

**TRƯỜNG ĐẠI HỌC TRÀ VINH**  
**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**BÁO CÁO MÔN HỌC**  
**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**  
**XÂY DỰNG WEB BÁN SÁCH**  
**BẰNG FRAMEWORK EMBERJS**

Giảng viên hướng dẫn:

TS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*

Nguyễn Đức Thịnh - 110121228

Huỳnh Quang Vinh -110121270

Nguyễn Hữu Trí -110121120

Lớp: DA21TTB

*Trà Vinh, tháng 6 năm 2024*

**TRƯỜNG ĐẠI HỌC TRÀ VINH**  
**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**BÁO CÁO MÔN HỌC**  
**CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI**  
**XÂY DỰNG WEB BÁN SÁCH**  
**BẰNG FRAMEWORK EMBERJS**

Giảng viên hướng dẫn:

TS. Nguyễn Bảo Ân

*Sinh viên thực hiện:*

Nguyễn Đức Thịnh - 110121228

Huỳnh Quang Vinh - 110121207

Nguyễn Hữu Trí – 110121120

Lớp: DA21TTB

*Trà Vinh, tháng 6 năm 2024*

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting or typing. There are no margins, text, or other markings on the page.

(Ký tên và ghi rõ họ tên)

## This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

**Giảng viên phản biện**  
(Ký tên và ghi rõ họ tên)

---

## LỜI CẢM ƠN

Nhóm em xin cảm ơn thầy Nguyễn Bảo Ân đã hướng dẫn nhóm em trong suốt học phần vừa qua. Cảm ơn với những kiến thức thầy mang đến cho nhóm em để nhóm em có thể hoàn thành bài báo cáo này một cách tốt nhất.

Thông qua học phần và thực hiện đồ án môn học này giúp chúng em hiểu rõ hơn về Công Nghệ Phần Mềm.

Nhóm chúng em đã cố gắng trong quá trình học cũng như nghiêm túc trong việc thực hiện đồ án này, nếu có thiếu sót nhóm em rất mong nhận được sự thông cảm từ quý thầy cô và hơn hết là những lời góp ý vô cùng quý báu từ quý thầy cô, để nhóm em rút kinh nghiệm và hoàn thiện hơn.

Nhóm em xin chân thành cảm ơn!

---

---

## MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU .....	8
1.1. Đặc tả ứng dụng .....	8
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	10
2.1. Mô hình Agile.....	10
2.1.1. Giới thiệu về mô hình Agile.....	10
2.1.2. Đặt trưng của Agile.....	10
2.1.3. Giá trị cốt lõi của Agile.....	11
2.1.4. Nguyên tắc của Agile.....	12
2.2. Mô hình Scrum .....	13
2.2.1. Giới thiệu về mô hình Scrum .....	13
2.2.2. Thuật ngữ Scrum.....	14
2.2.3. Các vai trò quan trọng trong Scrum .....	14
2.2.4. Các sự kiện trong Scrum .....	15
2.2.5. Tạo tác của Scrum.....	15
2.2.6. Quy trình Scrum.....	16
2.3. Công nghệ và kiến trúc được sử dụng .....	16
2.3.1. EmberJS .....	16
2.3.2. Điểm mạnh và điểm yếu của EmberJS .....	17
2.3.3. Kết luận .....	18
CHƯƠNG 3: XÁC ĐỊNH NHU CẦU .....	19
3.1. Mô tả bài toán .....	19
3.2. Đặc tả yêu cầu hệ thống.....	19
3.2.1. Yêu cầu chức năng.....	19
3.2.2. Yêu cầu phi chức năng.....	19
3.2.3. Yêu cầu kỹ thuật .....	20
CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM .....	21
4.1. Kế hoạch phát triển phần mềm .....	21
4.1.1. Xác định Product Backlog .....	21
4.1.2. Xác định Sprint Backlog .....	21
4.1.3. Phân công và kế hoạch.....	22
CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH.....	24
CHƯƠNG 6: KẾT LUẬN.....	30
DANH MỤC TÀI LIỆU THAM KHẢO .....	32

---

---

## DANH MỤC HÌNH ẢNH

Hình 1. Giao diện Figma của trang chủ.....	24
Hình 2. Giao diện Figma của trang đăng ký.....	24
Hình 3. Giao diện Figma của trang đăng nhập.....	25
Hình 4. Burndown chart tiến độ của Sprint 1 .....	25
Hình 5. Giao diện Figma của trang sản phẩm .....	26
Hình 6. Burndown chart tiến độ của Sprint 2 .....	26
Hình 7. Giao diện Figma của trang giỏ hàng .....	27
Hình 8. Burndown chart tiến độ của Sprint 3 .....	27
Hình 9. Kết quả triển khai ứng dụng lên docker .....	29
Hình 10. Burndown chart tiến độ của Sprint 4 .....	29

