**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

*Trà Vinh, ngày ….. tháng …… năm ……*

**Giáo viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

**NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG**

*Trà Vinh, ngày ….. tháng …… năm ……*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

**LỜI CẢM ƠN**

Chúng tôi xin gửi lời cảm ơn sâu sắc đến trường đại học Trà Vinh đã tạo điều kiện cho chúng tôi thực hiện dự án này, những người đã đưa ra sự hỗ trợ nhỏ nhất cho chúng tôi dù là một lời gợi ý. Nhờ mọi người mà dự án đã thành công tốt đẹp.

Chúng tôi biết ơn tất cả mọi người đã dành thời gian, công sức và kiến thức của mình để hỗ trợ chúng tôi. Sự giúp đỡ của các bạn là nguồn động lực giúp chúng tôi phát triển bản thân. Các ý kiến đóng góp và sự hợp tác của mọi người đã tạo nên một môi trường làm việc tích cực.

Không có gì quý báu hơn những người đồng đội, những người đã góp ý và những người đã hỗ trợ tôi như mọi người, mọi người đã dìu dắt chúng tôi trong dự án đầu tay của mình. Nhờ có những người bạn đã giúp đỡ mà dự án của chúng tôi mới có thể phát triển đến mức này. Chúng tôi hạnh phúc và tự hào về mối quan hệ mà chúng ta đã xây dựng cùng nhau và tin rằng những thành công trong tương lai sẽ tiếp tục phát triển từ sự đồng lòng, cống hiến và sự giúp đỡ của mọi người đối với chúng tôi.

Một lần nữa, xin chân thành cảm ơn mọi người đã giúp đỡ và chúng tôi mong sẽ tiếp tục nhận được sự ủng hộ của mọi người trong những dự án sắp tới.

Trân trọng.

**MỤC LỤC**

[CHƯƠNG 1: GIỚI THIỆU 8](#_Toc169117676)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 10](#_Toc169117677)

[2.1 Lý thuyết về Jira, Agile và Scrum 10](#_Toc169117678)

[1. Khái niệm về jira: . 10](#_Toc169117679)

[2. Tính năng chính của Jira là gì? 10](#_Toc169117680)

[3. Vì sao nên dùng Jira? 10](#_Toc169117681)

[4. Các thành phần cơ bản của Jira 11](#_Toc169117682)

[5. Ưu, nhược điểm của phần mềm Jira 12](#_Toc169117683)

[6. Các thuật ngữ cần biết khi sử dụng Jira 13](#_Toc169117684)

[7. Báo cáo 15](#_Toc169117685)

[2.2 Giới thiệu về HTML (HyperText Markup Language) 18](#_Toc169117686)

[2.3 Giới thiệu ngôn ngữ định dạng CSS 19](#_Toc169117687)

[2.4 Giới thiệu về ngôn ngữ lập trình Javascript 21](#_Toc169117688)

[2.5 Kiến thức cơ bản về Nodejs, Reactjs, Mysql 22](#_Toc169117689)

[2.5.1 Nodejs: Môi trường, nền tảng thực thi đa chức năng 22](#_Toc169117690)

[2.5.2 Reactjs: Framework thiết kế giao diện cực kì mạnh mẽ 24](#_Toc169117691)

[2.5.3 MySQL: Một hệ quản trị cơ sở dữ liệu quan hệ 25](#_Toc169117692)

[2.6 Lý luận thiết kế và phát triển Website 26](#_Toc169117693)

[2.6.1 Quy trình phát triển website 26](#_Toc169117694)

[2.6.2 Tích hợp giữa Frontend và Backend 26](#_Toc169117695)

[2.7 Giả thiết khoa học và phương pháp nghiên cứu đồ án cơ sở ngành 27](#_Toc169117696)

[2.7.1 Giả thiết khoa học: hiệu suất và sự linh hoạt 27](#_Toc169117697)

[2.7.2 Phương pháp nghiên cứu: phát triển liên tục và tiến hành kiểm thử 27](#_Toc169117698)

[CHƯƠNG 3: XÁC ĐỊNH NHU CẦU 27](#_Toc169117699)

[CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM 29](#_Toc169117700)

[CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH 32](#_Toc169117701)

[PHỤ LỤC 45](#_Toc169117702)

**DANH MỤC HÌNH ẢNH – BẢNG BIỂU**

[Hình 1 Bảng phân công công việc: 7](#_Toc169118968)

[Hình 2 Giao diện scrum board trong Jira 14](#_Toc169118969)

[Hình 3 HTML 18](#_Toc169118970)

[Hình 4 Đây là hình ảnh web dùng ngôn ngữ html 19](#_Toc169118971)

[Hình 5 CSS 20](#_Toc169118972)

[Hình 6 Hình kết hợp html css 21](#_Toc169118973)

[Hình 7 javascript 21](#_Toc169118974)

[Hình 8 Hình ảnh về Nodjes 23](#_Toc169118975)

[Hình 9 Reacjs 24](#_Toc169118976)

[Hình 10 Hình ảnh về hệ quản trị cơ sở dữ liệu MySQL 26](#_Toc169118977)

[Hình 11 Nhiệm vụ tuần 1 30](#_Toc169118978)

[Hình 12 Nhiệm vụ tuần 2 30](#_Toc169118979)

[Hình 13 Nhiệm vụ tuần 3 30](#_Toc169118980)

[Hình 14 Nhiệm vụ tuần 4 31](#_Toc169118981)

[Hình 15 Nhiệm vụ tuần 5 31](#_Toc169118982)

[Hình 16 Nhiệm vụ tuần 6 32](#_Toc169118983)

[Hình 17 Nhiệm vụ tuần 7 32](#_Toc169118984)

[Hình 18 Commit github 42](#_Toc169118985)

[Hình 19 Commit github frontend 42](#_Toc169118986)

Hình 1 Bảng phân công công việc:

|  |  |
| --- | --- |
| **Thành viên thực hiện** | **Công việc thực hiện** |
| Huỳnh Nhựt Huy – 110121033 | Frontend, Word, thuyết trình. Phân bố nhiệm vụ dựa trên quyền của SCRUM MASTER. |
| Nguyễn Hoàng Nhân -110121071 | Backend, API, fix bug, PowerPoint. Demo website. DEV. |
| Lâm Sơn Tùng - 110121155 | PowerPoint, thuyết trình, sửa lỗi phát sinh. DEV. |

# CHƯƠNG 1: GIỚI THIỆU

**Tên dự án: Website Bán Điện Thoại**

**Lý do chọn tài**:

Lý do chọn đề tài "Xây Dựng Website Bán Điện Thoại" vì nó phản ánh sự mong muốn của tôi khi cố gắng trở thành một lập trình viên full-stack đầy đủ kỹ năng cả về phía client và server. Trong môi trường đầy tính cạnh tranh hiện nay thì tôi cần phải tự mình phát triển những kĩ năng cần có của một lập trình viên fullstack và việc chọn một dự án website thương mại điện tử sẽ cung cấp cơ hội tuyệt vời để khám phá và thách thức bản thân trong việc xây dựng một ứng dụng hoàn chỉnh từ đầu đến cuối.

**Mục đích nghiên cứu:**

Mục đích của dự án không chỉ là xây dựng một trang web bán điện thoại hiệu quả mà còn là việc phát triển khả năng full-stack bằng cách làm việc với Node.js và React.js. Điều này sẽ giúp chúng tôi hiểu rõ hơn về quá trình tích hợp giữa frontend và backend, đồng thời nắm vững quản lý dữ liệu và tương tác người dùng.

**Công nghệ sử dụng**:

**- Frontend:** HTML, CSS, JavaScript, ReactJS.

**- Backend:** Node.js, Express.js, MySQL.

**- Công cụ quản lý mã nguồn:** GitHub.

**- Công cụ quản lý dự án:** Jira.

**Quy trình phát triển:**

**Xác định yêu cầu**: Dùng Jira để tạo và quản lý các yêu cầu. Phân loại và ưu tiên theo mức độ quan trọng.

**Thiết kế giao diện**: Dùng công cụ thiết kế để tạo mockup và wireframe. Lưu trữ tài liệu thiết kế trên GitHub.

**Phát triển**: Sử dụng GitHub để quản lý mã nguồn và kiểm soát phiên bản. Phân chia công việc thành các nhánh để làm việc song song.

**Kiểm thử**: Theo dõi quá trình kiểm thử và ghi nhận báo cáo lỗi trên Jira. Quản lý các issue trên GitHub.

**Triển khai**: Dùng công cụ triển khai liên tục (CI/CD) kết hợp với GitHub Actions để tự động hóa quy trình triển khai.

**Quản lý dự án:**

* Sử dụng Jira để quản lý backlog, lên kế hoạch sprint và theo dõi tiến độ công việc.
* GitHub sẽ được sử dụng để quản lý mã nguồn, kiểm soát phiên bản và theo dõi các issue và pull requests.

**Bảo trì và hỗ trợ:**

- Sử dụng Jira để theo dõi và ghi nhận các yêu cầu bảo trì và hỗ trợ từ người dùng.

- GitHub sẽ được sử dụng để quản lý mã nguồn và triển khai các bản vá.

**Nhóm phát triển:**

- Số lượng thành viên: 3 người.

- Vai trò: 1 Scrum Master, 2 Developers.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## Lý thuyết về Jira, Agile và Scrum

1. **Khái niệm về jira:**

Jira là một ứng dụng theo dõi và quản lý lỗi, vấn đề và dự án, được phát triển để làm quy trình này trở nên dễ dàng hơn cho mọi tổ chức. JIRA đã được thiết kế với trọng tâm vào kết quả công việc, có thể sử dụng ngay và linh hoạt khi sử dụng.

1. **Tính năng chính của Jira là gì?**

Quản lý, theo dõi tiến độ của dự án.

Quản lý lỗi, tính năng, công việc, những cải tiến hoặc bất kỳ vấn đề gì.

Tìm kiếm nhanh chóng với bộ lọc JIRA Query Language.

Xây dựng quy trình làm việc tương thích với yêu cầu của từng dự án.

Cung cấp nhiều loại báo cáo thống kê với rất nhiều loại biểu đồ khác nhau phù hợp với nhiều loại hình dự án, nhiều đối tượng người dùng.

Dễ dàng tích hợp với các hệ thống khác (như Email, Excel, RSS…).

Có thể chạy trên hầu hết các nền tảng phần cứng, hệ điều hành và cơ sở dữ liệu.

1. **Vì sao nên dùng Jira?**

Dễ dàng lựa chọn nhiều loại dự án với các mục đích khác nhau.

Dễ dàng lên kế hoạch, phân chia nhiệm vụ cho các thành viên.

Tạo ra quy trình làm việc hiệu quả.

Dễ dàng quản lý nhiều dự án phần mềm cùng lúc.

Bộ lọc JIRA Query Language giúp tìm kiếm nhanh chóng.

Dễ dàng đưa ra các báo cáo phân tích tình hình dự án một cách nhanh chóng, chính xác và hiệu quả.

Hơn 950 add-on mang đến các tính năng nâng cao

Giao diện thân thiện, rõ ràng, dễ thao tác, thích hợp với mọi đối tượng người sử dụng.

1. **Các thành phần cơ bản của Jira**

**Roles:** Xác lập các role của dự án, Mục này xác nhận ai tham gia vào dự án, những người add vào role thì mới có thể tạo Resource Allocation và project team sau này. Nhiều người có thể vào 1 role.

**Issue:** là các tasks, các bugs, các features hay bất kỳ các type khác của project work.

**Project**: Chức năng này dùng để phân quyền approve worklog cho thành viên của dự án. Ai là team lead của group nào thì sẽ được approve worklog cho member của group đó. Project management được quyền approve cho toàn bộ thành viên dự án.

**Component** là sản phẩm của dự án. Ở đây sẽ nhập tất cả sản phẩm của dự án lấy từ file kế hoạch doanh số. Nếu dự án làm theo Scrum thì sẽ là Product của Sprint tương ứng.

**Workflow:** Là một quản trị JIRA, bạn có thể cấu hình gây nên quy trình làmviệc, điều kiện, xác nhận, và sau chức năng. Trang này sẽ cung cấp một cái nhìn tổng quan và các bước cơ bản cho từng phần của công việc của bạn.

**Priority:** Là mức độ ưu tiên của một defect. Có 4 mức, chọn theo datalist

**Status:** Đại diện cho các vị trí của vấn đề trong workflow.

**Resolution.**

1. **Ưu, nhược điểm của phần mềm Jira**

**Ưu điểm**

* Jira có chức năng phân quyền cực kỳ chi tiết, không chỉ phân quyền trong dự án chung, mà còn phân quyền đối với từng nhiệm vụ, giúp team công nghệ bảo vệ thông tin độc quyền của mình.
* Dễ dàng tích hợp với các hệ thống khác (như Email, Excel, RSS,…)
* Hệ thống module và bộ công cụ phát triển bổ trợ cho phép tùy biến, mở rộng và tích hợp Jira vào trong hệ thống hiện tại.
* Jira được phát triển sử dụng chuẩn HTML và được thử nghiệm với tất cả các trình duyệt phổ biến hiện nay.
* Có thể chạy trên hầu hết các nền tảng phần cứng, hệ điều hành và cơ sở dữ liệu.
* Mỗi màn hình trong Jira có một phiên bản có thể in đảm bảo việc luân chuyển bản cứng một cách dễ dàng.
* Có thể tích hợp trực tiếp với code trên môi trường phát triển, là một công cụ hoàn toàn phù hợp với các [developer](https://topdev.vn/blog/duong-doi-cua-lap-trinh-vien/).

**Nhược điểm**

* Chi phí cao, sau 7 ngày dùng thử thì doanh nghiệp càng có quy mô lớn thì càng tốn nhiều chi phí: $10 mỗi tháng dành cho tối đa 10 tài khoản; từ 11-100 tài khoản là $7/tài khoản/tháng.
* Tốn nhiều thời gian và công sức để setup nên chỉ phát huy tối ưu hiệu quả với dự án lớn, không phù hợp với dự án vừa và nhỏ (dưới 3 tháng).
* Ngôn ngữ tiếng Anh với nhiều thuật ngữ khó sử dụng.
* Quy trình làm việc phức tạp đòi hỏi phải tìm hiểu kỹ lưỡng.

Tóm lại, Jira là một phần mềm quản lý dự án khá đặc thù, đây sẽ là lựa chọn hoàn hảo cho các team làm việc theo phương pháp Agile, đặc biệt là các team công nghệ hoặc phát triển phần mềm.

