TRƯỜNG ĐẠI HỌC BÁCH KHOA CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA CÔNG NGHỆ THÔNG TIN Độc lập - Tự do - Hạnh phúc

\*\*\* \*\*\*

**ĐỀ CƯƠNG CHI TIẾT ĐỒ ÁN TỐT NGHIỆP**

**Tên đề tài:** Xây dựng website hỗ trợ phân công công việc và theo dõi tiến độ dự án

**Sinh viên thực hiện:**

Trần Minh Nhật **MSSV:** 102210072 **Lớp :** 21TCLC\_DT1

**Tên người hướng dẫn:** ThS. Trần Hồ Thuỷ Tiên

1. **Lý do chọn đề tài**

Trong xu thế chuyển đổi số và sự phát triển mạnh mẽ của công nghệ thông tin, việc quản lý dự án phần mềm theo phương pháp Agile, đặc biệt là mô hình Scrum, ngày càng trở nên phổ biến. Các công cụ hỗ trợ quản lý như Jira đang đóng vai trò quan trọng trong việc tổ chức công việc, phân công nhiệm vụ, giám sát tiến độ và nâng cao hiệu suất làm việc nhóm.

Tuy nhiên, các nền tảng như Jira thường có giao diện phức tạp, nhiều tính năng nâng cao và chi phí sử dụng cao, gây khó khăn cho sinh viên, nhóm học thuật, hoặc các nhóm phát triển phần mềm quy mô nhỏ khi tiếp cận và sử dụng. Ngoài ra, việc tích hợp với các công cụ như Git hay CI/CD đôi khi không cần thiết đối với các nhóm nhỏ hoặc các dự án học thuật.

Xuất phát từ thực tế đó, đề tài "Xây dựng ứng dụng website hỗ trợ phân công công việc và theo dõi tiến độ dự án" được lựa chọn với mục tiêu xây dựng một ứng dụng website tập trung vào các chức năng quản lý công việc cốt lõi dựa theo Agile như backlog, sprint, phân công nhiệm vụ, theo dõi tiến độ, giao diện Kanban – nhằm mang đến một giải pháp trực quan, dễ sử dụng và phù hợp với các nhóm phát triển phần mềm nhỏ, sinh viên và môi trường giáo dục.

1. **Mục tiêu và nhiệm vụ đề tài**

**2.1. Mục tiêu tổng quát**

* + Xây dựng một nền tảng website hỗ trợ quản lý dự án phần mềm theo mô hình Agile, tập trung vào việc phân công công việc, theo dõi tiến độ, và quản lý quy trình làm việc nhóm một cách trực quan và hiệu quả.
  + Cung cấp công cụ giúp các nhóm phát triển phần mềm, nhóm sinh viên hoặc nhóm làm việc nhỏ tổ chức công việc khoa học, rõ ràng, và dễ theo dõi, thay thế việc quản lý thủ công hoặc sử dụng các công cụ phức tạp như Jira.

## **2.2. Nhiệm vụ cụ thể**

### **Xây dựng các chức năng cốt lõi của hệ thống quản lý dự án:**

* + Hệ thống đăng ký, đăng nhập, và phân quyền người dùng theo vai trò (Quản lý, Thành viên).
  + Giao diện người dùng hiện đại, trực quan, hỗ trợ thao tác kéo – thả (drag & drop) trong bảng Kanban để quản lý task.
  + Quản lý dự án: tạo, cập nhật, phân công nhiệm vụ, thiết lập deadline và theo dõi tiến độ từng task.
  + Tạo Sprint, quản lý Backlog, hỗ trợ workflow theo mô hình Scrum cơ bản.

### **Tích hợp các tính năng hỗ trợ thông minh:**

* + Tự động phân loại công việc theo trạng thái (To do – In progress – Done), và cho phép lọc, tìm kiếm, sắp xếp nhiệm vụ linh hoạt.
  + Gợi ý phân công công việc hoặc cảnh báo quá hạn dựa trên dữ liệu tiến độ thực tế
  + Thông báo (notification) cập nhật trạng thái công việc cho người dùng liên quan.