## DANH MỤC BẢNG CHIẾU

Bảng 1. Xác định Sprint Backlog .....	21
Bảng 2. Bảng phân công và kế hoạch.....	23

---

## CHƯƠNG 1: GIỚI THIỆU

### 1.1. Đặc tả ứng dụng

Sự bùng nổ phát triển của các ứng dụng web đang ngày càng phát triển mạnh trong những năm gần đây. Việc duy trì mã nguồn, quản lý dữ liệu và tương tác người dùng đòi hỏi một quy trình phát triển hiệu quả và linh hoạt. Trong bối cảnh này, EmberJS nổi lên như một công cụ mạnh mẽ để giúp giải quyết những thách thức này và mang lại trải nghiệm người dùng mượt mà.

EmberJS (có tên thật là SproutCore MVC framework) là một framework mã nguồn mở JavaScript phía client dùng cho phát triển các ứng dụng web và sử dụng mô hình Model-View-Controller. Mục tiêu của đề án này là tìm hiểu về cách EmberJS hoạt động, cách ứng dụng nó vào quá trình phát triển một trang web. Dưới đây là cách bước để hoàn thành đề án.

#### **Bước 1: Nghiên cứu EmberJS**

Tìm hiểu EmberJS: Nghiên cứu cơ bản về EmberJS, bao gồm cấu trúc components, services routing và các tính năng quan trọng để hiểu rõ cách sử dụng EmberJS trong phát triển front-end của trang web.

#### **Bước 2: Phân tích nhu cầu và thiết kế hệ thống**

Phân tích nhu cầu: Bắt đầu bằng việc xác định yêu cầu chính của dự án bao gồm chức năng của trang web, nhu cầu kinh doanh và yêu cầu về thông tin sản phẩm.

Thiết kế hệ thống: Xây dựng một thiết kế hệ thống tổng quan, bao gồm cấu trúc của trang web, các chức năng chính của web như hiển thị thông tin sản phẩm, đăng ký và đăng nhập vào hệ thống, tìm kiếm sản phẩm và giỏ hàng để thêm các sản phẩm.

#### **Bước 3: Thiết kế giao diện Thiết kế giao diện:**

Tập trung vào trải nghiệm trực quan, dễ nhìn và dễ sử dụng để người dùng duyệt, tìm kiếm sách và thực hiện giao dịch mua hàng. Trang chủ giới thiệu các sách mới phát hành và sách nổi bật kèm theo hình ảnh. Trang sản phẩm liệt kê các loại sách theo từng danh mục sách khác nhau. Nhấp vào sách



---

có thể thêm vào giỏ hàng. Tìm kiếm sản phẩm hoạt động bằng cách nhập từ cần tìm và hiện thị ra kết quả tên sách, giá bán.

#### **Bước 4: Hiện thực hóa nghiên cứu**

Cài đặt: Cài đặt NodeJs phiên bản LTS (Long-Term-Support) để có sự tương thích với EmberJs. Cài đặt Ember CLI để giúp tạo và quản lý dự án EmberJs.

Hiện thực hóa Front-end: Viết mã front-end bằng Html, Css, EmberJS, áp dụng các khái niệm đã nghiên cứu như components, services và routing để xây dựng giao diện người dùng tương tác.

Kiểm thử và Debug: Thực hiện kiểm thử chức năng và kiểm thử để đảm bảo tính ổn định và an toàn của trang web.

---

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Mô hình Agile

#### 2.1.1. Giới thiệu về mô hình Agile

Agile là một quy trình hỗ trợ một nhóm quản lý hiệu quả một dự án bằng cách chia nhỏ dự án thành nhiều giai đoạn, cho phép cộng tác nhất quán với những người tham gia để thúc đẩy những cải tiến ổn định ở mọi giai đoạn

Về bản chất, Agile giống như một phương pháp luận, một triết lý dựa trên nguyên tắc phân đoạn vòng lặp (iterative) và tăng trưởng (incremental) nên sở hữu tính linh hoạt cao. Tính chất này đi ngược lại với các phương pháp quản lý dự án truyền thống – vốn dĩ triển khai các giai đoạn một cách tuyến tính và vô cùng bị động trước các thay đổi bất ngờ.

Ngày nay, triết lý Agile đã vượt xa khỏi khu vực truyền thống của mình là phát triển phần mềm để đóng góp sự thay đổi trong cách thức làm việc, quản lý, sản xuất ở các ngành khác như sản xuất, dịch vụ, sales, marketing, giáo dục,... và trở thành một phương pháp quản lý dự án phổ biến nhất hiện nay với nhiều đại diện được gọi là các phương pháp “họ Agile”.