1. **Các thuật ngữ cần biết khi sử dụng Jira**

Jira là phần mềm với khá nhiều thuật ngữ công nghệ:

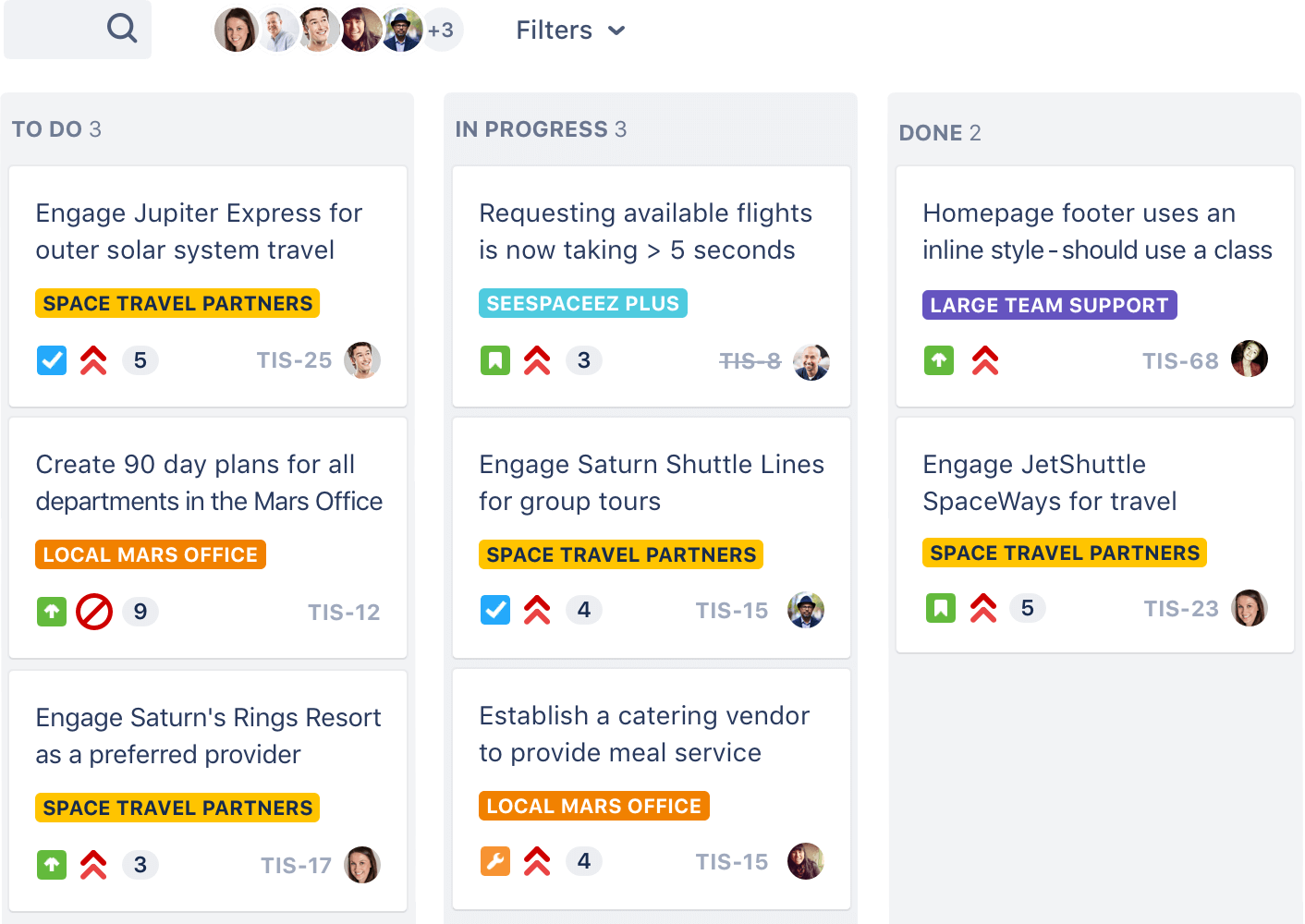
* **Sprint**: Một vòng lặp ngắn hạn (lý tưởng là 2-4 tuần) mà đội phát triển thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai để cho ra các phần nhỏ của sản phẩm.

**Active spint:** dùng để quản lý các giai đoạn của một phần mềm (vd: giai đoạn dầu tiên sẽ thiết kế UI, sau đó thiết kế DataBase (CSDL) j j đó) sau đó định thời gian để bắt đầu các cái spint đấy, các spint đang được kích hoạt (phát triển) sẽ nằm ở mục này, TO DO( các task đang được active) , IN PROCESS (những người join được vào dự án thì sẽ thực hiện kéo các task được giao vào ô này để sử lý đến khi hoàn thành công việc), DONE (các task đã được hoàn thành sẽ nằm ở ô này).

* **Backlog**: Danh sách tập hợp các user stories, bugs và tính năng cho một sản phẩm hoặc sprint.

Giao diện phía bên trái: gồm có backlog dùng để chứa các user story, hoặc gì gì đó (không rõ lắm) có hai hộp thoại: đầu tiên là EPIC (nơi chưa user story trong một user story sẽ có nhiều cái task khác nữa ( có thể chia nhỏ thành sub task)) trong một epic sẽ có nhiều task có hiện thị các task đã hoàn thành, các issue đang, và các task chưa ước tính thơi gian hoàn thành là khi nào. thứ hai là mục Version, (ví dụ khi ký một hợp đồng nào đó thì sẽc hẹn để demo trước một số chức năng và sau một khoản thời gian sẽ demo tiếp một số chức năng khác, có thể coi đây là một phương thức dùng để quản lý các version của phần mềm)

* **Scrum**: Một phương pháp Agile, nơi sản phẩm được xây dựng theo các lần lặp đi lặp lại trong một sprint.
* **Scrum of Scrums**: Một kỹ thuật để mở rộng quy mô Scrum, các dự án đa đội – theo truyền thống gọi là program management.
* **Board**: Công cụ dùng để hiển thị hoạt động công việc trong một quy trình làm việc cụ thể. Nó có thể thay đổi thích ứng với các phương pháp Agile khác nhau (ví dụ, một bảng Scrum sẽ hiển thị các công việc được di chuyển từ product backlog đến sprint backlog, trong khi đó một bảng Kanban thường có một quy trình làm việc ba bước: To do, In Progress, và Done).



Hình 2 Giao diện scrum board trong Jira

* **Burndown Chart**:Hiển thị số lượng ước tính và thực tế cho tổng số công việc phải hoàn thành trong một sprint.
* **Daily stand-up**: Là một cuộc họp nhỏ 15 phút trước khi bắt đầu ngày làm việc, giúp mọi thành viên nắm bắt toàn bộ công việc của ngày hôm qua.
* **Epic**: Đại điện cho một user story lớn và cần phải được chia thành các story nhỏ. Người dùng có thể phải chạy nhiều sprint để hoàn thành một epic.
* **Issue**: Một đơn vị công việc (task, bug, story, epic) trong Jira, hoạt động trong một quy trình từ khởi tạo đến khi hoàn thành.
* **Swimlane**: Phân loại các công việc để xem xét công việc nào nên tiến hành trước.
* **Velocity**: Đo lường khối lượng công việc mà một đội có thể xử lý trong một thời hạn nhất định.
* **Cumulative Flow Diagram (CFD)**: Một biểu đồ thể hiện các trạng thái khác nhau của các mục công việc trong một khoảng thời gian cụ thể. Cột ngang x trong CFD định nghĩa là thời gian, và cột dọc y là mục công việc (issue). Mỗi vùng màu của biểu đồ tương đương với trạng thái luồng công việc (ví dụ một cột trong bảng).
* **Iteration**: Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại.
* **Wallboard**: Một bảng (viết tay hoặc điện tử) lớn được đặt tại vị trí dễ thấy thể hiện dữ liệu quan trọng về hoạt động của đội development.

1. **Báo cáo**

Một trong những tính năng nổi trội của phần mềm Jira chính là việc cung cấp rất nhiều loại báo cáo khác nhau. Giúp những người quản lý dự án có cái nhìn tổng quan và chi tiết vừa nhanh vừa hiệu quả. Một số loại báo cáo quan trọng nhất là:

* Thống kê số lượng issue theo thời gian.
* Hiển thị số lượng issue đã được tạo và số lượng issue đã được giải quyết trong một khoảng thời gian nhất định.
* Hiển thị báo cáo thống kê dưới dạng biểu đồ hình tròn dựa trên tiêu chí thống kê do người dùng lựa chọn.
* Báo cáo số lượng issue được tạo trong một khoảng thời gian nhất định do người dùng lựa chọn.

**KHÁI NIỆM VỀ AGILE:**

Agile là một phương pháp quản lý dự án phát triển phần mềm phổ biến trong ngành công nghiệp công nghệ thông tin. Nó tập trung vào sự linh hoạt, tương tác và thích ứng, giúp các nhóm phát triển phần mềm làm việc hiệu quả hơn trong môi trường thay đổi nhanh chóng. Dưới đây là một bối cảnh tổng quan về Agile:

* **Nguyên tắc Agile:**

**- Tập trung vào con người:** Đặt ưu tiên cho việc tương tác trực tiếp và sự hợp tác giữa các thành viên trong nhóm.

**- Phản hồi liên tục**: Công nhận và thích ứng với sự thay đổi để phản hồi nhanh chóng.

**- Làm việc phức tạp qua việc chia nhỏ**: Phân chia công việc thành các phần nhỏ hơn để dễ quản lý và triển khai.

**- Tính hợp tác:** Các nhóm tự tổ chức và làm việc cùng nhau để đạt được mục tiêu chung.

**- Tạo giá trị liên tục:** Cung cấp giá trị sản phẩm cho khách hàng một cách liên tục và định kỳ.

* **Phương pháp Agile phổ biến:**

**- Scrum:** Một framework phổ biến trong Agile, với các sprints (chu kỳ phát triển ngắn hạn), các cuộc họp hàng ngày và sự tập trung vào sản phẩm hoàn thiện.

**- Kanban:** Tập trung vào quản lý luồng công việc và tối ưu hóa hiệu suất bằng cách hạn chế công việc đồng thời.

**- Extreme Programming (XP):** Tập trung vào việc phát triển phần mềm chất lượng cao qua các phương pháp như kiểm thử liên tục và lập trình đôi.

* **Lợi ích của Agile:**

- Phản hồi nhanh chóng và linh hoạt trong việc thích ứng với thay đổi.

- Tăng cường tương tác và hợp tác giữa các thành viên trong nhóm.

- Giảm thiểu rủi ro bằng cách chia nhỏ công việc và kiểm tra thường xuyên.

- Tăng cường chất lượng sản phẩm và sự hài lòng của khách hàng thông qua việc cung cấp giá trị liên tục.

Agile không chỉ được áp dụng trong phát triển phần mềm mà còn được sử dụng trong nhiều lĩnh vực khác như quản lý dự án, tiếp thị và quản lý sản phẩm. Đối với mỗi dự án, việc triển khai Agile có thể được điều chỉnh và tùy chỉnh để phù hợp với yêu cầu cụ thể của dự án và tổ chức.

**KHÁI NIỆM VỀ SCRUM:**

Scrum là một framework phổ biến trong phát triển phần mềm theo phương pháp Agile, tập trung vào việc tự tổ chức, tập trung vào sản phẩm và cung cấp giá trị liên tục cho khách hàng. Dưới đây là một cái nhìn tổng quan về Scrum:

* **Các phần tử chính của Scrum:**

**- Product Owner:** Người đại diện cho khách hàng hoặc người sở hữu sản phẩm, chịu trách nhiệm xác định yêu cầu và ưu tiên của sản phẩm.

**- Scrum Master:** Người đảm bảo rằng nhóm Scrum hiểu và tuân thủ theo quy trình Scrum, giúp loại bỏ các rủi ro và cản trở trong quá trình phát triển.

**- Development Team:** Nhóm các chuyên gia thực hiện công việc để tạo ra sản phẩm, tự quản lý và tự tổ chức.

* **Các sự kiện trong Scrum:**

**- Sprint:** Chu kỳ phát triển ngắn hạn, thường kéo dài từ 1 đến 4 tuần. Mỗi Sprint tập trung vào việc tạo ra một sản phẩm hoàn chỉnh, có thể triển khai hoặc phát hành.

**- Sprint Planning:** Cuộc họp dành cho toàn bộ nhóm Scrum để lập kế hoạch và đặt mục tiêu cho Sprint tới.

**- Daily Scrum (Daily Stand-up):** Cuộc họp hàng ngày trong 15 phút, mục đích là cập nhật tiến độ và tạo ra một kế hoạch làm việc cho ngày hôm đó.

**- Sprint Review:** Cuộc họp cuối Sprint, mục đích là xem xét sản phẩm đã hoàn thành và thu thập phản hồi từ các bên liên quan.

**- Sprint Retrospective:** Cuộc họp sau Sprint Review, mục đích là xem xét và cải thiện quy trình làm việc của nhóm.

* **Các bản vẽ trong Scrum:**

**- Product Backlog:** Danh sách tất cả các yêu cầu và tính năng cần thiết cho sản phẩm, được sắp xếp theo ưu tiên và có thể thay đổi.

**- Sprint Backlog:** Danh sách các công việc cụ thể cần hoàn thành trong một Sprint, được lựa chọn từ Product Backlog.

**- Burn-down Chart:** Biểu đồ theo dõi tiến độ của Sprint hoặc Product, hiển thị lượng công việc còn lại theo thời gian.

Scrum không chỉ áp dụng cho phát triển phần mềm mà còn được sử dụng trong nhiều lĩnh vực khác như quản lý dự án, tiếp thị và quản lý sản phẩm. Scrum cung cấp một phương pháp linh hoạt và hiệu quả để làm việc trong môi trường thay đổi và đáp ứng nhanh chóng.

## Giới thiệu về HTML (HyperText Markup Language)

HTML, viết tắt của "HyperText Markup Language" (Ngôn ngữ Đánh dấu Siêu văn bản), là một ngôn ngữ đánh dấu sử dụng để xây dựng và cấu trúc trang web. Nó là thành phần chính của mô hình web và được sử dụng để tạo ra các trang web và ứng dụng web.

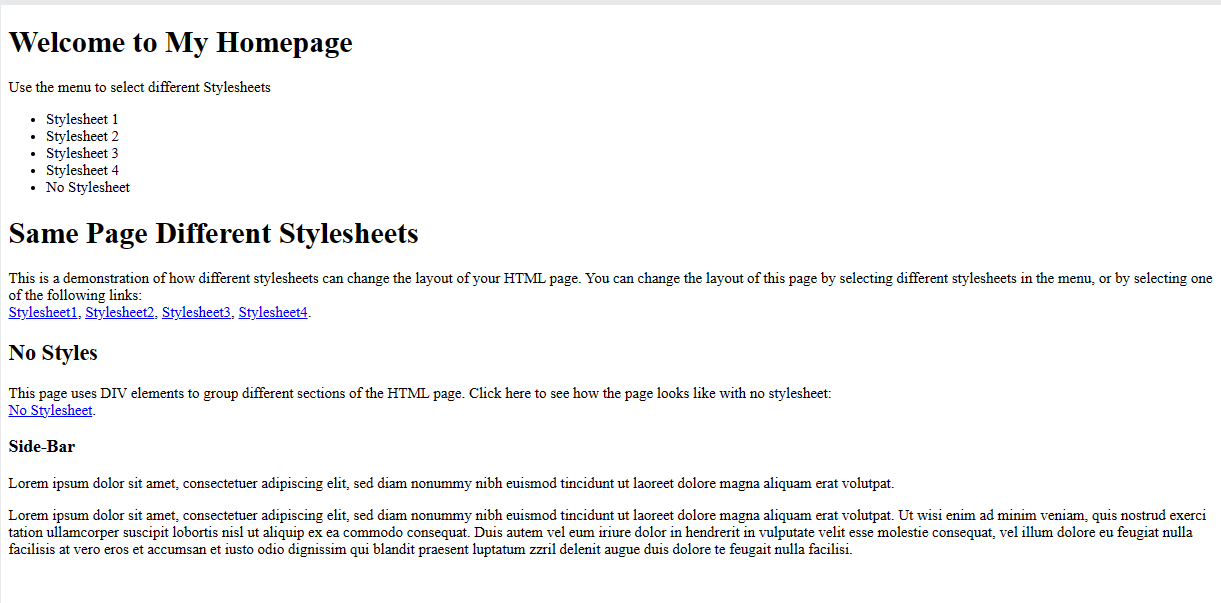


Hình 3 HTML

HTML mô tả cấu trúc của các trang web bằng cách sử dụng các thẻ và các phần tử được nhúng trong văn bản để xác định và hiển thị nội dung.

