

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC
CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
XÂY DỰNG ỨNG DỤNG WEB QUẢN LÝ CÔNG VIỆC

Giảng viên hướng dẫn:

TS. Nguyễn Bảo Ân

Sinh viên thực hiện:

Nguyễn Thị Ngọc Hân (110122069 – DA22TTD)

Huỳnh Khánh Duy (110122059 – DA22TTD)

Hồ Nguyễn Quốc Dũng (110122056 – DA22TTD)

Vĩnh Long, tháng 07 năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC
CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:
XÂY DỰNG ỨNG DỤNG WEB QUẢN LÝ CÔNG VIỆC

Giảng viên hướng dẫn:

TS. Nguyễn Bảo Ân

Sinh viên thực hiện:

Nguyễn Thị Ngọc Hân (110122069 – DA22TTD)

Huỳnh Khánh Duy (110122059 – DA22TTD)

Hồ Nguyễn Quốc Dũng (110122056 – DA22TTD)

Vĩnh Long, tháng 07 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

LỜI MỞ ĐẦU

Trong thời đại số hóa và làm việc theo nhóm ngày càng phổ biến, nhu cầu quản lý công việc một cách hiệu quả, minh bạch và có tổ chức trở nên vô cùng quan trọng. Đặc biệt trong môi trường phát triển phần mềm, việc phân chia nhiệm vụ, theo dõi tiến độ và phối hợp giữa các thành viên là yếu tố then chốt quyết định đến sự thành công của dự án. Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài *Quản lý công việc* nhằm xây dựng một ứng dụng hỗ trợ tối ưu hóa quá trình phân công và theo dõi nhiệm vụ trong nhóm.

Dự án được phát triển theo kiến trúc client/server với sự tách biệt rõ ràng giữa frontend và backend, đảm bảo tính mở rộng, dễ bảo trì cũng như phù hợp với định hướng phát triển phần mềm hiện đại. Các thành phần trong hệ thống được thiết kế để giao tiếp qua API, đồng thời nhóm cũng áp dụng đầy đủ các quy trình và công cụ cần thiết trong suốt vòng đời phát triển, từ lên kế hoạch, thiết kế, kiểm thử đến triển khai.

Thông qua dự án này, nhóm không chỉ củng cố kiến thức đã học về công nghệ phần mềm mà còn có cơ hội tiếp cận thực tiễn phát triển một hệ thống phần mềm hoàn chỉnh theo mô hình công nghiệp. Đây cũng là dịp để rèn luyện kỹ năng làm việc nhóm, giải quyết vấn đề và sử dụng thành thạo các công cụ hỗ trợ chuyên nghiệp trong quá trình phát triển phần mềm.

LỜI CẢM ƠN

Để hoàn thành tốt bài báo cáo này, trước hết chúng em xin cảm ơn đến quý thầy cô giáo trong khoa Kỹ Thuật và Công Nghệ của Trường Đại Học Trà Vinh em xin cảm ơn đến ban lãnh đạo nhà trường đã tạo mọi điều kiện tốt nhất để chúng em có cơ hội được trình bày và thể hiện ý tưởng của mình, từ đó áp dụng kiến thức, kỹ năng của mình để đưa ý tưởng ấy thành sản phẩm thực tế có thể áp dụng trong đời sống.

Đặc biệt, chúng em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Bảo Ân - Giảng viên trực tiếp hướng dẫn, thầy đã tận tình giúp đỡ chúng em trong quá trình học tập cũng như trong việc nâng cao kiến thức chuyên môn, phát huy tính tự học, tự tìm hiểu, từ đó hình thành một hướng đi rõ ràng cho chúng em. Do thời gian cũng như lượng kiến thức của nhóm còn hạn hẹp, nên không tránh khỏi những sai sót trong bài báo cáo của mình, kính mong quý Thầy/Cô thông cảm! Nhóm em cũng rất mong nhận được sự nhận xét của quý Thầy/Cô để nhóm em có thể khắc phục những sai sót cũng như phát triển hơn nữa trong tương lai!

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU	1
1.1 Tổng quan về dự án.....	1
1.2 Mục tiêu của ứng dụng.....	1
1.3 Lý do chọn đề tài.....	1
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU	2
2.1 Chức năng chính của hệ thống	2
2.2 Yêu cầu phi chức năng của hệ thống.....	3
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG.....	5
3.1 Kiến trúc tổng thể.....	5
3.2 Cấu trúc thư mục	5
3.3 Thiết kế cơ sở dữ liệu.....	6
3.4 Thiết kế API	7
3.5 Thiết kế giao diện (UI/UX)	8
3.5.1 Thiết kế giao diện UI	8
3.5.2 Thiết kế giao diện UX.....	11
CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG.....	20
4.1 Các công nghệ đã sử dụng	20
4.2 Quy trình CI/CD với GitHub Actions	21
4.3 Cấu hình Docker và quy trình triển khai ứng dụng	22
CHƯƠNG 5: QUẢN LÝ DỰ ÁN	24
5.1 Sử dụng Jira để lập kế hoạch và theo dõi tiến độ.....	24
5.2 Phân công nhiệm vụ	24
CHƯƠNG 6: KIỂM THỬ	28
6.1 Chiến lược kiểm thử và công cụ sử dụng.....	28
6.2 Kết quả kiểm thử API với Postman.....	29
CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN	33
7.1 Khó khăn	33
7.2 Bài học rút ra và đề xuất cải thiện trong tương lai.....	33
TÀI LIỆU THAM KHẢO	35
PHỤ LỤC	36

DANH MỤC HÌNH ẢNH

Hình 3.1 Cấu trúc thư mục	6
Hình 3.2 Mô hình ERD	7
Hình 3.3 Giao diện trang Tổng quan với Figma.....	8
Hình 3.4 Giao diện trang Công việc với Figma	9
Hình 3.5 Giao diện Thêm công việc với Figma	9
Hình 3.6 Giao diện trang Chi tiết công việc với Figma	10
Hình 3.7 Giao diện trang Đăng ký với Figma	10
Hình 3.8 Giao diện trang Đăng nhập với Figma	11
Hình 3.9 Giao diện trang Tổng quan	11
Hình 3.10 Giao diện trang Công việc	12
Hình 3.11 Giao diện khi Thêm công việc	13
Hình 3.12 Giao diện trang Chi tiết công việc	14
Hình 3.13 Giao diện khi Update công việc	15
Hình 3.14 Giao diện khi xóa công việc	15
Hình 3.15 Giao diện trang Đăng nhập.....	16
Hình 3.16 Giao diện trang Đăng ký.....	17
Hình 3.17 Giao diện thông báo khi đăng ký thành công.....	18
Hình 3.18 Giao diện email thông báo nhắc nhở.....	18
Hình 5.1 Phân công nhiệm vụ các thành viên trên Jira	25
Hình 5.2 Phân công nhiệm vụ các thành viên trên Jira	25
Hình 5.3 Phân công nhiệm vụ các thành viên trên Jira	25
Hình 5.4 Phân công nhiệm vụ các thành viên trên Jira	25
Hình 6.1 Thử nghiệm API đăng ký với Postman	29
Hình 6.2 Thử nghiệm API đăng nhập với Postman.....	29
Hình 6.3 Thử nghiệm API thêm danh mục công việc với Postman	30
Hình 6.4 Thử nghiệm API thêm công việc vào danh mục với Postman	30
Hình 6.5 Thử nghiệm API lấy danh sách danh mục công việc với Postman	31
Hình 6.6 Thử nghiệm API lấy danh sách công việc với Postman.....	31
Hình 6.7 Thử nghiệm API cập nhật công việc với Postman	32
Hình 6.8 Thử nghiệm API xóa công việc với Postman	32

DANH MỤC BẢNG BIỂU

Bảng 3.1 Các API liên quan Authentication	7
Bảng 3.2 Các API liên quan Categories	7
Bảng 5.1 Chi tiết và độ ưu tiên công việc	26

CHƯƠNG 1: GIỚI THIỆU

1.1 Tổng quan về dự án

Trong khuôn khổ môn học Công nghệ phần mềm, nhóm chúng em lựa chọn thực hiện đề tài **Quản lý công việc** với hình thức là một ứng dụng web. Chủ đề của dự án xoay quanh việc xây dựng một hệ thống cho phép người dùng quản lý, theo dõi và sắp xếp công việc cá nhân một cách hiệu quả. Dự án được phát triển theo mô hình kiến trúc client/server, đảm bảo tính phân tách rõ ràng giữa giao diện người dùng (frontend) và xử lý nghiệp vụ (backend). Giao tiếp giữa các phần được thực hiện thông qua API, giúp hệ thống dễ dàng mở rộng và bảo trì trong tương lai.

1.2 Mục tiêu của ứng dụng

Cung cấp cho người dùng một công cụ đơn giản nhưng hiệu quả để tổ chức công việc hằng ngày. Thông qua ứng dụng, người dùng có thể tạo mới công việc, cập nhật trạng thái thực hiện, đặt thời hạn hoàn thành và theo dõi toàn bộ danh sách công việc trong một giao diện trực quan. Ngoài ra, ứng dụng còn hướng đến việc hỗ trợ quản lý công việc theo thời gian, giúp người dùng chủ động hơn trong việc lập kế hoạch và hoàn thành nhiệm vụ đúng hạn. Từ đó, ứng dụng góp phần nâng cao ý thức kỷ luật cá nhân và cải thiện năng suất làm việc.

1.3 Lý do chọn đề tài

Xuất phát từ nhu cầu thực tế mà nhóm nhận thấy trong quá trình học tập cũng như trong cuộc sống hằng ngày. Ai cũng có lúc phải đối mặt với nhiều công việc cần giải quyết, và nếu không có một phương pháp quản lý phù hợp, việc quên nhiệm vụ, trễ hạn hay chồng chéo công việc là điều khó tránh khỏi. Do đó, việc xây dựng một công cụ hỗ trợ quản lý công việc là điều cần thiết, không chỉ có giá trị ứng dụng thực tiễn cao mà còn phù hợp để nhóm áp dụng các kiến thức đã học về phân tích hệ thống, thiết kế phần mềm, lập trình, kiểm thử và triển khai ứng dụng thực tế. Thông qua dự án này, nhóm mong muốn tạo ra một sản phẩm có thể ứng dụng lâu dài, đồng thời rèn luyện kỹ năng lập trình, tư duy hệ thống, làm việc theo quy trình, và làm quen với các công cụ hiện đại trong quy trình phát triển phần mềm chuyên nghiệp.

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU

2.1 Chức năng chính của hệ thống

Hệ thống “Quản lý công việc” được xây dựng nhằm hỗ trợ người dùng theo dõi và kiểm soát công việc của bản thân một cách hiệu quả, do đó các chức năng chính được thiết kế xoay quanh quá trình tạo, quản lý và cập nhật thông tin công việc. Các chức năng này đảm bảo tính logic, dễ sử dụng và đáp ứng đúng nhu cầu quản lý cá nhân của người dùng.

- **Chức năng thêm danh mục công việc:** khi tạo một danh mục công việc, người dùng chỉ cần nhập tên danh mục, và sau đó có thể thêm nhiều công việc khác nhau vào từng nhóm tương ứng. Việc phân loại này giúp dễ quản lý, theo dõi và phân biệt các loại công việc với nhau, đặc biệt khi khối lượng công việc ngày càng lớn.
- **Chức năng xóa danh mục công việc:** trong trường hợp một danh mục không còn cần thiết, người dùng có thể xóa nó khỏi hệ thống. Hệ thống sẽ đưa ra cảnh báo trước khi xóa, đặc biệt nếu trong danh mục đó vẫn còn các công việc đang hoạt động.
- **Chức năng thêm công việc:** người dùng có thể thêm mới công việc vào từng danh mục cụ thể. Mỗi công việc bao gồm tiêu đề, mô tả chi tiết, thời gian bắt đầu và hạn hoàn thành
- **Chức năng cập nhật công việc:** cho phép người dùng thay đổi thông tin của công việc đã tạo. Người dùng có thể chỉnh sửa tiêu đề, thay đổi nội dung điều chỉnh hạn hoàn thành hoặc chuyển trạng thái công việc (ví dụ: từ “Chưa hoàn thành” hoặc “Đã hoàn thành”). Điều này giúp hệ thống linh hoạt, phù hợp với tiến độ làm việc thực tế.
- **Chức năng xóa công việc:** người dùng có thể xóa những công việc không cần thiết. Việc này giúp danh sách công việc luôn gọn gàng, tránh tình trạng

tồn đọng những mục không còn giá trị sử dụng.

- **Chức năng thống kê công việc:** hệ thống cung cấp một số biểu đồ thống kê cơ bản để người dùng dễ dàng đánh giá tiến độ làm việc của mình trong tháng hoặc trong năm
- **Chức năng nhắc nhở:** khi có công việc sắp đến hạn, hệ thống sẽ gửi về mail thông báo cho người dùng để có thể nhanh chóng hoàn thành công việc đã được sắp xếp.

2.2 Yêu cầu phi chức năng của hệ thống

Các yêu cầu phi chức năng tuy không trực tiếp ảnh hưởng đến chức năng cụ thể, nhưng lại đóng vai trò rất quan trọng trong việc đánh giá chất lượng phần mềm.

- **Tính ổn định và hiệu năng cao:** khi số lượng công việc được tạo ra nhiều, hệ thống vẫn phải vận hành mượt mà, truy xuất dữ liệu nhanh và không bị gián đoạn. Điều này đặc biệt quan trọng với những người dùng có nhu cầu ghi chú, lên kế hoạch dài hạn hoặc sử dụng thường xuyên.
- **Giao diện thân thiện, dễ sử dụng:** toàn bộ hệ thống được thiết kế theo phong cách đơn giản, đảm bảo người dùng có thể dễ dàng làm quen và thao tác ngay từ lần sử dụng đầu tiên. Các thành phần giao diện như nút bấm, biểu tượng, màu sắc và bố cục đều được thiết kế đồng bộ, mang lại trải nghiệm người dùng tốt.
- **Tính bảo mật:** khi triển khai ở quy mô thực tế có lưu trữ tài khoản hoặc thông tin cá nhân người dùng. Dữ liệu công việc cần được lưu trữ an toàn, tránh mất mát hoặc bị truy cập trái phép.
- **Khả năng mở rộng và bảo trì:** ứng dụng được xây dựng theo kiến trúc tách biệt giữa frontend và backend, sử dụng API để giao tiếp, nên dễ dàng nâng cấp, thêm tính năng mới hoặc sửa lỗi mà không ảnh hưởng đến toàn hệ

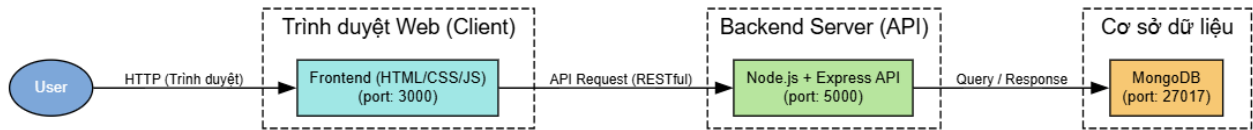
thống. Điều này rất quan trọng khi triển khai hệ thống lâu dài hoặc chuyển giao cho các nhóm phát triển tiếp theo.