### **Phát triển hệ thống thống kê và trực quan hóa tiến độ dự án:**

* Xây dựng dashboard tổng quan giúp người dùng đánh giá tiến độ chung của dự án, số lượng task theo trạng thái, hiệu suất làm việc của từng thành viên.
* Ứng dụng thư viện trực quan hóa dữ liệu (Chart.js, ApexCharts...) để thể hiện biểu đồ tiến độ, tỷ lệ hoàn thành công việc, lượng công việc tồn đọng...
* Đối tượng và phạm vi đề tài

**3. Đối tượng và phạm vi đề tài**

**3.1. Đối tượng sử dụng chính**

### **Nhóm đối tượng chính:**

* Sinh viên đại học/cao đẳng chuyên ngành Kinh Tế: Các nhóm sinh viên làm đồ án, dự án học kỳ hoặc nghiên cứu khoa học cần một công cụ hỗ trợ phân công nhiệm vụ, theo dõi tiến độ, và cộng tác hiệu quả trong môi trường học thuật.
* Nhóm phát triển phần mềm quy mô nhỏ (Startup, Freelancer): Các nhóm lập trình viên quy mô nhỏ muốn quản lý công việc theo mô hình Agile nhưng không muốn sử dụng các công cụ phức tạp như Jira hoặc Asana.
* Người quản lý dự án phần mềm cấp nhóm hoặc cá nhân làm side project: Cần một công cụ đơn giản, dễ sử dụng để lên kế hoạch, theo dõi công việc, và tổ chức sprint hiệu quả.

### **Nhóm đối tượng phụ:**

* Giảng viên hoặc cố vấn học thuật: theo dõi tiến độ làm việc của nhóm sinh viên trong các môn học yêu cầu làm việc nhóm hoặc đồ án.
* Người mới học Agile/Scrum: sử dụng như một công cụ thực hành để làm quen với quy trình phát triển phần mềm theo phương pháp Agile – từ lập kế hoạch, tạo backlog đến quản lý sprint.

## **3.2. Phạm vi áp dụng**

### **Lĩnh vực giáo dục:**

* Hỗ trợ sinh viên làm việc nhóm trong các môn học như phát triển phần mềm, đồ án môn học hoặc các hoạt động nghiên cứu.
* Là công cụ thực hành cho sinh viên khi học các phương pháp phát triển phần mềm Agile/Scrum, giúp tiếp cận quy trình quản lý dự án một cách trực quan và dễ hiểu.

### **Lĩnh vực doanh nghiệp:**

* Ứng dụng trong các nhóm có quy mô nhỏ hoặc startup, giúp tổ chức công việc, chia nhiệm vụ, theo dõi tiến độ một cách hiệu quả mà không cần công cụ phức tạp.
* Hỗ trợ freelancer hoặc nhóm cộng tác online trong việc lập kế hoạch, phân công công việc và kiểm soát quá trình thực hiện dự án.

**4. Phương pháp thực hiện, công nghệ sử dụng**

**4.1. Phương pháp thực hiện**

### **Giai đoạn 1: Xây dựng nền tảng hệ thống cơ bản**

**Mục tiêu:** Xây dựng hệ thống cốt lõi của ứng dụng, bao gồm chức năng quản lý người dùng, tạo dự án và phân công công việc.

**Các bước thực hiện:**

* Thiết kế cơ sở dữ liệu để lưu thông tin người dùng, dự án, task, phân công, trạng thái công việc, và tiến độ.
* Xây dựng backend RESTful API xử lý các nghiệp vụ chính: đăng ký, đăng nhập, tạo dự án, tạo và cập nhật task, phân quyền người dùng.
* Phát triển frontend với các tính năng cơ bản: trang dashboard tổng quan, giao diện danh sách công việc theo trạng thái (To Do – In Progress – Done).
* Thực hiện kiểm thử đơn vị (unit test) cho các module backend và frontend để đảm bảo hoạt động ổn định của từng thành phần.

### **Giai đoạn 2: Phát triển các tính năng nâng cao**

**Mục tiêu:** Bổ sung các tính năng quản lý quy trình Agile như Sprint, Backlog, phân loại công việc, thống kê và báo cáo tiến độ.

**Các bước thực hiện:**

* Xây dựng chức năng quản lý Sprint, Backlog và giai đoạn phát triển phần mềm theo mô hình Scrum.
* Phát triển giao diện bảng Kanban với tính năng kéo – thả task giữa các cột trạng thái.
* Thêm chức năng thống kê tiến độ, hiển thị biểu đồ công việc (Burn-down Chart, Task Distribution).
* Áp dụng các thư viện trực quan hóa dữ liệu (như Chart.js hoặc ApexCharts) để hiển thị biểu đồ trong dashboard.

