

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

## Project Proposal

Yêu cầu nhóm sinh viên hoàn thành tài liệu Project Proposal cho đồ án đã được giao theo biểu mẫu đính kèm.



Bộ môn Công nghệ phần mềm  
Khoa Công nghệ thông tin  
Đại học Khoa học tự nhiên TP HCM

# MỤC LỤC

Các nội dung chính	1
1 Bảng đánh giá thành viên	2
2 Phát biểu bài toán sơ lược	3
3 Giải pháp đề xuất	4
4 Kế hoạch phát triển	9
5 Kế hoạch nhân sự & chi phí	18

# Project Proposal

## Các nội dung chính

Mục tiêu tài liệu tập trung vào các chủ đề:

- ✓ Tạo ra tài liệu Project Proposal.
- ✓ Hoàn chỉnh tài liệu Project Proposal với các nội dung:
  - Phát biểu bài toán sơ lược
  - Giải pháp đề xuất
  - Kế hoạch phát triển
  - Kế hoạch nhân sự & chi phí
- ✓ Đọc hiểu tài liệu Project Proposal.

# 1

## Bảng đánh giá thành viên

MSSV	Họ Tên	% đóng góp (tối đa 100%)	Chữ ký
23120111	Phạm Quốc Nam Anh	20	
23120112	Thái Khắc Anh Tuấn	20	
23120121	Nguyễn Xuân Duy	20	
23120124	Nguyễn Minh Hiếu	20	
23120175	Huỳnh Thái Toàn	20	

# 2

## Phát biểu bài toán sơ lược

Trong xã hội hiện đại, việc lưu trữ và quản lý thông tin về gia đình, dòng họ thường được thực hiện bằng các phương pháp thủ công như sổ sách hoặc file Excel rời rạc. Cách làm này dễ dẫn đến tình trạng thất lạc dữ liệu, khó tra cứu, và đặc biệt gây khó khăn khi muốn hình dung mối quan hệ giữa các thành viên trong gia đình qua nhiều thế hệ.

Bài toán đặt ra là cần xây dựng một hệ thống phần mềm quản lý cây gia phả trực quan, giúp người dùng có thể:

- Dễ dàng lưu trữ, cập nhật và tra cứu thông tin của từng thành viên trong gia đình,
- Xây dựng và hiển thị sơ đồ quan hệ huyết thống giữa các thành viên,
- Chia sẻ thông tin với người thân hoặc các thành viên khác trong dòng họ,
- Đảm bảo an toàn và toàn vẹn dữ liệu khi có nhiều người cùng tham gia chỉnh sửa.

Hệ thống hướng tới việc cung cấp một nền tảng hiện đại, dễ sử dụng, giao diện thân thiện, giúp việc quản lý gia phả trở nên khoa học, trực quan và linh hoạt hơn, đồng thời góp phần lưu giữ và tôn vinh giá trị truyền thống gia đình Việt Nam.

# 3

## Giải pháp đề xuất

### 3.1 Phần mềm

#### 3.1.1. Danh sách các chức năng phần mềm

##### Yêu cầu chức năng (functional requirements)

Nhu cầu	Yêu cầu	Chi tiết
Là người quản lý, tôi muốn thêm thành viên mới vào gia phả (như con, vợ/chồng) để cập nhật hồ sơ gia tộc.	Tiếp nhận thành viên	Người dùng sẽ sử dụng biểu mẫu hồ sơ thành viên để nhập thông tin. Các trường bắt buộc bao gồm họ tên, ngày sinh, và phải chọn một "Thành viên cũ" để xác định mối quan hệ. Hệ thống phải tuân thủ các quy định nghiệp vụ: cung cấp danh sách chọn cho "Loại quan hệ" chỉ gồm 'Con' và 'Vợ Chồng' (có thể mở rộng thêm) , "Quê quán" , và "Nghề nghiệp". Hệ thống phải tự động xác thực "Ngày sinh" và "Ngày phát sinh" tuân theo quy luật tự nhiên (ví dụ: không phải là một ngày trong tương lai).
Là người quản lý, tôi muốn ghi lại thành tựu của các thành viên để lưu giữ cho các thế hệ sau.	Ghi nhận thành tích	Người dùng sử dụng biểu mẫu ghi nhận thành tích , trong đó phải chọn họ tên thành viên, loại thành tích và ngày phát sinh thành tích đó . Dựa trên quy định, hệ thống phải hiển thị một danh sách có định gồm các "Loại thành tích" để người dùng lựa chọn.
Là thành viên, tôi muốn xem danh sách các thành viên trong gia phả để biết về các thế hệ và quan hệ họ	Tra cứu thành viên	Người dùng chọn chức năng tra cứu. Hệ thống phải hiển thị một danh sách thành viên bao gồm các cột thông tin: Số thứ tự, Họ Tên, Ngày Sinh, thông tin về "Đời" (thế hệ), và thông tin "Cha/Mẹ" của thành viên đó.