- **Khả năng triển khai linh hoạt:** Với việc đóng gói bằng Docker, ứng dụng có thể triển khai nhanh chóng trên nhiều môi trường khác nhau, từ máy cá nhân đến máy chủ, mà không cần cấu hình phức tạp.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1 Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình Client – Server, gồm ba thành phần chính: giao diện người dùng (frontend), máy chủ API (backend) và cơ sở dữ liệu (MongoDB).



Luồng hoạt động của hệ thống:

Người dùng truy cập hệ thống thông qua trình duyệt và gửi yêu cầu HTTP đến giao diện frontend tại cổng 3000. Giao diện frontend xử lý tương tác và gửi yêu cầu API đến backend qua cổng 5000.

Backend xử lý logic nghiệp vụ, xác thực quyền truy cập, và tương tác với MongoDB để truy vấn hoặc cập nhật dữ liệu.

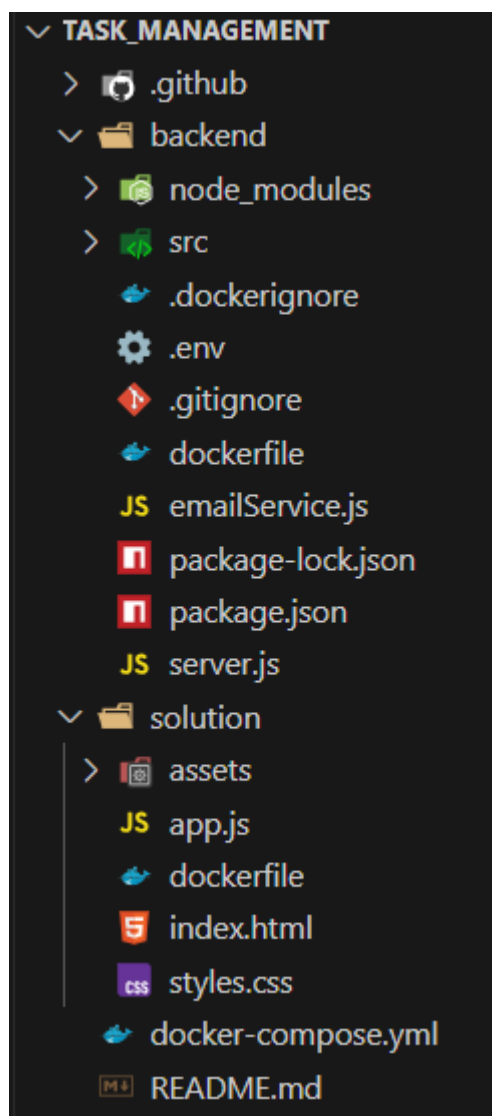
Sau khi có dữ liệu từ MongoDB, backend gửi phản hồi về cho frontend, sau đó frontend hiển thị thông tin cho người dùng.

3.2 Cấu trúc thư mục

Cấu trúc thư mục được xây dựng như sau:

Thư mục backend/ chứa toàn bộ mã nguồn phía server, nơi xử lý logic nghiệp vụ, kết nối cơ sở dữ liệu, xác thực người dùng và cung cấp các API cho frontend gọi đến. Thư mục backend/src/ gồm các thư mục như config/ để kết nối với cơ sở dữ liệu thông qua file môi trường .env, models/ để định nghĩa dữ liệu (MongoDB), controllers/ để xử lý request, routes/ để khai báo các endpoint, và services/ để xử lý logic nghiệp vụ. Backend đóng vai trò là trung tâm xử lý và giao tiếp với frontend thông qua RESTful API.

Thư mục solution/ là nơi xây dựng giao diện người dùng bằng ngôn ngữ HTML, CSS và Javascript. Đây là ứng dụng client-side, nơi người dùng tương tác trực tiếp thông qua trình duyệt, thực hiện các hành động như đăng ký, đăng nhập, thêm danh mục công việc, thêm công việc,... Solution gửi request tới backend qua HTTP (dùng axios), nhận dữ liệu JSON và hiển thị ra màn hình. Toàn bộ điều hướng, xử lý trạng thái và logic hiển thị đều nằm trong phần này, giúp tách biệt hoàn toàn khỏi backend.



Hình 3.1 Cấu trúc thư mục

3.3 Thiết kế cơ sở dữ liệu

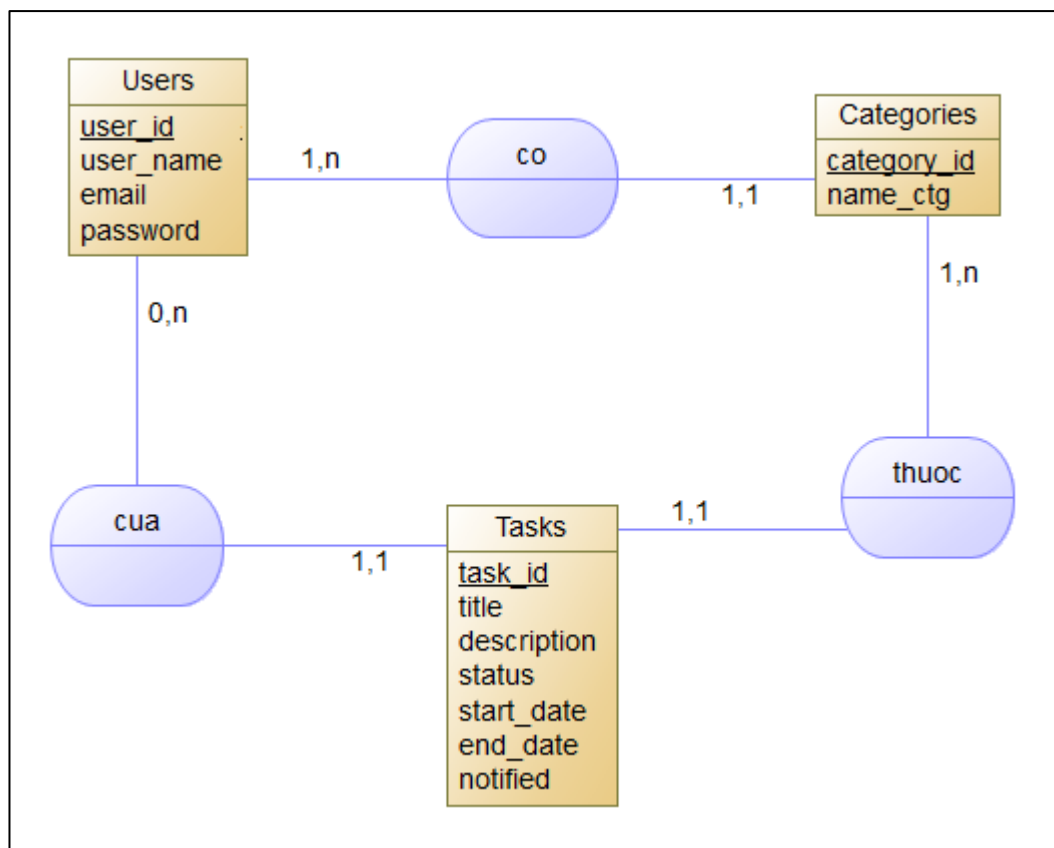
Xây dựng cơ sở dữ liệu để quản lý công việc gồm 3 bảng bao gồm:

Users: lưu thông tin đăng nhập người dùng trong hệ thống gồm có các thuộc tính như: user_id, user_name, email, password.

Categories: mỗi người dùng có thể tạo nhiều danh mục để tổ chức công việc gồm có các thuộc tính như: category_id, name_ctg, user_id.

Tasks: công việc cụ thể nằm trong từng danh mục, có thể bao gồm tiêu đề, mô tả, ngày bắt đầu, ngày đến hạn và trạng thái. Mỗi công việc thuộc về một người dùng và một danh mục cụ thể, gồm có các thuộc tính: task_id, title, description, status, start_date,

end_date, createdAt, updatedAt, notified, user_id, category_id.



Hình 3.2 Mô hình ERD

3.4 Thiết kế API

Ứng dụng Web quản lý công việc được xây dựng với các Endpoint chính như sau:

Bảng 3.1 Các API liên quan Authentication

Method	Endpoint	Mô tả
POST	/api/auth/login	Đăng nhập
POST	/api/auth/register	Đăng ký tài khoản mới

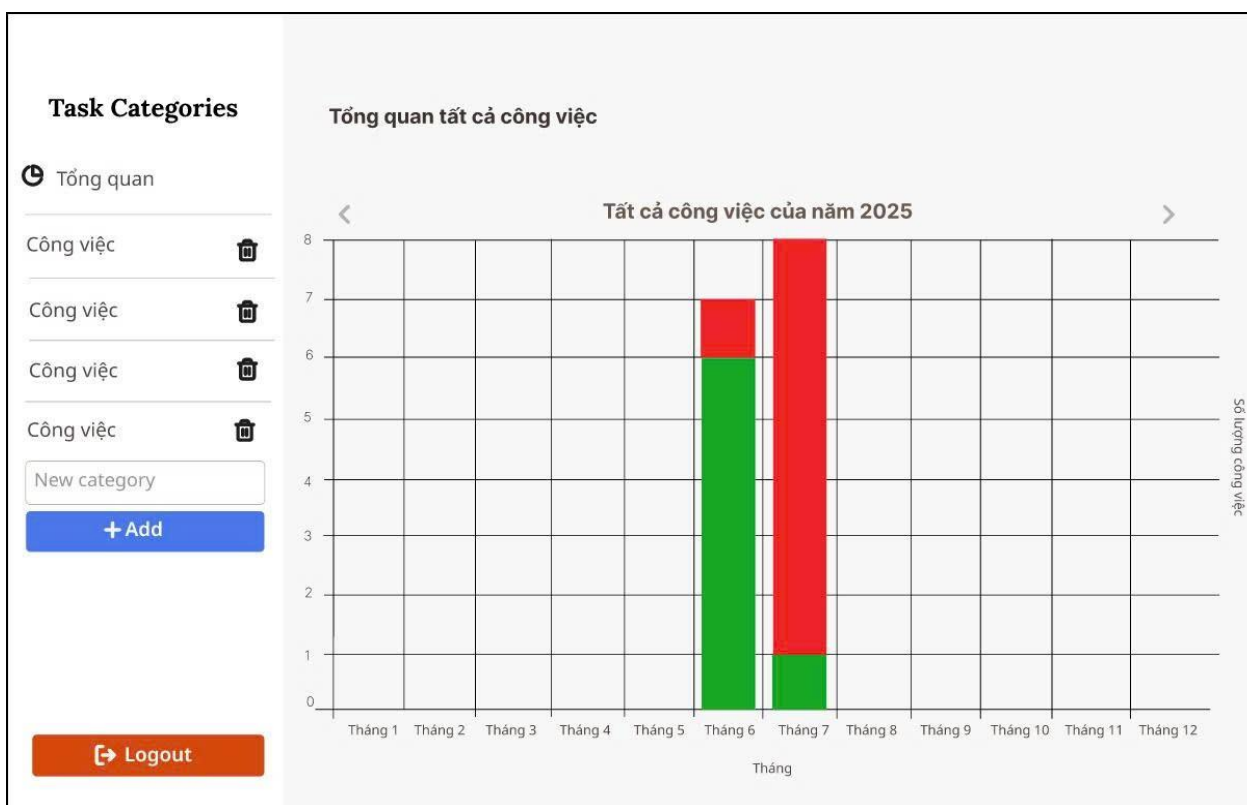
Bảng 3.2 Các API liên quan Categories

Method	Endpoint	Mô tả
POST	/api/categories	Thêm danh mục

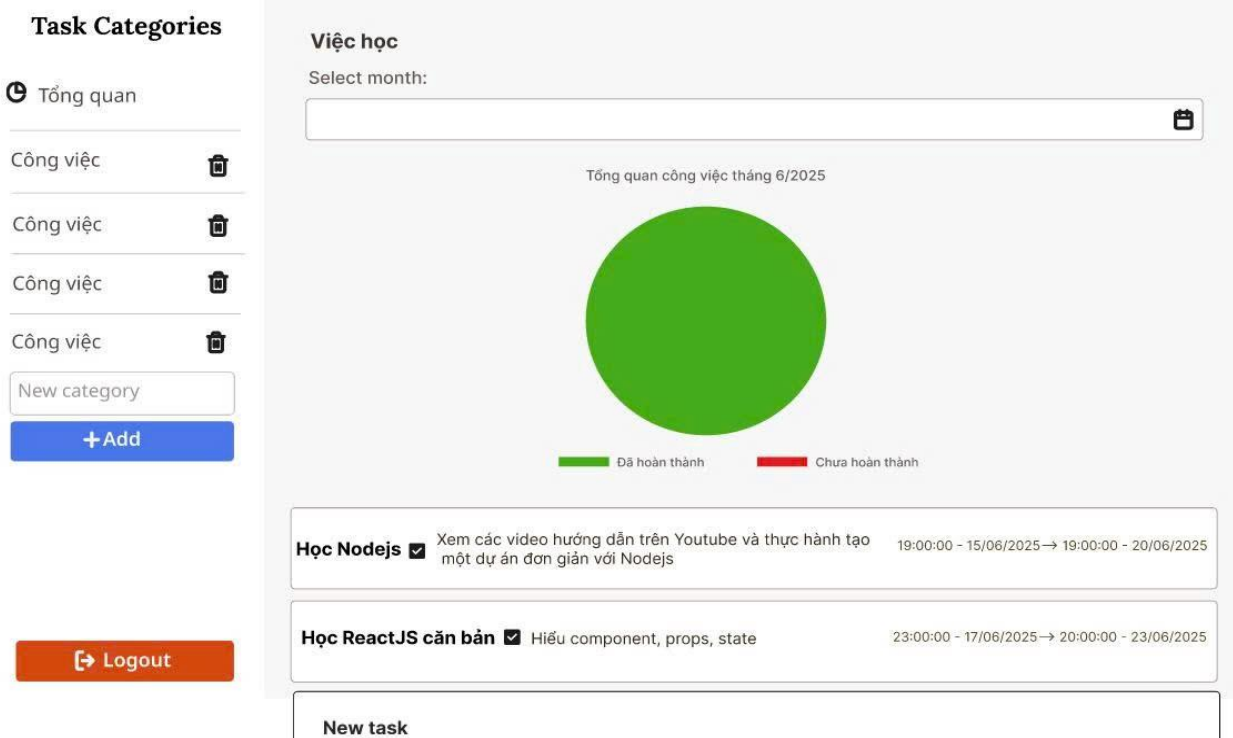
GET	/api/categories	Lấy danh mục
DELETE	/api/categories/:id	Xóa danh mục
GET	/api/categories/:id/tasks	Lấy các công việc thuộc danh mục
POST	/api/categories/:id/tasks	Thêm công việc vào danh mục
DELETE	/api/categories/:id/tasks/:tid	Xóa công việc khỏi danh mục
PATCH	/api/categories/:id/tasks/:tid	Cập nhật thông tin công việc