Dưới đây là một số điểm quan trọng về HTML: Thẻ và Thuộc tính: HTML sử dụng các thẻ để xác định các phần khác nhau của một trang web. Mỗi thẻ thường đi kèm với các thuộc tính để định cấu hình và trang trí nó.

Ví dụ: <p> là thẻ mở cho đoạn văn bản, </p> là thẻ đóng. Cấu trúc Cơ bản: Một trang HTML thông thường bắt đầu bằng thẻ <!DOCTYPE html> để xác định phiên bản HTML mà trang sử dụng. Phần chính của trang thường được bao gồm trong cặp thẻ <html>...</html>. Các phần như <head> chứa thông tin không hiển thị trực tiếp trên trình duyệt (ví dụ: tiêu đề trang), trong khi <body> chứa nội dung hiển thị. Thẻ Tiêu đề và Đoạn văn bản: Thẻ tiêu đề <h1> đến <h6> được sử dụng để xác định kích thước và mức độ quan trọng của tiêu đề. Thẻ đoạn văn bản <p> được sử dụng để định nghĩa các đoạn văn bản. Thẻ Liên kết và Hình ảnh: Thẻ <a> được sử dụng để tạo liên kết đến các trang web khác. Thẻ <img> được sử dụng để nhúng hình ảnh vào trang web. Thẻ danh sách: Có thẻ <ul> (unordered list) để tạo danh sách không có thứ tự và <ol> (ordered list) để tạo danh sách có thứ tự. Mỗi mục trong danh sách được đặt trong thẻ <li>. Biểu mẫu (Forms): Thẻ <form> được sử dụng để tạo biểu mẫu và thu thập thông tin từ người dùng. Các phần tử như <input>, <select>, và <textarea> được sử dụng để xây dựng các trường nhập liệu.



Hình 4 Đây là hình ảnh web dùng ngôn ngữ html

HTML là một phần quan trọng của triển khai web và thường được kết hợp với CSS (Cascading Style Sheets) và JavaScript để tạo ra các trang web động và thú vị.

## Giới thiệu ngôn ngữ định dạng CSS

CSS, hay "Cascading Style Sheets" (Bảng kiểu Tích hợp), là một ngôn ngữ định dạng được sử dụng để mô tả cách mà các trang web được hiển thị trên trình duyệt. CSS được thiết kế để tách rời cấu trúc và nội dung của trang web từ kiểu dáng và định dạng, giúp cho quá trình phát triển và bảo trì trở nên dễ dàng hơn. Dưới đây là một số điểm quan trọng về CSS:



Hình 5 CSS

- Chia cấu trúc và Kiểu dáng: CSS giúp phân chia cấu trúc HTML và kiểu dáng của trang web. Điều này mang lại sự linh hoạt và dễ bảo trì hơn, vì bạn có thể thay đổi giao diện mà không làm ảnh hưởng đến cấu trúc nội dung.

- Chọn phần tử HTML: CSS sử dụng các bộ chọn (selectors) để xác định các phần tử HTML cụ thể mà bạn muốn kiểu dáng. Các bộ chọn có thể là tên thẻ HTML, lớp, ID, hoặc một cách phức tạp hơn.

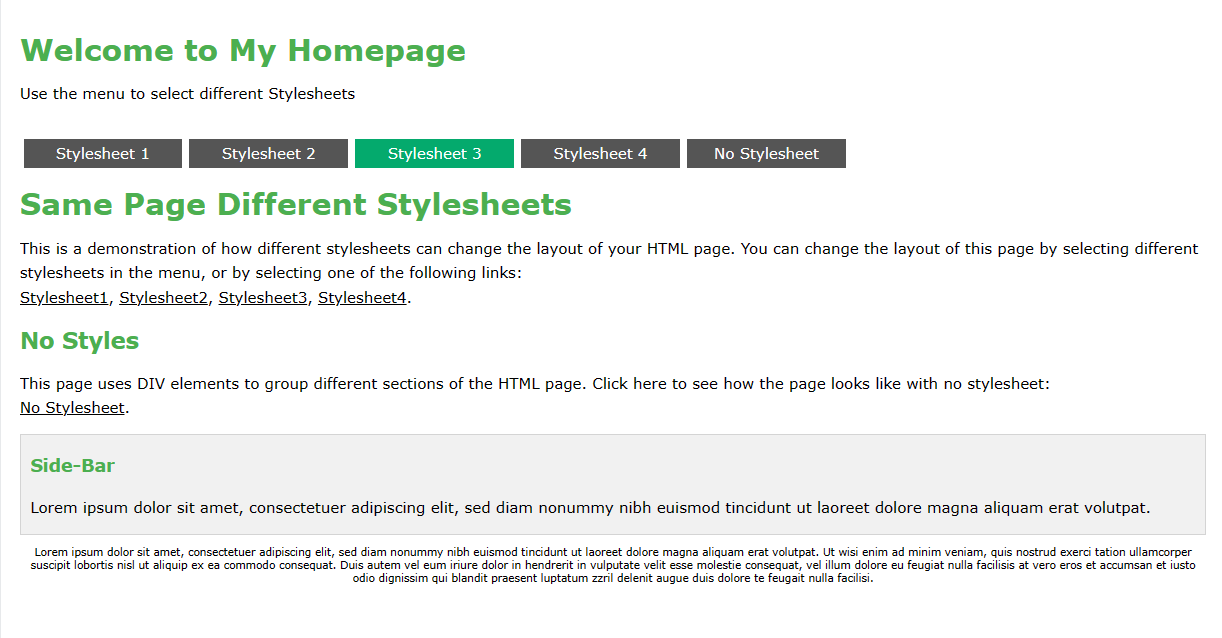
- Kiểu Dáng Màu sắc và Phông chữ: CSS cho phép bạn thiết lập màu sắc cho văn bản và nền của các phần tử. Bạn có thể điều chỉnh kiểu dáng của phông chữ, kích thước chữ, khoảng cách chữ và nhiều thuộc tính khác.

- Định dạng Hình ảnh và Đối tượng Đa phương tiện: CSS giúp kiểm soát vị trí, kích thước và định dạng hình ảnh và đối tượng đa phương tiện (như video và âm thanh) trên trang web.

- Box Model: Box model là một khái niệm quan trọng trong CSS, mô tả cách mà các phần tử HTML được bao quanh và tương tác với nhau. Gồm có kích thước của phần tử, các phần padding, border, và margin.

- Responsive Design: CSS cung cấp các kỹ thuật để tạo ra trang web phản ứng, tức là trang web có thể tự điều chỉnh và hiển thị đẹp trên nhiều thiết bị khác nhau, từ máy tính đến điện thoại di động.

- Animation và Transition: CSS hỗ trợ tạo hiệu ứng chuyển động và chuyển tiếp trên trang web thông qua các thuộc tính như animation và transition.

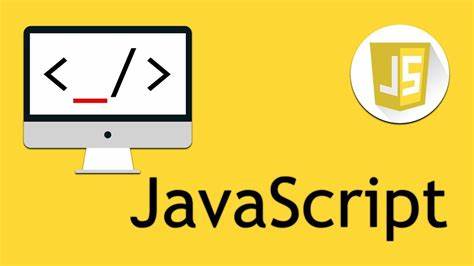


Hình 6 Hình kết hợp html css

CSS thường được sử dụng cùng với HTML và JavaScript để tạo ra trang web động, đẹp mắt và dễ bảo trì. Sự tích hợp của HTML, CSS và JavaScript là một phần quan trọng trong quá trình phát triển web hiện đại.

## Giới thiệu về ngôn ngữ lập trình Javascript

JavaScript là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu cho việc phát triển các ứng dụng web. Được thiết kế ban đầu để tương tác với trình duyệt web và thay đổi động nội dung trên trang, JavaScript ngày nay đã phát triển trở thành một ngôn ngữ đa nền tảng và đa mục đích.



Hình 7 javascript

Dưới đây là một số điểm quan trọng về JavaScript:

- Ngôn ngữ Chạy ở Phía Client: JavaScript thường chạy trực tiếp trên trình duyệt web của người dùng, giúp thay đổi nội dung và tương tác người dùng mà không cần tải lại trang.

- Đa Nhiệm và Bất đồng bộ: JavaScript hỗ trợ các tác vụ bất đồng bộ (asynchronous) thông qua các hàm callback và Promise, giúp xử lý các tác vụ như gọi API, tải ảnh mà không làm chậm trang web.

- DOM (Document Object Model) Manipulation: JavaScript cho phép thay đổi cấu trúc và nội dung của trang web thông qua truy cập và sửa đổi DOM, mô hình đối tượng của trang.

- Sự Kiện và Tương tác Người Dùng: JavaScript giúp xử lý sự kiện như click, hover, submit form, và tương tác người dùng khác để làm cho trang web trở nên động đà và thú vị.

- AJAX (Asynchronous JavaScript and XML): AJAX cho phép gửi và nhận dữ liệu từ máy chủ mà không cần phải tải lại toàn bộ trang, tạo ra trải nghiệm người dùng mượt mà và nhanh chóng.

- Frameworks và Thư viện: JavaScript có nhiều frameworks và thư viện mạnh mẽ như React, Angular, và Vue.js, giúp đơn giản hóa và tăng tốc quá trình phát triển ứng dụng web.

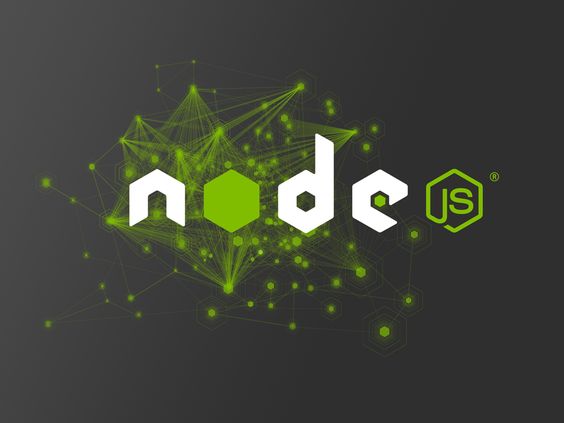
- Node.js: Node.js là một nền tảng cho phép chạy JavaScript ở phía server, mở ra khả năng phát triển cả ở phía client và phía server với cùng một ngôn ngữ. ES6 và Phiên bản

JavaScript đóng vai trò quan trọng trong cả quá trình phát triển front-end và back-end của ứng dụng web hiện đại, là một trong những ngôn ngữ lập trình phổ biến và quan trọng nhất trên thế giới web ngày nay.

## Kiến thức cơ bản về Nodejs, Reactjs, Mysql

### Nodejs: Môi trường, nền tảng thực thi đa chức năng

Nodejs, được xây dựng dựa trên Chrome's V8 JavaScript Engine, là một môi trường thực thi mã nguồn mở. Nó cung cấp khả năng thực thi mã JavaScript ở phía máy chủ (server-side) hỗ trợ việc xây dựng ứng dụng mạng có khả năng mở rộng cao.



Hình 8 Hình ảnh về Nodjes

Nodejs đặc trưng bởi khả năng không chỉ sử dụng mã nguồn mở mà còn xử lý đa nhiệm và không đồng bộ, giúp hệ thống xử lý nhiều yêu cầu cùng lúc mà không làm giảm hiệu suất. Với kiến trúc event-driven, Nodejs là một lựa chọn mạnh mẽ cho phần server của ứng dụng, đảm bảo khả năng mở rộng dễ dàng khi có thêm người dùng.

JavaScript ở Phía Server: Node.js cho phép việc sử dụng JavaScript ở phía máy chủ, không chỉ ở phía trình duyệt web. Điều này giúp đơn giản hóa quá trình phát triển bằng cách sử dụng cùng một ngôn ngữ lập trình ở cả hai phía.

Sự Kiện và Bất đồng bộ: Node.js được xây dựng với mô hình sự kiện và I/O bất đồng bộ (asynchronous), giúp xử lý hàng loạt yêu cầu mà không cần chờ đợi, tăng hiệu suất và thời gian đáp ứng.

Module và NPM (Node Package Manager): Node.js sử dụng hệ thống module để tách biệt chức năng và tái sử dụng mã nguồn. NPM là một công cụ quản lý gói (package manager) giúp cài đặt, quản lý và chia sẻ các thư viện và công cụ trong cộng đồng Node.js.

Web Server và API: Node.js thường được sử dụng để tạo ra các web server và API (Application Programming Interface) với khả năng xử lý đồng thời hàng nghìn kết nối.

Frameworks: Có nhiều frameworks phổ biến xây dựng trên Node.js như Express.js, Koa.js, và Nest.js, giúp đơn giản hóa quá trình phát triển ứng dụng web.

Real-time Applications: Node.js rất phù hợp cho các ứng dụng yêu cầu truyền thông thời gian thực như ứng dụng chat, trò chơi trực tuyến và ứng dụng đồng bộ.

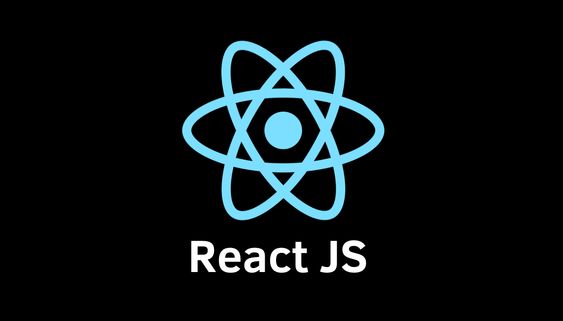
Dễ Mở Rộng: Node.js cho phép mở rộng dễ dàng bằng cách sử dụng các cổng kết nối và các giải pháp mở rộng như load balancing.

Community và Hỗ trợ: Node.js có một cộng đồng lớn và tích cực, với nhiều tài liệu, thư viện, và nguồn tư duy được chia sẻ.

Node.js đã trở thành một phần quan trọng trong ngôi nhà công nghệ web hiện đại, cung cấp một môi trường mạnh mẽ cho việc xây dựng các ứng dụng linh hoạt và hiệu suất cao.

### Reactjs: Framework thiết kế giao diện cực kì mạnh mẽ

Reactjs là một thư viện JavaScript được phát triển bởi Facebook, thiết kế để xây dựng giao diện người dùng đơn trang (Single Page Applications). Sự linh hoạt của React.js đến từ cách nó quản lý trạng thái và cập nhật giao diện người dùng mà không làm tải lại toàn bộ trang website, cung cấp trải nghiệm người dùng mượt mà và nhanh chóng.



Hình 9 Reacjs

React.js không chỉ giúp xây dựng giao diện người dùng đẹp mắt mà còn quản lý trạng thái của ứng dụng một cách hiệu quả. Cơ chế Virtual DOM và khả năng tái sử dụng component giúp giảm bớt độ phức tạp của ứng dụng, giúp duy trì mã nguồn dễ bảo trì và mở rộng.

Thành Phần (Components): React chia giao diện người dùng thành các thành phần nhỏ hơn, gọi là "components". Mỗi component đảm nhận một phần cụ thể của giao diện và có thể được sử dụng lại trong toàn bộ ứng dụng.

