chuyên trang Sức khỏe của Dân trí...
- Tin tức tổng hợp: Hơn 1 triệu lượt xem trang và 200 triệu lượt truy cập mỗi tháng: Dantri, VTV, Nguoilaodong, VCTV và SohaNews

#### **3.3.4 Thương mại điện tử**

VCCorp phát triển các nền tảng thương mại điện tử nổi bật:

- **Muachung.vn:** Trang mua sắm theo nhóm, mang lại lợi ích cho người dùng.
- **Rongbay.com:** Cổng thương mại điện tử phổ biến, hỗ trợ mua bán trực tuyến.
- **Enbac.com:** Nền tảng giao dịch trực tuyến với cộng đồng người dùng lớn.

#### **3.3.5 Phát hành game (SohaGame)**

SohaGame là đơn vị phát hành game di động lớn thứ ba tại Việt Nam, với hơn 14 triệu người chơi và 23 tựa game được ra mắt trong năm 2020. Các sản phẩm nổi bật bao gồm các game nhập vai và chiến thuật, thu hút đông đảo game thủ.

#### **3.3.6 Mạng xã hội (Lotus)**

Với tinh thần muốn giải các bài toán khó, muốn làm theo cách riêng và mong muốn tạo nên giá trị to lớn cho xã hội, VCCorp cùng các nhà đầu tư trong nước đã quyết tâm xây dựng một MXH make in Việt Nam, mạng xã hội do người Việt làm chủ. Sự kiện ra mắt Lotus được đông đảo cộng đồng quan tâm và ủng hộ, đạt 1 triệu người dùng chỉ sau 10 ngày, 3

triệu người dùng tính đến thời điểm hiện tại. Đặc biệt năm 2020, Lotus cũng thực hiện chiến dịch “Lá chắn virus Corona”, tạo được tiếng vang lớn trong xã hội, trở thành kênh truyền thông uy tín, nhanh nhạy về Phòng chống dịch Covid-19 được người đọc tin tưởng và lựa chọn.

### 3.3.7 Các sản phẩm và dịch vụ khác

Hệ sinh thái VCCorp còn bao gồm các thương hiệu như Solo, SohaPay, F139, GenK và Soha TV, cung cấp giải pháp thanh toán trực tuyến, tin tức công nghệ và nội dung truyền hình trực tuyến.

Hình 3.3: Các sản phẩm và dịch vụ tiêu biểu của hệ sinh thái VCCorp (Nguồn: <https://vccorp.vn/>)

## 3.4 Những đóng góp của doanh nghiệp trong lĩnh vực chuyên môn

Công ty Cổ phần VCCorp, cùng với khối Bizfly Cloud, đã ghi dấu ấn quan trọng trong lĩnh vực công nghệ thông tin tại Việt Nam thông qua các đóng góp nổi bật trong chuyển đổi số, bảo mật và phát triển nhân lực. Những nỗ lực này không chỉ nâng cao vị thế của doanh nghiệp mà còn thúc đẩy sự phát triển của ngành công nghệ thông tin và Internet trong nước.

#### ***3.4.1 Thúc đẩy chuyển đổi số***

Bizfly Cloud đã hỗ trợ hơn 4.000 doanh nghiệp và 100.000 khách hàng cá nhân trong hành trình chuyển đổi số, cung cấp các giải pháp hạ tầng công nghệ tiên tiến. Các dự án tiêu biểu bao gồm triển khai hệ thống cho sự kiện ra mắt VinFast tại Paris Motor Show 2018, mạng xã hội Lotus và các nền tảng như Coolmate, Thu Cúc, SSI và VNDirect. Những giải pháp như Cloud Server, CDN và Kubernetes Engine đã giúp doanh nghiệp tối ưu chi phí, quản lý dữ liệu hiệu quả và nâng cao trải nghiệm người dùng.

#### ***3.4.2 Làm chủ công nghệ điện toán đám mây***

Bizfly Cloud là một trong bốn nền tảng điện toán đám mây “Make in Vietnam” được Bộ Thông tin và Truyền thông công nhận vào năm 2020, đáp ứng đầy đủ 153 tiêu chí kỹ thuật phục vụ Chính phủ điện tử. Việc làm chủ mã nguồn mở và phát triển các giải pháp như Cloud Database, API Gateway và Simple Storage đã giúp doanh nghiệp cạnh tranh với các nhà cung cấp quốc tế, góp phần giảm phụ thuộc vào công nghệ nước ngoài.

#### ***3.4.3 Bảo mật và an toàn thông tin***

Với chứng nhận quốc tế ISO 27001:2013 và ISO 27017:2015, Bizfly Cloud đảm bảo an toàn thông tin cho khách hàng. Các giải pháp như Anti-DDoS và Web Application Firewall (WAF) đã giảm thiểu 99% rủi ro bảo mật, đặc biệt trong các ngành nhạy cảm như tài chính, ngân hàng và y tế. Sự đóng góp này đã củng cố niềm tin của doanh nghiệp và người dùng vào các dịch vụ nội địa.

#### ***3.4.4 Hỗ trợ cộng đồng và giáo dục***

Bizfly Cloud cung cấp dịch vụ Cloud LMS, hỗ trợ các tổ chức giáo dục triển khai hệ thống học trực tuyến, đặc biệt trong bối cảnh đại dịch COVID-19 và xu hướng học từ xa. Ngoài ra, các chiến dịch truyền thông của VCCorp, như “Lá chắn virus Corona” trên mạng xã hội Lotus, đã góp phần nâng cao nhận thức cộng đồng về các vấn đề xã hội và sức khỏe.



#### ***3.4.5 Đào tạo và phát triển nhân lực***

Với đội ngũ hơn 2.500 nhân sự, trong đó Bizfly Cloud sở hữu hơn 200 chuyên gia kỹ thuật cao cấp được đào tạo về quản trị mạng, bảo mật và các ngôn ngữ lập trình như Python, Java và C, VCCorp không chỉ cung cấp giải pháp công nghệ mà còn đóng góp vào việc đào tạo nhân lực chất lượng cao. Sự hợp tác với các trường đại học và tổ chức giáo dục đã tạo điều kiện cho sinh viên tiếp cận thực tế, chuẩn bị cho sự nghiệp trong ngành công nghệ thông tin.

#### ***3.4.6 Đóng góp vào sự phát triển của Internet Việt Nam***

Hệ sinh thái hơn 200 website của VCCorp, bao gồm Kenh14.vn, CafeF.vn và Soha.vn, tiếp cận 90% người dùng Internet tại Việt Nam, đóng vai trò quan trọng trong việc cung cấp thông tin và nội dung số hóa. Các nền tảng như Admicro và SohaGame cũng góp phần thúc đẩy quảng cáo trực tuyến và ngành công nghiệp game nội địa, khẳng định vị thế của VCCorp trong việc xây dựng một Internet Việt Nam bền vững.

Những đóng góp này không chỉ thể hiện năng lực chuyên môn của VCCorp và Bizfly Cloud mà còn mang lại giá trị thiết thực cho cộng đồng, doanh nghiệp và sự phát triển của ngành công nghệ thông tin tại Việt Nam.

## PHẦN 4 NHẬT KÝ THỰC TẬP

**Tuần 1** (từ ngày 2 tháng 7 năm 2025 đến ngày 6 tháng 7 năm 2025)

Ngày	Tóm tắt hoạt động thực tập	Quy định khung tham chiếu (TCVN, QCVN, ISO...)	Kết quả hoạt động	Phân tích, giải thích, kết luận	Xác nhận của CBHD
02/07/2025	Làm quen với hệ điều hành Ubuntu Server, tìm hiểu các lệnh cơ bản (quản lý user, file, quyền).	ISO/IEC 20000 - 1:2018	Nắm được các lệnh cơ bản như useradd, chmod, chown trên Ubuntu Server.	Tiến độ tốt, cần thực hành thêm để thành thạo.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên)
03/07/2025	Thực hành cài đặt và cấu hình máy chủ Ubuntu Server trên môi trường ảo.	TCVN 9261:2012	Hoàn thành cài đặt và cấu hình cơ bản máy chủ ảo.	Cấu hình thành công, nhưng cần chú ý đến tối ưu hóa hiệu suất.	
04/07/2025	Tìm hiểu sơ đồ mạng của Bizfly Cloud, nghiên cứu cách triển khai máy chủ trong mạng nội bộ.	QCVN 81:2014/BTTTT	Hiểu sơ đồ mạng và cách kết nối máy chủ với mạng nội bộ.	Cần thời gian để nắm rõ các thành phần mạng, nhưng đã có nền tảng cơ bản.	

**Tuần 2** (từ ngày 7 tháng 7 năm 2025 đến ngày 13 tháng 7 năm 2025)

<b>Ngày</b>	<b>Tóm tắt hoạt động thực tập</b>	<b>Quy định khung tham chiếu (TCVN, QCVN, ISO...)</b>	<b>Kết quả hoạt động</b>	<b>Phân tích, giải thích, kết luận</b>	<b>Xác nhận của CBHD</b>
07/07/2025	Tìm hiểu mô hình OSI và TCP/IP, nghiên cứu các lớp cơ bản của networking.	TCVN 11973:2018	Hiểu rõ mô hình OSI và các giao thức TCP/IP cơ bản.	Tiến độ tốt, cần nghiên cứu thêm về ứng dụng thực tế.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD <i>(Ký tên và họ tên)</i>
08/07/2025	Cấu hình máy chủ, tìm hiểu cách định nghĩa và kiểm tra thông số mạng.	ISO/IEC 20000 - 1:2018	Hoàn thành cấu hình mạng cơ bản và kiểm tra thông số.	Cấu hình ổn định, cần cải thiện kỹ năng kiểm tra lỗi.	
09/07/2025	Cài đặt và cấu hình IP trên máy chủ Ubuntu, thực hành các lệnh mạng.	TCVN 9261:2012	Cài đặt và cấu hình IP thành công, nắm được các lệnh như ifconfig.	Kỹ năng cải thiện, cần chú ý đến xung đột IP.	
10/07/2025	Cấu hình và sử dụng SSH để quản trị máy chủ từ xa.	QCVN 81:2014/BTTTT	Thực hiện thành công kết nối SSH từ xa.	Hiệu quả, cần học thêm cách bảo mật kết nối SSH.	
11/07/2025	Tìm hiểu và cấu hình tường lửa ufw trên Ubuntu, thực hành các quy tắc cơ bản.	ISO 27001:2013	Cấu hình ufw thành công, áp dụng các quy tắc cơ bản.	Tiến độ tốt, cần thực hành thêm để tối ưu hóa bảo mật.	

**Tuần 3** (từ ngày 14 tháng 7 năm 2025 đến ngày 20 tháng 7 năm 2025)

Ngày	Tóm tắt hoạt động thực tập	Quy định khung tham chiếu (TCVN, QCVN, ISO...)	Kết quả hoạt động	Phân tích, giải thích, kết luận	Xác nhận của CBHD
14/07/2025	Cài đặt và cấu hình web server (Apache/Nginx) trên Ubuntu, tìm hiểu các lệnh cơ bản.	TCVN 11973:2018	Cài đặt thành công Apache/Nginx, nắm được các lệnh cơ bản.	Tiến độ ổn định, cần học thêm cách tối ưu hóa web server.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên))
15/07/2025	Tạo Virtual Host trên web server, thực hành cấu hình domain ảo.	ISO/IEC 20000 - 1:2018	Tạo thành công Virtual Host, cấu hình domain ảo hoàn tất.	Hiệu quả, cần kiểm tra kỹ lưỡng lỗi cấu hình.	
16/07/2025	Cài đặt và cấu hình MySQL/MariaDB trên Ubuntu, tìm hiểu cơ sở dữ liệu.	TCVN 9261:2012	Cài đặt và cấu hình MySQL/MariaDB thành công.	Tiến độ tốt, cần thực hành thêm các câu lệnh SQL.	
17/07/2025	Thử nghiệm web server và database, kiểm tra kết nối giữa Apache/Nginx và MySQL.	QCVN 81:2014/BTTTT	Kết nối web server với database thành công.	Hiệu quả, cần chú ý bảo mật kết nối database.	
18/07/2025	Truy cập website, thử nghiệm các tính năng cơ bản và ghi nhận kết quả.	ISO 27001:2013	Truy cập website thành công, ghi nhận kết quả thử nghiệm.	Kết quả ổn định, cần cải thiện giao diện và bảo mật website.	

**Tuần 4** (từ ngày 21 tháng 7 năm 2025 đến ngày 27 tháng 7 năm 2025)

<b>Ngày</b>	<b>Tóm tắt hoạt động thực tập</b>	<b>Quy định khung tham chiếu (TCVN, QCVN, ISO...)</b>	<b>Kết quả hoạt động</b>	<b>Phân tích, giải thích, kết luận</b>	<b>Xác nhận của CBHD</b>
21/07/2025	Cài đặt và cấu hình PHP trên Ubuntu, tìm hiểu các phiên bản PHP.	TCVN 11973:2018	Cài đặt PHP thành công, nắm được các phiên bản chính.	Tiến độ tốt, cần học thêm cách tối ưu PHP.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên)
22/07/2025	Tích hợp PHP với web server (Apache/Nginx) và MySQL, kiểm tra kết nối.	ISO/IEC 20000 - 1:2018	Tích hợp thành công, kiểm tra kết nối ổn định.	Hiệu quả, cần chú ý bảo mật khi kết nối database.	
23/07/2025	Viết mã PHP cơ bản (kết nối database, xử lý form), thực hành trên máy chủ.	TCVN 9261:2012	Viết thành công mã PHP cơ bản, xử lý form hiệu quả.	Kỹ năng lập trình cải thiện, cần tối ưu hóa mã nguồn.	
24/07/2025	Tìm hiểu về WordPress, cài đặt và cấu hình môi trường WordPress.	QCVN 81:2014/BTTTT	Cài đặt WordPress thành công, cấu hình cơ bản hoàn tất.	Tiến độ ổn định, cần học thêm cách bảo mật WordPress.	
25/07/2025	Triển khai website WordPress trên máy chủ, thử nghiệm các tính năng cơ bản.	ISO 27001:2013	Triển khai website thành công, thử nghiệm các tính năng ổn định.	Kết quả tốt, cần kiểm tra bảo mật và hiệu suất website.	