hàng.		
Là người quản lý, tôi muốn cập nhật thông tin khi một thành viên qua đời để hồ sơ gia phả được đầy đủ.	Ghi nhận kết thúc	Người dùng sử dụng biểu mẫu ghi nhận kết thúc , chọn họ tên thành viên và nhập các thông tin: Ngày giờ mất, Nguyên nhân, Địa điểm mai táng . Theo quy định, hệ thống phải cung cấp các danh sách chọn cho "Nguyên nhân" và "Địa điểm mai táng". "Ngày giờ mất" phải được hệ thống kiểm tra để đảm bảo tuân thủ quy luật tự nhiên.
Là người quản lý, tôi muốn xem báo cáo tổng kết hàng năm về sự thay đổi (sinh, tử, kết hôn) và các thành tích của gia phả.	Lập báo cáo năm	Người dùng cần nhập khoảng thời gian "Từ năm" và "Đến năm" mà họ muốn xem báo cáo. Hệ thống sẽ kết xuất hai loại báo cáo riêng biệt: <ol style="list-style-type: none"> <li>1. Báo cáo tăng giảm thành viên: Thống kê số lượng sinh, số lượng kết hôn, và số lượng mất theo từng năm trong khoảng thời gian đã chọn.</li> <li>2. Báo cáo thành tích: Thống kê tổng số lượng của mỗi loại thành tích đã đạt được trong khoảng thời gian đó.</li> </ol>
Là người quản lý, tôi muốn tùy chỉnh được các danh mục (như nghề nghiệp, nguyên nhân mất) để phù hợp với thực tế.	Thay đổi quy định	Người dùng (với quyền quản trị) chọn chức năng thay đổi quy định . Hệ thống phải hiển thị giao diện cho phép người dùng (thêm/sửa/xóa) các giá trị trong các danh mục sau: <ul style="list-style-type: none"> <li>• Các mục liên quan đến hồ sơ: 'Loại quan hệ', 'Quê quán', 'Nghề nghiệp' .</li> <li>• Các mục liên quan đến ghi nhận kết thúc: 'Nguyên nhân mất', 'Địa điểm mai táng'.</li> </ul>

### Yêu cầu phi chức năng (Non-functional requirements)

Nhu cầu	Yêu cầu	Chi tiết

Là người sử dụng, tôi muốn hệ thống phải dễ nhìn và dễ sử dụng, và phải báo lỗi ngay nếu tôi nhập sai quy định (ví dụ: ngày mất trước ngày sinh).	Tính dễ sử dụng (Usability)	Giao diện của các biểu mẫu nhập liệu (như khi thêm thành viên, ghi nhận thành tích, hay ghi nhận kết thúc) phải được thiết kế trực quan và nhất quán. Hệ thống phải chủ động kiểm tra tính hợp lệ của dữ liệu ngay khi người dùng nhập, ví dụ: cảnh báo ngay nếu "Ngày giờ mất" được nhập trước "Ngày giờ sinh", thay vì đợi đến khi người dùng nhấn nút "Lưu".
Là người dùng, tôi muốn việc tra cứu và lập báo cáo phải nhanh chóng, kể cả khi gia phả có hàng ngàn thành viên.	Tính hiệu quả (Performance)	Hệ thống phải được tối ưu để đảm bảo tốc độ phản hồi nhanh. Thời gian xử lý cho các chức năng như tra cứu danh sách thành viên và lập báo cáo năm không được vượt quá 5 giây. Trong khi hệ thống đang tải dữ liệu, một thông báo "Đang xử lý..." phải được hiển thị rõ ràng.
Là người quản lý, tôi muốn dữ liệu gia phả phải chính xác và nhất quán; không thể có chuyện thành viên A là con của thành viên B, nhưng B lại sinh sau A.	Tính toàn vẹn dữ liệu (Integrity)	Hệ thống phải tự động thực thi các ràng buộc logic nghiệp vụ. Ví dụ, khi tiếp nhận một thành viên mới, hệ thống phải xác thực rằng "Ngày sinh" của con phải sau "Ngày sinh" của cha/mẹ. Tương tự, khi ghi nhận kết thúc, "Ngày giờ mất" phải luôn xảy ra sau "Ngày giờ sinh" của thành viên đó.
Là thành viên, tôi muốn thông tin cá nhân của mình phải được bảo vệ, và chỉ người quản trị mới có quyền thay đổi các thông tin quan trọng.	Tính bảo mật (Security)	Hệ thống bắt buộc phải có cơ chế đăng nhập để xác thực người dùng. Mật khẩu phải được lưu trữ trong cơ sở dữ liệu dưới dạng đã mã hóa. Cần thiết lập ít nhất hai vai trò: "Quản trị viên" và "Thành viên". Chỉ "Quản trị viên" mới có quyền truy cập chức năng thay đổi các quy định của hệ thống.