### **Giai đoạn 3: Tối ưu và triển khai hệ thống**

**Mục tiêu:** Cải thiện trải nghiệm người dùng, triển khai hệ thống lên nền tảng thực tế và đảm bảo tính ổn định.

**Các bước thực hiện:**

* Tối ưu giao diện người dùng (UI/UX) theo phản hồi từ người dùng thử nghiệm: bố cục, màu sắc, trải nghiệm kéo – thả...
* Triển khai backend lên nền tảng như Render, Railway, hoặc Azure App Service; triển khai frontend lên Vercel hoặc Netlify.
* Cấu hình cơ sở dữ liệu như MongoDB Atlas / PostgresSQL / MySQL trên nền tảng cloud để đảm bảo truy cập từ xa.
* Viết tài liệu hướng dẫn sử dụng, chuẩn bị bản demo để trình bày đề tài.

**4.2. Công nghệ sử dụng**

### **Backend:**

* **Spring Boot (Java):** Framework chính để xây dựng hệ thống backend, cung cấp các API RESTful phục vụ cho các chức năng như: quản lý người dùng, dự án, nhiệm vụ, phân quyền, và trạng thái công việc.
* **Spring Security:**Đảm bảo tính bảo mật của hệ thống thông qua cơ chế xác thực và phân quyền người dùng (sử dụng JWT – JSON Web Token).
* **Spring Data JPA/JDBC:** Tích hợp với cơ sở dữ liệu MySQL để thao tác với dữ liệu dự án, người dùng, nhiệm vụ... giúp đơn giản hóa quá trình truy vấn và lưu trữ.
* **PostgresSQL / MySQL:**Hệ quản trị cơ sở dữ liệu quan hệ dùng để lưu trữ toàn bộ dữ liệu của hệ thống (người dùng, dự án, task, sprint...).

### **Frontend:**

* **NextJS (TypeScript):** Thư viện chính để xây dựng giao diện người dùng tương tác, hiển thị bảng Kanban, dashboard quản lý tiến độ và các biểu mẫu tạo/cập nhật nhiệm vụ.
* **Axios:**Thư viện dùng để gọi API từ frontend đến backend, đảm bảo kết nối mượt mà và đồng bộ dữ liệu giữa client và server.
* **Chart.js / ApexCharts:**Thư viện trực quan hóa dữ liệu, dùng để xây dựng dashboard thể hiện tiến độ dự án, thống kê công việc theo từng trạng thái, hiệu suất làm việc của từng thành viên.

## **AI – Mô hình lựa chọn:**

* Ứng dụng Machine Learning để huấn luyện mô hình dự đoán dựa trên nhiều dự án lịch sử hơn (tập trung vào Regression hoặc Classification).
* Dùng các thuật toán như Decision Tree, Random Forest hoặc Gradient Boosting để đánh giá khả năng hoàn thành dự án sớm/trễ.

**Cơ sở dữ liệu và triển khai:**

* **Cơ sở dữ liệu:** PostgreSQL trên Azure hoặc Supabase để lưu thông tin người dùng, dự án, task, lịch sử hành vi.
* **Triển khai:**
  + - **Backend (Java Spring Boot)**: Triển khai bằng **Azure App Service**.
    - **Frontend (Next.js)**: Triển khai bằng **Azure Static Web Apps** hoặc **Vercel**.
* **Tích hợp AI đơn giản:** Tính toán điểm hoàn thành được xử lý trực tiếp trong backend.
* **Tích hợp AI nâng cao:** Huấn luyện mô hình riêng bằng Python, triển khai riêng qua Flask/FastAPI.

### **5. Dự kiến kết quả**

Triển khai được các chức năng chính của một nền tảng quản lý dự án theo mô hình Agile, bao gồm:

#### **5.1. Chức năng quản lý người dùng**

* Cho phép người dùng đăng ký và đăng nhập vào hệ thống.
* Xác thực và phân quyền người dùng (Quản trị viên, Thành viên dự án).
* Quản lý thông tin cá nhân của người dùng (tên, email, vai trò, ảnh đại diện).