Virtual DOM (Document Object Model): React sử dụng Virtual DOM để tối ưu hóa quá trình cập nhật giao diện người dùng. Thay vì cập nhật toàn bộ DOM khi có thay đổi, React tạo ra một bản sao ảo của DOM và chỉ cập nhật các phần thay đổi. Sau đó, nó so sánh và áp dụng sự thay đổi vào DOM thực tế, giảm tải cho trình duyệt và tăng hiệu suất.

JSX (JavaScript XML): React sử dụng JSX, một cú pháp mở rộng của JavaScript, để viết mã HTML trong JavaScript. JSX giúp tạo ra mã nguồn gọn gàng và dễ đọc hơn.

Unidirectional Data Flow: React thực hiện mô hình unidirectional data flow, có nghĩa là dữ liệu di chuyển một chiều từ component cha đến component con. Điều này giúp quản lý trạng thái ứng dụng một cách dễ dàng và dự đoán được.

React Router: Để xử lý định hướng (routing) trong ứng dụng React, có thể sử dụng React Router. Điều này giúp quản lý các trang và URL một cách hiệu quả.

State và Props: React sử dụng khái niệm "state" và "props" để quản lý dữ liệu. "State" thường được sử dụng để lưu trữ trạng thái của component, trong khi "props" được sử dụng để truyền dữ liệu từ component cha đến component con.

### MySQL: Một hệ quản trị cơ sở dữ liệu quan hệ

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS - Relational Database Management System). Được phát triển và duy trì bởi một công ty tên là MySQL AB, sau này là một phần của Oracle Corporation khi Oracle mua lại MySQL AB vào năm 2008. MySQL được thiết kế để lưu trữ và quản lý dữ liệu trong các bảng có cấu trúc quan hệ. Nó sử dụng ngôn ngữ truy vấn SQL (Structured Query Language) để thực hiện các thao tác như truy vấn, cập nhật, xoá và thêm dữ liệu.



Hình 10 Hình ảnh về hệ quản trị cơ sở dữ liệu MySQL

MySQL là một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất trên thế giới, được sử dụng rộng rãi cho các ứng dụng web và nhiều dự án phần mềm do tính nhẹ, hiệu suất cao và mã nguồn mở.

## Lý luận thiết kế và phát triển Website

### Quy trình phát triển website

Quy trình phát triển website yêu cầu sự linh hoạt, với điều tiên quyết cần phải tìm hiểu các thành phần của website từ đó mới tiến hành thiết kế, triển khai và quản lý dự án. Các bước từ việc tìm hiểu, xác định yêu cầu đến triển khai và duy trì đều quan trọng để đảm bảo một ứng dụng hoạt động mạnh mẽ và ổn định.

### Tích hợp giữa Frontend và Backend

Việc tích hợp giữa phần frontend và backend là chìa khóa để xây dựng một ứng dụng hoàn chỉnh. Truyền thông dữ liệu hiệu quả, quản lý trạng thái và xử lý yêu cầu đòi hỏi sự hiểu biết sâu sắc về cả hai phía của ứng dụng.

Việc tích hợp hiệu quả yêu cầu sự hiểu biết về RESTful APIs và MySQL. RESTful APIs cung cấp một cách tiếp cận đơn giản và linh hoạt, trong khi MySQL cho phép truy vấn dữ liệu theo nhu cầu, giúp tối ưu hóa việc truyền thông.

## Giả thiết khoa học và phương pháp nghiên cứu đồ án cơ sở ngành

### Giả thiết khoa học: hiệu suất và sự linh hoạt

Giả thiết cơ bản của dự án là sự kết hợp giữa Node.js và React.js sẽ tạo ra một hệ thống website thương mại điện tử mạnh mẽ với hiệu suất cao và trải nghiệm người dùng đáng chú ý. Điều này dựa trên khả năng của Node.js xử lý đa nhiệm và sự linh hoạt của React.js trong quản lý trạng thái.

### Phương pháp nghiên cứu: phát triển liên tục và tiến hành kiểm thử

Phương pháp nghiên cứu sẽ áp dụng tích hợp liên tục để đảm bảo sự linh hoạt trong quá trình phát triển và kiểm thử hiệu suất để đánh giá khả năng chịu tải và ổn định của hệ thống website thương mại điện tử trong môi trường thực tế.

# 

# **CHƯƠNG 3: XÁC ĐỊNH NHU CẦU**

**1. Mô tả tổng quan:**

- Dự án này nhằm xây dựng một website thương mại điện tử chuyên về việc bán điện thoại.

- Website sẽ cung cấp nền tảng cho người dùng mua hàng hình thức trực tuyến tiện lợi.

**2. Đặc điểm chính của website:**

- Giao diện thân thiện và dễ sử dụng: Người dùng có thể dễ dàng thao tác với giao diện trang web mà không gặp bất kỳ trở ngại khi mua hàng.

- Danh mục sản phẩm đa dạng: Bao gồm các loại điện thoại theo từng mức giá với những thương hiệu đứng top đầu trong ngành sản xuất điện thoại thông minh.

- Tính năng tìm kiếm: Cho phép người dùng tìm kiếm sản phẩm theo nhiều tiêu chí khác nhau, …

- Hệ thống thanh toán: Đảm bảo thông tin thanh toán của người dùng được lưu trữ đúng.

- Quản lý đơn hàng hiệu quả: Hệ thống quản lý đơn hàng cho phép người quản trị kiểm tra, xử lý và theo dõi các đơn hàng một cách hiệu quả.

- Tính năng đăng nhập và đăng ký.

**3. Yêu cầu chức năng:**

- Hiển thị danh sách sản phẩm theo danh mục và thương hiệu.

- Tính năng tìm kiếm sản phẩm theo nhiều tiêu chí.

- Xem chi tiết sản phẩm với thông tin đầy đủ và hình ảnh minh họa.

- Quản lý đơn hàng: Xem, xóa đơn hàng từ người dùng.

- Đăng nhập và đăng ký tài khoản.

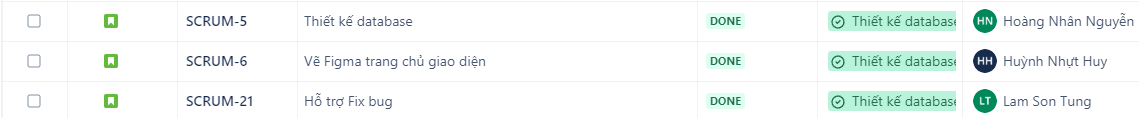
**4. Yêu cầu khác:**

- Giao diện thân thiện với người dùng.

- Khả năng mở rộng.

# CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM

**- Tuần 1: Từ 22/4 – 28/4: Thiết kế database, giao diện**



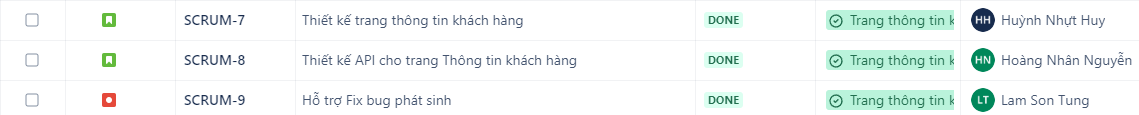
Hình 11 Nhiệm vụ tuần 1

- Nhiệm vụ của tuần đầu tiên sẽ là thiết kế database.

- Thiết kế giao diện cho trang chủ.

- Hỗ trợ fix bug trong lúc làm dự án.

**- Tuần 2: Từ 29/5 – 5/5: Trang thông tin khách hàng**

****

Hình 12 Nhiệm vụ tuần 2

- Thiết kế trang thông tin khách hàng

- Thiết kế các api cho rang thông tin khách hàng

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

**- Tuần 3: Từ 6/5 – 12/5: Trang chủ**

****

Hình 13 Nhiệm vụ tuần 3

- Thiết kế trang chủ

- Thiết kế các api cho trang chủ

- Tạo dữ liệu mẫu cho database

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

**- Tuần 4: Từ 13/5 – 19/5: Trang thông tin sản phẩm**

****

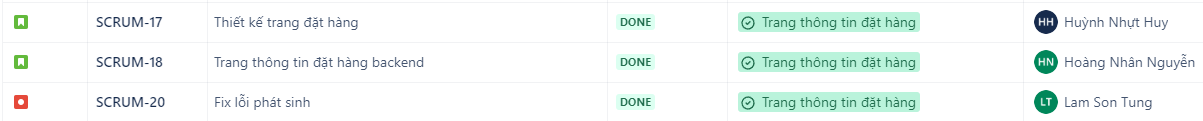
Hình 14 Nhiệm vụ tuần 4

- Thiết kế trang thông tin sản phẩm

- Thiết kế các api cho trang thông tin sản phẩm

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

**- Tuần 5: Từ 20/5 – 26/5: Trang thông tin đặt hàng**

****

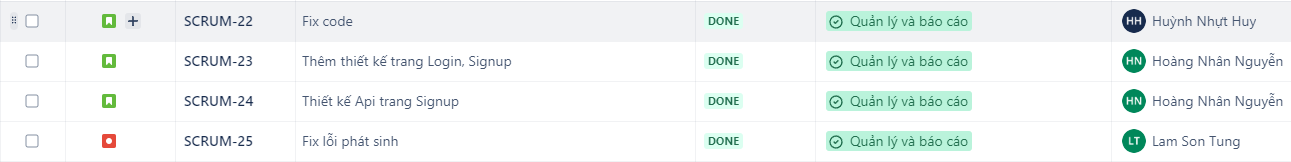
Hình 15 Nhiệm vụ tuần 5

- Thiết kế trang đặt hàng

- Trang thông tin đặt hàng backend

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

**- Tuần 6: Từ 26/5 – 6/6 : Quản lý và báo cáo**



Hình 16 Nhiệm vụ tuần 6

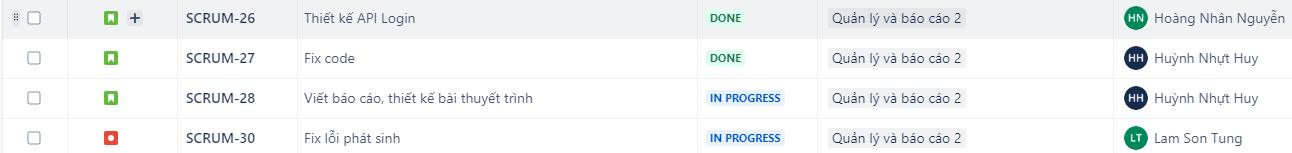
- Fix code phát sinh

- Thêm thiết kế trang Login và Sign up

- Thiết kế Api cho trang Signup

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

**- Tuần 7: Từ 7/5 – 13/6 : Quản lý và báo cáo 2**



Hình 17 Nhiệm vụ tuần 7

- Thiết kế Api cho trang Login

- Fix code

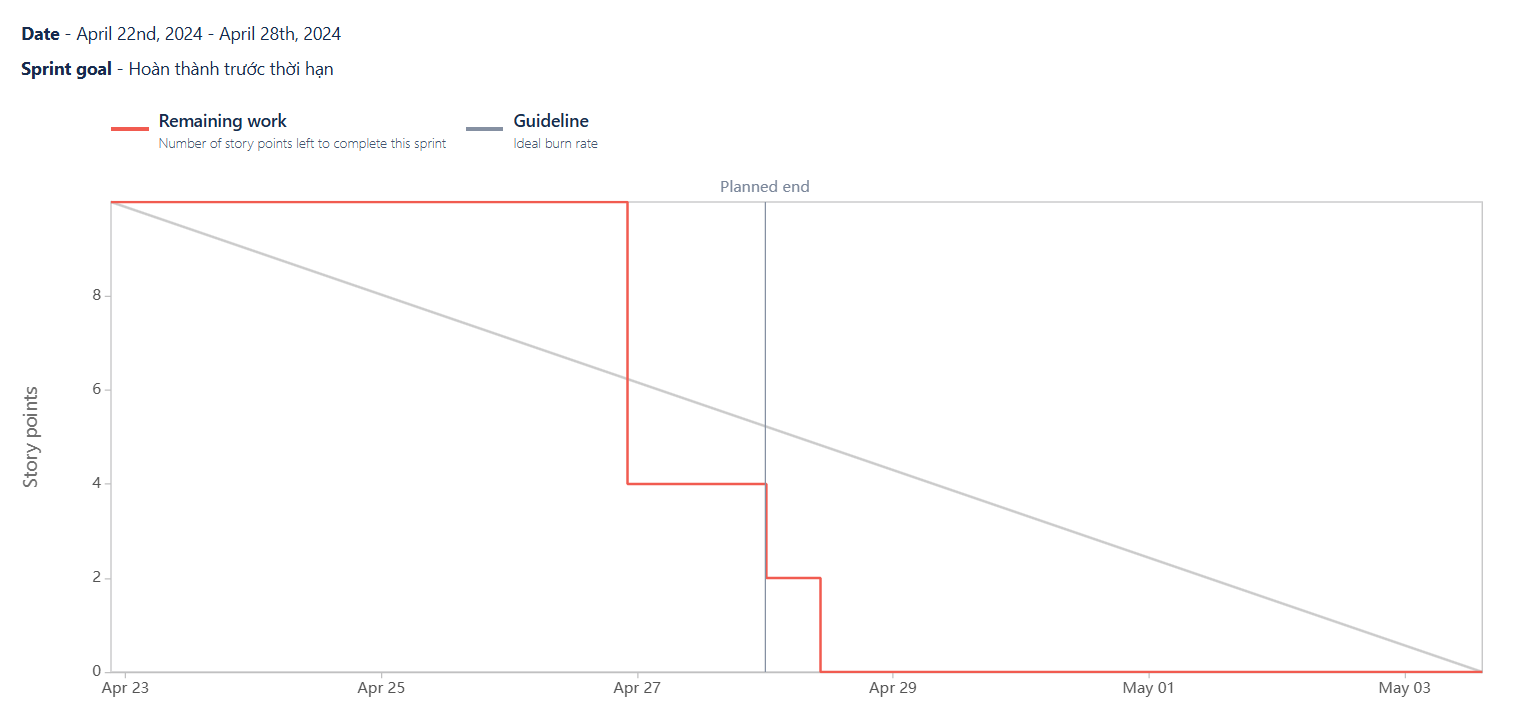
- Viết báo cáo và thuyết trình

- Ngoài ra scrum master sẽ hỗ trợ họp nhóm với các thành viên khác nhằm hỗ trợ sửa các lỗi nếu có phát sinh, các deverloper sẽ kiểm tra và thực hiện task này.