**Tuần 5** (từ ngày 28 tháng 7 năm 2025 đến ngày 3 tháng 8 năm 2025)

<b>Ngày</b>	<b>Tóm tắt hoạt động thực tập</b>	<b>Quy định khung tham chiếu (TCVN, QCVN, ISO...)</b>	<b>Kết quả hoạt động</b>	<b>Phân tích, giải thích, kết luận</b>	<b>Xác nhận của CBHD</b>
28/07/2025	Cài đặt và cấu hình WordPress Multisite trên Ubuntu, tìm hiểu các bước cơ bản.	TCVN 11973:2018	Cài đặt WordPress Multisite thành công, nắm được các bước cơ bản.	Tiến độ tốt, cần học thêm cách quản lý nhiều site.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên)
29/07/2025	Tích hợp domain cho WordPress Multisite, kiểm tra kết nối domain.	ISO/IEC 20000 - 1:2018	Tích hợp domain thành công, kết nối ổn định.	Hiệu quả, cần chú ý đến cấu hình DNS.	
30/07/2025	Tối ưu hóa website WordPress (cache, nén file, tối ưu hình ảnh).	TCVN 9261:2012	Áp dụng cache và nén file thành công, tối ưu hình ảnh cơ bản.	Kỹ năng cải thiện, cần thử nghiệm thêm với lưu lượng lớn.	
31/07/2025	Nghiên cứu về SEO, tìm hiểu các công cụ và kỹ thuật cơ bản.	QCVN 81:2014/BTTTT	Hiểu các khái niệm SEO cơ bản và công cụ như Google Analytics.	Tiến độ ổn định, cần thực hành áp dụng vào website.	
01/08/2025	Thực hành tối ưu hóa SEO cơ bản trên website WordPress, kiểm tra kết quả.	ISO 27001:2013	Thực hiện SEO cơ bản, cải thiện thứ hạng tìm kiếm ban đầu.	Kết quả khả quan, cần theo dõi lâu dài để đánh giá hiệu quả.	

**Tuần 6** (từ ngày 4 tháng 8 năm 2025 đến ngày 10 tháng 8 năm 2025)

<b>Ngày</b>	<b>Tóm tắt hoạt động thực tập</b>	<b>Quy định khung tham chiếu (TCVN, QCVN, ISO...)</b>	<b>Kết quả hoạt động</b>	<b>Phân tích, giải thích, kết luận</b>	<b>Xác nhận của CBHD</b>
04/08/2025	Nghiên cứu về Docker, tìm hiểu khái niệm và lợi ích cơ bản.	TCVN 11973:2018	Hiểu khái niệm Docker và các lợi ích cơ bản.	Tiến độ tốt, cần nghiên cứu thêm cách triển khai thực tế.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên)
05/08/2025	Cài đặt và cấu hình Docker trên Ubuntu, kiểm tra môi trường.	ISO/IEC 20000 - 1:2018	Cài đặt Docker thành công, kiểm tra môi trường ổn định.	Hiệu quả, cần học thêm cách quản lý Docker.	
06/08/2025	Tạo container cơ bản sử dụng Docker, thực hành các lệnh cơ bản.	TCVN 9261:2012	Tạo container thành công, nắm được các lệnh như docker run.	Kỹ năng cải thiện, cần thực hành thêm với nhiều container.	
07/08/2025	Tích hợp Docker với WordPress, cấu hình môi trường container.	QCVN 81:2014/BTTTT	Tích hợp Docker với WordPress thành công, cấu hình ổn định.	Tiến độ tốt, cần chú ý bảo mật container.	
08/08/2025	Thử nghiệm triển khai ứng dụng WordPress trên Docker, kiểm tra hiệu suất.	ISO 27001:2013	Triển khai ứng dụng thành công, hiệu suất ban đầu tốt.	Kết quả khả quan, cần tối ưu hóa thêm hiệu suất và bảo mật.	

**Tuần 7** (từ ngày 11 tháng 8 năm 2025 đến ngày 15 tháng 8 năm 2025)

<b>Ngày</b>	<b>Tóm tắt hoạt động thực tập</b>	<b>Quy định khung tham chiếu (TCVN, QCVN, ISO...)</b>	<b>Kết quả hoạt động</b>	<b>Phân tích, giải thích, kết luận</b>	<b>Xác nhận của CBHD</b>
11/08/2025	Nghiên cứu về Kubernetes, tìm hiểu khái niệm và lợi ích cơ bản.	TCVN 11973:2018	Hiểu khái niệm Kubernetes và các lợi ích cơ bản.	Tiến độ tốt, cần nghiên cứu thêm cách triển khai thực tế.	Dữ liệu, thông tin trung thực <input type="checkbox"/> / không trung thực <input type="checkbox"/> Xác nhận của CBHD (Ký tên và họ tên)
12/08/2025	Cài đặt và cấu hình Kubernetes trên môi trường ảo, kiểm tra môi trường.	ISO/IEC 20000 - 1:2018	Cài đặt Kubernetes thành công, kiểm tra môi trường ổn định.	Hiệu quả, cần học thêm cách quản lý cluster.	
13/08/2025	Triển khai ứng dụng WordPress trên Kubernetes, cấu hình các thành phần cơ bản.	TCVN 9261:2012	Triển khai WordPress trên Kubernetes thành công, cấu hình cơ bản hoàn tất.	Kỹ năng cải thiện, cần tối ưu hóa hiệu suất.	
14/08/2025	Kiểm tra hiệu suất ứng dụng WordPress trên Kubernetes, ghi nhận kết quả.	QCVN 81:2014/BTTTT	Kiểm tra hiệu suất thành công, ghi nhận kết quả ban đầu.	Tiến độ ổn định, cần cải thiện bảo mật và tối ưu hóa thêm.	
15/08/2025	Tổng kết thực tập, báo cáo kết quả với giảng viên hướng dẫn, nộp báo cáo cuối kỳ.	ISO 9001:2015	Hoàn thành báo cáo	Quá trình thực tập hiệu quả, đạt được mục tiêu đề ra.	



## PHẦN 5 BÁO CÁO PROJECT THỰC TẬP

**Tên dự án - project: Triển khai ứng dụng To-do List trên nền tảng Render theo mô hình 5 – 4 – 3 Cloud Computing NIST**

### 5.1 Tổng quan dự án

#### a. Xu hướng và vai trò của điện toán đám mây

Điện toán đám mây đang dẫn đầu xu hướng công nghệ năm 2025, mang lại khả năng triển khai ứng dụng nhanh chóng, tiết kiệm chi phí và dễ dàng mở rộng quy mô. Các mô hình dịch vụ như SaaS (Software as a Service) cung cấp ứng dụng trực tiếp cho người dùng cuối (ví dụ: Google Workspace), trong khi PaaS (Platform as a Service) hỗ trợ nhà phát triển triển khai mà không cần quản lý hạ tầng cơ sở (ví dụ: Render). Với sự gia tăng của làm việc từ xa và nhu cầu quản lý tác vụ cá nhân, điện toán đám mây trở thành giải pháp tối ưu, đặc biệt trong bối cảnh các doanh nghiệp và cá nhân tìm kiếm hiệu quả cao với chi phí thấp. Dự án này tập trung triển khai ứng dụng To-do List (SaaS) trên Render (PaaS), đánh giá theo chuẩn NIST SP 800-145, góp phần minh chứng vai trò thực tiễn của công nghệ đám mây trong phát triển ứng dụng hiện đại.

Các xu hướng nổi bật hiện tại:

- **Multi-cloud và Hybrid Cloud:** Các tổ chức ngày càng áp dụng chiến lược multi-cloud để tối ưu hóa hiệu suất, giảm chi phí và tránh phụ thuộc vào một nhà cung cấp duy nhất. Mô hình hybrid cloud cho phép doanh nghiệp kết hợp hạ tầng on-premises với public cloud để đạt được sự linh hoạt tối ưu.
- **Containerization và Microservices:** Công nghệ container như Docker và Kubernetes đã cách mạng hóa cách thức triển khai ứng dụng trên cloud, cho phép phát triển theo mô hình microservices với khả năng mở rộng và bảo trì tốt hơn.

- **Serverless Computing:** Mô hình Function-as-a-Service (FaaS) đang ngày càng phổ biến, cho phép developers tập trung vào logic ứng dụng mà không cần quan tâm đến quản lý server, từ đó giảm chi phí vận hành và tăng hiệu quả phát triển.
- **Edge Computing:** Xu hướng xử lý dữ liệu gần với nguồn phát sinh để giảm độ trễ và cải thiện trải nghiệm người dùng, đặc biệt quan trọng với các ứng dụng IoT và real-time.

b. Mục tiêu dự án

Xây dựng ứng dụng web SaaS đơn giản (To-do List) sử dụng Python Flask và PostgreSQL để quản lý tác vụ cơ bản.

Triển khai ứng dụng trên nền tảng PaaS Render (Public Cloud) với tính năng tự động triển khai từ GitHub.

Đánh giá ứng dụng dựa trên mô hình 5-4-3 của NIST (5 đặc tính, 4 mô hình triển khai, 3 mô hình dịch vụ), đảm bảo tính linh hoạt, khả năng mở rộng và tuân thủ chuẩn điện toán đám mây.

## 5.2 Cơ sở lý thuyết

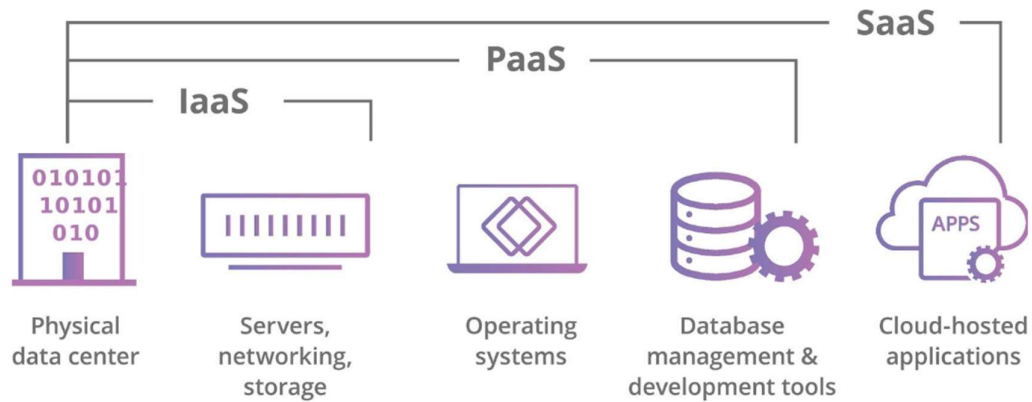
a. Điện toán đám mây theo NIST SP 800-145

Điện toán đám mây (Cloud Computing) là một trong những xu hướng quan trọng nhất của công nghệ thông tin hiện nay, được ứng dụng rộng rãi trong cả học tập, nghiên cứu và thực tiễn doanh nghiệp. Để thống nhất cách hiểu về khái niệm này, Viện Tiêu chuẩn và Công nghệ Hoa Kỳ (NIST – National Institute of Standards and Technology) đã công bố tài liệu NIST Special Publication 800-145. Trong tài liệu này, NIST đã đưa ra định nghĩa chuẩn mực, được nhiều quốc gia, tổ chức và doanh nghiệp công nghệ lớn trên thế giới áp dụng.

Theo NIST SP 800-145, điện toán đám mây được định nghĩa là:

*“Một mô hình cho phép truy cập theo nhu cầu (on-demand) qua mạng tới một tập hợp tài nguyên tính toán có thể cấu hình được (chẳng hạn như mạng, máy chủ, lưu trữ, ứng dụng và dịch vụ). Các tài nguyên này có thể được cung cấp và thu hồi nhanh chóng với sự can thiệp quản lý tối thiểu từ nhà cung cấp dịch vụ.”*

Nói cách khác, thay vì phải mua sắm và quản lý trực tiếp hệ thống phần cứng – phần mềm, người dùng có thể thuê và sử dụng tài nguyên điện toán dưới dạng dịch vụ qua Internet, tương tự như việc sử dụng điện, nước hay viễn thông.