3.5 Thiết kế giao diện (UI/UX)

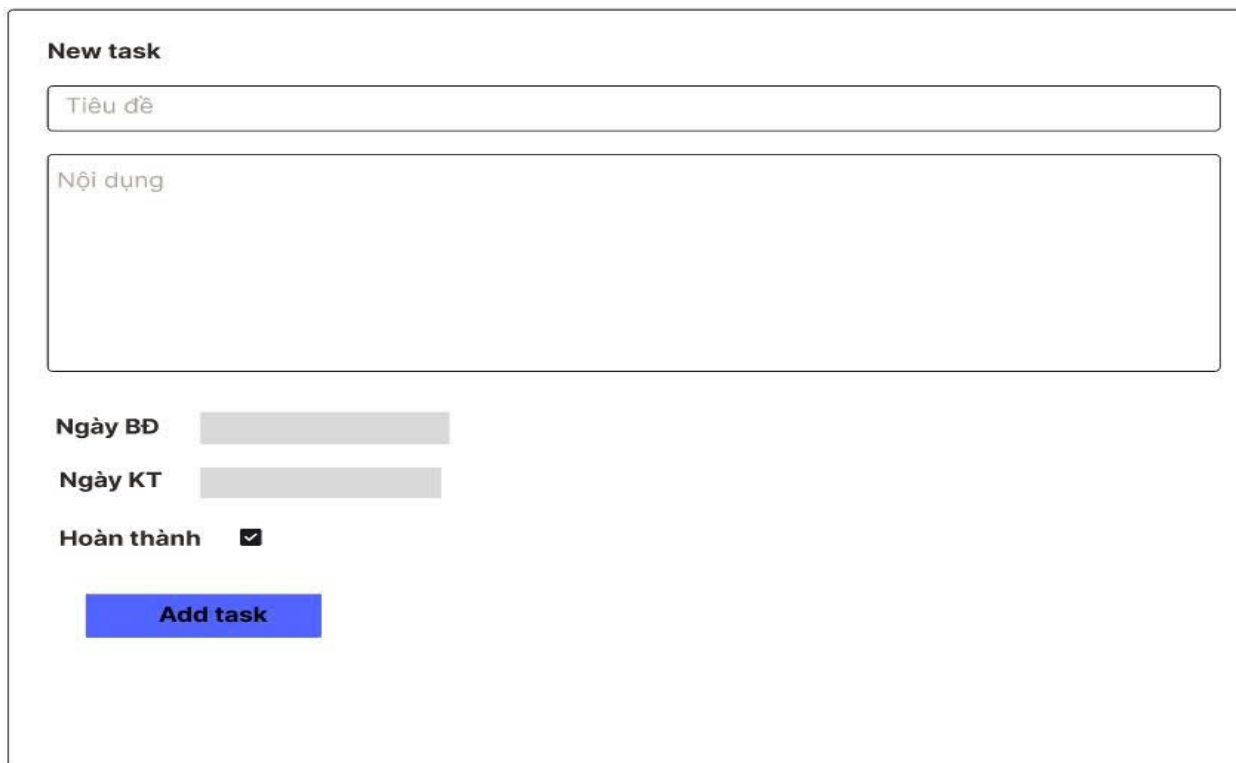
3.5.1 Thiết kế giao diện UI



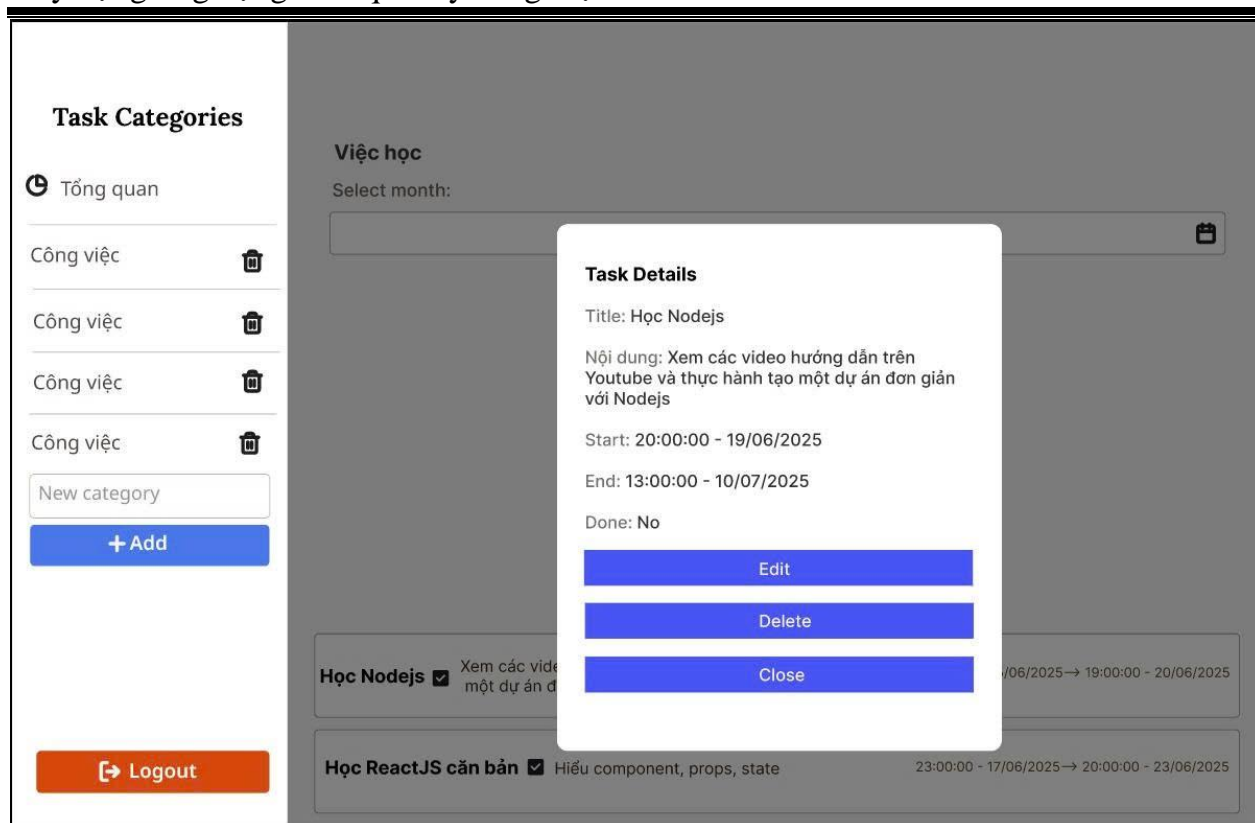
Hình 3.3 Giao diện trang Tổng quan với Figma



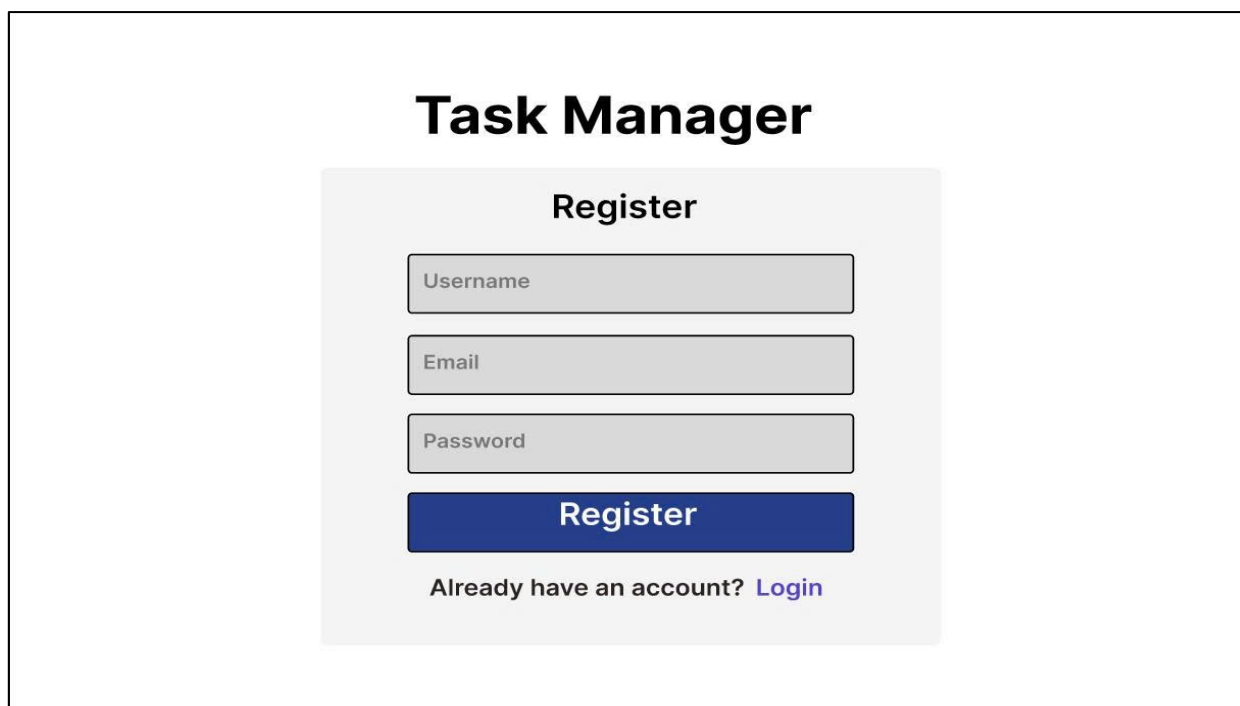
Hình 3.4 Giao diện trang Công việc với Figma



Hình 3.5 Giao diện Thêm công việc với Figma



Hình 3.6 Giao diện trang Chi tiết công việc với Figma



Hình 3.7 Giao diện trang Đăng ký với Figma

Task Manager

Login

Email

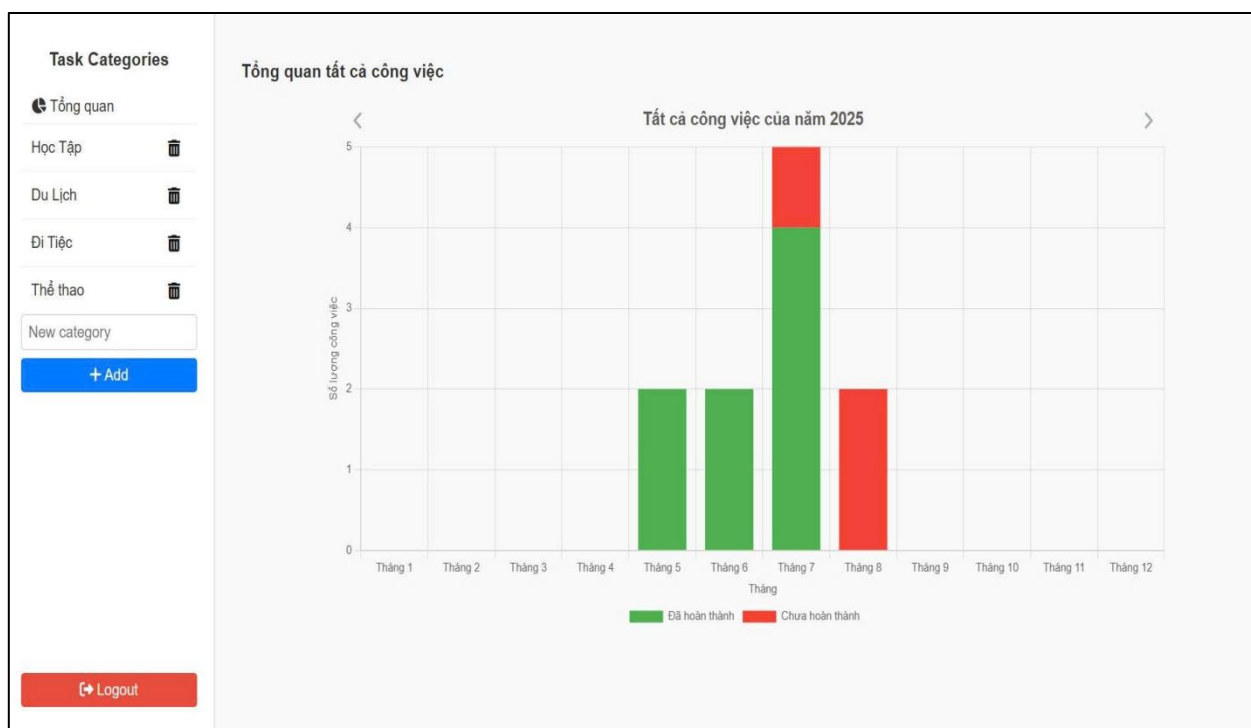
Password

Login

Don't have an account? [Register](#)