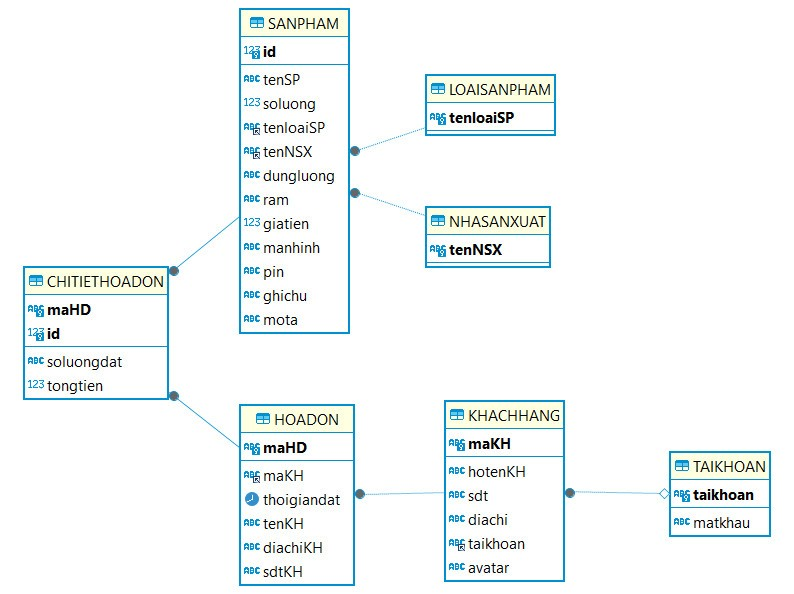
# CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH

**- Tuần 1: Từ 22/4 – 28/4: Thiết kế database, giao diện:**

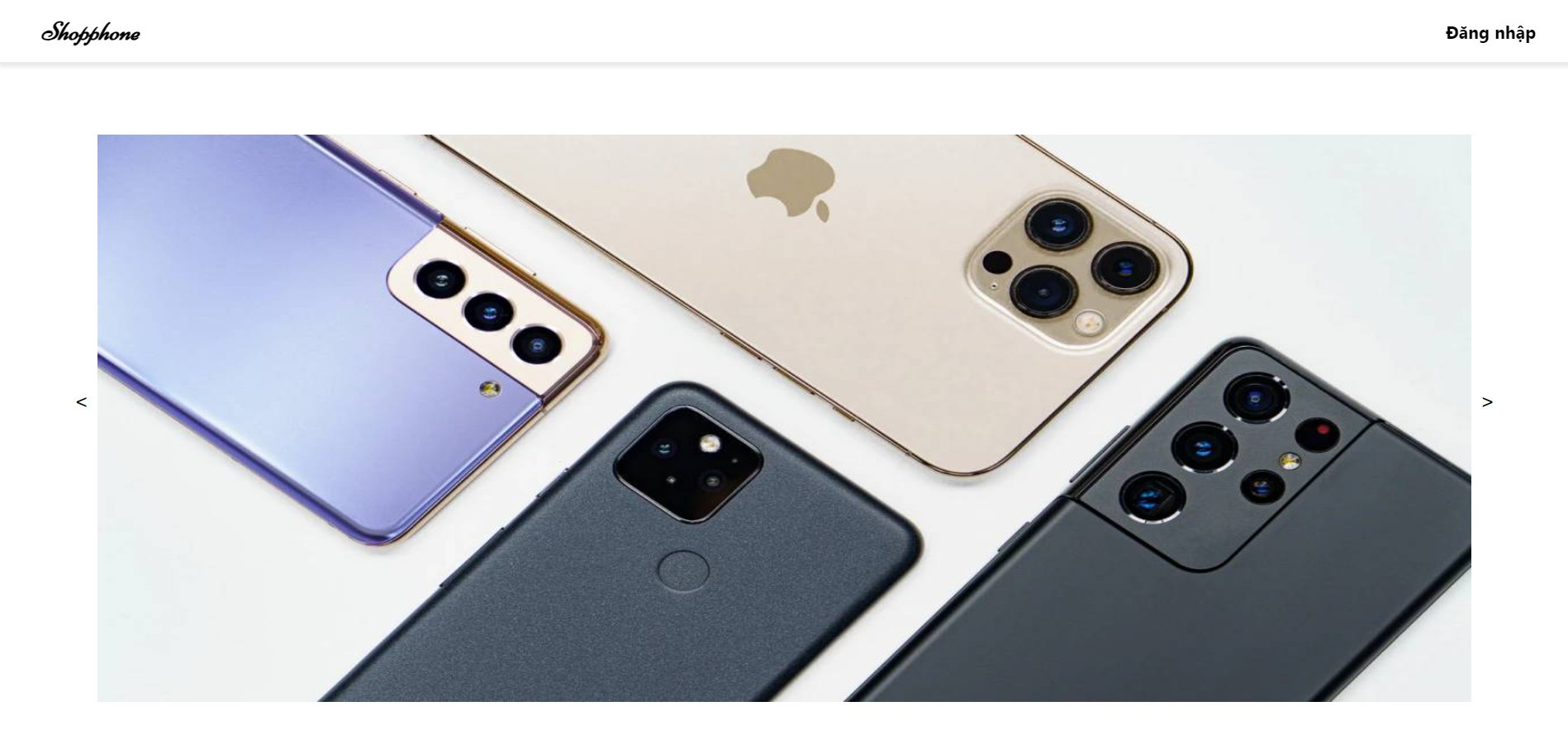
Đây là biểu đồ burndownchart tuần 1:

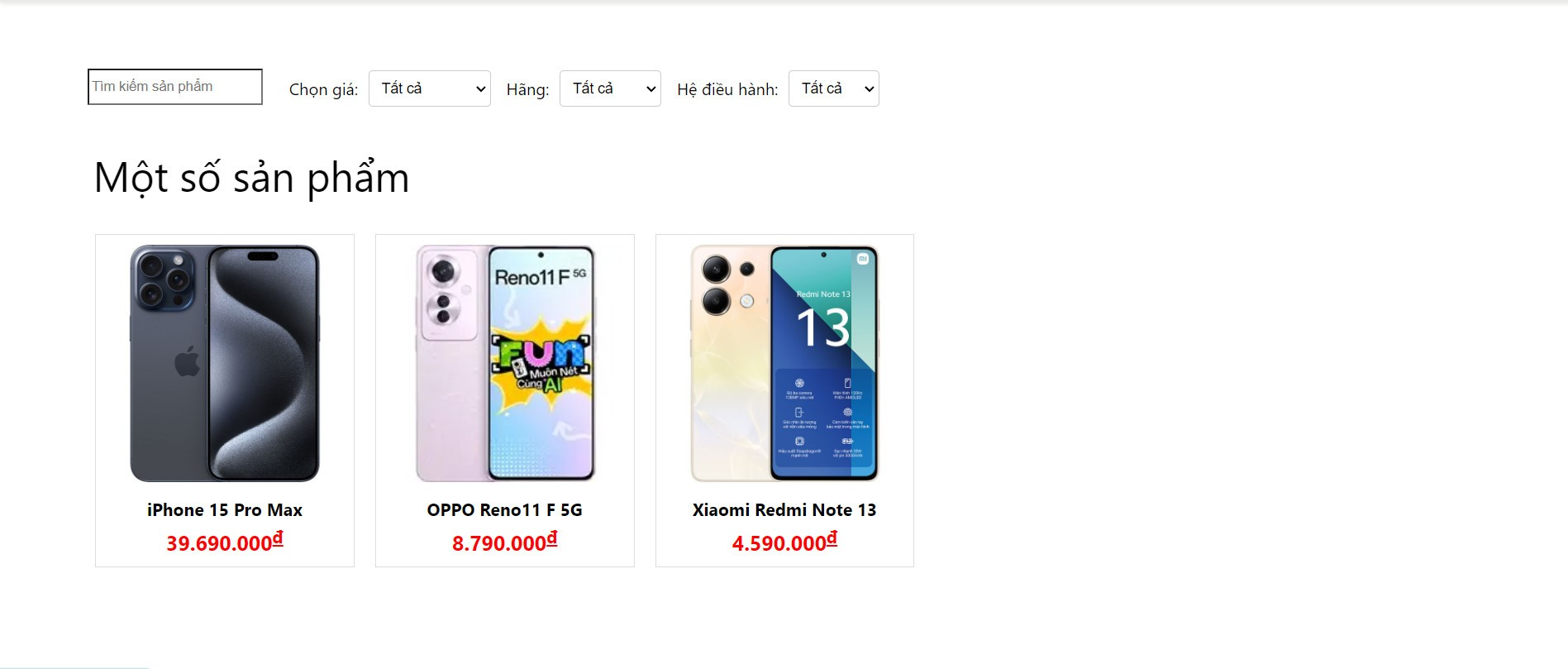


* Kết quả tuần đầu tiên thiết kế được database bằng cách tạo một docker compose với mô hình như sau:



* Vẽ Figma cho giao diện website:



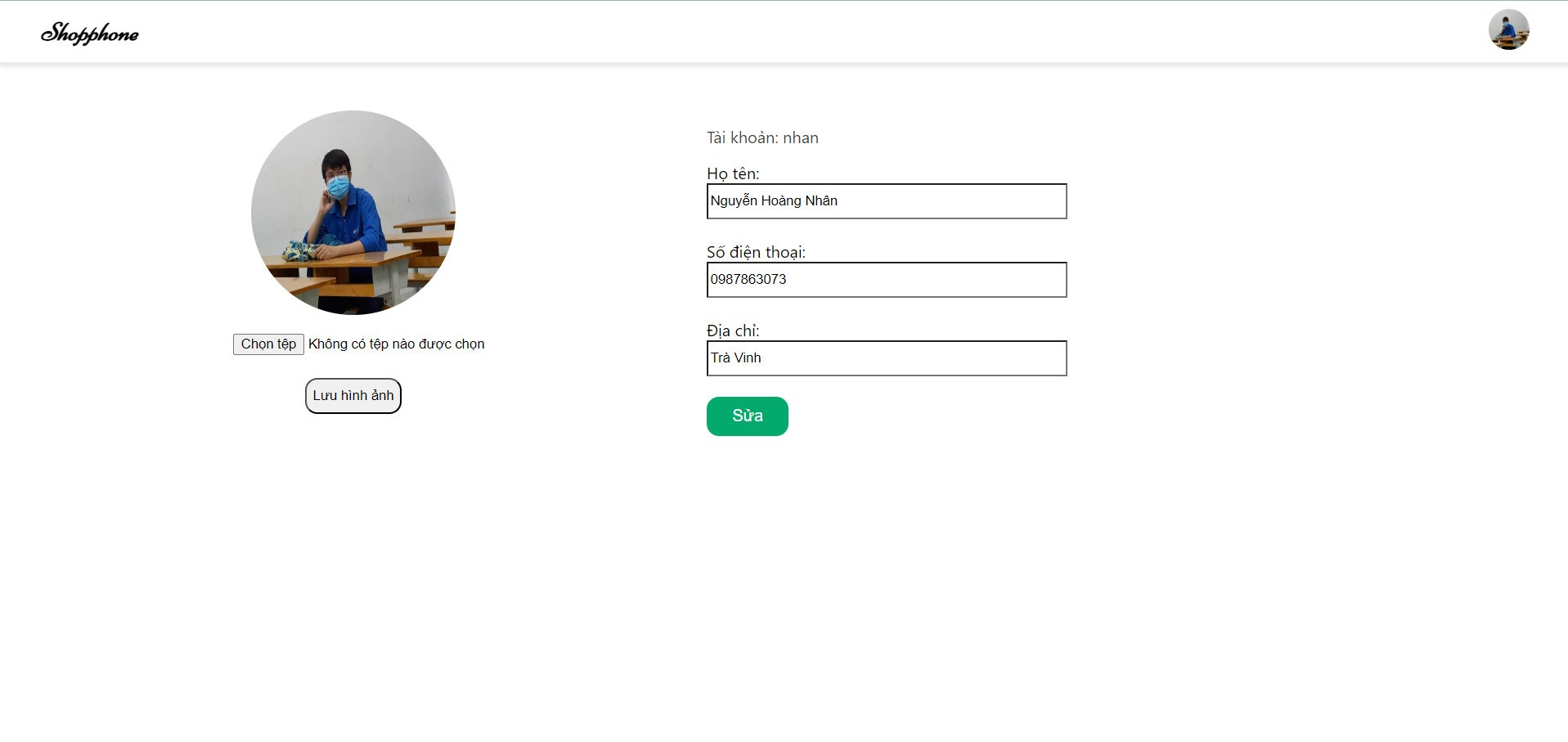


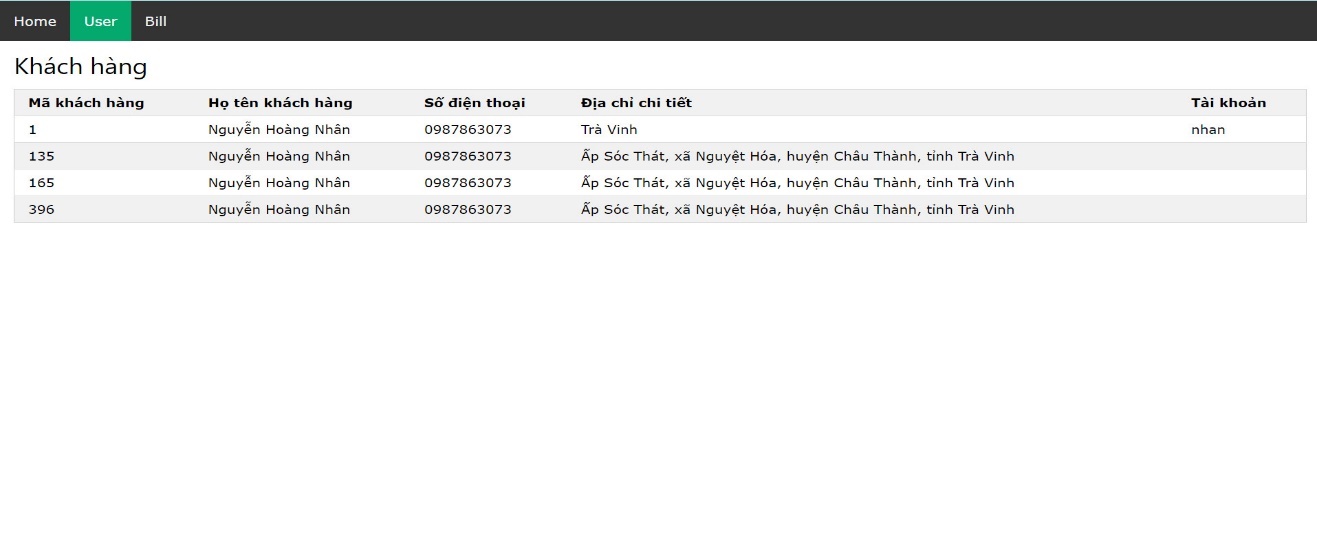
**- Tuần 2: Từ 29/5 – 5/5: Trang thông tin khách hàng**

**-** Đây là burndownchart tuần thứ 2 trang thông tin khách hàng:

****

- Kết quả đạt được sau tuần này là: thiết kế được trang thông tin khách hàng, tạo được API cho trang khách hàng.