#### **5.2. Chức năng quản lý dự án**

* Người dùng có thể tạo mới, cập nhật, xóa và xem chi tiết dự án.
* Phân quyền người dùng cho từng dự án.
* Giao diện hiển thị danh sách các dự án mà người dùng đang tham gia.

#### **5.3. Chức năng quản lý bảng công việc (Kanban Board)**

* Hiển thị các cột công việc theo trạng thái (To Do, In Progress, Done).
* Cho phép tạo, cập nhật, xóa công việc (task) trong từng cột.
* Hỗ trợ kéo thả task để thay đổi trạng thái một cách trực quan.
* Giao diện Kanban tương tự như JIRA.

#### **5.4. Chức năng quản lý task chi tiết**

* Giao diện hiển thị chi tiết thông tin task: tên, mô tả, người được gán, mức độ ưu tiên, thời hạn, file đính kèm,...
* Cho phép bình luận, cập nhật tiến độ.
* Ghi lịch sử chỉnh sửa và thay đổi trạng thái task.

#### **5.5. Chức năng thống kê & báo cáo**

* Hiển thị biểu đồ tiến độ thực hiện dự án (task completion chart).
* Thống kê số lượng task theo trạng thái, theo người phụ trách.
* Báo cáo tiến độ theo từng dự án và thời gian thực hiện.

#### **5.6. Chức năng realtime với WebSocket**

* Hệ thống cập nhật thay đổi task (di chuyển, chỉnh sửa, bình luận) theo thời gian thực giữa các người dùng đang cùng thao tác.
* Hạn chế tình trạng xung đột dữ liệu khi có nhiều người dùng thao tác đồng thời.

#### **5.7. Chức năng triển khai và vận hành trên Azure**

* Triển khai Backend (Spring Boot) trên Azure App Service.
* Triển khai Frontend (Next.js) trên Azure Static Web Apps hoặc Vercel.
* Cấu hình CORS, bảo mật JWT, tối ưu hiệu năng và đảm bảo an toàn dữ liệu.
* Hệ thống hoạt động ổn định, có thể mở rộng khi cần thiết

## **6. Ý nghĩa thực tiễn và khoa học**

**6.1. Về mặt thực tiễn**

* + Giải quyết nhu cầu cấp thiết: cung cấp một công cụ hỗ trợ quản lý công việc nhóm đơn giản, hiệu quả và dễ tiếp cận, đặc biệt phù hợp với sinh viên, nhóm học thuật, nhóm startup hoặc các dự án cá nhân, trong bối cảnh làm việc nhóm ngày càng phổ biến.
  + Nâng cao hiệu quả làm việc nhóm: giúp người dùng phân chia nhiệm vụ rõ ràng, theo dõi tiến độ, đồng thời phát hiện sớm các điểm nghẽn trong quy trình làm việc để có hướng xử lý kịp thời.
  + Thúc đẩy chuyển đổi số trong học tập và làm việc: đề tài góp phần ứng dụng quy trình Agile và công cụ quản lý hiện đại vào môi trường giáo dục, làm việc và khởi nghiệp tại Việt Nam, tăng tính chuyên nghiệp và tự chủ trong tổ chức công việc.

## **6.2. Về mặt khoa học**

* Mô hình hóa quy trình phát triển phần mềm theo Agile: áp dụng các khái niệm như Sprint, Backlog, Task Management, và Role-based Permission trong hệ thống, tạo điều kiện để sinh viên và người dùng hiểu và thực hành phương pháp phát triển phần mềm hiện đại.
* Thiết kế và triển khai hệ thống phân tán: kết hợp giữa Spring Boot (backend) và NextJS (frontend) để xây dựng mô hình client-server chuẩn RESTful API – là một kỹ thuật phổ biến và mang tính học thuật cao trong các hệ thống phần mềm hiện đại.
* Ứng dụng trực quan hóa dữ liệu:tích hợp các công cụ trực quan hóa như Chart.js, ApexCharts để phân tích hiệu suất làm việc, tiến độ dự án, giúp người dùng dễ dàng đưa ra quyết định – một hướng đi thực tiễn kết hợp giữa lập trình phần mềm và khoa học dữ liệu.