#### 2.1.2. Đặt trung của Agile

**Tính lặp (Iterative):** Dự án sẽ được thực hiện trong các phân đoạn lặp đi lặp lại, thường có khung thời gian ngắn (từ 1-4 tuần). Trong mỗi phân đoạn đó, nhóm phát triển dự án sẽ thực hiện đầy đủ các công việc cần thiết như lập kế hoạch, phân tích yêu cầu, thiết kế, triển khai, kiểm thử để cho ra các phần nhỏ của sản phẩm.

**Tính tăng trưởng và tiến hóa (Incremental & Evolutionary):** Cuối các phân đoạn, nhóm cho ra các phần nhỏ của sản phẩm cuối cùng, thường là đầy đủ, có khả năng chạy tốt, được kiểm thử cẩn thận và có thể sử dụng. Theo thời gian, phân đoạn này tiếp nối phân đoạn kia, các phần chạy được này sẽ được tích lũy, lớn dần lên cho tới khi toàn bộ yêu cầu của khách hàng được thỏa mãn.

**Tính thích nghi (adaptive):** Do các phân đoạn chỉ kéo dài trong một khoảng thời gian ngắn và việc lập kế hoạch cũng được điều chỉnh liên tục, nên các thay đổi

---

trong quá trình phát triển (yêu cầu thay đổi, thay đổi công nghệ, thay đổi định hướng về mục tiêu,...) đều có thể được đáp ứng theo cách thích hợp.

**Nhóm tự tổ chức và liên chức năng:** Các cấu trúc nhóm này tự phân công công việc mà không dựa trên các mô tả cứng nhắc về chức danh hay một sự phân cấp rõ ràng. Nhóm tự tổ chức đã đủ các kỹ năng cần thiết để có thể được trao quyền tự ra quyết định, tự quản lý và tổ chức lấy công việc của chính mình để đạt được hiệu quả cao nhất.

**Quản lý tiến trình thực nghiệm (Empirical Process Control):** Các nhóm Agile ra các quyết định dựa trên các dữ liệu thực tiễn (data-driven) thay vì tính toán lý thuyết hay các tiên giả định. Agile rút ngắn vòng đời phản hồi để dễ dàng thích nghi và gia tăng tính linh hoạt nhờ đó có thể kiểm soát được tiến trình, và nâng cao năng suất lao động.

**Giao tiếp trực diện (face-to-face communication):** Agile không phản đối việc tài liệu hóa, nhưng đánh giá cao hơn việc giao tiếp trực diện thay vì thông qua giấy tờ. Agile khuyến khích nhóm phát triển trực tiếp nói chuyện để hiểu rõ hơn về những gì khách hàng thực sự cần. Trong giao tiếp giữa nội bộ nhóm, Agile khuyến khích trực tiếp trao đổi và thống nhất với nhau về thiết kế của hệ thống và cùng nhau triển khai thành các chức năng theo yêu cầu.

**Phát triển dựa trên giá trị (value-based development):** Một trong các nguyên tắc cơ bản của Agile là “sản phẩm chạy tốt chính là thước đo của tiến độ”. Nhóm Agile thường cộng tác trực tiếp và thường xuyên với khách hàng để biết yêu cầu nào có độ ưu tiên cao hơn, mang lại giá trị hơn sớm nhất có thể cho dự án.

### 2.1.3. Giá trị cốt lõi của Agile

Bốn giá trị cốt lõi của tuyên ngôn Agile:

- + **Cá nhân và tương tác hơn là quy trình và công cụ:** sự quan trọng của sự giao tiếp và tương tác giữa các thành viên trong nhóm làm việc. Agile tập trung vào việc

---

xây dựng một môi trường làm việc linh hoạt và cởi mở, khuyến khích sự hợp tác và trao đổi thông tin trong nhóm để đạt được mục tiêu chung.

- + **Phần mềm hoạt động tốt hơn là tài liệu đầy đủ:** đề cao việc tập trung vào việc phát triển và cung cấp phần mềm có giá trị ngay từ giai đoạn sớm. Agile đặt sự ưu tiên vào việc tạo ra các sản phẩm phần mềm hoạt động chính xác và đáp ứng yêu cầu của khách hàng thay vì chỉ tập trung vào việc viết tài liệu chi tiết.
- + **Hợp tác với khách hàng hơn là đàm phán hợp đồng:** nhấn mạnh sự quan trọng của việc tương tác và hợp tác chặt chẽ với khách hàng. Agile khuyến khích sự tương tác thường xuyên với khách hàng để hiểu và đáp ứng các yêu cầu thay đổi, từ đó tạo ra giá trị và sự hài lòng cho khách hàng.
- + **Ứng phó, phản hồi với các thay đổi hơn là làm theo kế hoạch:** nhấn mạnh sự linh hoạt và khả năng thích nghi của phương pháp Agile. Agile tin rằng sự thay đổi là chắc chắn xuất hiện trong quá trình phát triển phần mềm, và thay vì tuân thủ kế hoạch ban đầu, Agile tập trung vào việc tương ứng và đáp ứng linh hoạt với sự thay đổi để đạt được kết quả tốt nhất.