Hình 3.8 Giao diện trang Đăng nhập với Figma

3.5.2 Thiết kế giao diện UX



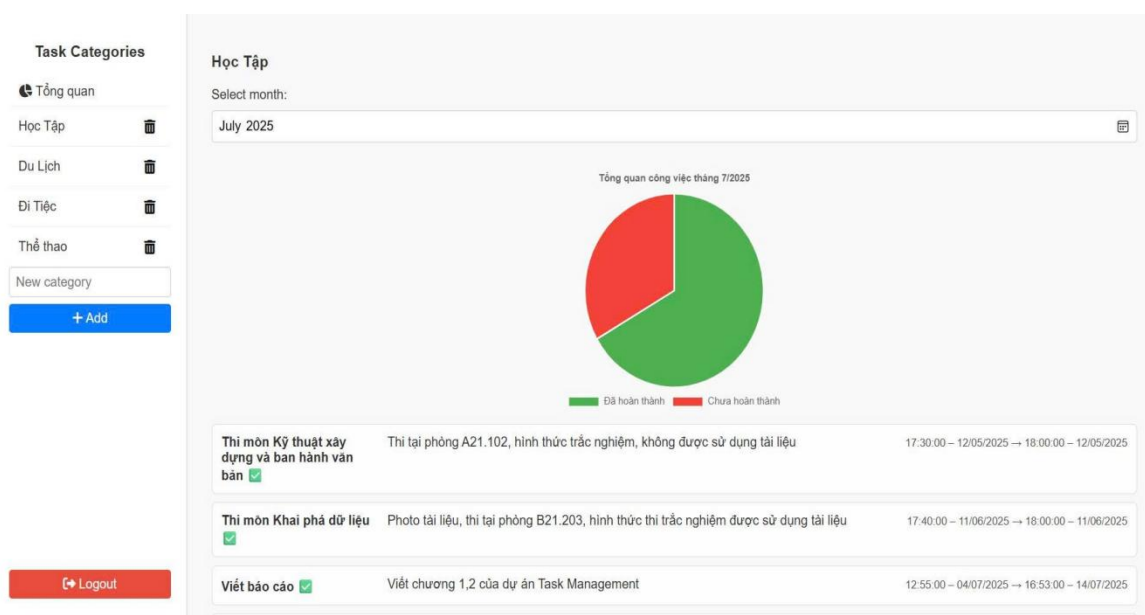
Hình 3.9 Giao diện trang Tổng quan

Giao diện thống kê tổng quan tất cả công việc của người dùng trong năm, được

trình bày dưới dạng biểu đồ cột trực quan. Giao diện được chia làm hai phần chính: thanh điều hướng danh mục bên trái và khu vực hiển thị biểu đồ bên phải.

- Khu vực danh mục công việc bên trái: là nơi người dùng có thể quản lý các danh mục công việc. Mỗi danh mục đều được hiển thị kèm biểu tượng thùng rác để xóa. Bên dưới là ô nhập liệu “New category” giúp người dùng dễ dàng thêm danh mục mới, cùng với nút + Add màu xanh nổi bật. Ngoài ra, còn có nút Logout màu cam để đăng xuất khỏi hệ thống, đảm bảo tính bảo mật tài khoản.
- Khu vực thống kê bên phải: biểu đồ cột thể hiện số lượng công việc theo từng tháng trong năm, được phân biệt theo trạng thái hoàn thành:
 - Cột màu xanh lá: biểu thị số công việc đã hoàn thành.
 - Cột màu đỏ: biểu thị số công việc chưa hoàn thành.

Ngoài ra, hệ thống cũng có biểu tượng mũi tên trái/phải ở đầu tiêu đề để chuyển đổi giữa các năm khác nhau, giúp người dùng dễ dàng xem thống kê theo từng năm.



Hình 3.10 Giao diện trang Công việc

Giao diện hiển thị công việc thuộc danh mục “**Học Tập**”, với chức năng thống kê tiến độ công việc theo từng tháng. Khi nhấn vào danh mục công việc này sẽ thấy:

- Phía trên là mục chọn tháng (Select month), cho phép người dùng lọc và xem công việc theo từng tháng cụ thể. Ngay bên dưới là biểu đồ hình tròn thể hiện tỷ lệ

công việc đã hoàn thành và chưa hoàn thành trong tháng đó. Màu xanh lá đại diện cho các công việc đã hoàn tất, trong khi màu đỏ biểu thị những công việc vẫn chưa được thực hiện xong. Biểu đồ được thiết kế trực quan giúp người dùng có thể đánh giá tổng quan tiến độ làm việc của mình trong một khoảng thời gian cụ thể.

- Bên dưới biểu đồ là danh sách chi tiết các công việc. Mỗi công việc đều có tiêu đề nổi bật, mô tả nội dung cụ thể và thời gian bắt đầu và kết thúc rõ ràng. Các công việc đã hoàn thành được đánh dấu bằng biểu tượng tick xanh, giúp người dùng dễ dàng theo dõi tình trạng hiện tại của từng nhiệm vụ.

The screenshot shows a web interface for adding a new task. On the left, there's a sidebar with 'Task Categories' including 'Tổng quan', 'Học Tập', 'Du Lịch', 'Đi Tiệc', and 'Thể thao', each with a trash icon. Below these is a 'New category' input field and a '+ Add' button. At the bottom left is a 'Logout' button. The main area is titled 'New task' and contains a 'Title' input field, a 'Content' text area, 'Start Date' and 'End Date' date pickers, a 'Done' checkbox, and an 'Add Task' button.

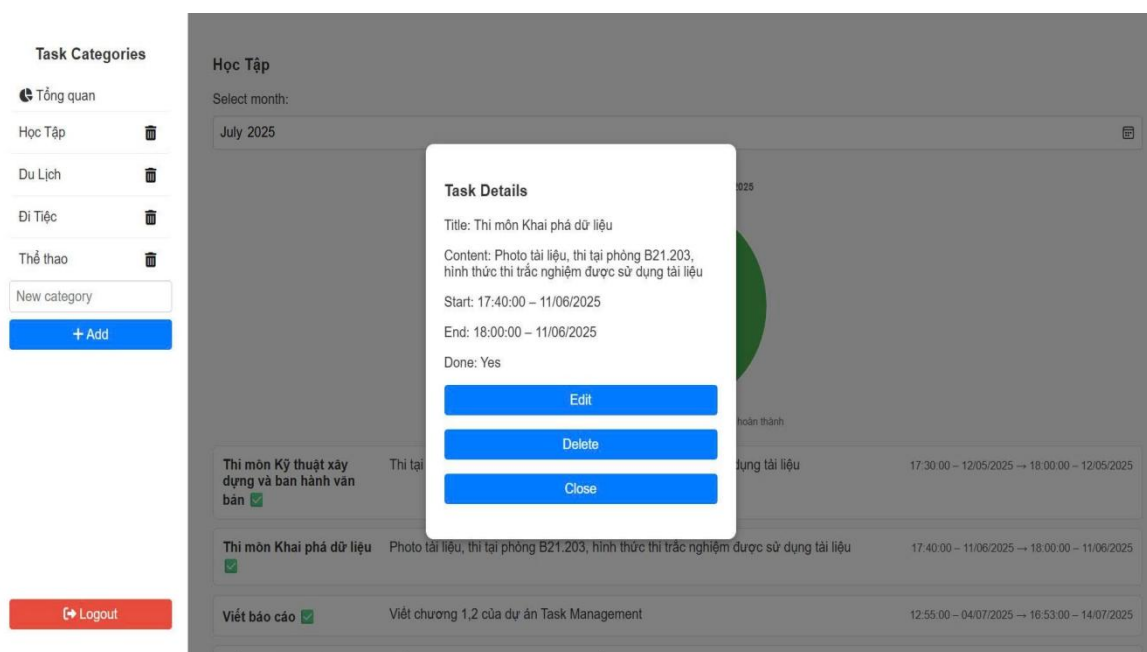
Hình 3.11 Giao diện khi Thêm công việc

Kéo xuống phía dưới cùng của trang Công việc, người dùng sẽ thấy chỗ để thêm mới công việc gồm các trường thông tin cơ bản cần thiết:

- **Title:** ô nhập tiêu đề công việc, giúp định danh nhanh nhiệm vụ cần làm.
- **Content:** ô nhập mô tả chi tiết nội dung công việc, để người dùng dễ nhớ mục tiêu cụ thể.
- **Start Date và End Date:** chọn ngày giờ bắt đầu và kết thúc công việc, giúp hệ thống theo dõi đúng hạn, người dùng chọn từ giao diện lịch sổ xuống.
- **Done:** ô checkbox dùng để đánh dấu trạng thái hoàn thành của công việc nếu muốn cập nhật ngay khi thêm mới.

- **Nút Add Task:** sau khi điền đầy đủ thông tin, người dùng bấm nút này để lưu công việc mới vào cơ sở dữ liệu tương ứng với danh mục đang chọn.

Giao diện được bố trí hợp lý, dễ sử dụng, đặc biệt phù hợp với những người dùng cần một công cụ quản lý đơn giản, gọn nhẹ nhưng vẫn đảm bảo đủ thông tin quan trọng. Việc sử dụng các thành phần như date picker hay checkbox giúp tăng tính tương tác và hạn chế lỗi khi nhập liệu thủ công.



Hình 3.12 Giao diện trang Chi tiết công việc

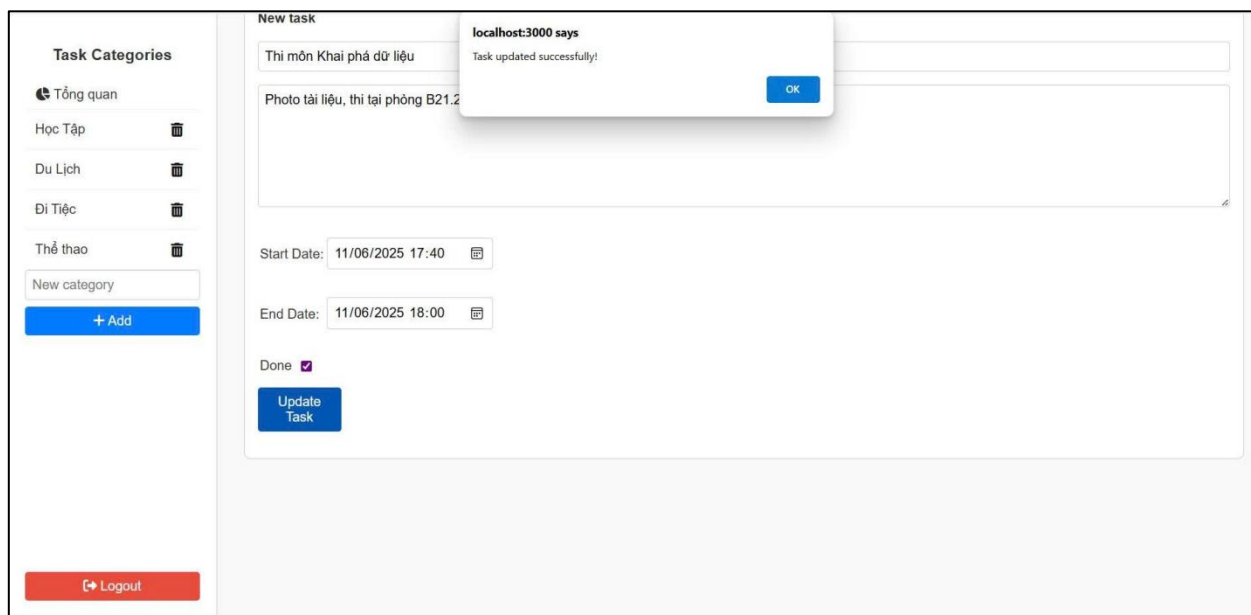
Hình ảnh trên mô tả hộp thoại (popup) hiển thị chi tiết của một công việc khi người dùng bấm chọn vào một nhiệm vụ trong danh sách. Giao diện được thiết kế dưới dạng cửa sổ nổi (modal), giúp người dùng dễ dàng theo dõi nội dung mà không cần chuyển sang trang khác, từ đó tăng tính tiện dụng và giữ sự liền mạch trong trải nghiệm.

Trong hộp thoại “**Task Details**”, toàn bộ thông tin của công việc được trình bày rõ ràng, ngay bên dưới là ba nút thao tác chính:

- **Edit:** cho phép người dùng chỉnh sửa lại nội dung công việc, chuyển hướng sang form cập nhật.
- **Delete:** cho phép xóa công việc đang xem khỏi danh sách.
- **Close:** đóng hộp thoại và quay về giao diện danh sách công việc ban đầu.

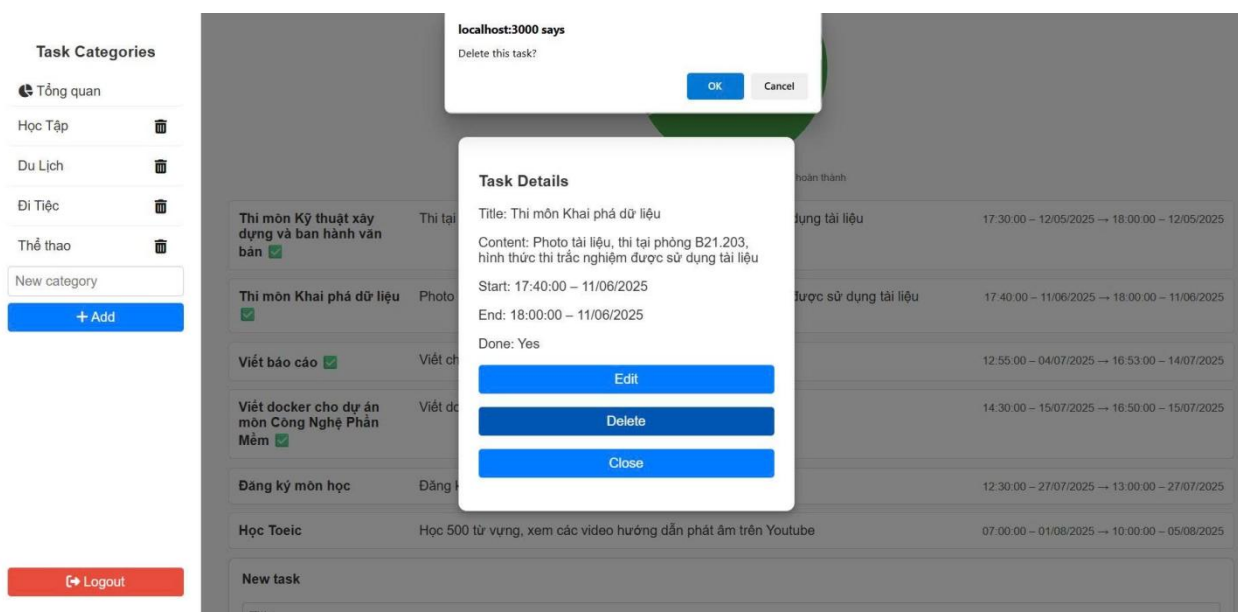
Xây dựng ứng dụng web quản lý công việc

Việc sử dụng popup giúp người dùng dễ dàng xem và thao tác nhanh chóng trên từng công việc mà không làm mất ngữ cảnh đang thao tác.