NIST chỉ ra rằng để một hệ thống được coi là “Cloud Computing” đúng nghĩa, nó cần có đủ **5 đặc tính cốt lõi** sau:

- **Tự phục vụ theo nhu cầu (On-demand self-service):** Người dùng có thể tự động yêu cầu và sử dụng tài nguyên điện toán (ví dụ: máy ảo, dung lượng lưu trữ, ứng dụng) mà không cần thông qua nhân viên quản trị hệ thống.  
*Ví dụ:* Khi đăng ký sử dụng Render hoặc AWS, người dùng có thể tạo server hoặc cơ sở dữ liệu mới chỉ bằng vài thao tác, không cần cấu hình máy chủ thủ công.
- **Truy cập diện rộng qua mạng (Broad network access):** Dịch vụ đám mây có thể truy cập từ nhiều thiết bị khác nhau như máy tính, điện thoại, máy tính bảng... thông qua kết nối Internet. Điều này giúp tăng tính linh hoạt và di động.  
*Ví dụ:* Người dùng có thể quản lý ứng dụng To-do List được triển khai trên Render bằng laptop hoặc điện thoại mà không cần cài đặt thêm gì.
- **Gộp tài nguyên (Resource pooling):** Các tài nguyên như máy chủ, dung lượng lưu trữ, băng thông được gộp chung và phân phối linh hoạt cho nhiều khách hàng khác nhau. Người dùng không cần quan tâm tài nguyên đó được đặt ở đâu, mà chỉ quan tâm đến việc có thể sử dụng dịch vụ.  
*Ví dụ:* Hệ thống Render sử dụng hạ tầng máy chủ tập trung, chia sẻ cho nhiều dự án khác nhau của sinh viên, doanh nghiệp, nhưng mỗi người dùng vẫn thấy mình có “tài nguyên riêng”.
- **Khả năng mở rộng nhanh chóng (Rapid elasticity):** Điện toán đám mây cho phép tăng hoặc giảm tài nguyên sử dụng một cách linh hoạt tùy theo nhu cầu. Về phía người dùng, tài nguyên này gần như “vô hạn”.

*Ví dụ:* Một ứng dụng To-do List có thể ban đầu chỉ cần 1 server nhỏ, nhưng khi số lượng người dùng tăng lên hàng nghìn, hệ thống Render có thể mở rộng để đáp ứng tải mà không cần cài đặt thêm máy chủ thủ công.

- **Dịch vụ đo lường (Measured service):** Mọi tài nguyên sử dụng đều được giám sát, đo lường và tính phí theo mức độ sử dụng (theo giờ chạy, dung lượng lưu trữ, băng thông...). Nhờ vậy, người dùng chỉ trả tiền cho những gì họ thực sự dùng.

*Ví dụ:* Render có gói miễn phí giới hạn tài nguyên. Khi người dùng cần nâng cấp (thêm RAM, CPU, băng thông), hệ thống sẽ tính phí dựa trên mức tăng thêm.

Theo NIST, có 4 mô hình triển khai điện toán đám mây:

- **Public Cloud (Đám mây công cộng):** Tài nguyên được sở hữu và quản lý bởi nhà cung cấp (như Render, AWS, Google Cloud). Người dùng thuê và sử dụng tài nguyên qua Internet.
- **Private Cloud (Đám mây riêng):** Được triển khai cho một tổ chức duy nhất, thường đặt tại trung tâm dữ liệu nội bộ.
- **Community Cloud (Đám mây cộng đồng):** Dùng chung cho một nhóm tổ chức có nhu cầu hoặc mục tiêu tương đồng (ví dụ: các trường đại học dùng chung hệ thống).
- **Hybrid Cloud (Đám mây lai):** Kết hợp Public và Private để tận dụng ưu điểm của cả hai.

NIST phân loại dịch vụ đám mây thành 3 mức:

- **Infrastructure as a Service (IaaS):** Cung cấp hạ tầng (máy ảo, lưu trữ, mạng). Người dùng tự cài hệ điều hành, middleware và ứng dụng.
- **Platform as a Service (PaaS):** Cung cấp nền tảng (runtime, framework, cơ sở dữ liệu) để người dùng triển khai ứng dụng mà không quản lý hạ tầng.
- **Software as a Service (SaaS):** Cung cấp trực tiếp phần mềm hoàn chỉnh cho người dùng cuối.

Trong dự án triển khai ứng dụng To-do List trên Render sử dụng PaaS để triển khai ứng dụng Django và PostgreSQL, vì vậy chỉ cần tập trung vào code và logic, không cần quản lý máy chủ phức tạp.

#### b. Mô hình SaaS và PaaS

Trong ba mô hình dịch vụ điện toán đám mây mà NIST định nghĩa (IaaS, PaaS, SaaS), thì SaaS (Software as a Service) và PaaS (Platform as a Service) là hai mô hình phổ biến nhất đối với sinh viên, lập trình viên và doanh nghiệp khi phát triển và triển khai ứng dụng.

- **SaaS – Software as a Service**

SaaS là mô hình cung cấp phần mềm dưới dạng dịch vụ. Thay vì phải cài đặt phần mềm trên máy tính cá nhân hoặc máy chủ, người dùng chỉ cần truy cập qua Internet (thường thông qua trình duyệt web). Nhà cung cấp dịch vụ SaaS chịu trách nhiệm toàn bộ từ hạ tầng, nền tảng, đến việc vận hành và bảo trì ứng dụng.

Đặc điểm chính của SaaS:

- Người dùng không cần quản lý gì ngoài việc sử dụng phần mềm.
- Truy cập dễ dàng qua Internet, hỗ trợ đa nền tảng (máy tính, điện thoại).
- Thanh toán linh hoạt theo gói thuê bao hoặc mức độ sử dụng.
- Cập nhật và bảo mật do nhà cung cấp đảm nhận.

*Ví dụ:*

- Gmail, Google Docs, Microsoft 365 → người dùng chỉ cần đăng nhập để sử dụng.
- Zoom, Slack → cung cấp trực tiếp dịch vụ liên lạc, cộng tác.
- Trello, Notion → phần mềm quản lý công việc dưới dạng dịch vụ SaaS.

Liên hệ với dự án: Nếu ứng dụng To-do List sau khi triển khai trên Render được đóng gói và cung cấp trực tiếp cho người dùng cuối (ví dụ: sinh viên chỉ cần đăng ký tài khoản và đăng nhập để quản lý công việc), thì khi đó ứng dụng đang vận hành theo mô hình SaaS.

- **PaaS – Platform as a Service**

PaaS là mô hình cung cấp nền tảng dưới dạng dịch vụ. Nhà cung cấp PaaS chịu trách nhiệm về hạ tầng vật lý, máy chủ, hệ điều hành và các phần mềm trung gian (middleware). Người dùng (lập trình viên, doanh nghiệp) chỉ cần tập trung vào viết code, phát triển và triển khai ứng dụng, mà không phải lo lắng về việc cài đặt server hay cấu hình mạng phức tạp.

Đặc điểm chính của PaaS:

- Cung cấp môi trường phát triển và triển khai ứng dụng sẵn sàng.
- Hỗ trợ nhiều ngôn ngữ lập trình và framework (Node.js, Python/Django, Ruby on Rails, Java Spring...).
- Tích hợp các dịch vụ cơ sở dữ liệu, load balancing, giám sát hiệu năng.
- Cho phép mở rộng dễ dàng theo nhu cầu.

*Ví dụ:*

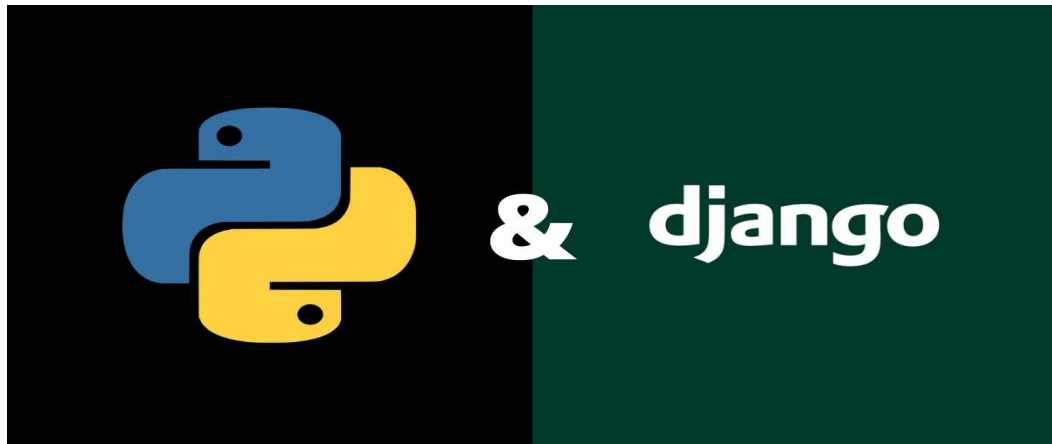
- Render, Heroku, Google App Engine, AWS Elastic Beanstalk → cho phép deploy ứng dụng web chỉ bằng vài lệnh hoặc vài click chuột.
- Firebase (Google) → PaaS chuyên cho mobile/web app, tích hợp sẵn authentication, database, hosting.

**Liên hệ với dự án:** Dự án To-do List sử dụng Render để triển khai ứng dụng Django (backend) và PostgreSQL (cơ sở dữ liệu). Đây là ví dụ điển hình của PaaS, vì:

- Render cung cấp sẵn môi trường server, Python runtime và PostgreSQL database.
- Nhóm không cần tự quản lý máy ảo, không phải cấu hình OS, network, firewall thủ công.
- Toàn bộ thời gian được tập trung vào viết code và triển khai ứng dụng.

c. Công nghệ triển khai

- **Django Framework:**



Python là một ngôn ngữ lập trình bậc cao, hướng đối tượng, thông dịch (interpreted language). Tính chất thông dịch nghĩa là Python không biên dịch thành mã máy trước khi chạy, mà mã nguồn được thực thi trực tiếp qua trình thông dịch. Điều này giúp tăng tính linh hoạt, dễ thử nghiệm và phát triển nhanh, nhưng đổi lại tốc độ có thể chậm hơn ngôn ngữ biên dịch như C/C++.

Ngoài ra, Python sở hữu hệ sinh thái thư viện phong phú, trong đó các thư viện web như **Django** hay **Flask** là các công nghệ quan trọng để xây dựng ứng dụng web hiện đại.

Django hoạt động dựa trên mô hình MVT (Model – View – Template), trong đó:

- **Model:** biểu diễn dữ liệu và logic nghiệp vụ, ánh xạ với cơ sở dữ liệu thông qua ORM.
- **View:** xử lý yêu cầu từ người dùng, gọi Model để lấy dữ liệu và trả kết quả.
- **Template:** định nghĩa giao diện hiển thị cho người dùng cuối.

Django sử dụng **ORM (Object Relational Mapping)** để chuyển đổi dữ liệu từ dạng bảng quan hệ (SQL) sang đối tượng Python. Đây là kỹ thuật giúp lập trình viên thao tác dữ liệu bằng ngôn ngữ lập trình mà không cần viết SQL trực tiếp.

Django tuân theo nguyên tắc **DRY (Don't Repeat Yourself)**: hạn chế lặp lại mã nguồn, tăng khả năng tái sử dụng.

Django là framework monolithic (nguyên khối), tức là nhiều tính năng được tích hợp sẵn (authentication, admin site, session, middleware). So với micro-framework như Flask, Django phù hợp hơn cho các ứng dụng cần hệ thống chặt chẽ và bảo mật cao.

- **PostgreSQL Database:**



PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, tuân thủ chuẩn **ACID**:

- **Atomicity**: mọi giao dịch đều là nguyên tử (hoặc thành công toàn bộ, hoặc thất bại hoàn toàn).
- **Consistency**: dữ liệu luôn ở trạng thái hợp lệ trước và sau giao dịch.
- **Isolation**: các giao dịch đồng thời không ảnh hưởng lẫn nhau.
- **Durability**: dữ liệu đã ghi nhận sẽ được lưu trữ bền vững ngay cả khi hệ thống gặp sự cố.

Điểm khác biệt giữa PostgreSQL và các RDBMS khác:

- **So với MySQL**: PostgreSQL hỗ trợ nhiều kiểu dữ liệu phức tạp hơn (JSONB, HSTORE, UUID), thích hợp cho các ứng dụng cần linh hoạt dữ liệu.
- **So với SQLite**: PostgreSQL hỗ trợ nhiều người dùng, nhiều kết nối đồng thời và khả năng mở rộng trên môi trường server. SQLite thường chỉ phù hợp cho ứng dụng đơn giản, không yêu cầu đa người dùng.

Ngoài ra, PostgreSQL sử dụng cơ chế MVCC (Multi-Version Concurrency Control) để quản lý song song, cho phép nhiều giao dịch đọc/ghi mà không cần khóa (lock) dữ liệu thường xuyên, giúp hiệu năng cao hơn.

- **Render Platform (PaaS):**



Render là một nền tảng Platform as a Service (PaaS). Khác với IaaS (Infrastructure as a Service) như AWS EC2 (nơi người dùng phải tự cài đặt hệ điều hành, web server, cơ sở dữ liệu), PaaS cung cấp một lớp trừu tượng cao hơn: người dùng chỉ cần đưa code, Render sẽ lo việc:

- Cung cấp máy chủ ảo hóa sẵn.
- Cài đặt runtime môi trường (Python, Node.js, v.v).
- Quản lý cơ sở dữ liệu PostgreSQL.
- Cấu hình tên miền, chứng chỉ SSL.
- Theo dõi và giám sát (monitoring).

Điều này giúp giảm đáng kể thời gian thiết lập môi trường, đồng thời phù hợp với mô hình học tập hoặc phát triển nhanh MVP (Minimum Viable Product).

Về kiến trúc, khi người dùng push code lên GitHub, Render sẽ:

- Tự động pull code,
- Build theo file requirements.txt (đối với Python)
- Deploy thành một Web Service.

Đây là mô hình CI/CD (Continuous Integration/Continuous Deployment) đơn giản nhưng hiệu quả, giúp ứng dụng được cập nhật liên tục mà không cần cấu hình thủ công phức tạp.