### 3.1.2. Kiến trúc tổng thể phần mềm

Hệ thống phần mềm được xây dựng dựa trên kiến trúc microservices và mô hình 3 lớp (client-server-database) để đảm bảo tính linh hoạt, khả năng mở rộng và dễ bảo trì. Các công nghệ



chính được sử dụng trong dự án bao gồm React, TailwindCSS, Next.js cho giao diện người dùng và logic phía máy khách, MySQL cho cơ sở dữ liệu, và Microsoft Azure cho triển khai. GitHub sẽ được sử dụng để quản lý mã nguồn và Figma cho thiết kế giao diện.

Hệ thống sẽ được thiết kế theo mô hình 3 lớp:

- **Presentation Layer (Client-side):**
  - **Next.js:** Là framework React cho phép phát triển ứng dụng web với khả năng kết xuất phía máy chủ (SSR) và tạo trang tĩnh (SSG), giúp cải thiện hiệu suất và SEO. Next.js sẽ đóng vai trò là nền tảng chính để xây dựng giao diện người dùng.
  - **React:** Thư viện JavaScript để xây dựng giao diện người dùng tương tác và hiệu quả.
  - **TailwindCSS:** Một framework CSS utility-first giúp xây dựng giao diện nhanh chóng và linh hoạt mà không cần viết CSS tùy chỉnh từ đầu.
- **Logic Layer (Server-side/API routes):**
  - **Next.js API Routes:** Sẽ được sử dụng để xây dựng các API backend, xử lý nghiệp vụ, xác thực người dùng và tương tác với cơ sở dữ liệu. Điều này giúp giữ cho toàn bộ ứng dụng (frontend và backend) trong một dự án Next.js duy nhất, đơn giản hóa việc phát triển và triển khai.
- **Data Layer (Database):**
  - **MySQL:** Hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở phổ biến, được lựa chọn để lưu trữ dữ liệu cây gia phả, thông tin người dùng, thành viên, quan hệ, thành tích, nghề nghiệp và địa điểm.

## 3.2 Phần cứng

### 3.2.1 Yêu cầu phần cứng cho môi trường phát triển:

**Máy tính cá nhân (Development Workstations):** Thiết bị cần chạy được [Node.js](#) và trình duyệt hiện đại

- **CPU:** Intel Core i5 thế hệ thứ 8 trở lên hoặc tương đương.
- **RAM:** Tối thiểu 4GB (khuyến nghị 8GB).
- **Ổ cứng:** 500MB dung lượng.
- **Hệ điều hành:** Windows 10/11 (64-bit), macOS Catalina trở lên, hoặc Ubuntu 20.04 LTS trở lên.

### 3.2.2 Yêu cầu phần cứng cho môi trường triển khai (Microsoft Azure):

Vì hệ thống sẽ được triển khai trên nền tảng đám mây Microsoft Azure, các yêu cầu về phần cứng vật lý sẽ được trừu tượng hóa và quản lý bởi Azure.

- **Máy chủ ứng dụng (Next.js Application Hosting):**

- **CPU:** Tối thiểu 1 vCore.
- **RAM:** Tối thiểu 1GB.
- **Storage:** Tối thiểu 20GB.
- **Máy chủ cơ sở dữ liệu (MySQL Database):**
  - **CPU:** Tối thiểu 1 vCore.
  - **RAM:** Tối thiểu 1GB.
  - **Storage:** Tối thiểu 20GB.