Hình 3.13 Giao diện khi Update công việc

Sau người dùng muốn chỉnh sửa một số chi tiết của công việc ví dụ: nội dung, tiêu đề hoặc ngày hoàn thành,.. thì hãy nhấn vào nút “Update Task”, lúc này công việc sẽ được update và hệ thống sẽ gửi một hộp thoại thông báo rằng công việc của bạn đã được update thành công, chọn “Ok” để kết thúc thao tác.



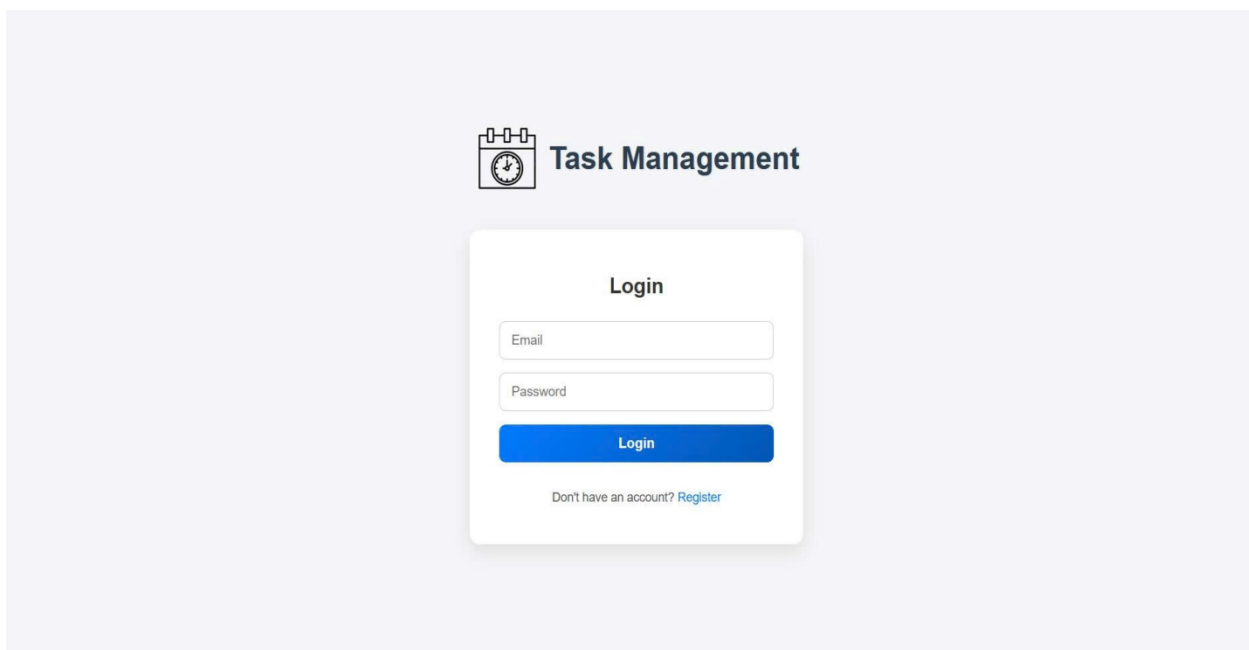
Hình 3.14 Giao diện khi xóa công việc

Khi người dùng nhấn nút Delete trong cửa sổ chi tiết công việc (Task Details), hệ thống sẽ ngay lập tức hiển thị một hộp thoại xác nhận (confirm) với nội dung "Delete this task?".

Hộp thoại xác nhận này có hai lựa chọn:

- **OK:** nếu người dùng đồng ý, công việc sẽ bị xóa khỏi danh sách và cơ sở dữ liệu.
- **Cancel:** nếu người dùng thay đổi ý định, thao tác xóa sẽ bị hủy và không ảnh hưởng đến dữ liệu.

Thiết kế này đảm bảo tính an toàn và tránh thao tác nhầm, giúp người dùng có cơ hội xem lại trước khi thực hiện hành động không thể hoàn tác. Đây là một điểm quan trọng trong các hệ thống quản lý, đặc biệt khi xử lý dữ liệu cá nhân hoặc dữ liệu quan trọng.



Hình 3.15 Giao diện trang Đăng nhập

Phía trên cùng là tên ứng dụng “Task Management” cùng biểu tượng lịch có đồng hồ, giúp người dùng dễ nhận diện mục đích sử dụng ngay từ đầu.

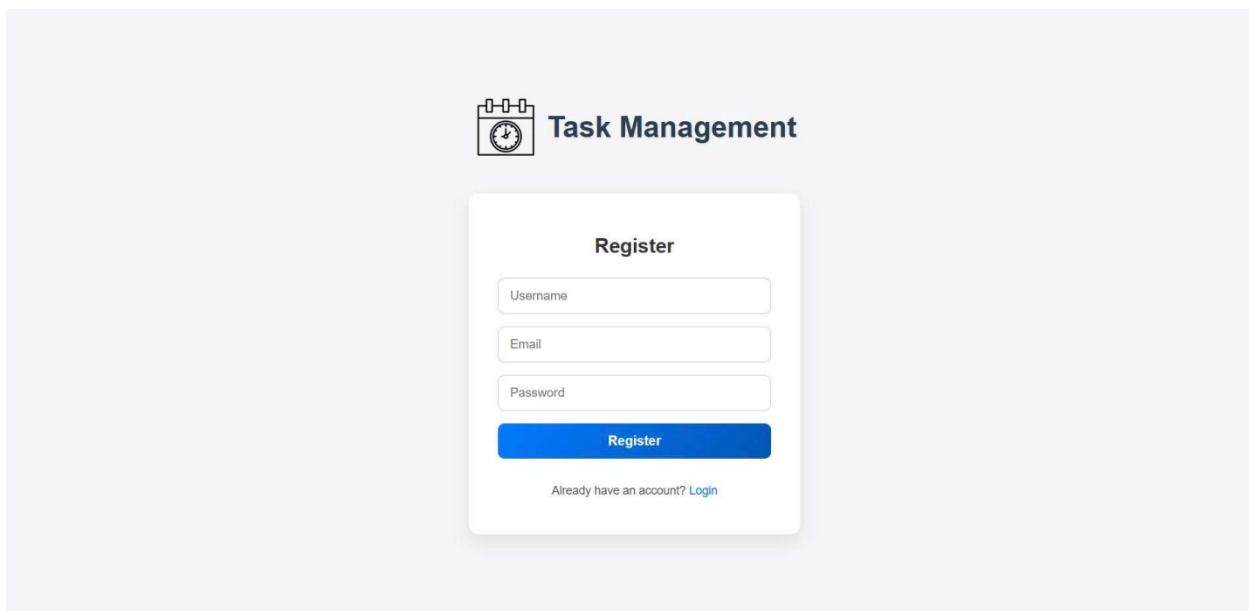
Phần chính giữa trang là form đăng nhập, bao gồm:

- **Email:** nơi người dùng nhập địa chỉ email đã đăng ký.
- **Password:** nhập mật khẩu tương ứng với tài khoản.

- **Nút “Login”**: xác thực thông tin và chuyển hướng đến giao diện chính nếu đăng nhập thành công.

Ngay bên dưới là dòng chữ “Don’t have an account? Register”, cho phép người dùng mới dễ dàng tạo tài khoản bằng cách nhấn vào liên kết “Register”. Thiết kế này đảm bảo quá trình chuyển đổi giữa đăng nhập và đăng ký được liền mạch, không gây rối.

Ngoài ra, giao diện sử dụng màu sắc tươi sáng và nút bấm màu xanh đậm nổi bật, giúp tăng khả năng nhận biết hành động chính của người dùng.

The image shows a web interface for a 'Task Management' application. At the top, there is a logo consisting of a calendar icon and the text 'Task Management'. Below the logo is a white rectangular form titled 'Register'. Inside the form, there are three input fields labeled 'Username', 'Email', and 'Password'. Below these fields is a prominent blue button labeled 'Register'. At the bottom of the form, there is a link that says 'Already have an account? Login'.

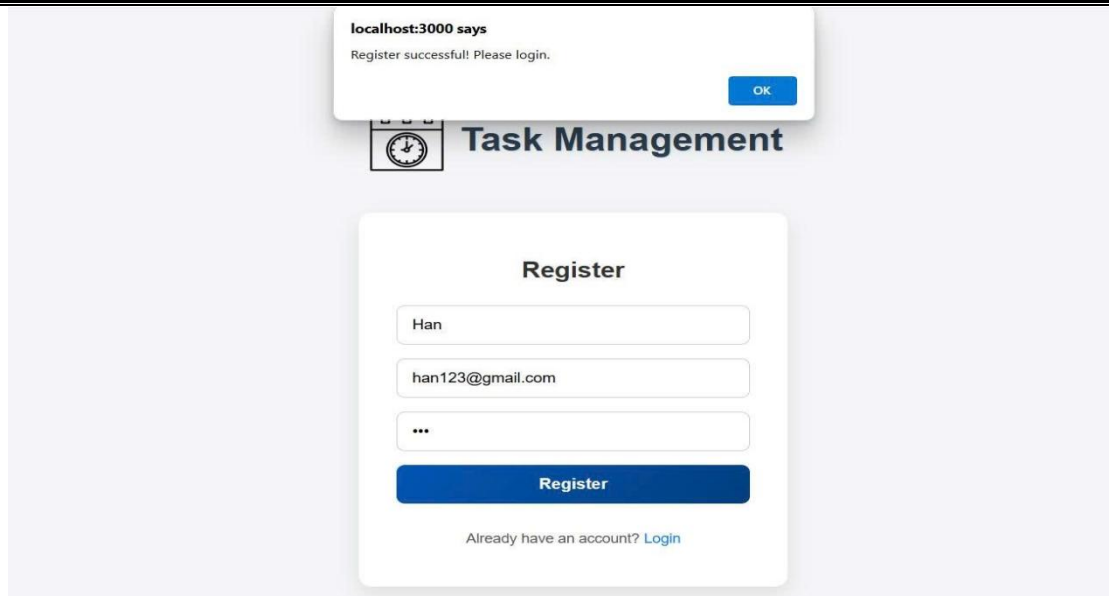
Hình 3.16 Giao diện trang Đăng ký

Trên cùng là logo và tên ứng dụng “Task Management”, tạo cảm giác nhận diện thương hiệu đồng bộ với trang đăng nhập.

Phần trung tâm của trang là form đăng ký gồm ba trường chính:

- **Username**: tên hiển thị của người dùng trong hệ thống.
- **Email**: địa chỉ email hợp lệ được sử dụng để xác thực tài khoản và đăng nhập.
- **Password**: mật khẩu bảo mật để đăng nhập vào hệ thống.

Sau khi điền đầy đủ thông tin, người dùng nhấn nút “Register” để tiến hành tạo tài khoản. Nếu đã có tài khoản, hệ thống cung cấp liên kết “Login” ngay phía dưới để quay lại trang đăng nhập một cách thuận tiện.



Hình 3.17 Giao diện thông báo khi đăng ký thành công

Sau khi người dùng điền đầy đủ thông tin và nhấn nút “Register”, hệ thống sẽ tiến hành kiểm tra hợp lệ, lưu trữ tài khoản vào cơ sở dữ liệu, sau đó hiển thị thông báo xác nhận dưới dạng hộp thoại: "Register successful! Please login." Thông báo này xuất hiện rõ ràng ở giữa màn hình, giúp người dùng dễ dàng nhận biết rằng quá trình đăng ký đã thành công. Sau khi nhấn nút “OK”, hệ thống sẽ điều hướng hoặc yêu cầu người dùng chuyển sang trang đăng nhập để sử dụng tài khoản vừa tạo.



Hình 3.18 Giao diện email thông báo nhắc nhở

Khi một công việc sắp đến hạn hoàn thành, hệ thống sẽ gửi thông báo nhắc nhở cho người dùng dưới dạng email cùng với các nội dung rõ ràng về công việc đó gồm có tiêu đề, nội dung và thời hạn hoàn thành của công việc với mục đích là giúp người dùng không bỏ lỡ các nhiệm vụ quan trọng, duy trì tiến độ công việc, đồng thời tăng trải nghiệm sử dụng bằng cách nhắc nhở kịp thời.

CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

4.1 Các công nghệ đã sử dụng

Ngôn ngữ lập trình:

- Node.js (JavaScript): Dùng cho backend.
- HTML, CSS, JavaScript: Dùng cho frontend.
- MongoDB Query Language (MQL): Truy vấn dữ liệu trong MongoDB.

Framework/ Thư viện:

Frontend:

- Axios: Gửi các yêu cầu HTTP đến backend (đăng nhập, lấy danh mục, công việc...)
- Chart.js: Hiển thị biểu đồ cột và biểu đồ tròn thống kê công việc theo tháng, năm.

Backend:

- Express: Xây dựng REST API.
- Mongoose: Kết nối MongoDB và định nghĩa schema.
- JWT (json web token): Xác thực người dùng qua JWT.
- bcryptjs: Mã hóa mật khẩu.
- dotenv: Load biến môi trường từ .env.
- cors: Cho phép frontend truy cập tài nguyên backend.
- nodemailer: Gửi email.

Cơ sở dữ liệu:

MongoDB: NoSQL Database dùng để lưu trữ thông tin người dùng, danh mục công việc và công việc,..

Kiểm thử:

Swagger: mô tả, thiết kế và trực quan hóa tài liệu API.

Postman: kiểm thử API.

GitHub Actions: kiểm thử khi có thay đổi mã nguồn.

Triển khai:

Docker: đóng gói và triển khai ứng dụng.