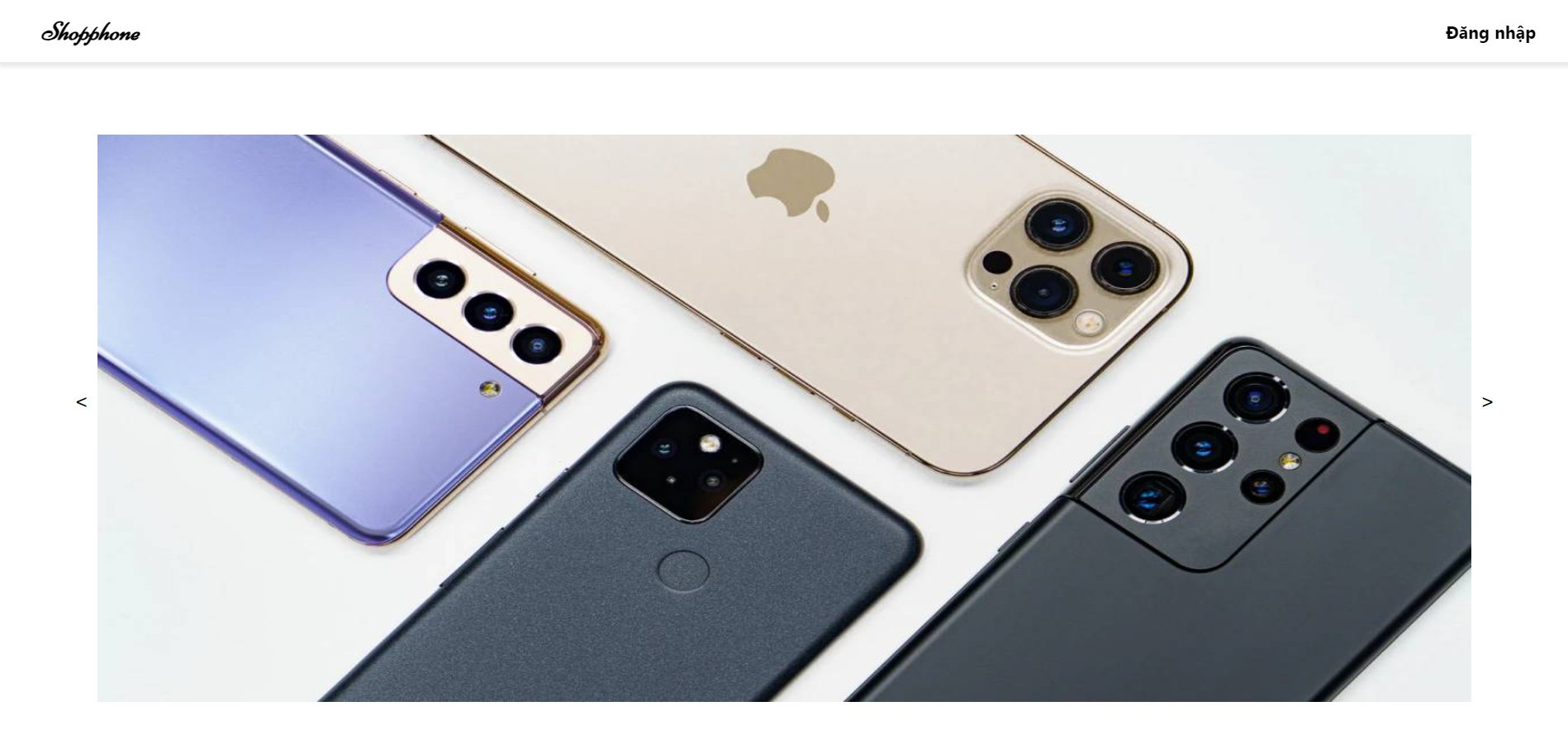
****

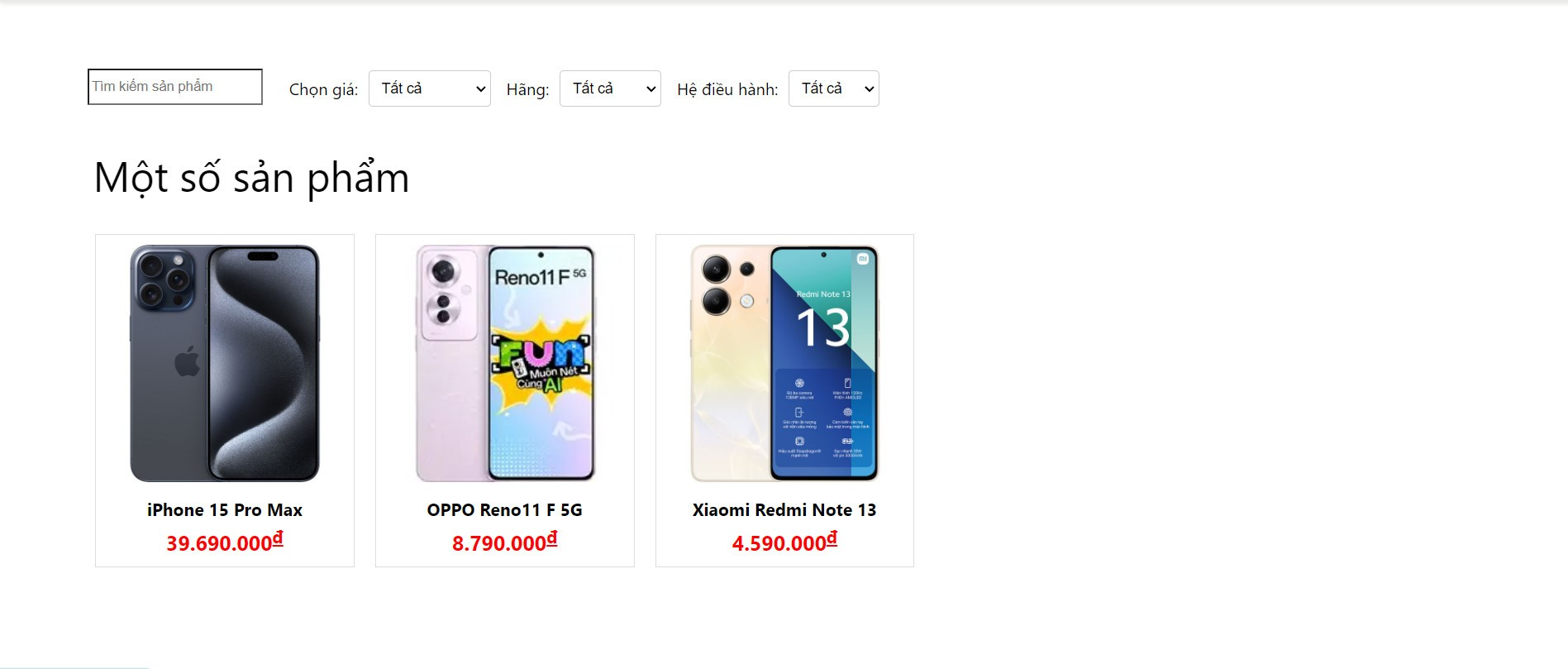
**Tuần 3: Từ 6/5 – 12/5: Trang chủ**

Đây là burndownchart của tuần 3:

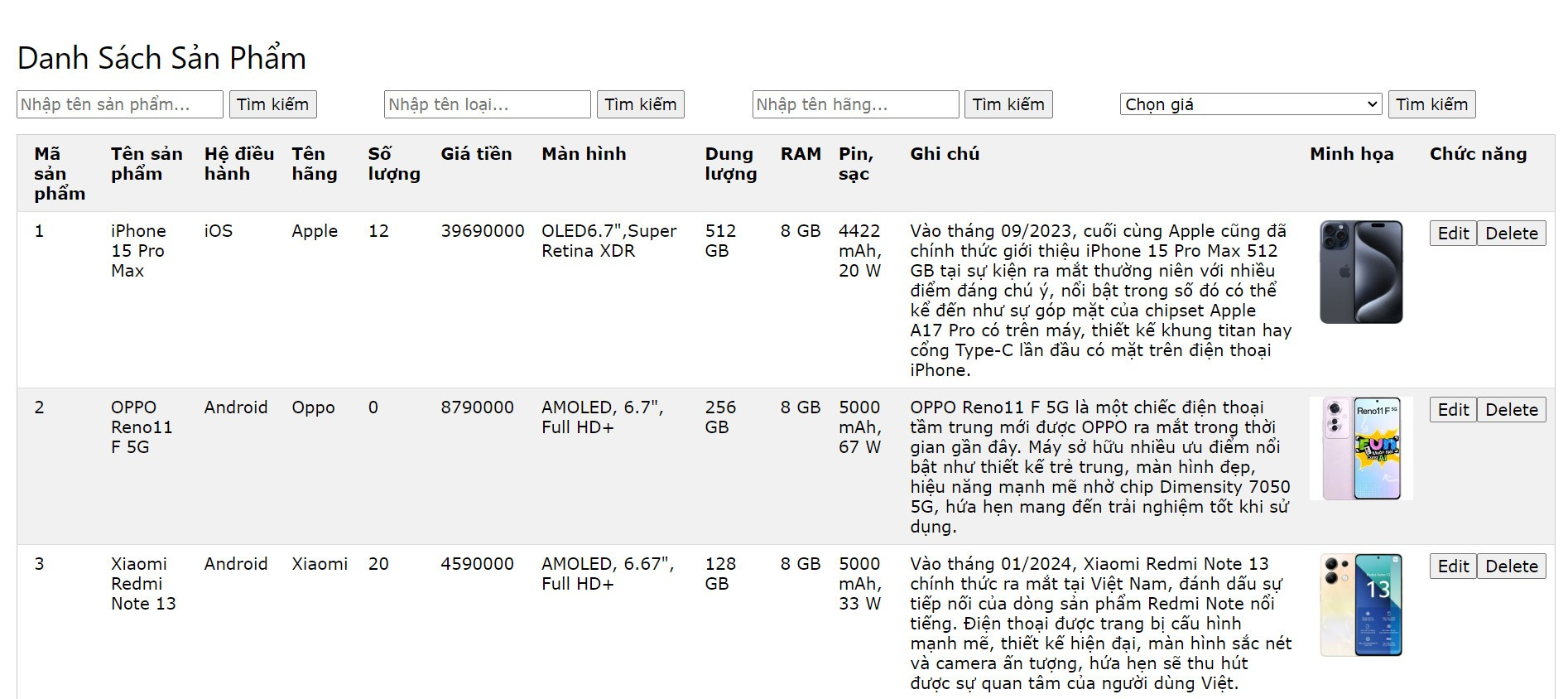


Kết quả tuần thứ 3 đạt được là tạo được Giao diện trang chủ, viết được API cho trang chủ, tạo dữ liệu mẫu cho database.



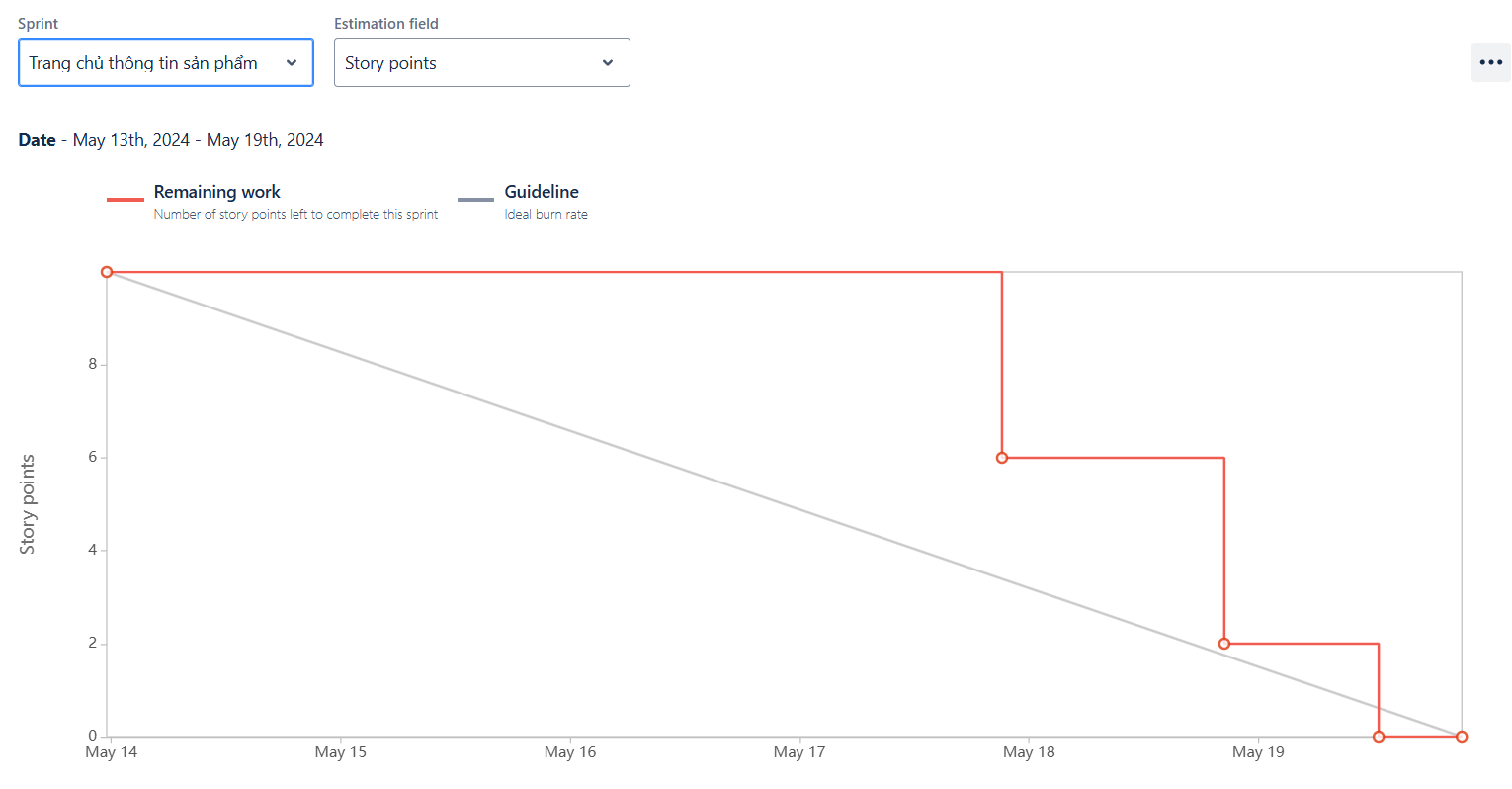


* Tạo database mẫu:

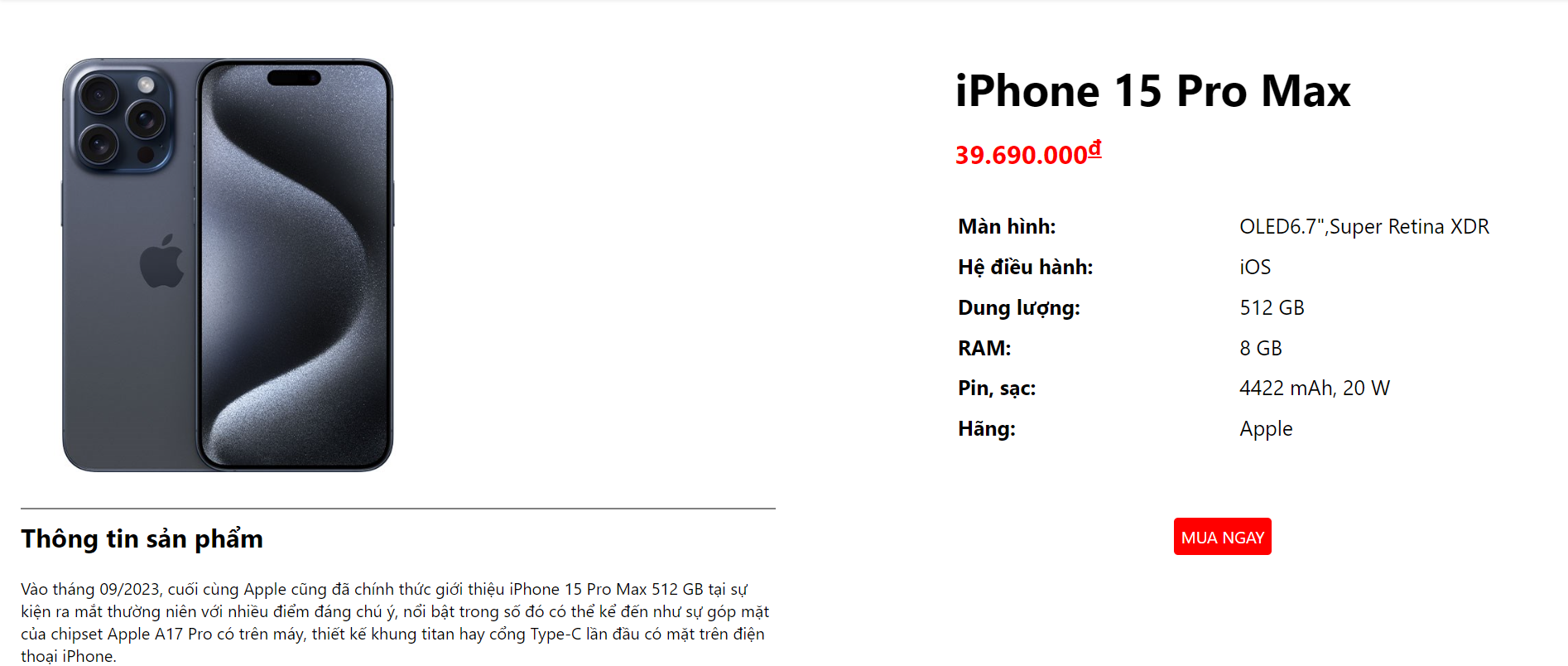


**- Tuần 4: Từ 13/5 – 19/5: Trang thông tin sản phẩm**

Đây là burndownchart của tuần 4 trang thông tin sản phẩm:

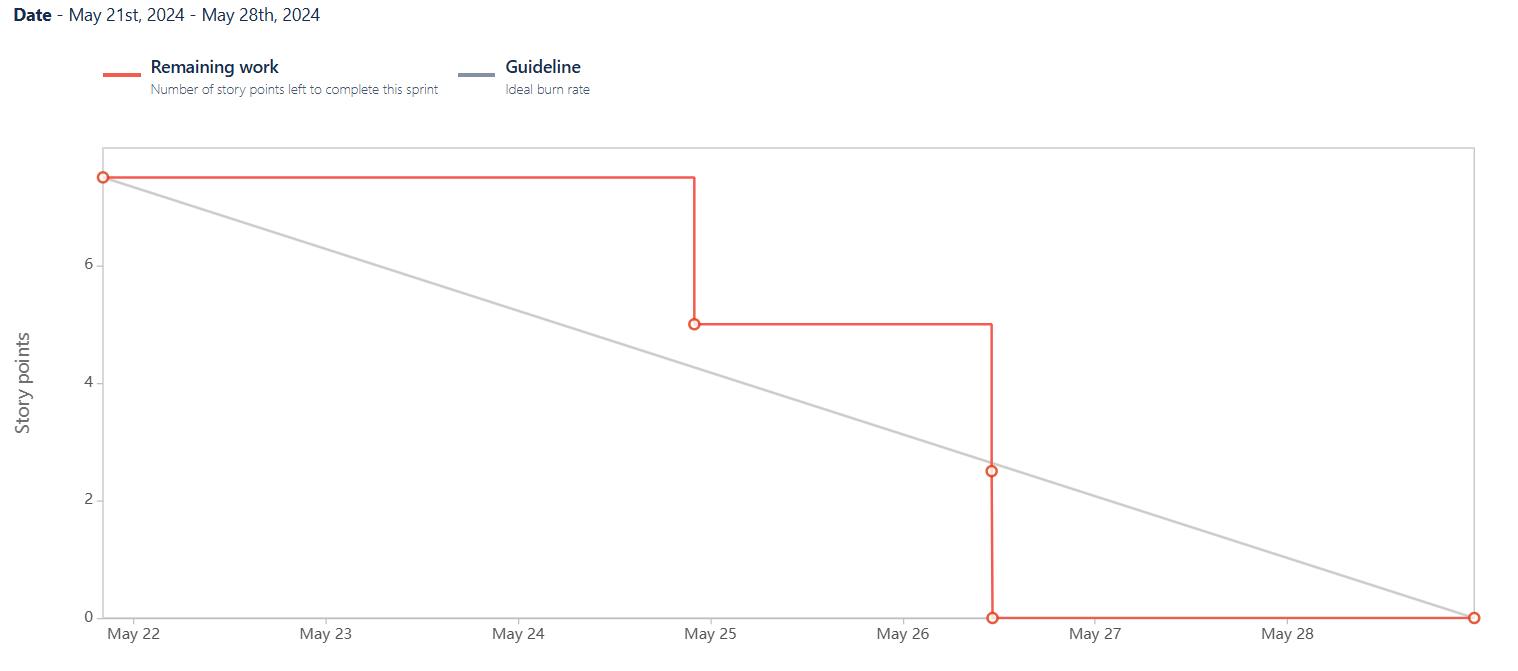


Kết quả đạt được ở tuần này là tạo được trang thông tin sản phẩm, tạo được API cho thông tin sản phẩm ở phía backend.

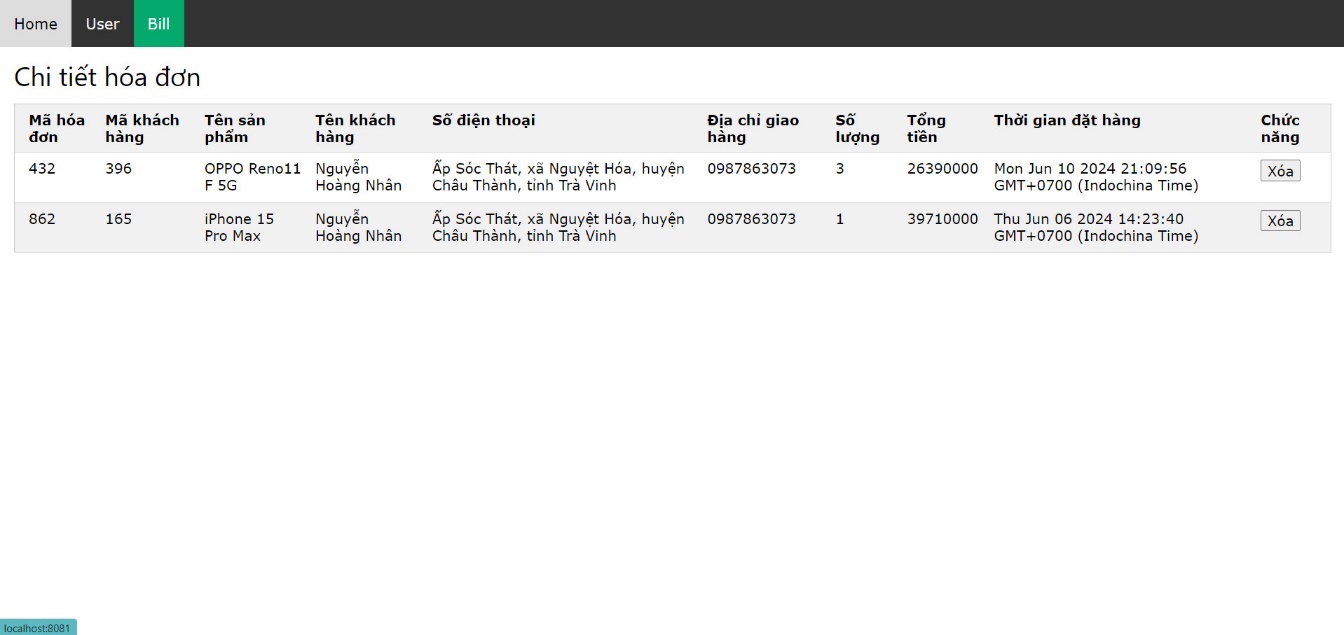
****

**- Tuần 5: Từ 20/5 – 26/5: Trang thông tin đặt hàng**

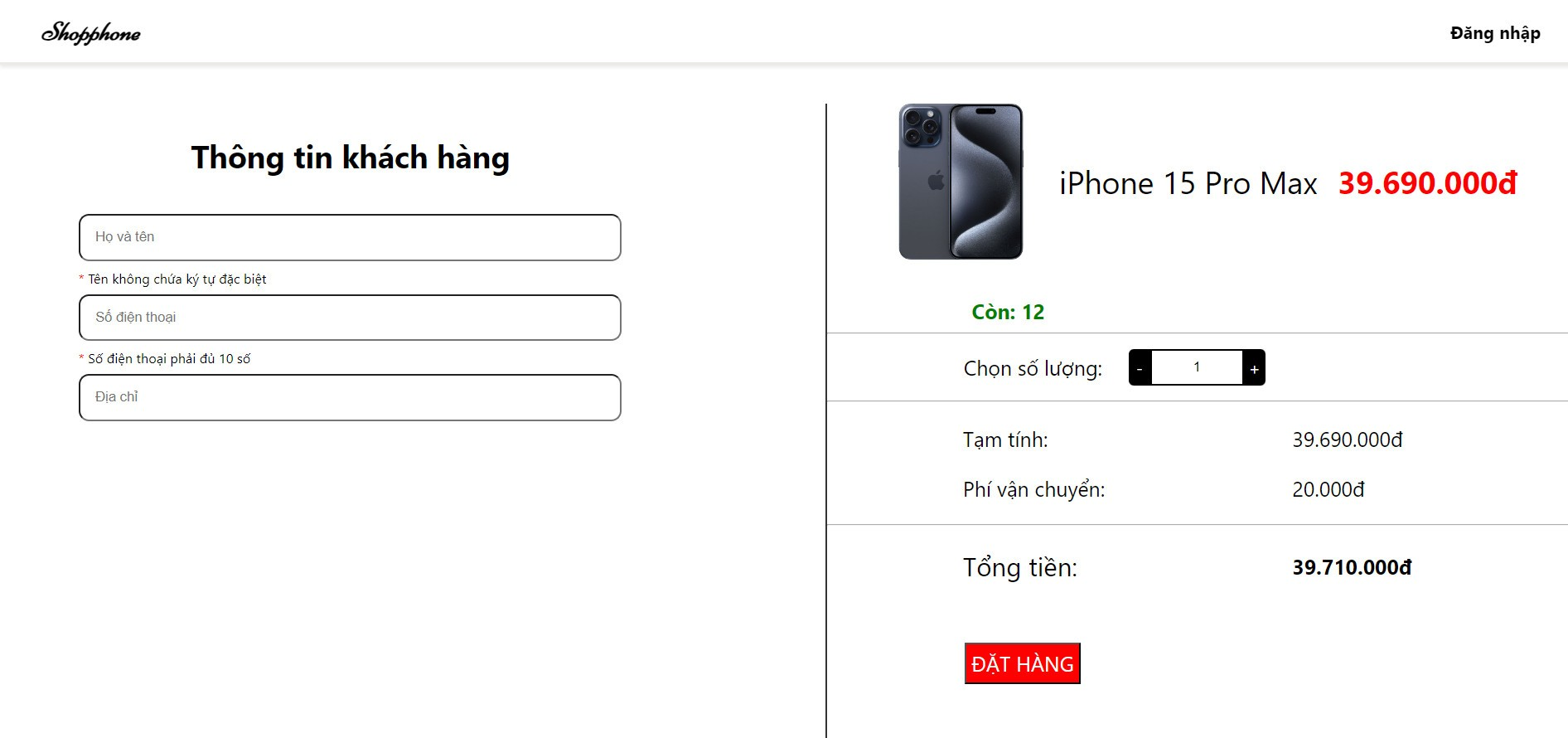
Đây là burndownchart của tuần 5 thông tin đặt hàng:



Thiết kế được trang thông tin đặt hàng, thông tin đặt hàng ở phía backend:

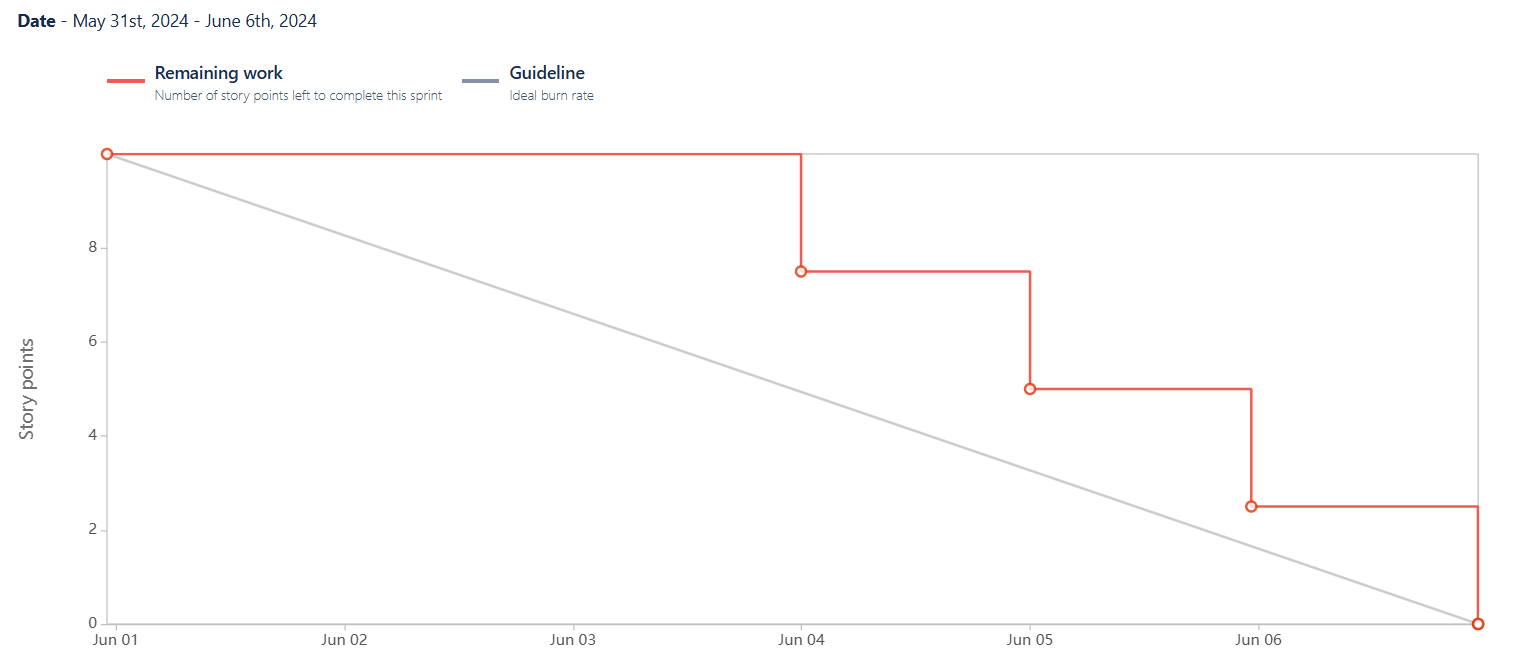


Trang thông tin đặt hàng: khi đặt hàng thì thông tin khách hàng sẽ được lưu xuống backend.