○

# 4 Kế hoạch phát triển

## 4.1 Phân tích yêu cầu

Quy trình thực hiện gồm 6 hoạt động chính:

### 4.1.1. Tiếp nhận biểu mẫu và quy định của đồ án

- Mục tiêu: Hiểu rõ yêu cầu, quy chuẩn trình bày và định dạng của đồ án từ Bộ môn.
- Mô tả công việc:
  - Nhận biểu mẫu và các hướng dẫn liên quan từ giảng viên.
  - Phân tích các mục cần hoàn thiện (phạm vi, chức năng, kế hoạch phát triển, nhân sự...).
- Thời gian: Tuần 3

### 4.1.2. Soạn thảo danh sách câu hỏi phỏng vấn

- Mục tiêu: Chuẩn bị bộ câu hỏi nhằm thu thập yêu cầu và làm rõ các điểm chưa xác định.
- Mô tả công việc:
  - Mỗi thành viên phụ trách một nhóm câu hỏi: chức năng, giao diện, người dùng, logic cây, phi chức năng...
  - Tổng hợp thành một danh sách hoàn chỉnh (khoảng 20–25 câu).
  - Rà soát ngôn ngữ để câu hỏi phù hợp với buổi phỏng vấn chính thức.
- Thời gian: Tuần 4

### 4.1.3. Tổ chức buổi phỏng vấn với giảng viên (Customer Interview)

- Mục tiêu: Làm rõ những yêu cầu còn mơ hồ, xác định phạm vi thật sự của hệ thống.
- Mô tả công việc:
  - Gửi mail hẹn lịch và chuẩn bị tài liệu phỏng vấn.
  - Tiến hành phỏng vấn trực tuyến qua Google Meet.
  - Ghi biên bản buổi trao đổi, thống nhất các điểm quan trọng về loại người dùng, quyền hạn, cách hiển thị, và yêu cầu giao diện.
- Thời gian: Tuần 5 (Ngày 1)

#### 4.1.4. Xác định đối tượng sử dụng và vai trò

- Mục tiêu: Xác định các loại người dùng của hệ thống và quyền hạn tương ứng.
- Mô tả công việc:
  - Phân tích thông tin thu được từ buổi phỏng vấn.
  - Xác định ba nhóm người dùng chính: Quản trị viên, Người tạo cây, Người xem.
  - Mô tả quyền và chức năng của từng nhóm (xem, chỉnh sửa, chia sẻ, quản lý dữ liệu...).
- Thời gian: Tuần 5 (Ngày 1 – 2)

#### 4.1.5. Tổng hợp và phân loại yêu cầu

- Mục tiêu: Tổ chức lại toàn bộ thông tin thành các nhóm yêu cầu cụ thể để dễ theo dõi và thiết kế.
- Mô tả công việc:
  - Phân loại yêu cầu thành 3 nhóm: Yêu cầu chức năng: tạo cây, thêm/sửa thành viên, chia sẻ quyền truy cập, xuất dữ liệu, thống kê,...
  - Yêu cầu phi chức năng: tốc độ phản hồi, độ bảo mật, khả năng sao lưu, dễ sử dụng.
  - Yêu cầu giao diện (UI/UX): giao diện sáng, bố cục rõ ràng, dễ đọc
- Thời gian: Tuần 5 (Ngày 3)

#### 4.1.6. Xây dựng tài liệu Project Proposal

- Mục tiêu: Hoàn thiện bản đề xuất chính thức của nhóm dựa trên kết quả phân tích yêu cầu.
- Mô tả công việc:
  - Phân chia công việc viết từng phần trong Proposal (Problem Statement, Functional Features, Architecture, Plan, Resources...).
  - Tổng hợp lại thành một tài liệu thống nhất, định dạng theo biểu mẫu của Bộ môn.
  - Kiểm tra, chỉnh sửa và gửi bản hoàn chỉnh lên hệ thống hoặc cho giảng viên duyệt.
- Thời gian: Tuần 5 (Ngày 4 - 7)

#### 4.2.7 Sản phẩm bàn giao:

- Bộ câu hỏi phỏng vấn và biên bản ghi nhận thông tin.
- Danh sách và phân loại các yêu cầu chức năng, phi chức năng, giao diện.
- Mô tả vai trò và quyền hạn của từng loại người dùng.
- Tài liệu Project Proposal hoàn chỉnh

## 4.2 Thiết kế phần mềm

Quy trình thực hiện gồm 4 hoạt động chính:

#### 4.2.1. Thiết kế kiến trúc tổng thể và cơ sở dữ liệu

##### Mục tiêu:

Xác định cách các thành phần của hệ thống (frontend, backend, database) tương tác với nhau và thiết kế cấu trúc lưu trữ dữ liệu đáp ứng yêu cầu nghiệp vụ.