#### 2.1.4. Nguyên tắc của Agile

➤ **12 nguyên tắc của tuyên ngôn Agile:**

1. Ưu tiên sự hài lòng khách hàng thông qua việc giao sản phẩm có giá trị sớm và liên tục.
2. Chào đón việc thay đổi trong suốt quá trình phát triển. Tránh sự chậm trễ khi có yêu cầu thay đổi yêu cầu.
3. Chuyển giao sản phẩm thường xuyên. Chuyển giao phần mềm sử dụng được định kỳ, từ một vài tuần đến một vài tháng, với một thời gian ngắn hơn.

- 
4. Khách hàng và đội phát triển phải làm việc cùng nhau hàng ngày trong suốt dự án.
  5. Xây dựng các dự án xoay quanh các cá nhân có động lực. Tạo cho họ một môi trường và hỗ trợ họ những thứ cần thiết và tin tưởng họ để công việc được hoàn thành.
  6. Phương pháp hiệu quả nhất là truyền tải thông tin đến vào bên trong đội phát triển là hội thoại mặt-đối-mặt.
  7. Phần mềm làm việc được là thước đo của quá trình.
  8. Các quy trình Agile thúc đẩy sự phát triển bền vững. Các nhà tài trợ cho dự án, các nhà phát triển và người dùng cuối có thể duy trì một tốc độ vô hạn định.
  9. Liên tục quan tâm đến kỹ thuật xuất sắc và thiết kế tốt giúp nâng cao tính linh hoạt.
  10. Tính đơn giản – nghệ thuật tối đa hoá khối lượng công việc chưa hoàn thành – là điều thiết yếu.
  11. Các kiến trúc, yêu cầu và thiết kế tốt nhất xuất hiện từ các nhóm tự tổ chức.
  12. Ở thời điểm kết thúc sprint, các team nên xem lại làm thế nào để hiệu quả hơn, sau đó cùng tự cải thiện ứng xử, cải tiến quy trình, kỹ thuật của mình sao cho phù hợp.

## **2.2. Mô hình Scrum**

### **2.2.1. Giới thiệu về mô hình Scrum**

Scrum là một khung tổ chức công việc (framework) là một phương pháp phát triển phần mềm linh hoạt, được dựa theo cơ chế lặp và tăng trưởng. Scrum được thiết kế để hỗ trợ việc tạo ra, phân phối và cải tiến các sản phẩm phức tạp.

---

Điểm nổi bật của Scrum chính là các Sprint – các giai đoạn phát triển và vòng lặp và việc tối đa hóa thời gian phát triển sản phẩm để có thể đạt được mục tiêu – các Product Goal.

Scrum được xem là mô hình phổ biến nhất trong các phương pháp vận hành doanh nghiệp theo phương pháp luận Agile.

### 2.2.2. Thuật ngữ Scrum

➤ Các thuật ngữ cơ bản trong Scrum:

- + **Daily Scrum:** Một cuộc họp nhóm hàng ngày trong đó tiến độ được xem xét và công việc phải hoàn thành vào ngày hôm đó như đã thảo luận và thống nhất.
- + **Sprint:** Một khoảng thời gian ngắn, thường là từ hai đến bốn tuần, khi phát triển phần gia tăng của sản phẩm.
- + **ScrumMaster:** Một huấn luyện viên nhóm hướng dẫn nhóm sử dụng hiệu quả Scrum.
- + **Product:** Sản phẩm phần mềm đang được phát triển bởi nhóm Scrum.
- + **ProductOwner:** Một thành viên trong nhóm chịu trách nhiệm xác định các tính năng và thuộc tính của sản phẩm. Họ xem xét công việc đã hoàn thành và giúp kiểm tra sản phẩm.
- + **Product backlog:** Danh sách việc cần làm bao gồm các hạng mục như lỗi, tính năng và cải tiến sản phẩm mà nhóm Scrum chưa hoàn thành.
- + **Development team:** Một nhóm nhỏ tự tổ chức từ năm đến tám người chịu trách nhiệm phát triển sản phẩm.
- + **Potentially shippable product increment:** Đầu ra của một lần chạy nước rút có chất lượng đủ cao để triển khai cho khách hàng sử dụng.
- + **Velocity (Vận tốc):** Ước tính khối lượng công việc mà một nhóm có thể thực hiện trong một lần chạy nước rút.

### 2.2.3. Các vai trò quan trọng trong Scrum

- **Product Owner:** Chịu trách nhiệm đảm bảo rằng nhóm phát triển luôn tập trung vào sản phẩm mà họ đang xây dựng thay vì chuyển hướng sang công

---

việc thú vị về mặt kỹ thuật nhưng ít liên quan hơn. Trong quá trình phát triển sản phẩm, người quản lý sản phẩm thường đảm nhận vai trò Chủ sở hữu sản phẩm.