- **Công cụ hỗ trợ**

- **Git/GitHub:** quản lý phiên bản và lưu trữ mã nguồn. Về mặt lý thuyết, Git hoạt động theo mô hình phân tán (distributed version control), giúp mỗi thành viên có bản sao đầy đủ của repository.
- **Postman:** dùng để test API RESTful theo các phương thức HTTP (GET, POST, PUT, DELETE). Đây là bước đảm bảo tính đúng đắn của backend.
- **DevTools (Chrome, Firefox):** công cụ kiểm tra performance, đo tốc độ tải trang, số lượng request và dung lượng dữ liệu.



## 5.3 Phân tích và thiết kế hệ thống

### a. Yêu cầu hệ thống

Với mục tiêu xây dựng một ứng dụng web To-do List hỗ trợ người dùng quản lý công việc một cách hiệu quả. Em đã phân tích các yêu cầu chức năng (liên quan đến tính năng cụ thể) và phi chức năng (liên quan đến chất lượng hệ thống) để đảm bảo ứng dụng đáp ứng nhu cầu thực tế và phù hợp với mô hình điện toán đám mây.

#### Chức năng:

- **Xác thực người dùng:**

- **Đăng ký:** Người dùng có thể tạo tài khoản mới thông qua form đăng ký trên giao diện web (endpoint /register). Form yêu cầu nhập username, email và password, được xử lý bởi `django.contrib.auth` và lưu vào bảng `auth_user` trong PostgreSQL. Em thiết kế để đảm bảo username duy nhất và password được mã hóa (sử dụng PBKDF2 với salt).
- **Đăng nhập:** Người dùng đăng nhập bằng username và password qua endpoint /login, sử dụng `authenticate` để kiểm tra thông tin. Sau khi đăng nhập, hệ thống tạo session để theo dõi trạng thái người dùng.
- **Đăng xuất:** Người dùng đăng xuất qua endpoint /logout, xóa session và redirect về trang đăng nhập. Chức năng này đảm bảo bảo mật và trải nghiệm mượt mà.

- **Quản lý task:**

- **Thêm task:** Người dùng đã đăng nhập có thể thêm task mới thông qua form tại endpoint POST /add. Form yêu cầu nhập nội dung task (`CharField`, tối đa 200 ký tự) và hệ thống tự động ghi nhận thời gian tạo (`created_at`, `DateTimeField`). Task được lưu vào bảng `Task` với liên kết đến người dùng (`user`, `ForeignKey`).
- **Chỉnh sửa task:** Người dùng có thể chỉnh sửa nội dung task qua endpoint POST /edit/<id>, cập nhật record trong PostgreSQL. Chỉ người tạo task được phép chỉnh sửa, đảm bảo quyền riêng tư.
- **Đánh dấu hoàn thành/chưa hoàn thành:** Người dùng có thể thay đổi trạng thái task (`completed`, `BooleanField`) qua endpoint POST /toggle/<id>. Task hoàn thành được hiển thị với giao diện khác (ví dụ: gạch ngang), giúp dễ theo dõi.
- **Xóa task:** Người dùng xóa task qua endpoint GET /delete/<id>, loại bỏ record khỏi bảng `Task`. Hệ thống kiểm tra quyền sở hữu để tránh xóa nhầm.

- **API endpoints:**

- **Lấy thông tin người dùng:** Endpoint `/api/profile_info/user=<username>/` trả về JSON chứa thông tin người dùng (username, email, ngày tham gia), sử dụng `djangorestframework`.

- **Lấy danh sách task của người dùng:**

Endpoint `/api/user_tasks/user=<username>/` trả về JSON chứa danh sách task của người dùng, bao gồm nội dung, thời gian tạo và trạng thái hoàn thành. API được bảo vệ bằng xác thực (chỉ user đã đăng nhập truy cập được).

- **Hiển thị danh sách task:** Endpoint GET / hiển thị danh sách task của người dùng hiện tại, sắp xếp theo thời gian tạo (mới nhất ở đầu). Danh sách bao gồm nội dung task, thời gian tạo (`created_at`) và trạng thái hoàn thành (`completed`), được lấy từ PostgreSQL bằng `Task.objects.filter(user=request.user).order_by('-created_at')`. Giao diện hiển thị rõ ràng, dễ đọc trên mọi thiết bị.

Các chức năng này được thiết kế đơn giản và hỗ trợ kiểm thử đồng thời với 5 user để đánh giá các đặc tính NIST như Resource Pooling và Broad Network Access.

**Phi chức năng:**

Các yêu cầu phi chức năng tập trung vào chất lượng hệ thống, đảm bảo ứng dụng hoạt động ổn định, an toàn và dễ tiếp cận trong môi trường điện toán đám mây:

- Truy cập từ mọi thiết bị: Ứng dụng phải hoạt động mượt mà trên PC, laptop, tablet và điện thoại di động, đáp ứng đặc tính Broad Network Access của NIST. Em sử dụng thiết kế responsive trong template HTML (`index.html`, `login.html`, `register.html`) với CSS media queries để tự động điều chỉnh kích thước form và danh sách task. Ví dụ, trên mobile, nút "Add Task" được phóng to để dễ chạm và danh sách task hiển thị dạng cột dọc. Em sẽ kiểm thử với 5 user trên PC (Chrome) và mobile (Safari) để xác nhận tính năng này.
- Thời gian phản hồi < 2 giây: Mọi thao tác (đăng nhập, thêm/xóa task, gọi API) phải hoàn thành trong dưới 2 giây để đảm bảo trải nghiệm tốt. Em sử dụng PostgreSQL với index trên trường `created_at` để tối ưu truy vấn và Render tự động phân bổ tài nguyên để giảm độ trễ. Nếu thời gian vượt quá, em sẽ xem xét thêm cache (như Django cache framework với Redis). Yêu cầu này hỗ trợ Measured Service của NIST, được đánh giá qua Metrics của Render (request latency).
- Bảo mật cơ bản: Dữ liệu người dùng (username, email, password) và task (nội dung, thời gian, trạng thái) phải được lưu an toàn trên PostgreSQL. Em sử dụng:
  - Kết nối mã hóa: PostgreSQL kết nối qua `DATABASE_URL` với SSL.
  - Django ORM: Ngăn chặn SQL injection bằng cách sử dụng parameterized queries.
  - CSRF protection: Bảo vệ form thêm/chỉnh sửa task bằng CSRF token.
  - Mã hóa password: Password được lưu dưới dạng hash (PBKDF2) trong bảng `auth_user`.
  - Khóa bí mật: Sử dụng `SECRET_KEY` (tạo bằng `secrets.token_hex()`) để mã hóa session và dữ liệu nhạy cảm.

- HTTPS: Render cung cấp HTTPS tự động, đảm bảo an toàn khi truyền dữ liệu. Yêu cầu này đảm bảo dữ liệu không bị lộ khi nhiều user truy cập đồng thời, hỗ trợ Resource Pooling.
- Khả năng mở rộng: Hệ thống phải hỗ trợ nhiều người dùng đồng thời (ít nhất 5 user) mà không gặp lỗi, đáp ứng Rapid Elasticity của NIST. Render tự động scale tài nguyên khi tải tăng và PostgreSQL hỗ trợ truy vấn đồng thời hiệu quả.
- Dễ triển khai và sử dụng: Ứng dụng phải dễ cài đặt local (qua DJANGO\_ENV=DEV) và triển khai trên Render. Các bước cài đặt (migrate, createsuperuser) được tối ưu đơn giản, hỗ trợ On-demand Self-service.

b. Lựa chọn mô hình dịch vụ

Dựa trên yêu cầu chức năng và phi chức năng, em đã lựa chọn mô hình dịch vụ và triển khai phù hợp để đảm bảo ứng dụng To-do List hoạt động hiệu quả, tiết kiệm chi phí. Lựa chọn này dựa trên mô hình **5-4-3** của NIST SP 800-145, bao gồm 5 đặc tính, 4 mô hình triển khai và 3 mô hình dịch vụ.

- **Service Model:**

- **SaaS (Phần mềm như một Dịch vụ):** To-do List là một ứng dụng SaaS, cung cấp phần mềm quản lý task trực tiếp cho người dùng cuối qua trình duyệt web. Người dùng chỉ cần truy cập <https://todo-app.onrender.com> để đăng ký, đăng nhập và quản lý task mà không cần cài đặt phần mềm hay quản lý hạ tầng. Chức năng này đáp ứng **Broad Network Access** (truy cập đa thiết bị) và **Measured Service** (theo dõi hiệu suất qua Metrics của Render). API endpoints (/api/profile\_info, /api/user\_tasks) cũng là một phần của SaaS, cung cấp dữ liệu theo định dạng JSON cho các ứng dụng bên ngoài nếu cần.
- **PaaS (Nền tảng như một Dịch vụ):** Em sử dụng **Render** làm nền tảng PaaS để triển khai ứng dụng Django và cơ sở dữ liệu PostgreSQL. Render cung cấp môi trường chạy ứng dụng (Web Service), quản lý database và tự động scale tài nguyên, giúp em tập trung vào phát triển code mà không cần cấu hình máy chủ. PaaS hỗ trợ **On-demand Self-service** (tự tạo Web Service/DB qua Dashboard) và **Rapid Elasticity** (scale khi 5 user truy cập đồng thời).

- **Deployment Model:**

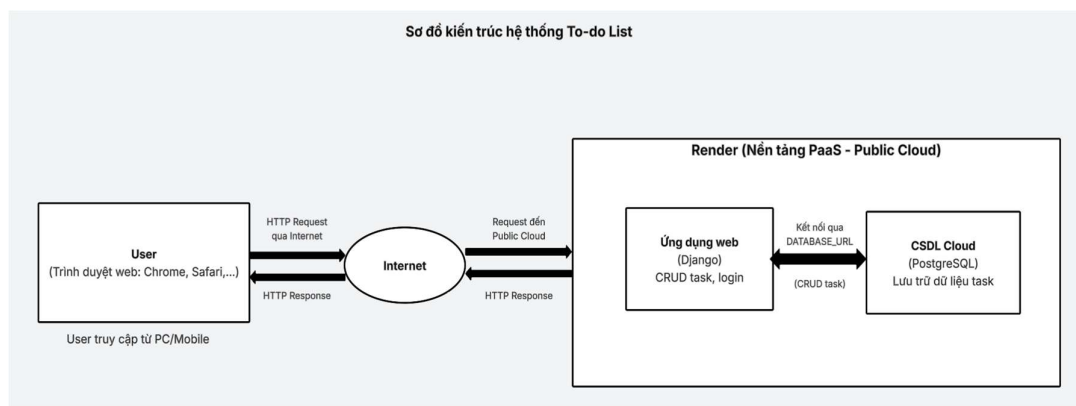
- **Public Cloud (Đám mây công cộng):** Em chọn triển khai trên Render, một nền tảng Public Cloud, nơi tài nguyên (CPU, bộ nhớ, database) được chia sẻ giữa nhiều người dùng. Public Cloud phù hợp với dự án vì miễn phí (free tier 750 giờ/tháng), dễ sử dụng và hỗ trợ **Resource Pooling** (chia sẻ tài nguyên an toàn khi 5 user test). So với Private Cloud (chi phí cao, cần quản lý hạ tầng) hoặc Hybrid Cloud (phức tạp), Public Cloud là lựa chọn tối ưu với thời gian và ngân sách hạn chế.

- **Lý do lựa chọn:**

- **SaaS** phù hợp với người dùng cuối, giúp họ quản lý task mà không cần kiến thức kỹ thuật. Giao diện responsive và API endpoints đảm bảo tính linh hoạt và khả năng tích hợp.
- **PaaS (Render)** giúp em triển khai nhanh (push code lên GitHub, deploy trong 5-10 phút), cung cấp Metrics/Logs để đánh giá NIST và tự động quản lý tài nguyên. Free tier đủ để test.
- **Public Cloud** tiết kiệm chi phí, hỗ trợ truy cập đa thiết bị và dễ mở rộng khi số user tăng, phù hợp với yêu cầu kiểm thử NIST (5 user, ~20 request).

Lựa chọn này đảm bảo ứng dụng To-do List đáp ứng các đặc tính NIST, như **On-demand Self-service** (tự tạo dịch vụ trên Render), **Broad Network Access** (truy cập từ PC/mobile) và **Rapid Elasticity** (scale tự động). Đồng thời, nó phù hợp vì Render có giao diện Dashboard thân thiện và không yêu cầu cấu hình server phức tạp.

### c. Sơ đồ kiến trúc hệ thống



Mô tả:

Hệ thống To-do List được triển khai trên nền tảng Render (PaaS – Public Cloud), kết hợp với dịch vụ CSDL Cloud (PostgreSQL). Sơ đồ kiến trúc cho thấy các thành phần và luồng xử lý chính như sau:

1. User (người dùng)
  - Người dùng có thể truy cập hệ thống từ nhiều thiết bị khác nhau (PC, laptop, smartphome) thông qua trình duyệt web (Chrome, Safari, Firefox,...).
  - Người dùng gửi yêu cầu (HTTP Request) qua Internet đến ứng dụng web.
2. Internet
  - Đóng vai trò là môi trường truyền tải, kết nối giữa người dùng và dịch vụ cloud.