##### Mô tả công việc:

- Xây dựng **kiến trúc tổng thể hệ thống (System Architecture Diagram)** theo mô hình 3 lớp:
  - *Presentation Layer (Next.js)*: hiển thị giao diện và tương tác với người dùng.
  - *Logic Layer (API routes)*: xử lý nghiệp vụ và xác thực người dùng.
  - *Data Layer (MySQL)*: lưu trữ dữ liệu cây gia phả, thông tin người dùng, thành viên, quan hệ,...
- Xác định các thực thể chính: Person, User, FamilyTree, Relationship, Achievement, Career, Location.
- Thiết kế ERD (Entity Relationship Diagram) thể hiện quan hệ giữa các thực thể.
- Chuyển đổi sang lược đồ dữ liệu quan hệ (Relational Schema).
- Thiết lập các ràng buộc dữ liệu

**Thời gian:** Tuần 6 (Ngày 1 → 2)

#### 4.2.2. Thiết kế chức năng và luồng xử lý

##### Mục tiêu:

Biểu diễn rõ cách người dùng tương tác với hệ thống, các luồng thao tác và cách dữ liệu di chuyển giữa các thành phần.

##### Mô tả công việc:

- Xây dựng **Use Case Diagram** để mô tả toàn bộ các chức năng chính: Đăng nhập, Đăng ký, Tạo cây, Thêm/Xóa thành viên, Chia sẻ cây, Xuất dữ liệu, Xem thông tin.
- Mô tả chi tiết luồng xử lý cho từng chức năng bằng **Activity Diagram**
- Vẽ **Data Flow Diagram (DFD)**

**Thời gian thực hiện:** Tuần 6 (Ngày 3 → 4)

#### 4.2.3. Thiết kế giao diện người dùng (UI/UX Design)

**Mục tiêu:**

Tạo ra các bản thiết kế giao diện trực quan, dễ sử dụng, thân thiện với nhiều đối tượng người dùng

**Mô tả công việc:**

- Thiết kế **wireframe và prototype** trên Figma
- Phát triển các màn hình chính:
  - Trang đăng nhập / đăng ký tài khoản.
  - Trang chủ quản lý cây gia phả.
  - Giao diện hiển thị cây (theo sơ đồ hoặc danh sách).
  - Trang chỉnh sửa thông tin thành viên.
  - Trang thống kê và xuất dữ liệu.
- Xây dựng **thống nhất phong cách thiết kế**:
  - Giao diện **tươi sáng, hiện đại, dễ đọc**, sử dụng font rõ ràng.
  - Màu nền nhẹ, biểu tượng và nút dễ thao tác.
- Ghi chú chi tiết về kích thước, màu sắc, và bố cục cho lập trình viên frontend.

**Thời gian thực hiện:** Tuần 6 (Ngày 5 ->7)

#### 4.2.4. Hoàn thiện và rà soát bản thiết kế

**Mục tiêu:**

Kiểm tra tính nhất quán, đầy đủ, và khả năng triển khai của tất cả các sơ đồ và bản thiết kế trước khi chuyển sang giai đoạn lập trình.

**Mô tả công việc:**

- Tổ chức buổi họp nhóm để **rà soát lại toàn bộ tài liệu thiết kế**: ERD, Use Case, DFD, Activity, giao diện.
- Sửa lỗi hoặc mâu thuẫn giữa các sơ đồ (nếu có).
- Viết tài liệu **API Specification** liệt kê các endpoint và định dạng dữ liệu.
- Tổng hợp toàn bộ bản thiết kế thành một **System Design Document (SDD)** thống nhất.
- Lưu trữ tài liệu và sơ đồ trên GitHub Repository để nhóm lập trình truy cập khi triển khai.

**Thời gian thực hiện:** Tuần 7 (Ngày 1 → 2)

#### 4.2.5 Sản phẩm bàn giao

- Sơ đồ kiến trúc tổng thể hệ thống (System Architecture Diagram).
- Mô hình cơ sở dữ liệu (ERD + Relational Schema).

- Sơ đồ Use Case, Activity Diagram, DFD hoàn chỉnh.
- File thiết kế giao diện (Figma).
- Tài liệu API Specification mô tả các endpoint chính.
- System Design Document (SDD) tổng hợp toàn bộ thiết kế.

### 4.3 Cài đặt (implement) phần mềm

Quy trình thực hiện gồm 3 hoạt động chính:

#### 4.3.1. Thiết lập môi trường phát triển

**Mục tiêu:**

Chuẩn bị đầy đủ công cụ, môi trường và cấu trúc dự án để đảm bảo quá trình lập trình diễn ra thuận lợi, thống nhất giữa các thành viên.