- **Scrum Master:** Có nhiệm vụ hướng dẫn nhóm sử dụng hiệu quả phương pháp Scrum. Các nhà phát triển Scrum nhấn mạnh rằng ScrumMaster không phải là người quản lý dự án thông thường mà là huấn luyện viên cho nhóm. Họ có thẩm quyền trong nhóm về cách sử dụng Scrum. Trong nhiều công ty sử dụng Scrum, ScrumMaster cũng có một số trách nhiệm quản lý dự án.
- **Development Team (Nhóm phát triển):** Là đội ngũ trực tiếp làm ra sản phẩm, họ bao gồm các chuyên gia có nhiệm vụ chuyển giao phần tăng trưởng ở cuối mỗi Sprint

#### 2.2.4. Các sự kiện trong Scrum

- + **Sprint:** Khoảng thời gian cố định (thường là 2-4 tuần) trong đó một phần hoàn chỉnh và có giá trị của sản phẩm được tạo ra.
- + **Sprint Planning (Lập kế hoạch Sprint):** Cuộc họp đầu Sprint để xác định công việc sẽ được thực hiện trong Sprint.
- + **Daily Scrum:** nhóm có các cuộc họp hàng ngày để xem xét tiến độ và cập nhật danh sách các hạng mục công việc chưa hoàn thành.
- + **Sprint Review:** Cuộc họp cuối Sprint để trình bày và kiểm tra phần sản phẩm hoàn chỉnh với các bên liên quan.
- + **Sprint Retrospective:** Cuộc họp cuối Sprint để nhóm tự đánh giá và tìm cách cải thiện quy trình và hiệu suất làm việc.

#### 2.2.5. Tạo tác của Scrum

Scrum sử dụng ba tạo tác chính để quản lý công việc và đảm bảo sự minh bạch và hiệu quả trong quá trình phát triển sản phẩm. Các tạo tác này bao gồm:

- + **Product Backlog (Danh sách công việc sản phẩm):** là một danh sách động chứa tất cả các công việc cần thiết để phát triển và cải tiến sản phẩm. Đây là một tài liệu sống, liên tục được cập nhật và sắp xếp theo thứ tự ưu tiên.

- 
- + **Sprint Backlog (Danh sách công việc Sprint):** là danh sách các mục công việc được chọn từ Product Backlog sẽ được thực hiện trong Sprint hiện tại. Nó cũng bao gồm một kế hoạch chi tiết về cách nhóm phát triển sẽ thực hiện công việc trong Sprint.
  - + **Increment (Phần tăng trưởng):** là phần sản phẩm hoàn chỉnh và có giá trị được tạo ra sau mỗi Sprint. Nó phải đáp ứng "Definition of Done" (Định nghĩa của hoàn thành) và có thể cung cấp ngay lập tức nếu cần.

#### 2.2.6. Quy trình Scrum

- + **Lập Kế Hoạch Sprint (Sprint Planning):** Product Owner và nhóm phát triển chọn các mục tiêu từ Product Backlog để đưa vào Sprint Backlog. Xác định mục tiêu của Sprint và lập kế hoạch công việc để đạt được mục tiêu đó.
- + **Thực Hiện Sprint:** Nhóm phát triển làm việc để hoàn thành các công việc trong Sprint Backlog. Daily Scrum được tổ chức hàng ngày để đảm bảo tiến độ và giải quyết các vấn đề phát sinh.
- + **Kết Thúc Sprint:** Sprint Review để trình bày phần sản phẩm hoàn chỉnh. Sprint Retrospective để đánh giá và cải thiện quy trình làm việc cho Sprint tiếp theo

### 2.3. Công nghệ và kiến trúc được sử dụng

#### 2.3.1. EmberJS

EmberJS là một framework mã nguồn mở JavaScript phía client dùng cho phát triển các ứng dụng web và sử dụng mô hình MVC (Model-View-Controller).

Trong EmberJS, route được sử dụng như model, handlebar template giống như view còn controller điều khiển dữ liệu trong model

EmberJS tên thật là SproutCore MVC framework. Được phát triển bởi Yehuda Katz và bắt đầu phát hành vào tháng 12 năm 2011

#### Tại sao nên sử dụng Ember

EmberJS là một framework mã nguồn mở JavaScript theo MIT license.



---

Linh hoạt cho các trang web tốc độ cao, tăng hiệu suất của ứng dụng mà không cần reload trang.

Có các thư viện handlebars templating tương tự như HTML.

Có kích thước nhỏ hơn các thư viện Javascript khác.

Data binding được hỗ trợ đầy đủ. Có sự liên kết giữa 2 thuộc tính, khi một thuộc tính thay đổi mà tác động đến thuộc tính còn lại thì thuộc tính còn lại cũng được update theo.