- Yêu cầu từ người dùng được chuyển đến ứng dụng web trên Render và phản hồi (HTTP Response) sẽ quay trở lại trình duyệt.
3. Ứng dụng web (Django trên Render)
    - Ứng dụng To-do List được xây dựng bằng framework Django và triển khai trên Render theo mô hình Platform-as-a-Service (PaaS).
    - Ứng dụng cung cấp các chức năng chính:
      - Đăng nhập/đăng xuất (login/logout)
      - CRUD (Create, Read, Update, Delete) task: thêm, sửa, xóa và hiển thị danh sách công việc.
    - Ứng dụng kết nối đến CSDL Cloud để lưu trữ và truy xuất dữ liệu.
  4. CSDL Cloud (PostgreSQL trên Render)
    - Toàn bộ dữ liệu công việc (task) của người dùng được lưu trữ trong hệ quản trị cơ sở dữ liệu PostgreSQL.
    - Kết nối giữa ứng dụng web và CSDL được thực hiện thông qua biến môi trường DATABASE\_URL mà Render cung cấp.
    - Các thao tác CRUD từ ứng dụng web được chuyển thành câu lệnh SQL để xử lý trong CSDL.
  5. Luồng xử lý tổng quan
    - Người dùng nhập thao tác → Gửi HTTP Request qua Internet.
    - Render nhận request → Ứng dụng Django xử lý logic → Gửi truy vấn đến PostgreSQL.
    - PostgreSQL xử lý dữ liệu và trả kết quả về Django.
    - Django sinh HTTP Response → Trả kết quả hiển thị cho người dùng trên trình duyệt.

**Nhận xét:** Kiến trúc này thể hiện rõ mô hình 5-4-3 NIST Cloud Computing:

- 5 đặc tính: On-demand, Broad network access, Resource pooling, Elasticity, Measured service.
- 4 mô hình triển khai: Ở đây là Public Cloud.
- 3 mô hình dịch vụ: Sử dụng PaaS (Render) để triển khai ứng dụng và DBaaS (PostgreSQL) cho dữ liệu.

## 5.4 Triển khai

- a. Phát triển ứng dụng

**Cấu trúc thư mục:**

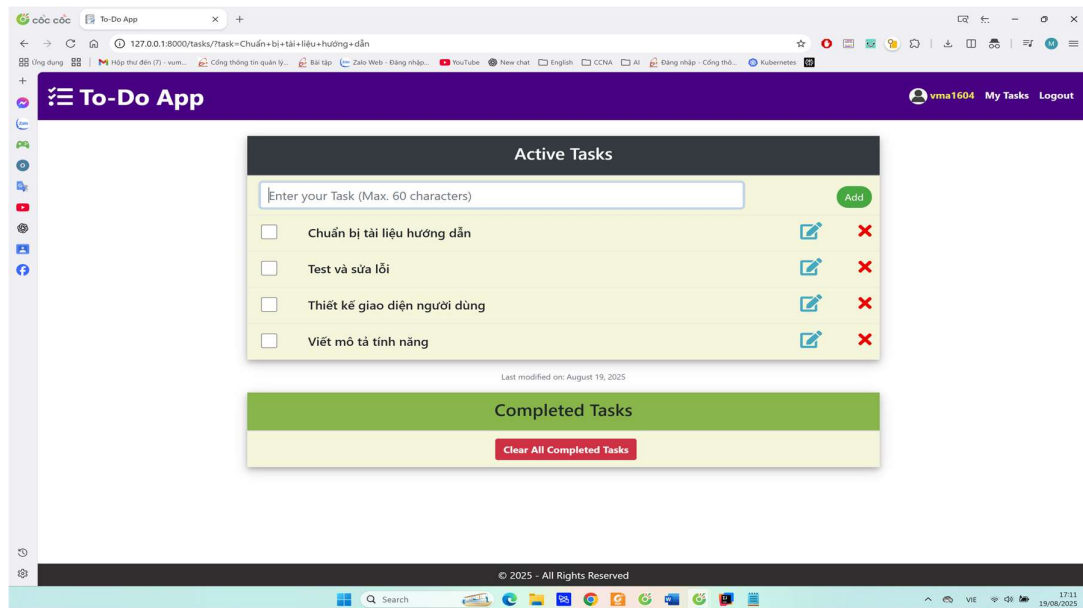
- project\_root/
  - .gitignore
  - .vscode/
    - launch.json
  - Procfile
  - README.md
  - manage.py
  - requirements.txt
  - media/
    - default.png
  - to\_do/
    - \_\_init\_\_.py
    - admin.py
    - apps.py
    - context\_processors.py
    - models.py
    - serializers.py
    - sitemaps.py
    - test.html
    - tests.py
    - urls.py
    - views.py
    - migrations/
      - \_\_init\_\_.py
    - static/
      - to\_do\_app/
        - body\_background.png
        - favicon.png
        - main.css
        - script.js
        - to\_do\_logo.png
        - base.html
        - index.html
        - tasks.html
        - test.html
- to\_do\_app/
  - \_\_init\_\_.py
  - asgi.py
  - urls.py
  - wsgi.py
  - media/
    - default.png
  - settings/
    - development.py
    - production.py
- users/
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - forms.py
  - models.py
  - serializers.py
  - signals.py
  - tests.py
  - views.py
  - migrations/
    - \_\_init\_\_.py
  - templates/
    - users/
      - edit\_profile.html
      - login.html
      - logout.html
      - password\_reset.html
      - password\_reset\_complete.html
      - password\_reset\_confirm.html
      - password\_reset\_done.html
      - profile.html
      - register.html

## Endpoint chính:

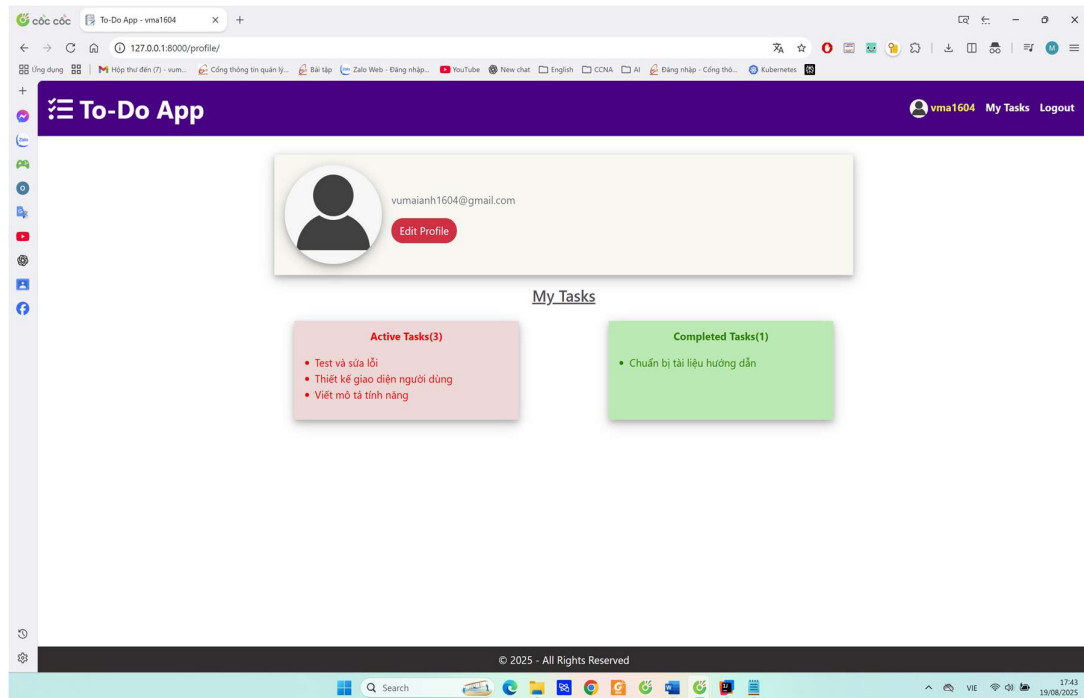
- GET /: Hiện thị danh sách task của người dùng hiện tại, sắp xếp theo created\_at (mới nhất ở đầu). Dữ liệu lấy từ `Task.objects.filter(user=request.user).order_by('-created_at')`.
- POST /add: Thêm task mới, lấy nội dung từ form (`request.POST['task']`), lưu vào bảng Task với `user=request.user`.
- POST /edit/<id>: Cập nhật nội dung task, kiểm tra quyền sở hữu (`Task.user == request.user`).
- POST /toggle/<id>: Thay đổi trạng thái completed, cập nhật giao diện (gạch ngang nếu hoàn thành).
- GET /delete/<id>: Xóa task, kiểm tra quyền sở hữu trước khi gọi `Task.objects.get(id=id).delete()`.
- GET /api/profile\_info/user=<username>/: Trả về JSON với thông tin người dùng (username, email, ngày tham gia), dùng `rest_framework.serializers`.
- GET /api/user\_tasks/user=<username>/: Trả về JSON danh sách task của người dùng, bao gồm nội dung, thời gian tạo, trạng thái.

## Giao diện:

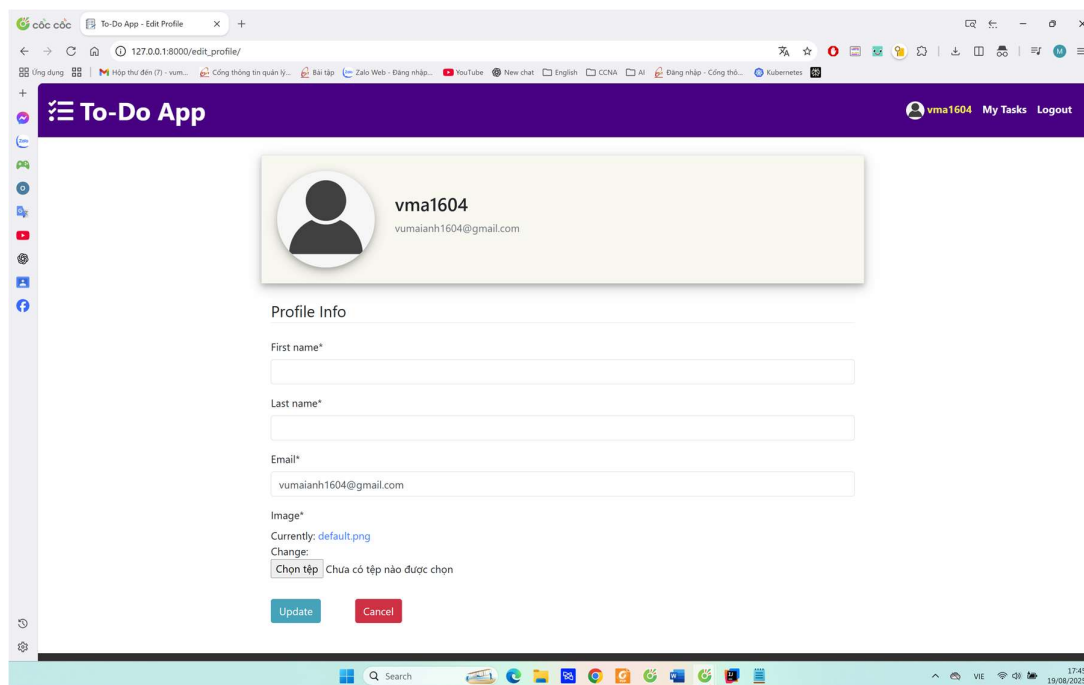
Giao diện responsive, hỗ trợ PC và mobile, sử dụng django-crispy-forms cho form đăng ký/đăng nhập và CSS media queries trong styles.css để điều chỉnh kích thước nút/form trên thiết bị nhỏ.



Hình 1: Giao diện To-do App trên localhost



Hình 2: Giao diện profile & my tasks trên localhost

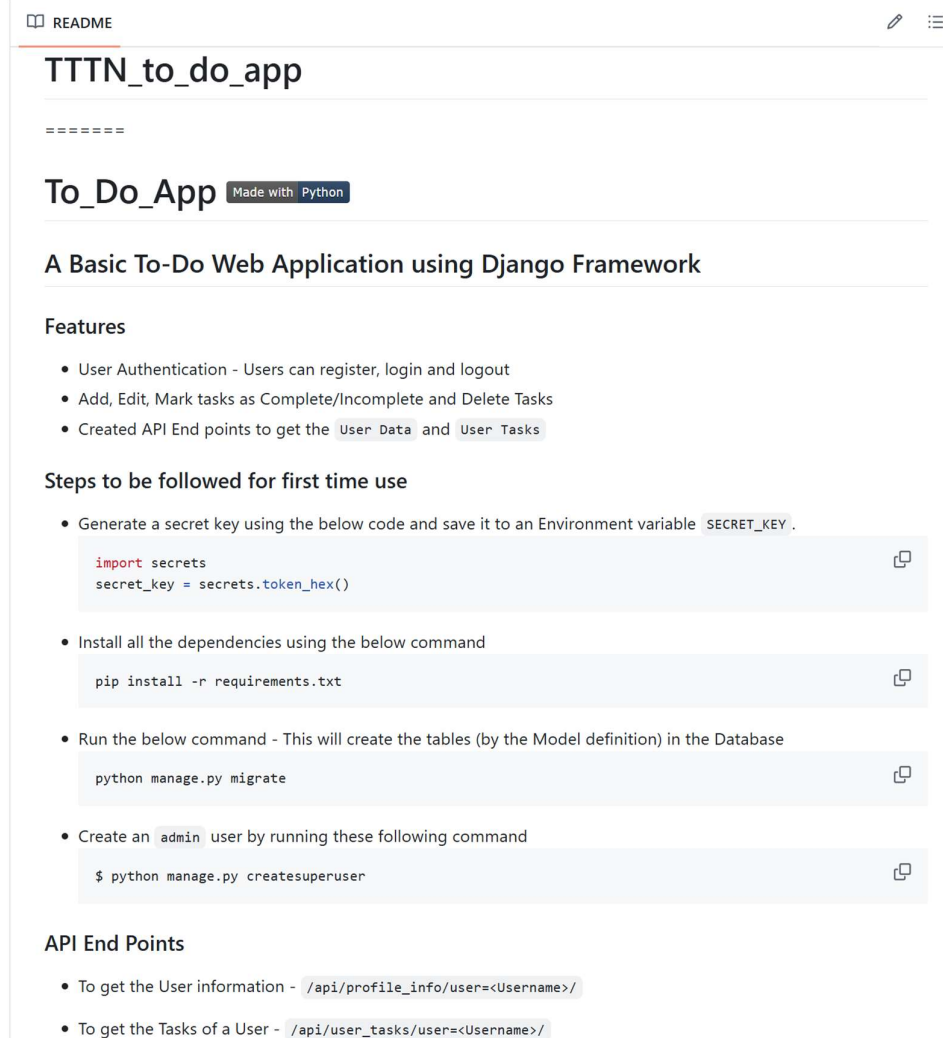


Hình 3: Giao diện edit\_profile trên localhost



## Quy trình triển khai:

1. **Clone mã nguồn:** Tải source code từ GitHub ([https://github.com/Vma-1122/TTTN\\_Todo\\_app.git](https://github.com/Vma-1122/TTTN_Todo_app.git)) bằng IntelliJ IDEA.
2. **Cấu hình môi trường local:**



The screenshot shows the README file for a project named 'TTTN\_to\_do\_app'. It is a 'To\_Do\_App' made with Python, described as 'A Basic To-Do Web Application using Django Framework'. The 'Features' section lists: User Authentication (register, login, logout), Add/Edit/Mark tasks as Complete/Incomplete/Delete, and API endpoints for User Data and User Tasks. The 'Steps to be followed for first time use' section includes: 1. Generating a secret key using a provided code snippet and saving it as SECRET\_KEY. 2. Installing dependencies with 'pip install -r requirements.txt'. 3. Running 'python manage.py migrate' to create database tables. 4. Creating an 'admin' user with 'python manage.py createsuperuser'. The 'API End Points' section lists: '/api/profile\_info/user=<Username>/' for user information and '/api/user\_tasks/user=<Username>/' for user tasks.