**Mô tả công việc:**

- Cài đặt các công cụ cần thiết: **Node.js, Next.js, MySQL, VSCode, Git.**
- Khởi tạo project Next.js
- Kết nối cơ sở dữ liệu MySQL

**Thời gian thực hiện:** Tuần 7 (Ngày 3)

#### 4.3.2. Xây dựng cơ sở dữ liệu và API Backend

**Mục tiêu:**

Phát triển các bảng dữ liệu và API chính để xử lý thông tin trong hệ thống.

**Mô tả công việc:**

- Tạo các bảng trong MySQL theo mô hình đã thiết kế
- Viết **API routes**
- Kiểm tra các API bằng **Postman** để đảm bảo phản hồi đúng định dạng JSON.
- Thêm các chức năng kiểm soát truy cập và xử lý lỗi.

**Thời gian thực hiện:** Tuần 7 (Ngày 3 → 7)

#### 4.3.3. Phát triển giao diện người dùng (Frontend Development)

**Mục tiêu:**

Hiện thực các màn hình giao diện theo thiết kế UI/UX và kết nối với backend qua API.

#### Mô tả công việc:

- Tạo trang **Đăng nhập / Đăng ký**, liên kết với API `/api/login` và `/api/register`.
- Xây dựng **trang chủ (Dashboard)** hiển thị danh sách cây gia phả người dùng sở hữu.
- Phát triển **giao diện hiển thị cây gia phả** dạng sơ đồ Xây dựng các form chức năng: thêm thành viên, chỉnh sửa thông tin, thống kê, chia sẻ.
- Áp dụng **TailwindCSS** để tạo bố cục gọn, màu tươi sáng và thân thiện.
- Kiểm tra giao diện trên trình duyệt desktop và mobile để đảm bảo tính phản hồi (responsive).

**Thời gian thực hiện:** Tuần 8 (Ngày 1 → 7)

#### 4.3.4 Sản phẩm bàn giao:

- Mã nguồn đầy đủ (Frontend + Backend) trên GitHub Repository.
- Cơ sở dữ liệu MySQL đã tạo và liên kết thành công.
- Phiên bản demo nội bộ chạy thử trên localhost.

### 4.4 Kiểm thử phần mềm

Quy trình thực hiện gồm 4 hoạt động chính:

#### 4.4.1. Chuẩn bị kế hoạch và dữ liệu kiểm thử

##### Mục tiêu:

Xây dựng kế hoạch kiểm thử chi tiết, xác định phạm vi, mục tiêu, dữ liệu test và tiêu chí đánh giá thành công.

##### Mô tả công việc:

- Xây dựng **Test Plan** nêu rõ: mục tiêu, phạm vi, loại test, công cụ sử dụng, tiêu chí pass/fail.
- Chuẩn bị **Test Data**: dữ liệu mẫu cho người dùng, thành viên, cây gia phả, quan hệ cha mẹ, vợ chồng,...
- Thiết lập môi trường kiểm thử
- Phân công nhiệm vụ test cho từng thành viên (frontend, backend, database).

**Thời gian thực hiện:** Tuần 9 (Ngày 1 → 2)

#### 4.4.2. Kiểm thử đơn vị (Unit Testing)



**Mục tiêu:**

Đảm bảo từng module, hàm hoặc API đúng theo yêu cầu kỹ thuật trước khi tích hợp.

**Mô tả công việc:**

- Kiểm tra từng API route riêng lẻ
- Kiểm thử chức năng xử lý logic
- Dùng công cụ **Postman** để kiểm tra phản hồi JSON.
- Ghi nhận lỗi (nếu có), chỉnh sửa và cập nhật lại mã nguồn.

**Thời gian thực hiện:** Tuần 9 (Ngày 2 → 3)

#### 4.4.3. Kiểm thử tích hợp (Integration Testing)

**Mục tiêu:**

Đảm bảo các thành phần frontend – backend – database ăn khớp, dữ liệu được trao đổi chính xác.

**Mô tả công việc:**

- Kết nối các module đã kiểm thử đơn vị và chạy thử toàn luồng:  
Đăng ký → Đăng nhập → Tạo cây → Thêm thành viên → Chia sẻ → Xuất dữ liệu.
- Kiểm tra tính đồng bộ dữ liệu giữa UI và database.
- Cập nhật **Integration Test Report** để ghi lại lỗi và cách khắc phục.