### **Tính năng**

EmberJS được sử dụng cho việc tạo và bảo trì các ứng dụng web.

View của EmberJS được tạo bởi các handlebars templates nên rất thuận tiện cho việc thiết kế front-end.

Tự động xác định route và controller.

Loại bỏ những khuôn mẫu cũ và thay bằng các kiến trúc ứng dụng tiêu chuẩn.

Ember.js có HTML và CSS là cốt lõi của mô hình phát triển. Routes là tính năng cốt lõi để quản lý URL.

Hỗ trợ chuyên sâu nhiều loại view.

### **2.3.2. Điểm mạnh và điểm yếu của EmberJS**

#### **Về điểm mạnh:**

Kiến trúc MVC rõ ràng: Ember.js tuân thủ nghiêm ngặt mô hình MVC giúp tách biệt logic xử lý và trình bày giao diện. Điều này làm cho code được tổ chức tốt hơn, dễ bảo trì và mở rộng hơn.

Hỗ trợ các Component mạnh: Ember hỗ trợ xây dựng các Component độc lập, có thể tái sử dụng dễ dàng. Điều này giúp giảm thiểu duplicating code, tăng khả năng mở rộng và bảo trì cho ứng dụng.

---

LiveReload nhanh chóng: Ember hỗ trợ tính năng LiveReload giúp lập trình viên có thể thấy ngay sự thay đổi khi code thay đổi, đặc biệt hữu ích trong quá trình phát triển nhanh ban đầu.

Cộng đồng đông đảo và nhiều tài liệu học tập: EmberJs có một cộng đồng rộng lớn trên toàn cầu, luôn cập nhật những kiến thức và phát triển mới nhất. Điều này giúp người dùng mới dễ học và tiếp cận với framework.

### **Về điểm yếu:**

Khởi tạo dự án phức tạp: Quá trình khởi tạo ban đầu của Ember yêu cầu nhiều cấu hình và cài đặt thư viện hơn so với các framework nhẹ hơn.

Tạo ra nhiều callback phức tạp: Khi không quản lý tốt, Ember có thể dẫn đến nhiều callback lồng nhau và logic phức tạp.

Tốc độ xử lý chậm hơn: So với các framework nhẹ hơn như VueJS, tốc độ xử lý của Ember thường chậm hơn do phụ thuộc nhiều mối phụ thuộc.

Dung lượng bundle lớn: Nếu không tối ưu hóa, kích thước bundle của các ứng dụng Ember thường lớn hơn so với các framework khác.

Học quá nhiều thứ cho lần đầu: Để làm quen hoàn toàn với Ember, lập trình viên phải học đủ kiến thức về MVC cũng như các thành phần riêng của Ember.

Cập nhật chậm hơn so với các framework mới: Ember đã ra mắt lâu năm nên tốc độ cập nhật và phát triển các tính năng mới có phần chậm hơn so với các

### **2.3.3. Kết luận**

EmberJS là một framework JavaScript của SproutCore giúp xây dựng ứng dụng web động. Nó sử dụng mô hình MVC để quản lý dữ liệu và tương tác với người dùng, kết hợp với khả năng liên kết dữ liệu giúp tối ưu hóa hiệu suất phát triển. Có thể xây dựng nên một trang web linh hoạt mang lại trải nghiệm tốt cho người dùng và quản lý dữ liệu một cách hiệu quả.

---

## CHƯƠNG 3: XÁC ĐỊNH NHU CẦU

### 3.1. Mô tả bài toán

Bài toán có thể được mô tả như sau: Xây dựng một trang web giới thiệu các sản phẩm sách trong đó sử dụng EmberJs để hiển thị danh sách các sản phẩm sách và cung cấp thông tin chi tiết cho mỗi cuốn sách khi người dùng chọn. Website gồm có 4 trang: Trang chủ dùng để hiển thị sơ lược các cuốn sách để người dùng có thể hình dung được nội dung mà trang web hướng đến. Ở trang Sản phẩm sẽ hiển thị đầy đủ tất cả các cuốn sách người dùng có thể tìm hiểu những thông tin cụ thể hơn bằng cách nhấn vào bất kì một cuốn sách nào đó sẽ được hiển thị thông tin như tên và giá bán, ngoài ra trang sản phẩm cũng có thêm chức năng tìm kiếm giúp người dùng có thể tìm kiếm một cuốn sách nào đó nhanh hơn. Trang đăng ký và đăng nhập giúp người dùng tạo tài khoản và đăng nhập vào web một dễ dàng và nhanh chóng. Cuối cùng là trang giỏ hàng giúp người dùng có thể thêm những sản phẩm sách có thể cập nhật số lượng và đặt hàng.