TTTN\_to\_do\_app

=====

To\_Do\_App Made with Python

A Basic To-Do Web Application using Django Framework

Features

- User Authentication - Users can register, login and logout
- Add, Edit, Mark tasks as Complete/Incomplete and Delete Tasks
- Created API End points to get the `User Data` and `User Tasks`

Steps to be followed for first time use

- Generate a secret key using the below code and save it to an Environment variable `SECRET_KEY`.

```
import secrets
secret_key = secrets.token_hex()
```

- Install all the dependencies using the below command

```
pip install -r requirements.txt
```

- Run the below command - This will create the tables (by the Model definition) in the Database

```
python manage.py migrate
```

- Create an `admin` user by running these following command

```
$ python manage.py createsuperuser
```

API End Points

- To get the User information - `/api/profile_info/user=<Username>/`
- To get the Tasks of a User - `/api/user_tasks/user=<Username>/`

- **Kết quả:**

- Ứng dụng chạy ổn định trên local, hỗ trợ đầy đủ chức năng (xác thực, CRUD task, API).
- Giao diện responsive, hoạt động tốt trên PC (Chrome) và mobile (Safari).
- Sẵn sàng triển khai lên Render, với cấu hình PostgreSQL và dependencies tối ưu.

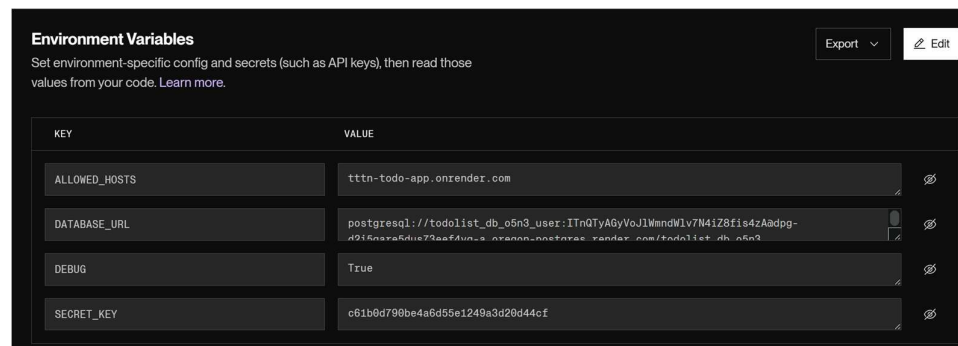
b. Triển khai trên Render

1. **Chuẩn bị mã nguồn:**

- Đảm bảo các file cần thiết (requirements.txt, Procfile, settings.py) đã được cấu hình sẵn.
- Push mã nguồn lên GitHub:
  - `git add .`
  - `git commit -m "Todoapp_update"`
  - `git push origin main`

## 2. Tạo Web Service trên Render:

- Truy cập dashboard.render.com, đăng nhập bằng tài khoản GitHub.
- Chọn **New > Web Service**, liên kết với repository GitHub (todo-app).
- Cấu hình Web Service:
  - Name: TTTN\_Todo\_app.
  - Environment: Python.
  - Build Command: `./build.sh`.
  - Start Command: `gunicorn to_do_app.wsgi`
  - Plan: Free (750 giờ/tháng, đủ để test NIST trong 1 ngày).
  - Environment Variables:



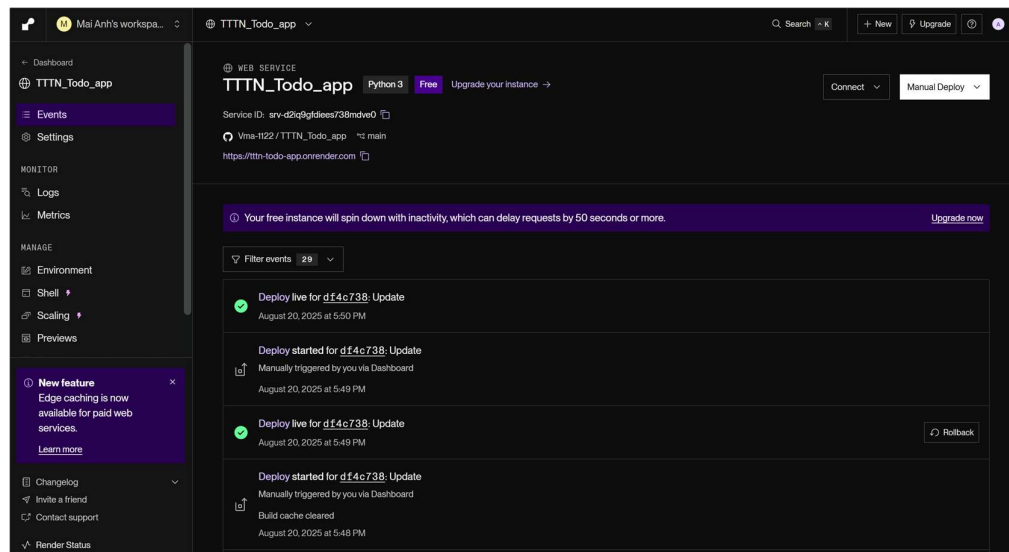
## 3. Tạo cơ sở dữ liệu PostgreSQL:

- Trong Render Dashboard, chọn **New > PostgreSQL**.
- Cấu hình:
  - Name: todolist\_db.
  - Database: todolist\_db.
  - User: admin.
- Copy **External Database URL**:
- Thêm DATABASE\_URL vào Environment Variables của Web Service.

- Chạy migrations trong Render Shell: *python manage.py migrate*

#### 4. Kiểm tra triển khai:

- Truy cập URL ứng dụng: <https://ttn-todo-app.onrender.com>
- Đăng nhập bằng admin user (tạo local hoặc qua createsuperuser trên Render Shell).
- Test các chức năng: đăng ký, đăng nhập, thêm/xóa task, gọi API (/api/profile\_info, /api/user\_tasks).
- Giao diện trên Render:



Hình: Giao diện deploy trên Render dashboard

- Kiểm tra Logs và Metrics trên Render Dashboard để xác nhận hoạt động ổn định.

#### Kết quả:

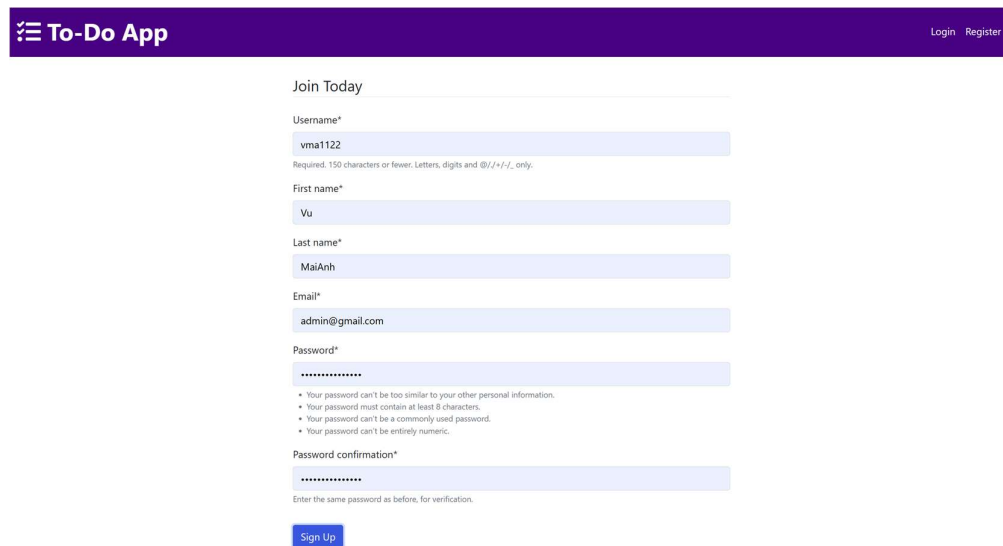
- Ứng dụng triển khai thành công trên Render, hỗ trợ xác thực người dùng, CRUD task và API endpoints.
- Giao diện responsive thời gian phản hồi < 2 giây (trừ cold start ~20-30s trên free tier).
- Metrics/Logs sẵn sàng cho kiểm thử NIST
- Hệ thống đáp ứng On-demand Self-service (tự tạo Web Service/DB) và Broad Network Access (truy cập qua HTTPS).

## 5.5 Kiểm thử và đánh giá

### a. Kiểm thử tính năng

Ứng dụng To-do List đã được kiểm thử trên cả môi trường local (localhost) và Render (free tier, public URL). Kết quả kiểm thử cho thấy các chức năng cơ bản hoạt động đúng như yêu cầu ban đầu:

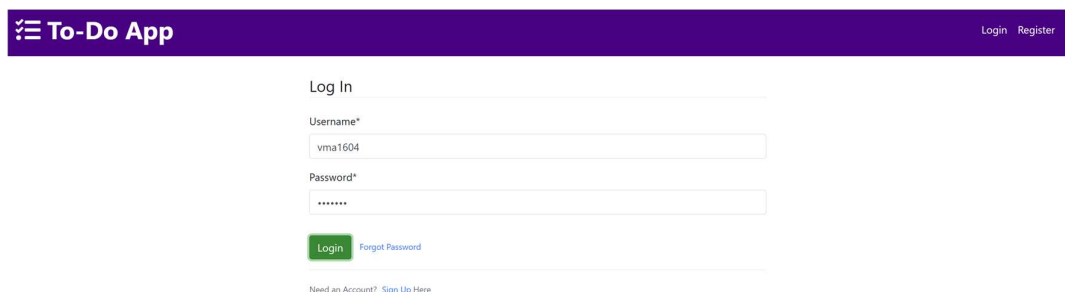
- **Đăng ký tài khoản mới:** Người dùng nhập username, email, password → hệ thống lưu vào cơ sở dữ liệu PostgreSQL. Test với nhiều user, dữ liệu được ghi nhận đúng.



The screenshot shows the 'Join Today' registration form for the 'To-Do App'. The form includes fields for Username, First name, Last name, Email, Password, and Password confirmation. The Username field contains 'vma1122', First name is 'Vu', Last name is 'MaiAnh', Email is 'admin@gmail.com', and Password is masked with asterisks. Below the Password field, there are four bullet points providing password requirements: 'Your password can't be too similar to your other personal information.', 'Your password must contain at least 8 characters.', 'Your password can't be a commonly used password.', and 'Your password can't be entirely numeric.' The Password confirmation field also contains masked text. A 'Sign Up' button is located at the bottom of the form. The top navigation bar is purple with the app name 'To-Do App' and links for 'Login' and 'Register'.

Hình : giao diện form đăng kí tài khoản mới

- **Đăng nhập/đăng xuất:** Xác thực thông tin người dùng, chỉ user đã đăng ký mới đăng nhập thành công. Session hết hiệu lực khi logout hoặc tắt trình duyệt.



The screenshot shows the 'Log In' form for the 'To-Do App'. It includes fields for Username and Password. The Username field contains 'vma1604' and the Password field is masked with asterisks. Below the Password field, there are two buttons: 'Login' (green) and 'Forgot Password' (blue). At the bottom, there is a link 'Need an Account? Sign Up Here'. The top navigation bar is purple with the app name 'To-Do App' and links for 'Login' and 'Register'.