**Thời gian thực hiện:** Tuần 9 (Ngày 3 → 4)

#### 4.4.4. Kiểm thử hệ thống (System Testing)

**Mục tiêu:**

Đánh giá toàn bộ hệ thống như một sản phẩm hoàn chỉnh, kiểm tra tất cả các yêu cầu chức năng và phi chức năng.

**Mô tả công việc:**

- Kiểm thử các nhóm chức năng chính:
  - Đăng ký/đăng nhập và quản lý tài khoản.
  - Tạo, chỉnh sửa, xóa và xem cây gia phả.
  - Giao diện hiển thị sơ đồ và danh sách thành viên.
  - Chức năng chia sẻ và quyền truy cập.
  - Xuất dữ liệu ra định dạng PDF hoặc CSV.

- Kiểm tra **các yêu cầu phi chức năng**: tốc độ tải trang, khả năng phản hồi, độ ổn định, bảo mật đăng nhập.
- Ghi nhận toàn bộ lỗi, sự cố và đánh giá mức độ nghiêm trọng (Minor / Major / Critical).

**Thời gian thực hiện:** Tuần 9 (Ngày 5 → 6)

#### 4.4.5. Kiểm thử chấp nhận người dùng (User Acceptance Test – UAT)

**Mục tiêu:**

Xác nhận hệ thống đáp ứng đúng mong đợi của giảng viên và người dùng cuối.

**Mô tả công việc:**

- Chuẩn bị kịch bản test mô phỏng các hành vi người dùng thực tế.
- Mời giảng viên hoặc nhóm bạn đóng vai người dùng để sử dụng thử hệ thống.
- Thu thập phản hồi về:
  - Dễ sử dụng (usability).
  - Tính chính xác của dữ liệu hiển thị.
  - Giao diện và trải nghiệm tổng thể.
- Ghi lại tất cả các đề xuất, phản hồi, và đề xuất chỉnh sửa nhỏ trước buổi trình bày cuối.

**Thời gian thực hiện:** Tuần 9 (Ngày 7)

#### 4.4.6 Sản phẩm bàn giao

- Test Plan
- Test Case & Test Data cho từng loại test.
- Unit Test Report và Integration Test Report.
- System Test Result ghi nhận kết quả và lỗi đã khắc phục.
- Bản Demo hoàn chỉnh sẵn sàng trình bày.

### 4.5 *Triển khai, bảo trì*

#### 4.5.1. Chuẩn bị môi trường triển khai

**Mục tiêu:**

Cấu hình môi trường triển khai ổn định để demo hệ thống trực tiếp hoặc online.

**Mô tả công việc:**

- Xác định hình thức triển khai: sử dụng Azure

- Cấu hình biến môi trường (.env) cho môi trường production.
- Di chuyển dữ liệu từ database test sang database production.
- Kiểm tra việc hoạt động ổn định của API, frontend và database sau khi triển khai.
- Đảm bảo hệ thống có thể truy cập dễ dàng cho giảng viên xem thử.

**Thời gian thực hiện:** Tuần 10 (Ngày 1 → 3)

#### 4.5.2. Kiểm tra sau triển khai (Post-deployment Testing)

**Mục tiêu:**

Đảm bảo hệ thống hoạt động đúng trên môi trường triển khai, không phát sinh lỗi ngoài dự kiến.

**Mô tả công việc:**

- Kiểm thử lại toàn bộ chức năng: đăng ký, đăng nhập, tạo cây, thêm thành viên, chia sẻ, xuất dữ liệu.
- Kiểm tra hiệu năng (thời gian tải trang, độ phản hồi API).
- Xác minh dữ liệu hiển thị khớp với database thực tế.
- Ghi nhận mọi lỗi hoặc sự cố còn lại và xử lý kịp thời trước khi trình bày.

**Thời gian thực hiện:** Tuần 10 (Ngày 4 → 7)

#### 4.5.3. Sản phẩm bàn giao:

- Link GitHub Repository và link demo online.

# 5 Kế hoạch nhân sự & chi phí

## 5.1. Cơ cấu nhân sự

Nhóm gồm 5 thành viên, tất cả đều tham gia xuyên suốt vào toàn bộ các giai đoạn của dự án — từ phân tích, thiết kế, lập trình đến kiểm thử và báo cáo. Mọi người cùng làm việc, thảo luận và hỗ trợ lẫn nhau trong từng nhiệm vụ, không có sự tách biệt vai trò.