### 3.2. Đặc tả yêu cầu hệ thống

#### 3.2.1. Yêu cầu chức năng

**Hiển thị danh sách và chi tiết sản phẩm:** Chức năng là hiển thị danh sách các sản phẩm và cho phép người dùng xem chi tiết từng sản phẩm như tên và giá bán.

**Tìm kiếm và lọc sản phẩm:** Chức năng là người dùng có thể tìm kiếm sản phẩm theo tên hoặc danh mục và cũng có thể lọc sản phẩm theo giá, phân loại hoặc các tiêu chí khác.

**Thêm sản phẩm giỏ hàng:** Chức năng thêm sản phẩm là có thể thêm mới sản phẩm vào giỏ hàng, có thể tăng và giảm số lượng sản phẩm và hiển thị hình ảnh của sản phẩm, có thể xóa sản phẩm khỏi giỏ hàng, tính tổng các sản phẩm hiện có trong giỏ hàng và thanh toán.

#### 3.2.2. Yêu cầu phi chức năng

**Giao diện người dùng thân thiện:** Yêu cầu một giao diện người dùng phải dễ sử dụng và thân thiện giúp người dùng dễ dàng tìm kiếm, lọc và xem thông tin chi tiết của sản phẩm.

---

**Hiệu suất và tốc độ:** Đảm bảo trang web hoạt động mượt mà và có thời gian phản hồi nhanh nhằm cung cấp trải nghiệm người dùng tốt nhất có thể.

**Tính năng tương tác mạnh mẽ:** Cung cấp tính năng thực hiện các hành động mà không cần tải lại trang giúp tăng trải nghiệm người dùng.

**Kiểm thử đầy đủ:** Tiến hành kiểm thử chất lượng để đảm bảo tính ổn định và hiệu suất của trang web cũng như xác nhận rằng tất cả các chức năng hoạt động đúng như mong đợi.

### **3.2.3. Yêu cầu kỹ thuật**

Ứng dụng sử dụng ngôn ngữ lập trình JavaScript và framework Ember.js để xây dựng phần back-end.

Sử dụng HTML, CSS và JavaScript để phát triển giao diện người dùng phía front-end.

Sử dụng Jira để quản lý các đề quản lý các Issue và Sprint trong dự án.

Sử dụng Github để lưu trữ và quản lý kho lưu trữ mã nguồn của dự án

Sử dụng Docker để triển khai và chạy ứng dụng trong môi trường độc lập, giúp dễ dàng trong việc phát triển, kiểm thử và triển khai.

---

## CHƯƠNG 4: LẬP KẾ HOẠCH SCRUM

### 4.1. Kế hoạch phát triển phần mềm

#### 4.1.1. Xác định Product Backlog

- + **Giao diện và trải nghiệm người dùng:** thiết kế giao diện cho trang chủ, trang sản phẩm, trang đăng ký và đăng nhập, đảm bảo tính tương thích và tối ưu hóa giao diện cho các thiết bị di động và máy tính bảng.
- + **Chức năng giỏ hàng:** phát triển các chức năng cơ bản của giỏ hàng như thêm sản phẩm, cập nhật số lượng, xóa sản phẩm, và hiển thị modal giỏ hàng.
- + **Tìm kiếm và phân loại sản phẩm:** xây dựng chức năng tìm kiếm sách theo từ khóa và phân loại sách theo danh mục để người dùng dễ dàng tìm kiếm và duyệt sản phẩm.
- + **Triển khai và kiểm thử:** triển khai ứng dụng web lên môi trường Docker để kiểm thử và chuẩn bị cho việc đưa vào sử dụng thực tế.

#### 4.1.2. Xác định Sprint Backlog

ID	Nội dung
SP1	<b>Sprint 1: Xây dựng giao diện trang chủ, đăng ký và đăng nhập:</b> thiết kế giao diện người dùng cơ bản để thiết lập nền tảng cho ứng dụng web.
SP1	<b>Sprint 2: Xây dựng các chức năng và giao diện:</b> Phát triển chức năng giỏ hàng cơ bản và hoàn thiện giao diện của các trang.
SP3	<b>Sprint 3: Xây dựng các chức năng và giao diện khác:</b> Hoàn thiện và nâng cao các chức năng của giỏ hàng, cũng như tối ưu hóa giao diện người dùng.
SP4	<b>Sprint 4: Triển khai và kiểm thử và hoàn thành dự án:</b> Triển khai ứng dụng lên môi trường Docker, tiến hành kiểm thử toàn diện và tối ưu hóa giao diện cho các thiết bị khác nhau.