Hình: giao diện đăng nhập/ đăng xuất

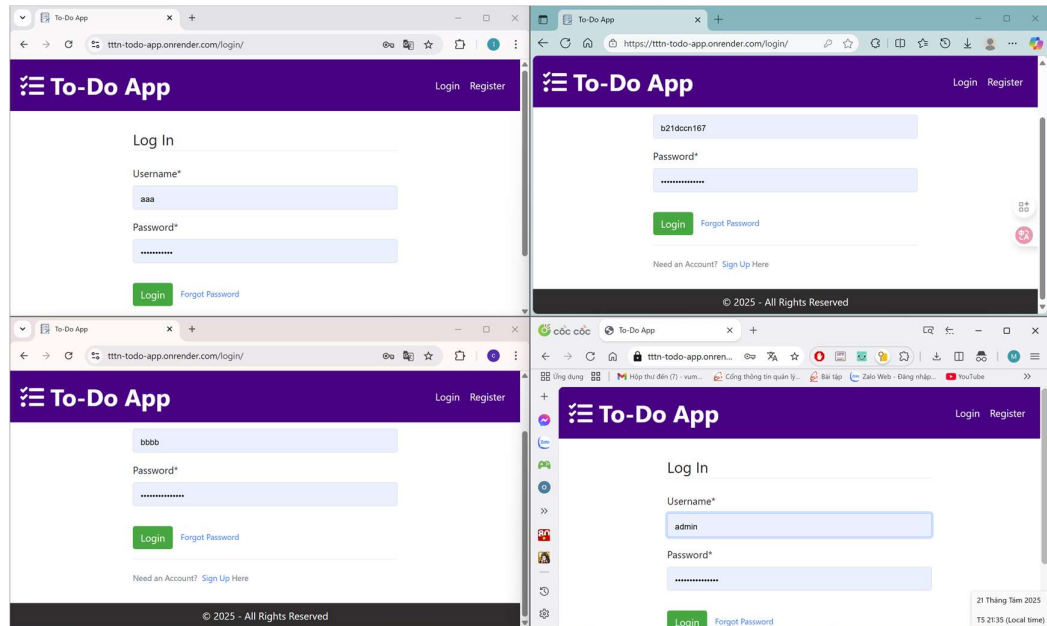
- **Quản lý hồ sơ cá nhân:** User có thể chỉnh sửa thông tin, đổi mật khẩu, reset password qua email.

*Hình: giao diện edit profile*

- **Quản lý task:** CRUD đầy đủ (tạo, xem danh sách, cập nhật, xóa). Các thao tác phản hồi nhanh, dữ liệu đồng bộ tức thì trong DB.

*Hình: Giao diện quản lý task*

- **Truy cập đa nền tảng:** Ứng dụng được test trên Chrome (Windows), Safari (iOS) và Firefox (Ubuntu). Giao diện responsive, các chức năng chạy ổn định.

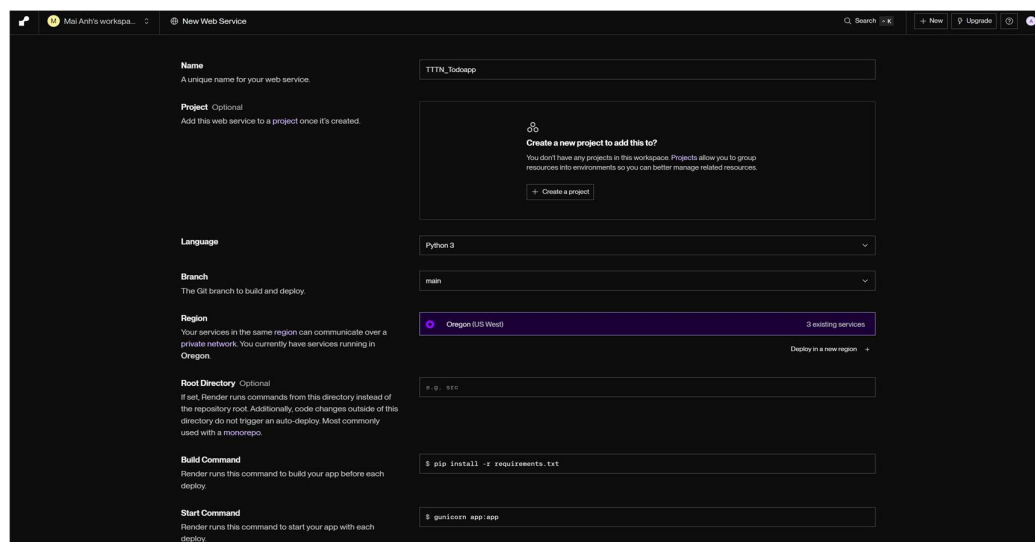


Kết quả kiểm thử xác nhận rằng toàn bộ luồng sử dụng cơ bản đều hoạt động đúng yêu cầu, không phát hiện lỗi logic nghiêm trọng trong quá trình thao tác.

b. Kiểm thử theo 5 đặc điểm cloud theo chuẩn NIST

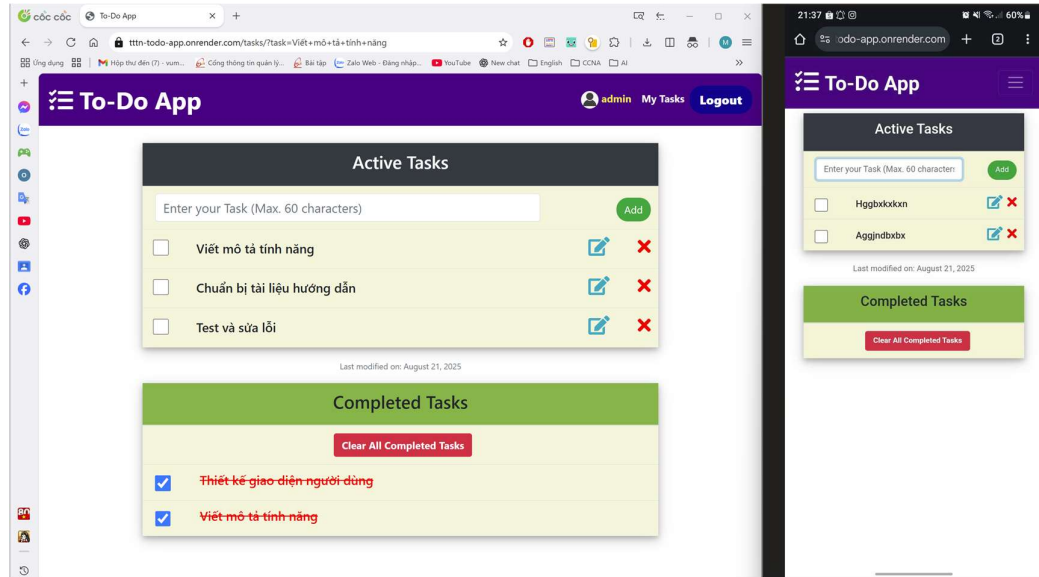
Để đánh giá ứng dụng To-do List triển khai trên Render trong bối cảnh điện toán đám mây, quá trình kiểm thử tập trung vào năm đặc điểm cơ bản theo chuẩn NIST:

- 1) **On-demand self-service (Tự phục vụ theo nhu cầu):** Người phát triển có thể khởi tạo dịch vụ web và cơ sở dữ liệu PostgreSQL hoàn toàn độc lập thông qua giao diện Render Dashboard, mà không cần quản trị viên can thiệp vào hạ tầng vật lý. Thao tác tạo, dừng hoặc khởi động lại dịch vụ đều thực hiện nhanh chóng và tự động, chứng minh đặc tính “tự phục vụ theo nhu cầu” của cloud.



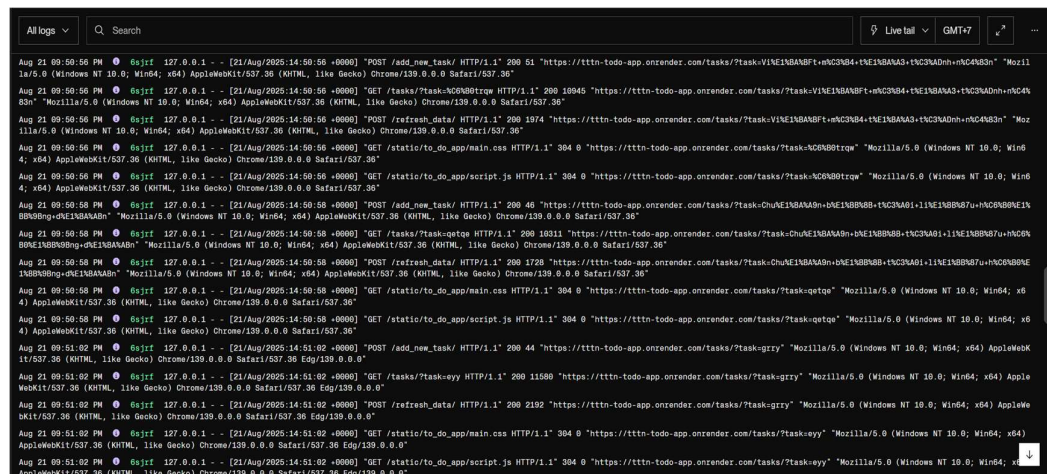
Hình 1 : Render Dashboard khi khởi tạo web service và DB

- 2) **Broad network access (Truy cập mạng rộng):** Ứng dụng được truy cập từ nhiều thiết bị và trình duyệt khác nhau, bao gồm Chrome trên Windows, Firefox trên Ubuntu và Safari trên iOS. Kiểm thử với 5 user đồng thời cho thấy tất cả các thao tác CRUD task đều hoạt động ổn định, giao diện responsive, dữ liệu đồng bộ tức thì trên PostgreSQL. Điều này phản ánh khả năng tiếp cận dịch vụ từ bất cứ nơi đâu có kết nối Internet.



Hình 2: Giao diện trên PC và mobile

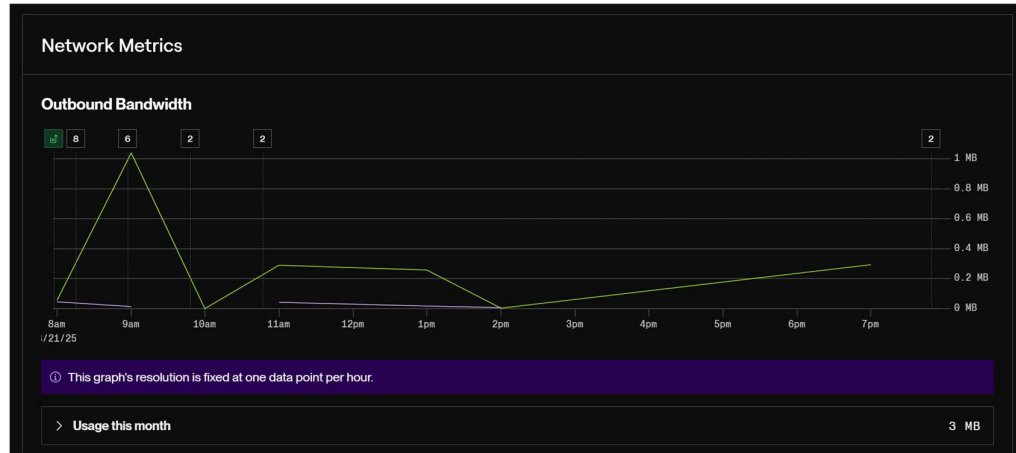
- 3) **Resource pooling (Chia sẻ tài nguyên):** Render quản lý tài nguyên tập trung và phân bổ động cho các ứng dụng. Trong quá trình kiểm thử, khi 5 user cùng thực hiện thao tác thêm/xóa task, Render Logs ghi nhận nhiều request đồng thời. Điều này chứng minh rằng ứng dụng chia sẻ tài nguyên backend một cách hiệu quả, đồng thời duy trì sự ổn định cho các user.



Hình 3: Render Logs hiển thị request đồng thời

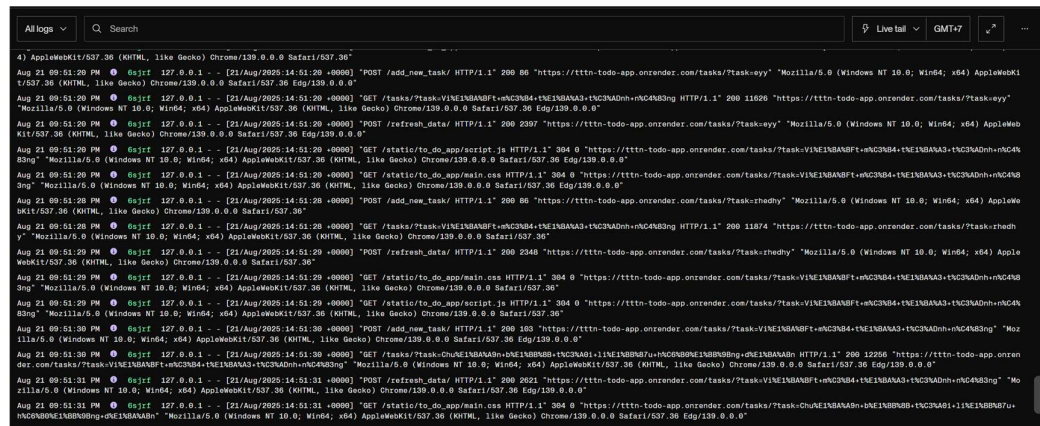


- 4) **Rapid elasticity (Khả năng co giãn nhanh):** Mặc dù gói free tier của Render không hỗ trợ auto-scale chi tiết, nhưng khi 5 user mở nhiều tab và thao tác liên tục, hệ thống vẫn phản hồi ổn định. Đây là minh chứng rằng trong phạm vi tài nguyên sẵn có, ứng dụng có thể phục vụ nhiều yêu cầu đồng thời mà không gặp lỗi. Cold start (khởi động khi idle) mất khoảng 20–30 giây, nhưng sau đó mọi thao tác diễn ra mượt mà.