4.2 Quy trình CI/CD với GitHub Actions

Kiểm thử Backend(CI)

```
name: CI - Test Backend

on:
  push:
    paths:
      - 'backend/**'
  pull_request:
    paths:
      - 'backend/**'

jobs:
  test-backend:
    name: Test Backend
    runs-on: ubuntu-latest

    steps:
      - name: Checkout code
        uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: '18'

      - name: Install backend dependencies
        working-directory: ./backend
        run: npm install

      - name: Run backend tests
        working-directory: ./backend
        run: npm test
```

Quy trình:

- Kích hoạt khi có push hoặc pull request với các thay đổi trong thư mục back-end/.
- Thiết lập môi trường Node.js phiên bản 18.
- Cài đặt phụ thuộc cho backend bằng lệnh npm install.
- Chạy lệnh kiểm thử với npm test.

Kiểm thử Frontend(CI)

name: CI - Frontend (JS thuần)

on:

```
push:
  paths:
    - 'solution/**'
pull_request:
  paths:
    - 'solution/**'
```

jobs:

```
test-frontend:
  name: Frontend Check
  runs-on: ubuntu-latest

  steps:
    - name: Checkout code
      uses: actions/checkout@v3

    - name: Kiểm tra file frontend (JS thuần)
      run: echo "No automated tests. This is pure JS. Manual testing recommended."
```

Quy trình:

- Kích hoạt khi có push hoặc pull request với các thay đổi trong thư mục solution/.
- Thiết lập môi trường CI sử dụng hệ điều hành Ubuntu.
- Thực hiện bước checkout mã nguồn từ repository.
- Kiểm tra file frontend: do frontend sử dụng JavaScript thuần (không framework) nên không có kiểm thử tự động, quá trình kiểm thử cần thực hiện thủ công (manual testing).

4.3 Cấu hình Docker và quy trình triển khai ứng dụng

Cấu hình:

Dockerfile (backend): Định nghĩa môi trường Node.js, copy source code, cài đặt dependencies và khởi động server backend.

Dockerfile (frontend): dùng Nginx để phục vụ static files (index.html, app.js,...).

docker-compose.yml: Quản lý và chạy nhiều container cùng lúc, bao gồm:

- backend: chạy Node.js Express API (port 5000), sử dụng biến môi trường từ .env, kết nối MongoDB.
- frontend: chạy giao diện JavaScript thuần (port 3000, map vào cổng 80 trong container).
- mongo: dịch vụ MongoDB (port 27017), lưu trữ dữ liệu qua volume mongo_data.

Triển khai:

Ứng dụng sẽ được triển khai với:

- Backend API chạy tại: `http://localhost:5000`
- Frontend chạy tại: `http://localhost:3000`
- MongoDB: `mongodb://mongo:27017/db`

CHƯƠNG 5: QUẢN LÝ DỰ ÁN

5.1 Sử dụng Jira để lập kế hoạch và theo dõi tiến độ

Lập kế hoạch Sprint và phân công công việc

Trong quá trình phát triển phần mềm, nhóm dự án chia toàn bộ lộ trình thành nhiều Sprint – mỗi Sprint thường kéo dài từ 1 đến 2 tuần, tùy vào quy mô và yêu cầu cụ thể. Mỗi Sprint được xem là một giai đoạn phát triển độc lập, với mục tiêu rõ ràng và sản phẩm có thể kiểm thử được. Dựa vào Product Backlog – danh sách tập hợp tất cả các tính năng, yêu cầu và đề xuất của dự án, nhóm lựa chọn các mục tiêu ưu tiên để đưa vào Sprint Backlog. Mỗi task được tạo thành một Issue trong Jira, được biểu diễn dưới dạng User Story – mô tả cụ thể yêu cầu của người dùng hoặc hành vi của hệ thống cần hoàn thiện.

Mỗi Issue đều được phân công rõ ràng cho các thành viên trong nhóm kèm theo thời hạn hoàn thành (deadline), mức độ ưu tiên (Priority), và có thể đính kèm tài liệu, hình ảnh liên quan để thuận tiện cho việc thực hiện. Việc phân công này giúp đảm bảo trách nhiệm cá nhân và tối ưu hiệu suất làm việc trong nhóm.

Theo dõi tiến độ

Để trực quan hóa quy trình làm việc và kiểm soát tiến độ, nhóm sử dụng Scrum Board trong Jira, với các cột chính: To Do → In Progress → Done. Mỗi thành viên sẽ chủ động cập nhật trạng thái công việc của mình khi bắt đầu hoặc hoàn thành một tính năng, giúp toàn nhóm nắm được tiến độ tổng thể của Sprint.






















Công cụ Jira còn hỗ trợ tự động tạo các báo cáo trực quan như Burndown Chart, thể hiện số lượng công việc còn lại theo thời gian, giúp nhóm dễ dàng theo dõi tiến độ và phát hiện sớm các nguy cơ chậm trễ. Ngoài ra, sau mỗi Sprint, Jira cung cấp Sprint Report để tổng hợp kết quả đạt được, so sánh giữa kế hoạch và thực tế, từ đó nhóm có thể rút kinh nghiệm, điều chỉnh chiến lược và ước lượng tốt hơn cho các Sprint tiếp theo.

Việc kết hợp quy trình Scrum và công cụ Jira giúp nhóm làm việc hiệu quả, minh bạch và kiểm soát chất lượng sản phẩm một cách chuyên nghiệp.[1]






















5.2 Phân công nhiệm vụ

Dự án web được hoàn thành trong 5 Sprint với mỗi nhiệm vụ phân công cho từng thành viên cụ thể như sau:

Xây dựng ứng dụng web quản lý công việc

Work	Assignee	Reporter	Status	Sprint	Comments	+
<input checked="" type="checkbox"/> CNPM-29 Tạo API CRUD cho Task (create/read/update/delete)	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	
<input checked="" type="checkbox"/> CNPM-19 Thiết kế giao diện trang "Danh sách công việc"	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	
<input checked="" type="checkbox"/> CNPM-17 Bảo vệ các route riêng tư (middleware auth)	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 2	 Add comment	
<input checked="" type="checkbox"/> CNPM-16 Thêm chức năng Đăng xuất	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 2	 Add comment	
<input checked="" type="checkbox"/> CNPM-15 Tạo API xác thực JWT (Đăng ký, Đăng nhập, Xác thực token)	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 2	 Add comment	
<input checked="" type="checkbox"/> CNPM-14 Tạo form Đăng nhập người dùng	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 2	 Add comment	
<input checked="" type="checkbox"/> CNPM-13 Tạo form Đăng ký người dùng	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 2	 Add comment	












Hình 5.1 Phân công nhiệm vụ các thành viên trên Jira

Work	Assignee	Reporter	Status	Sprint	Comments	+
<input checked="" type="checkbox"/> CNPM-27 Tạo biểu đồ thống kê tiến độ (hoàn thành/chưa hoàn thành)	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 4	 Add comment	
<input checked="" type="checkbox"/> CNPM-26 Hiện thị số lượng công việc theo trạng thái	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 4	 Add comment	
<input checked="" type="checkbox"/> CNPM-25 Tạo API gửi thông báo nhắc nhở (notification system)	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 4	 Add comment	
<input checked="" type="checkbox"/> CNPM-24 Thêm chức năng xóa công việc	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	
<input checked="" type="checkbox"/> CNPM-23 Thêm chức năng sửa nội dung công việc	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	
<input checked="" type="checkbox"/> CNPM-22 Thêm chức năng tạo mới công việc	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	
<input checked="" type="checkbox"/> CNPM-21 Hiện thị danh sách công việc của người dùng (theo trạng thái: done)	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 3	 Add comment	

Hình 5.2 Phân công nhiệm vụ các thành viên trên Jira

Work	Assignee	Reporter	Status	Sprint	Comments	+
<input checked="" type="checkbox"/> CNPM-34 Triển khai backend lên Docker	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-33 Kiểm thử chức năng frontend	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-32 Viết unit test cho backend	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-31 Cải thiện trải nghiệm người dùng (loading, thông báo, form validate)	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 5	 Add comment	
<input checked="" type="checkbox"/> CNPM-30 Responsive cho điện thoại và máy tính bảng	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 5	 Add comment	
<input checked="" type="checkbox"/> CNPM-29 Thiết kế layout tổng thể (Header, Sidebar, Main content)	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 5	 Add comment	
<input checked="" type="checkbox"/> CNPM-28 Thiết lập nhắc nhở công việc sắp hết hạn (qua email)	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 4	 Add comment	

Hình 5.3 Phân công nhiệm vụ các thành viên trên Jira

Work	Assignee	Reporter	Status	Sprint	Comments	+
<input checked="" type="checkbox"/> CNPM-36 Kiểm tra toàn bộ hệ thống	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-35 Triển khai frontend lên Docker	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-34 Triển khai backend lên Docker	 Huỳnh Khánh Duy	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-33 Kiểm thử chức năng frontend	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-32 Viết unit test cho backend	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 6	 Add comment	
<input checked="" type="checkbox"/> CNPM-31 Cải thiện trải nghiệm người dùng (loading, thông báo, form validate)	 Nguyễn Thị Ngọc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 5	 Add comment	
<input checked="" type="checkbox"/> CNPM-30 Responsive cho điện thoại và máy tính bảng	 Hồ Nguyễn Quốc...	 Huỳnh Khánh Duy	DONE ▾	CNPM Sprint 5	 Add comment	

Hình 5.4 Phân công nhiệm vụ các thành viên trên Jira

Bảng 5.1 Chi tiết và độ ưu tiên công việc

Sprint	Mô tả	Story Points	Độ ưu tiên
Sprint 1	Tạo form đăng ký người dùng	5	Trung bình
Sprint 1	Tạo form đăng nhập người dùng	5	Trung bình
Sprint 1	Tạo API xác thực JWT (Đăng ký, Đăng nhập, Xác thực token)	5	Cao
Sprint 1	Thêm chức năng Đăng xuất	5	Trung bình
Sprint 1	Bảo vệ các route riêng tư (middleware auth)	5	Cao
Sprint 2	Thiết kế giao diện trang "Danh sách công việc"	5	Trung bình
Sprint 2	Tạo API CRUD cho Task (create/read/update/delete)	5	Trung bình
Sprint 2	Hiển thị danh sách công việc của người dùng (theo trạng thái: done)	5	Trung bình
Sprint 2	Thêm chức năng tạo mới công việc	5	Cao
Sprint 2	Thêm chức năng sửa nội dung công việc	5	Cao
Sprint 2	Thêm chức năng xóa công việc	5	Cao
Sprint 3	Tạo API gửi thông báo nhắc nhở (notification system)	5	Cao

Xây dựng ứng dụng web quản lý công việc

Sprint 3	Hiển thị số lượng công việc theo trạng thái	5	Trung bình
Sprint 3	Tạo biểu đồ thống kê tiến độ (hoàn thành/chưa hoàn thành)	5	Cao
Sprint 3	Thiết lập nhắc nhở công việc sắp hết hạn (qua email)	5	Cao
Sprint 4	Thiết kế layout tổng thể (Header, Sidebar, Main content)	5	Trung bình
Sprint 4	Responsive cho điện thoại và máy tính bảng	5	Trung bình
Sprint 4	Cải thiện trải nghiệm người dùng (loading, thông báo, form validate)	5	Cao
Sprint 5	Viết GittHub Action test cho backend	5	Cao
Sprint 5	Kiểm thử chức năng frontend	3	Trung bình
Sprint 5	Triển khai backend lên Docker	5	Cao
Sprint 5	Triển khai frontend lên Docker	5	Cao
Sprint 5	Kiểm tra toàn bộ hệ thống	3	Trung bình

CHƯƠNG 6: KIỂM THỬ

6.1 Chiến lược kiểm thử và công cụ sử dụng

Áp dụng chiến lược kiểm thử nhiều lớp để đảm bảo chất lượng phần mềm. Các API được kiểm thử bằng Postman, đảm bảo các chức năng backend hoạt động đúng. Ngoài ra, nhóm tích hợp kiểm thử tự động vào quy trình CI thông qua GitHub Actions, giúp kiểm tra mã nguồn tự động mỗi khi có thay đổi. Việc kiểm thử được thực hiện định kỳ trong suốt quá trình phát triển để kịp thời phát hiện và sửa lỗi. Nhờ đó, phần mềm duy trì được tính ổn định và đáp ứng đúng yêu cầu đặt ra.

Kiểm thử API (Postman)

Mục tiêu: Đảm bảo các API RESTful hoạt động chính xác, phản hồi đúng dữ liệu và xử lý hợp lệ các trường hợp thành công cũng như lỗi.

Quy trình thực hiện:

- Sử dụng Postman để gửi các yêu cầu HTTP như GET, POST, PATCH, DELETE tới các endpoint của hệ thống.
- Kiểm tra mã phản hồi (status code), cấu trúc dữ liệu trả về, và các thông báo lỗi trong các tình huống đầu vào hợp lệ và không hợp lệ.

Các chức năng đã được kiểm thử:

- Xác thực người dùng: Đăng ký, Đăng nhập.
- Quản lý danh mục: Lấy danh sách danh mục, thêm, xóa danh mục,...
- Quản lý công việc theo danh mục: Thêm, chỉnh sửa, xóa công việc,...

Thông qua kiểm thử thủ công bằng Postman, nhóm đảm bảo các chức năng backend đều hoạt động ổn định trước khi tích hợp với frontend.