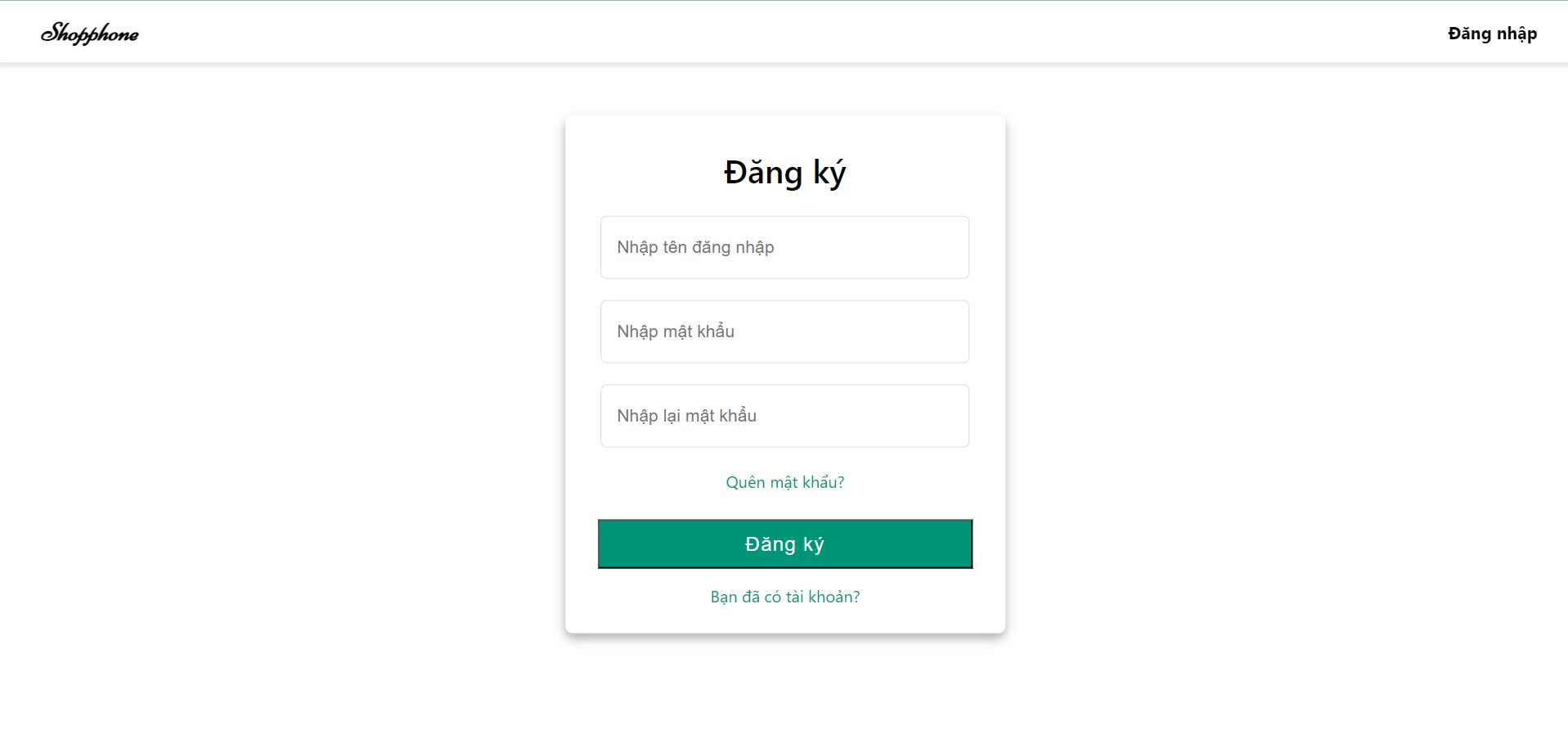
**- Tuần 6: Từ 26/5 – 6/6: Quản lý và báo cáo**

Đây là burndownchart của tuần 6 quản lý và báo cáo:

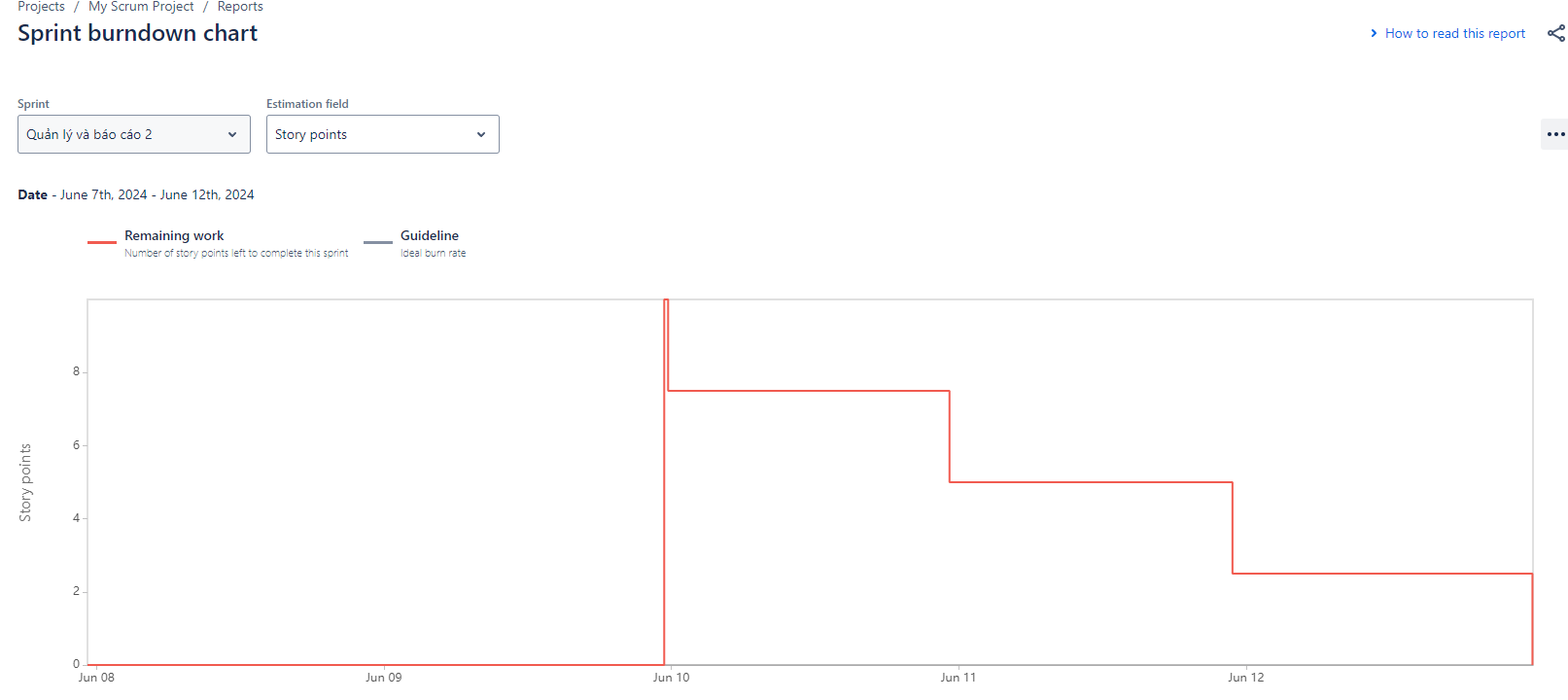


Kết quả ở tuần này ta đặt được là: fix các lỗi phát sinh, thiết kế được trang Login và Signup.

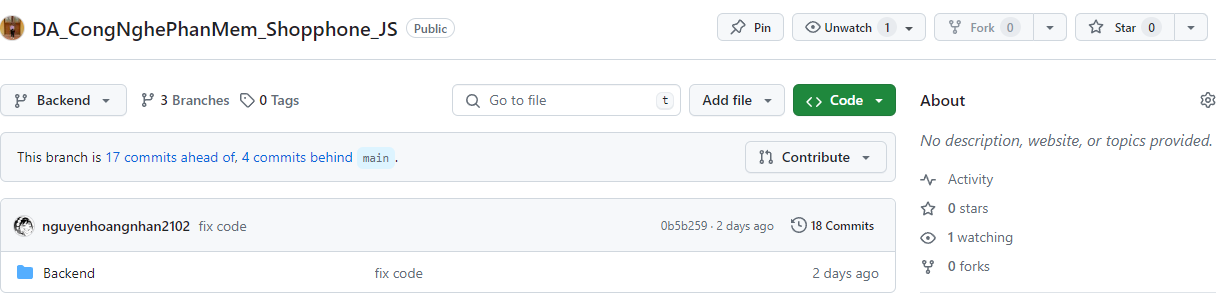




**- Tuần 7: Từ 7/5 – 12/6 : Quản lý và báo cáo 2**

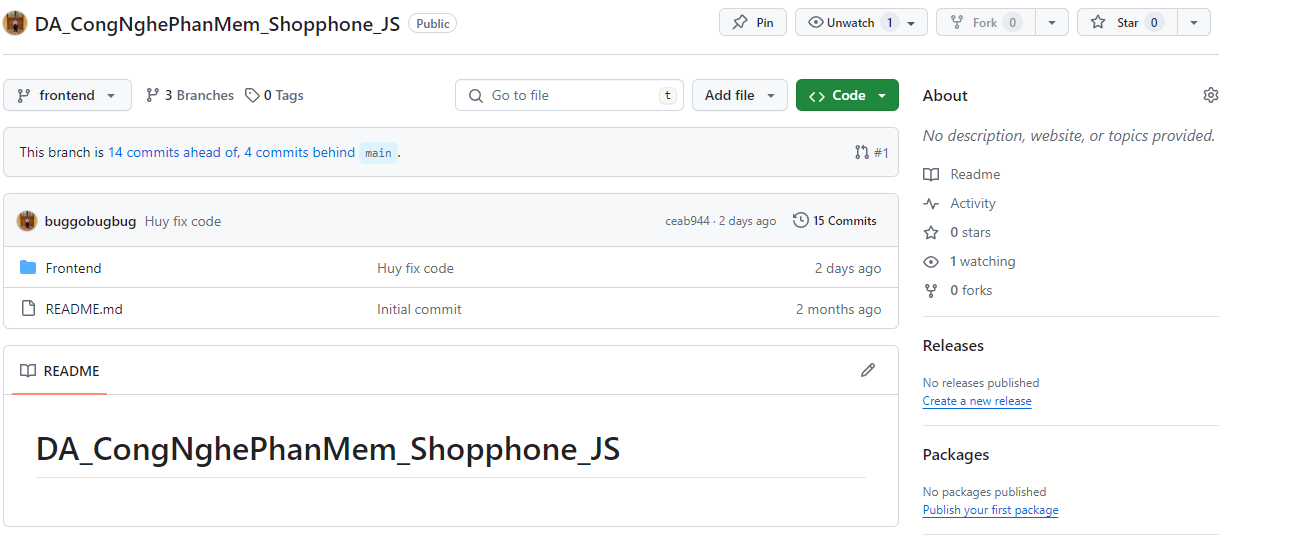


-Trong quá trình làm thì nhóm đã thực hiện được việc quản lý source code của mình thông qua github:

+Phía Backend: thực hiện 18 commit liên quan đến các task khi phân chia nhiệm vụ. Đây là lịch sử commit.

Hình 18 Commit github

+Phía Frontend: thực hiện 15 commit liên quan đến các task khi phân chia nhiệm vụ công nghệ phần mềm. Đây là lịch sử commit.



Hình 19 Commit github frontend

**CHƯƠNG 6: KẾT LUẬN**

**KẾT LUẬN:**

Việc lập trình trang web bán điện thoại không chỉ mang lại những thành quả về mặt kỹ thuật mà còn đem đến những tác động tích cực cho hoạt động kinh doanh và thị trường thương mại điện tử. Đây là những điểm kết luận quan trọng từ dự án này dưới góc độ thương mại điện tử:

+ Giúp doanh nghiệp mở rộng kinh doanh.

+ Tăng doanh thu.

+ Tiện ích cho khách hàng.

**HẠN CHẾ:**

Mặc dù dự án đã đạt được nhiều thành công, nhưng vẫn còn một số hạn chế cần được cải thiện trong tương lai. Một số hạn chế có thể bao gồm:

- Thiếu tính năng tiện ích: Có thể cần phát triển thêm các tính năng như chức năng đánh giá sản phẩm, hệ thống phản hồi từ khách hàng, thiếu thanh toán qua các ngân hàng, v.v.

- Hiệu suất và tối ưu hóa: Cải thiện hiệu suất của website và tối ưu hóa trải nghiệm người dùng để giảm thiểu thời gian tải trang và tăng tỷ lệ chuyển đổi.

- Thiếu nhiều chức năng.

-Chưa được hoàn thiện UI-UX.

**HƯỚNG PHÁT TRIỂN:**

Để cải thiện và phát triển dự án trong tương lai, có một số hướng phát triển có thể được xem xét:

**- Mở rộng danh mục sản phẩm**: Tăng cường danh mục sản phẩm với các loại điện thoại mới và các thương hiệu phổ biến hơn.

**- Tích hợp tính năng xã hội:** Tích hợp các tính năng xã hội để tạo ra một cộng đồng mua sắm trực tuyến đa dạng và sôi động.

**- Phát triển thêm để có thể trải nghiệm trên di động:** Phát triển ứng dụng di động và tối ưu hóa website cho trải nghiệm di động tốt hơn.

**- Phát triển chiến lược tiếp thị:** Xây dựng chiến lược tiếp thị mạnh mẽ để tăng cường sự nhận biết thương hiệu và tăng cường doanh số bán hàng.

Những hướng phát triển này sẽ giúp dự án tiếp tục phát triển và cung cấp giá trị ngày càng lớn hơn cho người dùng và khách hàng. Góp phần phát triển thêm trong lĩnh vực thương mại điện tử.

# DANH MỤC TÀI LIỆU THAM KHẢO

# 

[1]W3SCHOOL: <https://www.w3schools.com/>

[2]NODEJS: <https://nodejs.org/en>

[3]REACTJS: <https://react.dev/>

[4]ALASIAN: <https://www.atlassian.com/company>

|  |  |
| --- | --- |
| [5] | Atlassian, "Scrum Software Guide," [Online]. Available: www.atlassian.com. |
| [6] | w3schools, "Node.js Tutorial," [Online]. Available: https://www.w3schools.com/nodejs/. |
| [7] | w3schools, "React Tutorial - W3Schools," React Tutorial - W3Schools, [Online]. Available: https://www.w3schools.com/react/. |
| [8] | w3schools, "SQL Tutorial - W3Schools," SQL Tutorial - W3Schools, [Online]. Available: https://www.w3schools.com/sql/.xóa |

# PHỤ LỤC

**Đề tài**: Xây dựng website bán điện thoại  
**Giáo viên hướng dẫn**: Nguyễn Bảo Ân  
**Sinh viên thực hiện**:

Huỳnh Nhựt Huy – DA21TTA

Nguyễn Hoàng Nhân – DA21TTA

Lâm Sơn Tùng – DA21TTA

**Email**: nhathuy23122003@gmail.com  
**Điện thoại**: 0327434821

* **Hướng dẫn cài đặt dự án bao gồm máy chủ Node.js và React:**
* Reactjs
* Node.js
* Npm
* Dbeaver

1. **Clone Repository:**

Git clone: https://github.com/buggobugbug/DA\_CongNghePhanMem\_Shopphone\_JS.git

1. **Cài Đặt Dependencies:**
   * Chuột phải vào thư mục src\_nodejs/package.json chọn Open in integrated Terminal

npm install

* + Chuột phải vào thư mục src\_Reactjs/package.json chọn Open in integrated Terminal

npm install

1. **Cấu Hình Môi Trường:**
   * Tạo một tệp .env từ tệp .env.example và điền thông tin cần thiết (env có sẵn trong dự án).
2. **Chạy Ứng Dụng Nodejs:**
   * Mở visual studio code Open dự án
   * Chuột phải vào thư mục src\_nodejs/package.json chọn Open in integrated Terminal

npm run dev

* + Hoặc sử dụng nodemon để tự động khởi động lại ứng dụng khi có thay đổi (đã được cài sẵn):

npm install -g nodemon

1. **Chạy Ứng Dụng Reactjs:**
   * Mở visual studio code Open dự án
   * Chuột phải vào thư mục src\_reactjs/package.json chọn Open in integrated Terminal

npm run start