Hình 4: Render Metrics

- 5) **Measured service (Dịch vụ đo đếm được):** Gói free tier không cung cấp thông tin chi tiết về CPU hoặc RAM. Tuy nhiên, thông qua Render Logs và Task Manager trên máy local, có thể theo dõi số lượng request và mức độ sử dụng CPU khi 5 user thực hiện thao tác đồng thời. Dữ liệu này cung cấp cơ sở để đánh giá tải cơ bản và giám sát dịch vụ, phù hợp với đặc tính “dịch vụ đo đếm được” trong cloud.



Hình 5: Render Logs



Name	Status	12% CPU	50% Memory	1% Disk	0% Network
<b>Apps (7)</b>					
CocCoc (27)	Running	0,1%	3.772,3 MB	0 MB/s	0 Mbps
Google Chrome (14)	Running	0,7%	576,0 MB	0,1 MB/s	0 Mbps
IntelliJ IDEA Ultimate Edition	Running	0,2%	2.858,4 MB	0,1 MB/s	0 Mbps
Microsoft Edge (12)	Running	0,1%	781,8 MB	0,1 MB/s	0 Mbps
Microsoft Word	Running	0,1%	138,2 MB	0 MB/s	0 Mbps
Task Manager	Running	0,3%	85,1 MB	0 MB/s	0 Mbps
UltraViewer_Desktop.exe (32 b...	Running	0,1%	8,5 MB	0 MB/s	0 Mbps
<b>Background processes (108)</b>					
Antimalware Core Service	Running	0%	6,1 MB	0 MB/s	0 Mbps
Antimalware Service Executable	Running	0,1%	163,9 MB	0 MB/s	0 Mbps
Application Frame Host	Running	0%	1,2 MB	0 MB/s	0 Mbps

Hình 6: Task Manager local

c. Đánh giá SaaS/PaaS trong thực tế

Ứng dụng To-do List khi triển khai trên Render mang đặc trưng rõ rệt của mô hình **PaaS (Platform as a Service)**. Người triển khai chỉ cần tập trung phát triển ứng dụng với Django, còn Render chịu trách nhiệm cung cấp và quản lý môi trường thực thi (web server, database, hạ tầng).

- **Về SaaS:** Ở góc độ người dùng cuối, hệ thống hoạt động như một ứng dụng SaaS vì chỉ cần truy cập qua trình duyệt là có thể sử dụng ngay. Người dùng không cần cài đặt phần mềm, toàn bộ dữ liệu và chức năng đều chạy trên nền tảng Render.
- **Về PaaS:** Đối với người phát triển, Render cung cấp môi trường tích hợp sẵn (dịch vụ database PostgreSQL, runtime Python/Django, cơ chế triển khai tự động từ GitHub). Điều này giúp giảm bớt khối lượng công việc quản trị hệ thống, tập trung vào phát triển logic nghiệp vụ.

So sánh với việc tự triển khai trên VPS hoặc server vật lý, PaaS (Render) giúp triển khai nhanh, đơn giản, tiết kiệm thời gian quản trị hệ thống; SaaS (To-do List) mang lại tiện ích thực tế cho người dùng cuối. Tuy nhiên, để dùng trong môi trường sản xuất lớn, cần nâng cấp gói dịch vụ và bổ sung giải pháp giám sát hiệu năng.

**Bảng so sánh SaaS và PaaS trong dự án**

<b>Tiêu chí</b>	<b>SaaS – Todo List</b>	<b>PaaS – Render</b>
<b>Định nghĩa</b>	Ứng dụng hoàn chỉnh cung cấp dịch vụ quản lý công việc trực tuyến qua trình duyệt, người dùng không cần cài đặt thêm phần mềm.	Nền tảng cho phép nhà phát triển triển khai, chạy và quản lý ứng dụng Django + PostgreSQL mà không cần quan tâm đến hạ tầng vật lý.
<b>Vai trò</b>	To-do List là dịch vụ SaaS: cung cấp giao diện quản lý task, API và xác thực user cho người dùng cuối.	Render là PaaS: cung cấp môi trường triển khai, runtime Python/Django, cơ sở dữ liệu PostgreSQL, quản lý logs và tài nguyên backend.
<b>Đối tượng sử dụng</b>	Người dùng cuối (sinh viên, nhân viên) truy cập bằng trình duyệt để quản lý công việc cá nhân.	Nhà phát triển triển khai, quản trị, cập nhật ứng dụng và database.
<b>Yêu cầu cài đặt</b>	Không yêu cầu cài đặt phần mềm, chỉ cần truy cập URL công khai.	Không cần cấu hình server thủ công; deploy tự động từ GitHub, Render build và chạy ứng dụng.
<b>Quản lý tài nguyên</b>	Người dùng không thấy hạ tầng vật lý, chỉ thao tác trên giao diện và dữ liệu được lưu trữ trên server.	Render quản lý CPU, RAM, database, logs; có dashboard quan sát trạng thái và số liệu cơ bản (free tier) hoặc chi tiết (paid tier).
<b>Khả năng mở rộng</b>	Giới hạn theo hạ tầng PaaS: với free tier, phục vụ được 5–10 user đồng thời; nâng cấp gói trả phí để phục vụ hàng trăm user.	Hỗ trợ mở rộng qua scale-out và nâng cấp gói; free tier có “scale-to-zero” nhưng không auto-scale chi tiết.
<b>Ưu điểm</b>	<ul style="list-style-type: none"> <li>- Dễ dùng, chỉ cần browser.</li> <li>- Quản lý task cá nhân nhanh chóng và trực quan.</li> </ul>	<ul style="list-style-type: none"> <li>- Triển khai nhanh, không cần quản lý server.</li> <li>- Tích hợp PostgreSQL, logs, backup cơ bản.</li> </ul>
<b>Hạn chế</b>	<ul style="list-style-type: none"> <li>- Chưa có tính năng nâng cao (nhắc nhở, phân loại, chia sẻ nhóm).</li> <li>- Hiệu năng phụ thuộc vào hạ tầng PaaS.</li> </ul>	<ul style="list-style-type: none"> <li>- Free tier giới hạn CPU/RAM, thiếu metrics chi tiết.</li> <li>- Dừng dịch vụ khi idle (scale-to-zero).</li> <li>- Không hỗ trợ auto-scale chi tiết cho free tier.</li> </ul>

## 5.6 Kết luận và hướng phát triển

### a. Kết luận

Quá trình triển khai ứng dụng **To-do List** được xây dựng bằng **Django framework** và cơ sở dữ liệu **PostgreSQL** trên nền tảng **Render** đã giúp hệ thống hoạt động ổn định, đáp ứng các chức năng cốt lõi. Ứng dụng hiện tại đã có các tính năng cơ bản như đăng ký, đăng nhập, đăng xuất, quản lý hồ sơ cá nhân, thêm/xóa/sửa công việc và đánh dấu trạng thái hoàn thành. Các tính năng này đảm bảo tính thực tiễn và phù hợp với nhu cầu sử dụng của một hệ thống quản lý công việc cá nhân.

Trong giai đoạn kiểm thử, ứng dụng cho thấy khả năng vận hành ổn định, giao diện phản hồi nhanh chóng và duy trì tính sẵn sàng khi truy cập từ nhiều thiết bị khác nhau. Việc triển khai trên Render đã giúp ứng dụng đạt được tính **Broad Network Access** – đặc trưng quan trọng trong định nghĩa cloud theo chuẩn NIST – vì hệ thống có thể truy cập qua trình duyệt với một đường dẫn cố định mà không cần cài đặt thêm phần mềm.

Ngoài ra, quá trình triển khai và sử dụng Render cũng phản ánh rõ nét một số đặc điểm khác của điện toán đám mây:

- **On-demand self-service:** dịch vụ cho phép khởi tạo, triển khai và quản lý ứng dụng hoàn toàn qua giao diện web của Render.
- **Resource pooling:** tài nguyên được Render quản lý tập trung, người dùng chỉ cần thao tác ở mức cấu hình ứng dụng.
- **Rapid elasticity:** dù ở gói miễn phí, Render vẫn hỗ trợ việc scale cơ bản và có thể nâng cấp dễ dàng khi cần thiết.
- **Measured service:** ở mức free tier chưa có số liệu chi tiết về CPU, RAM, nhưng có thể theo dõi log request và trạng thái dịch vụ để đánh giá hoạt động.

Kết quả triển khai và kiểm thử cho thấy mô hình này phù hợp với việc học tập, nghiên cứu cũng như phát triển ứng dụng ở quy mô cá nhân hoặc nhóm nhỏ. Việc triển khai thành công ứng dụng Django trên Render đã củng cố thêm kiến thức về lập trình web, triển khai ứng dụng trên môi trường cloud, đồng thời giúp vận dụng trực tiếp các lý thuyết về SaaS/PaaS và năm đặc điểm cloud theo chuẩn NIST vào thực tiễn.

### b. Hướng phát triển

- Nâng cấp và đa dạng hóa tính năng quản lý công việc:
  - Bổ sung phân loại công việc theo nhãn (tag), mức độ ưu tiên (priority) hoặc deadline cụ thể.
  - Tích hợp tính năng nhắc việc tự động qua email hoặc thông báo trên trình duyệt.
  - Cho phép chia sẻ danh sách công việc với người khác, hướng đến ứng dụng quản lý nhóm.
- Cải thiện giao diện và trải nghiệm người dùng (UX)
  - Ứng dụng hiện mới sử dụng HTML/CSS thuần; trong tương lai có thể áp dụng các thư viện giao diện như Bootstrap, TailwindCSS để tăng tính trực quan.

- Tích hợp JavaScript framework (React, VueJS) để giao diện hiện đại hơn, đồng thời hỗ trợ cập nhật dữ liệu theo thời gian thực (AJAX hoặc WebSocket).
- Tăng cường bảo mật và xác thực người dùng
  - Nâng cấp tính năng đăng nhập bằng cách thêm xác thực hai lớp (2FA), chính sách mật khẩu mạnh, hoặc đăng nhập thông qua OAuth2 (Google, GitHub).
  - Triển khai HTTPS với chứng chỉ SSL để tăng tính an toàn khi truyền dữ liệu.
- Kiểm thử tải và tối ưu hệ thống
  - Sử dụng công cụ kiểm thử tải như Apache JMeter hoặc Locust để đánh giá hiệu năng khi nhiều người dùng truy cập đồng thời.
  - Tối ưu hóa truy vấn cơ sở dữ liệu PostgreSQL và caching nhằm cải thiện tốc độ phản hồi.
  - Khi nâng cấp lên gói trả phí của Render, có thể tận dụng các số liệu chi tiết về CPU, RAM để tối ưu hệ thống dựa trên dữ liệu thực tế.
- CI/CD và triển khai đa nền tảng
  - Tích hợp GitHub Actions để xây dựng quy trình CI/CD, tự động deploy ứng dụng khi có thay đổi trên repository.
  - Thử nghiệm triển khai trên các nền tảng PaaS khác (Heroku, Railway, Vercel) để so sánh hiệu năng, độ ổn định và chi phí.
- Tích hợp dịch vụ bổ sung
  - Thêm tính năng đồng bộ công việc với các ứng dụng phổ biến như Google Calendar, Microsoft To Do.
  - Tích hợp API gửi email hoặc thông báo đẩy (push notification) để nâng cao trải nghiệm người dùng.

Tóm lại, ứng dụng To-do List trên Django triển khai qua Render tuy mới ở mức cơ bản, nhưng đã đạt được những tiêu chí quan trọng của một dịch vụ SaaS trong môi trường cloud. Đây là nền tảng quan trọng để tiếp tục mở rộng nghiên cứu, đồng thời phát triển ứng dụng theo hướng hiện đại, chuyên nghiệp và phù hợp hơn với nhu cầu thực tế.

**Tài liệu tham khảo:**

1. Bộ Thông tin và Truyền thông, 2020. *Báo cáo đánh giá nền tảng điện toán đám mây phục vụ Chính phủ điện tử*. [Online] Available at: <https://mic.gov.vn/Upload/BC-Danhgia-Nentang-Dien-toan-Dam-may-2020.pdf> [Accessed 12 August 2025]
2. Bizfly Cloud, 2023. *Giới thiệu về Bizfly Cloud: Giải pháp chuyển đổi số toàn diện*. [Online] Available at: <https://bizflycloud.vn/gioi-thieu> [Accessed 12 August 2025].
3. VCCorp, 2023. *Báo cáo thường niên 2022: Hành trình phát triển công nghệ và truyền thông*. [Online] Available at: <https://vccorp.vn/bao-cai-thuong-nien-2022.pdf> [Accessed 12 August 2025].
4. VCCorp, n.d. *Giới thiệu Công ty Cổ phần VCCorp*. [Online] Available at: <https://vccorp.vn/about-us> [Accessed 12 August 2025].
5. Django Software Foundation, 2023. Django Documentation. [online] Available at: <https://docs.djangoproject.com/> [Accessed 12 August 2025].
6. Python Software Foundation, 2023. Python Documentation. [online] Available at: <https://docs.python.org/> [Accessed 12 August 2025].
7. TCVN, 2012. TCVN 9261:2012 - Quản trị hệ thống CNTT. Hà Nội: Tổng cục Tiêu chuẩn Đo lường Chất lượng.
8. ISO/IEC, 2018. ISO/IEC 20000-1:2018 - Hệ thống quản lý dịch vụ CNTT. Geneva: Tổ chức Tiêu chuẩn hóa Quốc tế.
9. NIST SP 800-145, "The NIST Definition of Cloud Computing," 2011, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>.
10. Render Documentation, 2025, <https://render.com/docs>.