Kiểm thử tự động với GitHub Actions (CI)

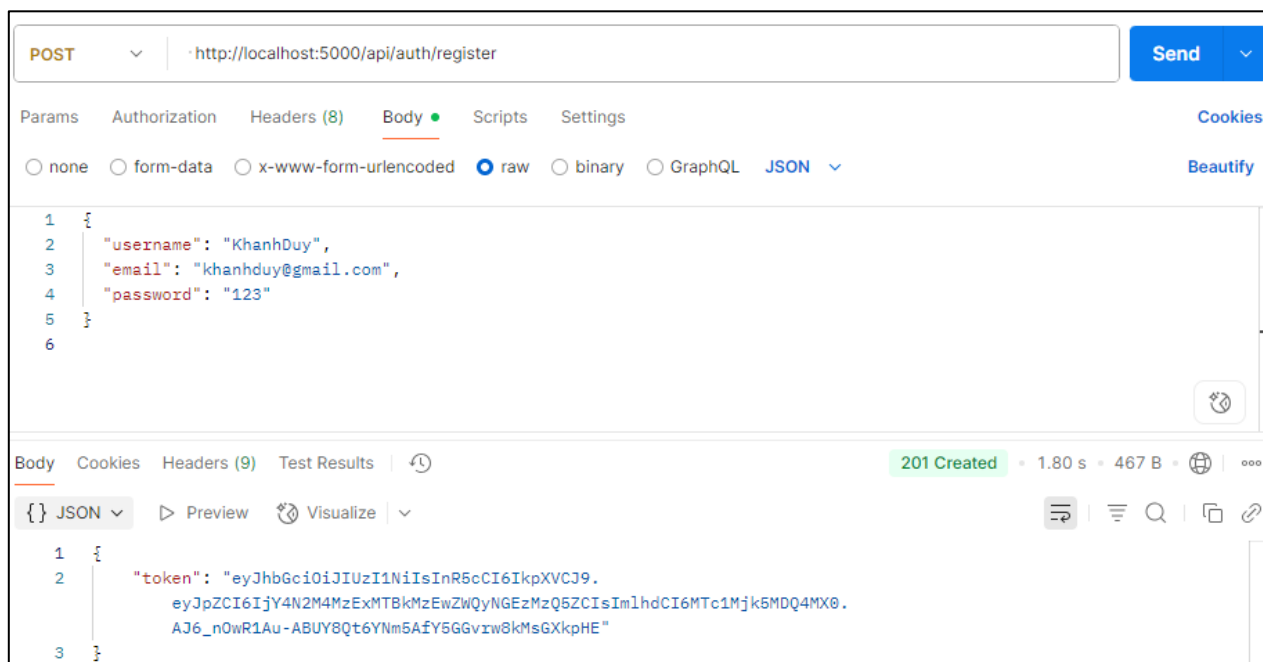
Mục tiêu: Tự động kiểm tra mã nguồn mỗi khi có thay đổi, giúp phát hiện lỗi sớm trong quy trình phát triển.

Khi có push hoặc pull request lên bất kì nhánh nào, GitHub Actions sẽ:

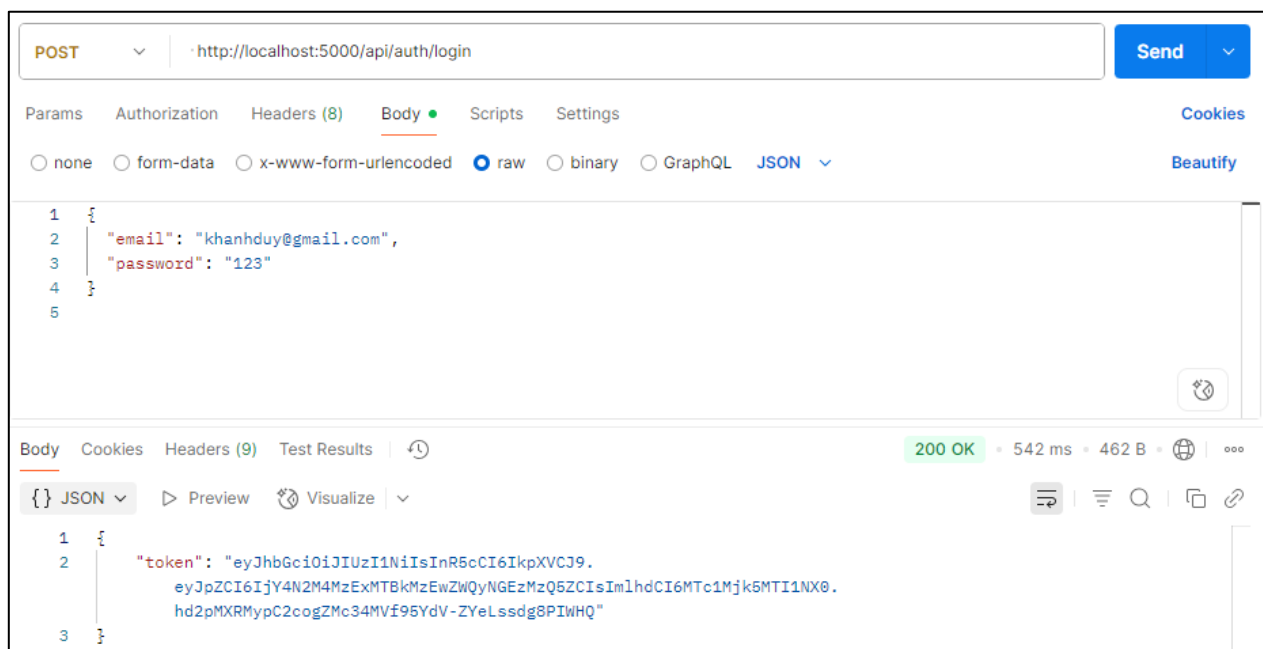
- Tự động cài dependencies.
- Chạy lệnh kiểm thử (npm test) cho cả frontend và backend.
- Thông báo lỗi nếu có lỗi test xảy ra.

6.2 Kết quả kiểm thử API với Postman

Các API liên quan xác thực:

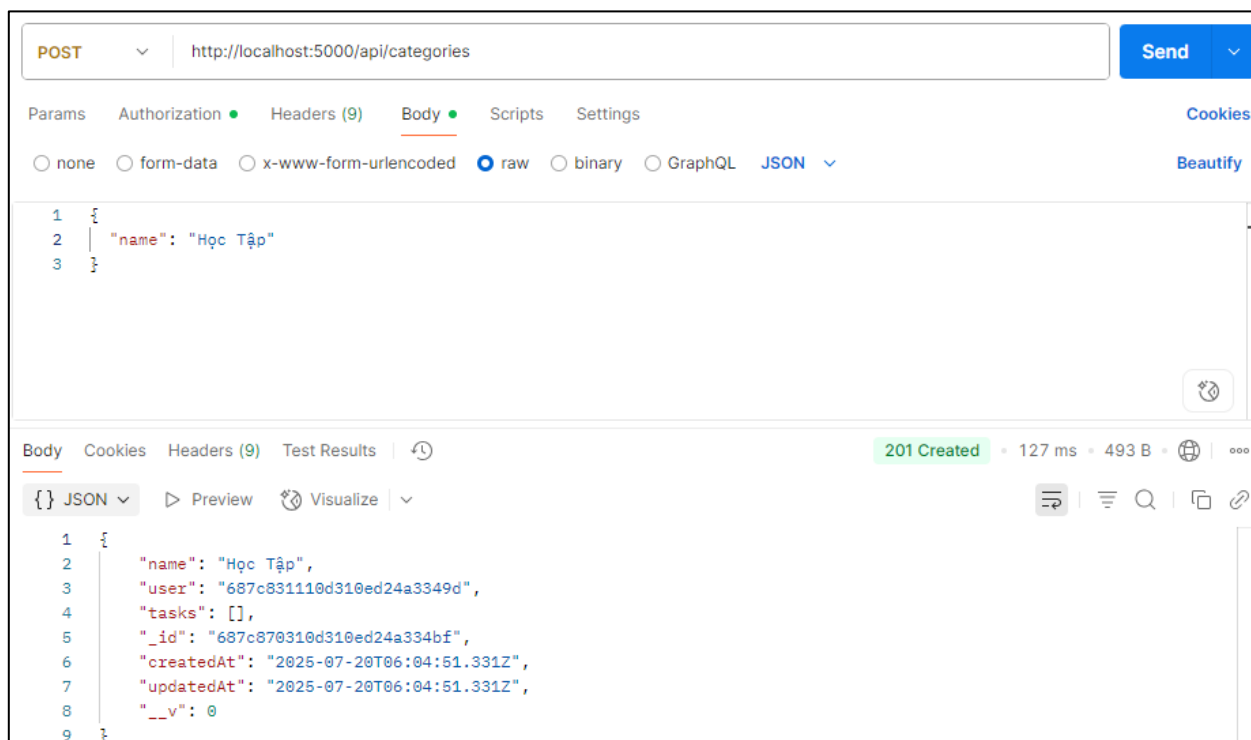


Hình 6.1 Thử nghiệm API đăng ký với Postman

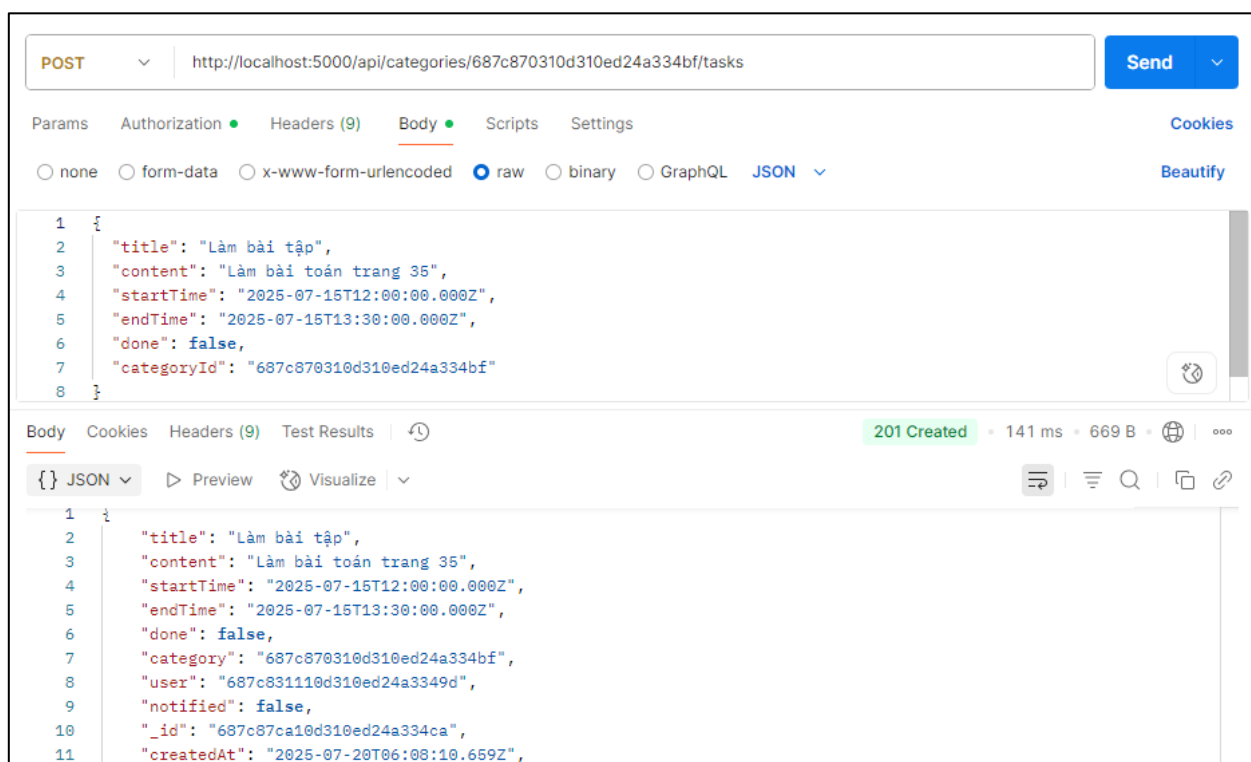


Hình 6.2 Thử nghiệm API đăng nhập với Postman

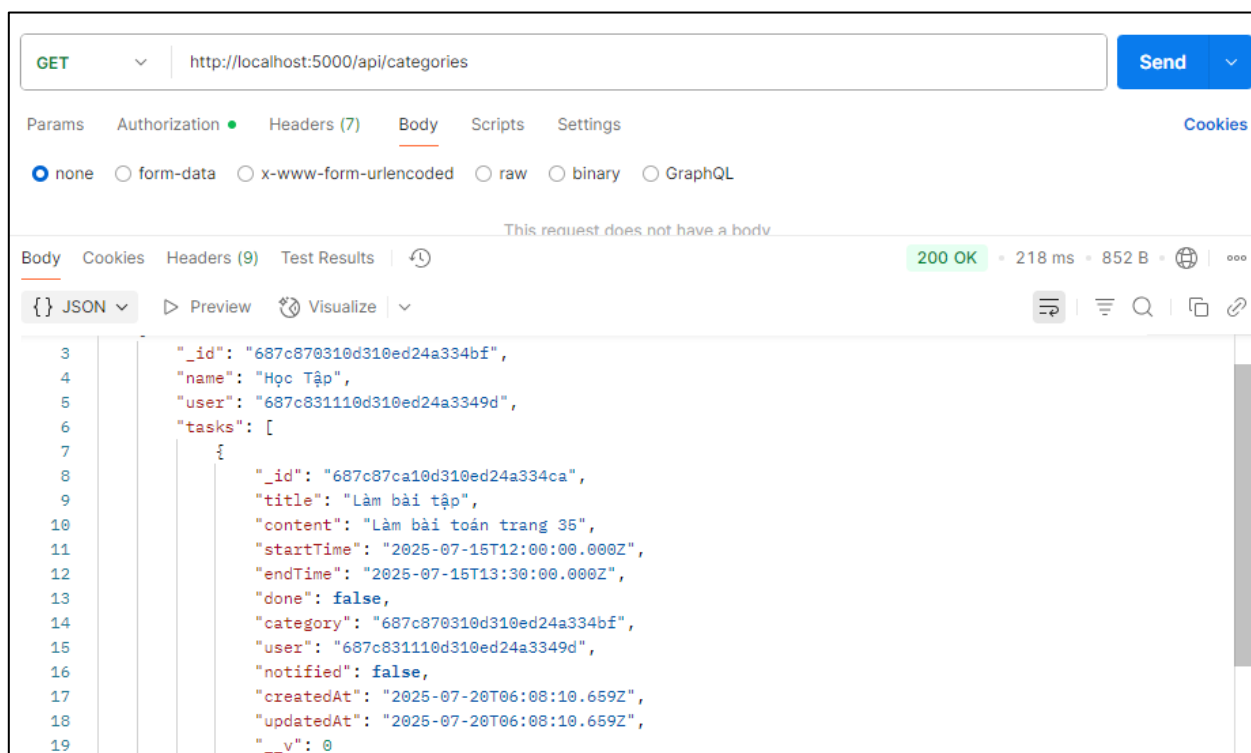
Các API liên quan đến Categories



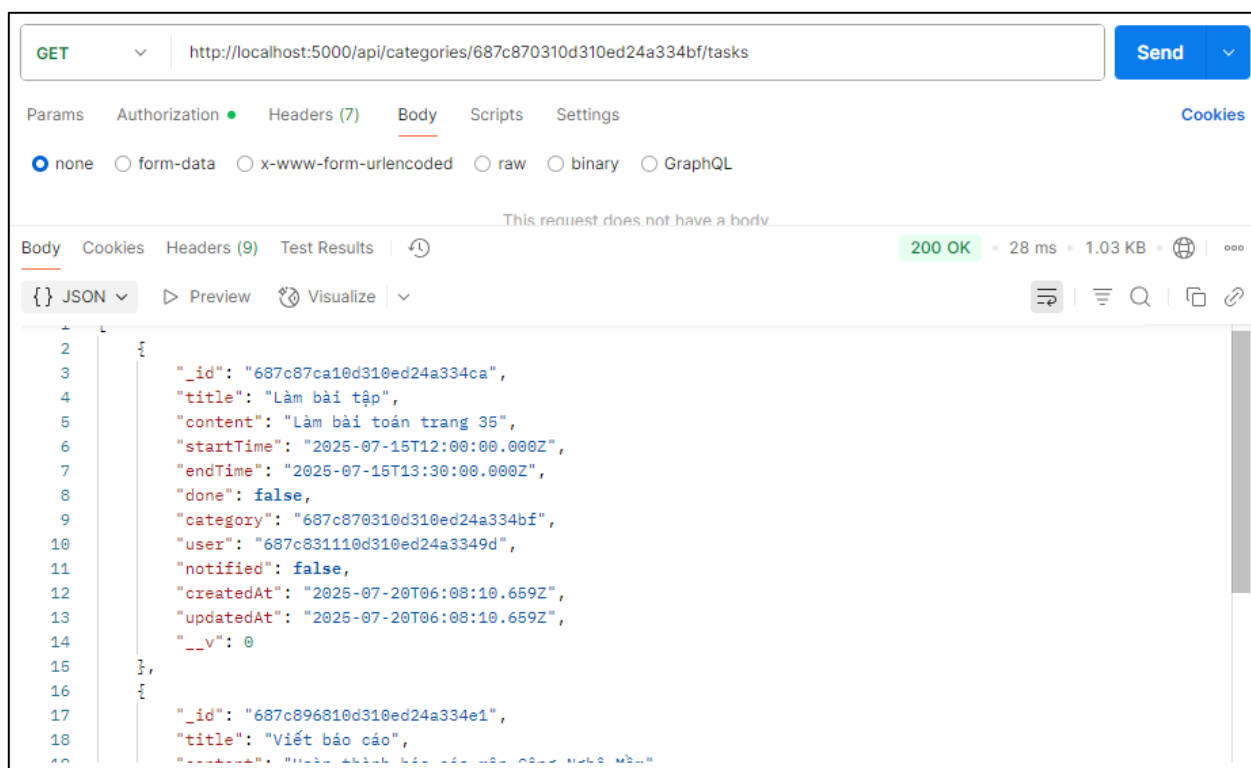
Hình 6.3 Thử nghiệm API thêm danh mục công việc với Postman



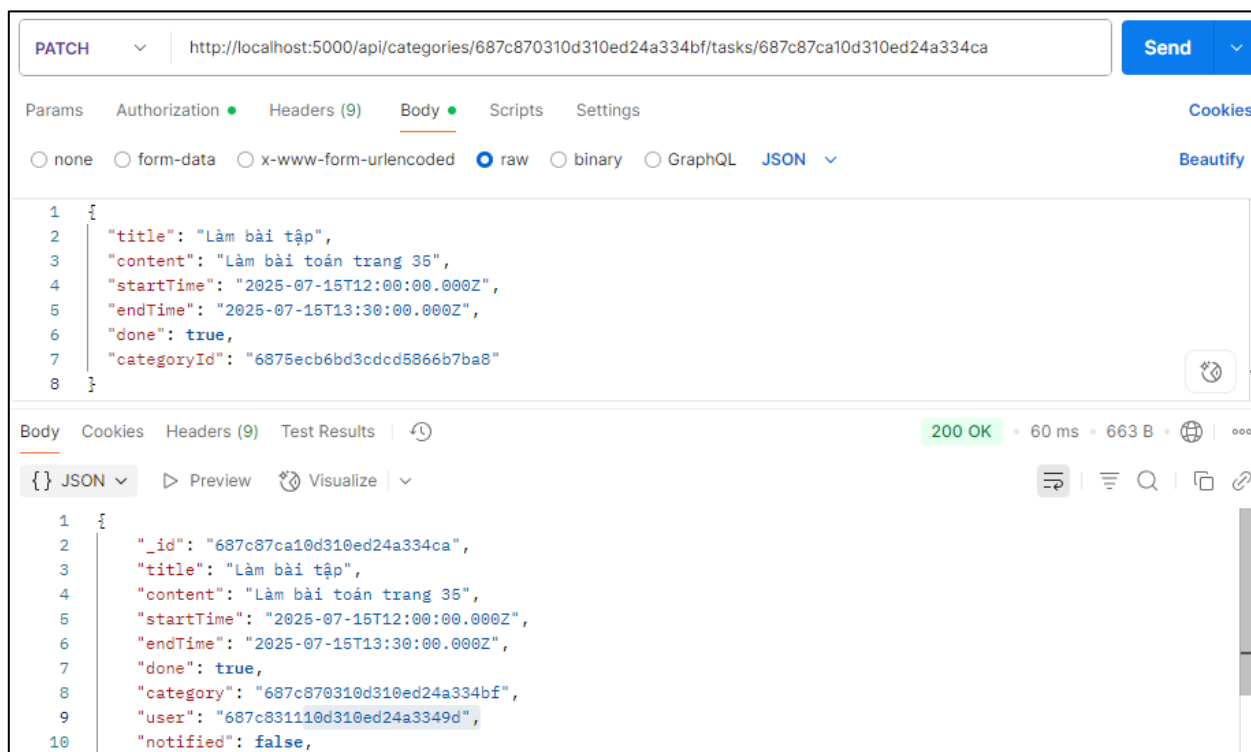
Hình 6.4 Thử nghiệm API thêm công việc vào danh mục với Postman



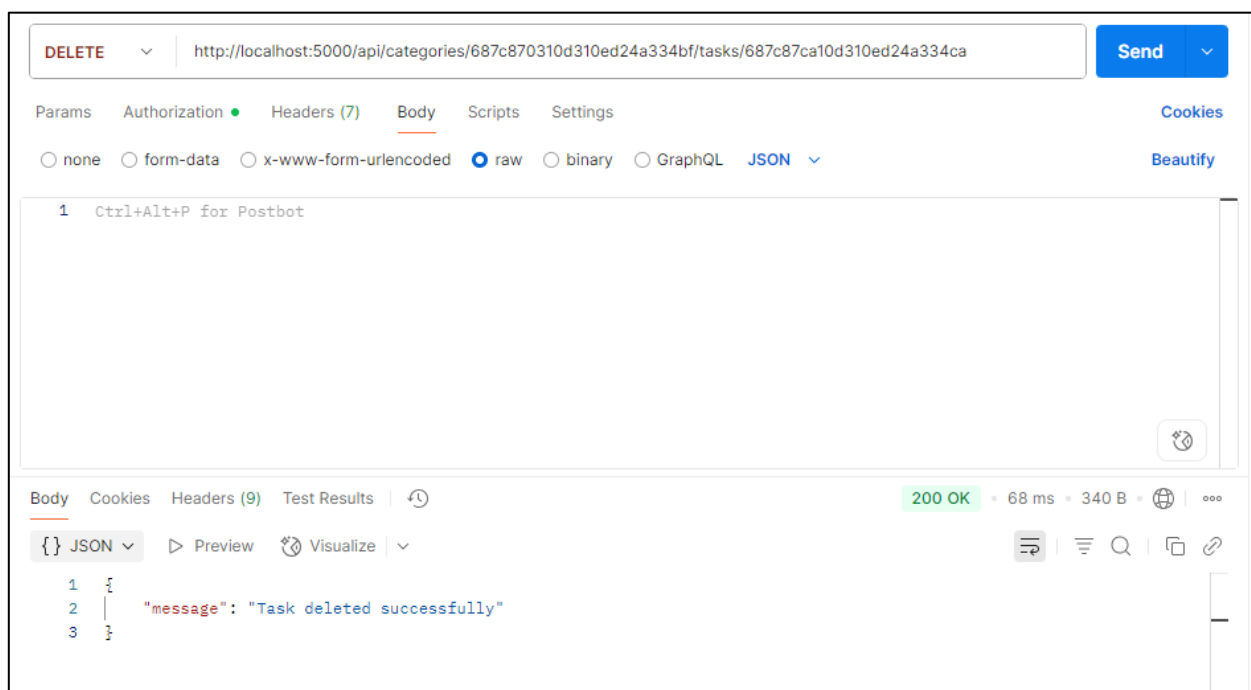
Hình 6.5 Thử nghiệm API lấy danh sách danh mục công việc với Postman



Hình 6.6 Thử nghiệm API lấy danh sách công việc với Postman



Hình 6.7 Thử nghiệm API cập nhật công việc với Postman



Hình 6.8 Thử nghiệm API xóa công việc với Postman

CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN

7.1 Khó khăn

Một trong những khó khăn lớn nhất là việc lựa chọn công nghệ phù hợp và làm chủ chúng trong thời gian ngắn. Đề tài yêu cầu áp dụng nhiều công cụ và công nghệ hiện đại như Docker, Swagger, GitHub Actions, Postman, kiến trúc RESTful API,... Trong khi đó, phần lớn thành viên trong nhóm chưa từng sử dụng những công cụ này trước đây. Vì vậy, nhóm phải tự học, tra cứu tài liệu, xem hướng dẫn từ nhiều nguồn khác nhau, vừa học vừa áp dụng vào dự án thực tế. Điều này gây áp lực khá lớn về mặt thời gian và dễ dẫn đến sai sót kỹ thuật ở các giai đoạn đầu.

Ngoài ra, nhóm cũng gặp thách thức trong việc phối hợp làm việc nhóm và quản lý tiến độ. Mỗi thành viên đều có lịch học và công việc cá nhân khác nhau, nên không dễ để sắp xếp thời gian làm việc chung. Dù đã sử dụng Jira để phân chia công việc và quản lý sprint, nhưng vẫn có thời điểm tiến độ bị chậm hoặc trễ do không đồng bộ được nhịp làm việc giữa các thành viên.

7.2 Bài học rút ra và đề xuất cải thiện trong tương lai

Trong quá trình thực hiện đồ án không chỉ giúp chúng em củng cố kiến thức đã học, mà còn mang lại nhiều trải nghiệm thực tế trong việc triển khai một hệ thống phần mềm hoàn chỉnh. Nhóm hiểu rõ hơn về tầm quan trọng của việc phân tích yêu cầu đúng ngay từ đầu, lên kế hoạch phát triển hợp lý và giữ vững tiến độ làm việc theo từng giai đoạn. Việc áp dụng các công nghệ mới tuy ban đầu còn bỡ ngỡ, nhưng đã giúp các thành viên rèn luyện được kỹ năng tự học, tự xử lý vấn đề và phối hợp nhóm một cách linh hoạt hơn.

Bên cạnh đó, việc xây dựng kiến trúc frontend – backend tách biệt cũng là một bước tiến quan trọng giúp nhóm hình dung rõ hơn về cách các hệ thống thực tế vận hành và giao tiếp thông qua API. Thông qua quá trình phát triển, nhóm cũng rút ra được kinh nghiệm trong việc kiểm thử chức năng, xử lý lỗi và cải thiện giao diện sao cho thân thiện với người dùng nhất.

Trong tương lai, để hoàn thiện hệ thống hơn nữa, nhóm dự định cải thiện hiệu suất của backend khi xử lý dữ liệu lớn, đặc biệt là trong các tình huống có nhiều người dùng

hoặc danh sách công việc phức tạp. Đồng thời, hệ thống sẽ được bổ sung các biểu đồ thống kê trực quan về tiến độ và hiệu quả làm việc của người dùng, giúp họ có cái nhìn tổng quan hơn về thói quen và năng suất cá nhân. Giao diện cũng cần được cân chỉnh để tương thích tối ưu hơn với các thiết bị di động, và nếu có thời gian nhóm sẽ nghiên cứu thêm việc lưu trữ liệu ngoại tuyến và đồng bộ sau khi kết nối mạng trở lại.

TÀI LIỆU THAM KHẢO

- [1] T. Blog, “Jira là gì - Sử dụng JIRA để tối ưu quy trình như nào?,” TopDev. Accessed: Jul. 21, 2025. [Online]. Available: <https://topdev.vn/blog/jira-la-gi/>

PHỤ LỤC

1. Yêu cầu hệ thống

Để có thể chạy được hệ thống quản lý công việc, máy tính cần đảm bảo các yêu cầu sau:

- Đã cài đặt Node.js phiên bản 14 trở lên.
- Có trình quản lý gói npm (đi kèm với Node.js).
- Đã cài đặt MongoDB (nếu chạy cơ sở dữ liệu cục bộ).
- Đã cài đặt VS code và Docker.
- Trình duyệt web hiện đại như Google Chrome hoặc Firefox.
- Công cụ Git để clone mã nguồn từ GitHub.

2. Clone mã nguồn từ GitHub

Người dùng cần mở terminal hoặc Git Bash và thực hiện lệnh:

➔ git clone https://github.com/HuynhKhanhDuy2912/Task_Management.git

➔ cd Task_Management

3. Cài đặt và cấu hình backend

Truy cập vào thư mục backend và tiến hành cài đặt các thư viện cần thiết:

➔ cd backend

➔ npm install

4. Khởi chạy ứng dụng bằng Docker

Người dùng có thể sử dụng Docker để chạy toàn bộ hệ thống. Trong thư mục gốc của dự án, sử dụng tệp docker-compose.yml có sẵn và chạy lệnh:

➔ docker-compose up --build

Truy cập và sử dụng ứng dụng

Backend chạy tại địa chỉ: <http://localhost:5000>

Frontend (giao diện người dùng) chạy tại địa chỉ: <http://localhost:3000>

Tại giao diện người dùng, người dùng có thể thực hiện các chức năng:

Đăng ký/ Đăng nhập tài khoản. Tạo, xóa danh mục công việc. Thêm, sửa, xóa công việc và nhận thông báo khi công việc đến hạn. Xem biểu đồ thống kê công việc theo tháng, năm.

Link GitHub: https://github.com/HuynhKhanhDuy2912/Task_Management.git