### **7. Dự kiến nội dung đồ án tốt nghiệp**

### **Mở đầu**

* Lý do chọn đề tài
* Phạm vi và đối tượng áp dụng
* Mục tiêu và ý nghĩa
* Kết quả mong đợi
  + - Về mặt thực tiễn
    - Về mặt lý thuyết
    - Về mặt ứng dụng
* Phương pháp phát triển phần mềm: Agile (Scrum)
* Các bước triển khai thực hiện

### **Chương I. Cơ sở lý thuyết và công nghệ sử dụng**

* Tổng quan về hệ thống quản lý dự án (PM tool)
* Tìm hiểu JIRA – các tính năng chính
* Mô hình Agile/Scrum
* Kiến trúc tổng quan hệ thống
* Các công nghệ sử dụng:
  + - Frontend: Next.js
    - Backend: Java Spring Boot + PostgreSQL
    - Quản lý task: Drag & Drop, Realtime với Socket.io / WebSocket
    - Xác thực: JWT
    - Quản lý mã nguồn: GitHub

### **Chương II. Phân tích và thiết kế hệ thống**

* Phân tích yêu cầu chức năng
  + - Đăng nhập / đăng ký
    - Quản lý người dùng, phân quyền
    - Quản lý dự án
    - Quản lý task theo board (To do / In progress / Done)
    - Bình luận, đính kèm file
    - Lọc, tìm kiếm task
* Phân tích yêu cầu phi chức năng
* Mô hình hệ thống  
  + - Biểu đồ Use Case
    - Biểu đồ tuần tự
    - Biểu đồ lớp
    - ERD
    - Kiến trúc hệ thống

### **Chương III. Triển khai hệ thống**

* Thiết lập môi trường phát triển
* Xây dựng Backend với Spring Boot: API cho user, project, task, comment
* Xây dựng giao diện người dùng với Next.js:
* Tích hợp xác thực và phân quyền người dùng
* Kiểm thử và đánh giá

### **Kết luận và hướng phát triển**

* Kết quả đạt được
* Hạn chế còn tồn tại
* Hướng phát triển: thêm ứng dụng mobile

**8. Dự kiến tiến độ**

Thời gian từng giai đoạn và nội dung triển khai

|  |  |  |
| --- | --- | --- |
| **STT** | **Thời gian** | **Nội dung triển khai** |
| 1 | 17/02/2025 – 24/03/2025 | * Xác định đề tài và mục tiêu thực hiện * Phân tích yêu cầu chức năng, phi chức năng * Viết báo cáo tổng quan và đề cương chi tiết |
| 2 | 25/03/2025 – 07/04/2025 | * Thiết kế UI/UX: giao diện Dashboard, Project, Task, Kanban board... * Thiết kế cơ sở dữ liệu (ERD) * Thiết kế API cho user, task, project |
| 3 | 08/04/2025 – 14/04/2025 | * Xây dựng tính năng đăng ký, đăng nhập, phân quyền người dùng (JWT) trên backend * Xây dựng giao diện Login/Register/Dashboard |
| 4 | 15/04/2025 – 06/05/2025 | * Xây dựng các chức năng chính backend: tạo Project, tạo Task, cập nhật trạng thái Task * Giao diện quản lý Project và Task (drag-drop Kanban) |
| 5 | 07/05/2025 – 24/05/2025 | * Tích hợp tính năng Realtime (Socket.io/WebSocket) cho cập nhật Task * Chức năng bình luận, gán người phụ trách Task |
| 6 | 25/05/2025 – 01/06/2025 | * Kiểm thử các chức năng * Hoàn thiện báo cáo chương Triển khai và Kết luận |
| 7 | 02/06/2025 – 07/06/2025 | * Tối ưu hiệu năng, bảo mật cơ bản * Viết tài liệu hướng dẫn sử dụng * Kiểm thử hệ thống toàn diện |
| 8 | 08/06/2025 – 23/06/2025 | * Triển khai hệ thống lên môi trường thực tế (VD: Azure, Vercel, Render...) * Nộp báo cáo hoàn chỉnh, chuẩn bị báo cáo bảo vệ |

*Đà Nẵng, ngày 27 tháng 03 năm 2025*

XÁC NHẬN CỦA NGƯỜI HƯỚNG DẪN SINH VIÊN THỰC HIỆN

(Ký và ghi rõ họ tên) (Ký và ghi rõ họ tên)