MSSV	Họ và tên	Vai trò	Nhiệm vụ
23120124	Nguyễn Minh Hiếu	Leader	Điều phối, phân công nhiệm vụ. Tham gia toàn bộ quy trình phát triển phần mềm: phân tích yêu cầu, thiết kế, lập trình, kiểm thử và triển khai.
23120111	Phạm Quốc Nam Anh	Thành viên	Tham gia toàn bộ quy trình phát triển phần mềm: phân tích yêu cầu, thiết kế, lập trình, kiểm thử và triển khai.
23120112	Thái Khắc Anh Tuấn	Thành viên	Tham gia toàn bộ quy trình phát triển phần mềm: phân tích yêu cầu, thiết kế, lập trình, kiểm thử và triển khai.
23120121	Nguyễn Xuân Duy	Thành viên	Tham gia toàn bộ quy trình phát triển phần mềm: phân tích yêu cầu,

			thiết kế, lập trình, kiểm thử và triển khai.
23120175	Huỳnh Thái Toàn	Thành viên	Tham gia toàn bộ quy trình phát triển phần mềm: phân tích yêu cầu, thiết kế, lập trình, kiểm thử và triển khai.

## 5.2. Phân bổ công việc và thời gian thực hiện

Cả nhóm cùng thực hiện các giai đoạn chính theo kế hoạch phát triển đã đề ra ở phần 4:

Giai đoạn	Nội dung chính	Thời gian (tuần)	Giờ công nhóm (ước tính)
Phân tích yêu cầu	Tìm hiểu, thu thập yêu cầu, viết đề xuất	Tuần 3-5	40 giờ
Thiết kế hệ thống	Kiến trúc, sơ đồ ERD, giao diện	Tuần 6-7	50 giờ
Lập trình & tích hợp	Xây dựng cơ sở dữ liệu, API, giao diện	Tuần 7-8	60 giờ
Kiểm thử & hiệu chỉnh	Chạy thử, sửa lỗi, hoàn thiện hệ thống	Tuần 9	30 giờ
Triển khai & báo cáo	Demo, viết tài liệu và báo cáo cuối kỳ	Tuần 10	20 giờ

Tổng thời gian làm việc: ~200 giờ /nhóm (tương đương 40 giờ mỗi thành viên).

### 5.3. Chi phí nhân công (tính theo mức lương Intern)

Giả định mức lương thực tập sinh ngành CNTT tại TP.HCM là 30.000 VNĐ/giờ. Tổng chi phí nhân công được tính như sau:

Hạng mục	Giờ công nhóm	Đơn giá (VNĐ/giờ)	Thành tiền (VNĐ)
Phân tích yêu cầu	40	30.000	1.200.000
Thiết kế hệ thống	50	30.000	1.500.000
Lập trình & tích hợp	60	30.000	1.800.000
Kiểm thử & sửa lỗi	30	30.000	900.000
Triển khai & báo cáo	20	30.000	600.000
<b>Tổng chi phí nhân công</b>	<b>200</b>		<b>6.000.000 VNĐ</b>

Phân bổ bình quân:  $6.000.000 / 5 = 1.200.000$  VNĐ/người.

### 5.4. Chi phí công cụ và hạ tầng

Hạng mục	Mô tả	Chi phí (VNĐ)
Microsoft Azure	Hosting web và MySQL database (gói sinh viên miễn phí)	0

Figma	Thiết kế giao diện UI/UX (gói miễn phí)	0
GitHub	Lưu trữ mã nguồn, quản lý phiên bản (miễn phí)	0
Tên miền demo (nếu có)	Dự kiến đăng ký tên miền dùng cho demo	200.000
Chi phí khác (in ấn, trình bày, mạng, di chuyển)	In ấn tài liệu báo cáo, tiền họp nhóm,..	300.000
<b>Tổng chi phí hạ tầng &amp; công cụ</b>		<b>500.000 VNĐ</b>

### 5.5. Tổng chi phí dự án

<b>Loại chi phí</b>	<b>Số tiền (VNĐ)</b>
Chi phí nhân công	6.000.000
Chi phí công cụ & hạ tầng	500.000
<b>Tổng cộng</b>	<b>6.500.000 VNĐ</b>

### 5.6. Nhận xét

Tất cả thành viên đều tham gia đồng đều trong suốt quá trình phát triển, không phân chia cứng vai trò. Tổng chi phí được xây dựng dựa trên mức lương intern thực tế và giả định sử dụng công cụ miễn phí cho sinh viên. Mức chi phí này hợp lý, phản ánh đúng khối lượng công việc và thời gian nhóm đầu tư cho dự án học phần.