*Bảng 1. Xác định Sprint Backlog*

### 4.1.3. Phân công và kế hoạch

Sau giai đoạn phân tích có thể đưa các product backlogs vào 4 sprints và lập bảng phân công công việc như sau:

ID	Công việc	Người thực hiện	Story Points	Sprint	Ngày bắt đầu	Ngày kết thúc
1	Thiết kế trang chủ cho web	Nguyễn Đức Thịnh	3	Sprint 1	06/05/2024	12/05/2024
2	Thiết kế giao diện trang sản phẩm	Huỳnh Quang Vinh	3			
3	Thiết kế trang đăng ký và đăng nhập	Nguyễn Hữu Trí	3			
4	Thiết kế modal cho giỏ hàng	Nguyễn Đức Thịnh	3	Sprint 2	13/05/2024	19/05/2024
5	Xây dựng chức năng thêm vào giỏ hàng	Huỳnh Quang Vinh	5			
6	Xây dựng chức năng cập nhật giỏ hàng	Nguyễn Hữu Trí	5			
7	Thêm CSS cho trang chủ	Nguyễn Đức Thịnh	2			
8	Thêm CSS cho trang đăng ký và đăng nhập	Nguyễn Hữu Trí	2			
9	Thêm CSS cho trang sản phẩm	Huỳnh Quang Vinh	2			
10	Thêm CSS cho giỏ hàng	Nguyễn Đức Thịnh	2	Sprint 3	20/05/2024	27/05/2024
11	Xây dựng chức năng phân loại sách	Huỳnh Quang Vinh	4			
12	Xây dựng chức năng thay đổi số lượng trong giỏ hàng	Nguyễn Đức Thịnh	3			
13	Xây dựng chức năng xóa sản phẩm trong giỏ hàng	Nguyễn Hữu Trí	3			

14	HTML chức năng xóa sản phẩm	Huỳnh Quang Vinh	2	Sprint 4	27/05/2024	02/06/2024
15	Thêm CSS chức năng xóa sản phẩm	Nguyễn Đức Thịnh	2			
16	Xây dựng chức năng tìm kiếm sách	Nguyễn Đức Thịnh	4			
17	Reponsive Tablet trang Product	Nguyễn Hữu Trí	3			
18	Reponsive Mobile trang Product	Nguyễn Đức Thịnh	3			
19	Xây dựng chức năng thêm sản phẩm vào giỏ hàng	Nguyễn Hữu Trí	2			
20	Triển khai web lên Docker	Huỳnh Quang Vinh	4			

*Bảng 2. Bảng phân công và kế hoạch*

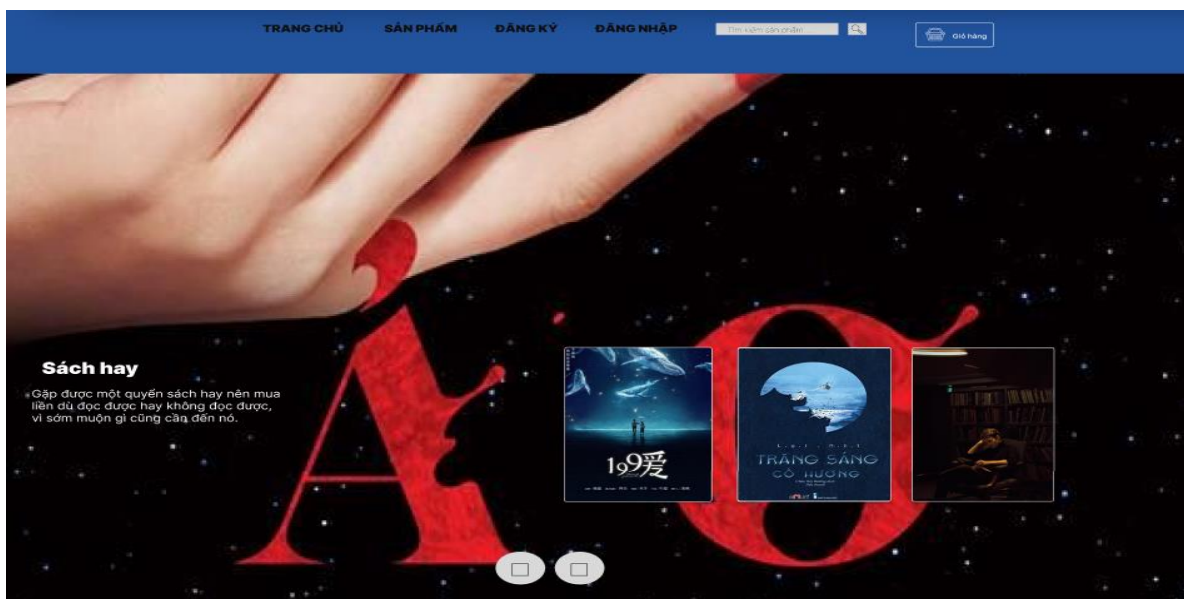
## CHƯƠNG 5: HIỆN THỰC HÓA KẾ HOẠCH

### 5.1. Kết quả của các Sprint

**Sprint 1.** Xây dựng giao diện trang chủ, đăng ký và đăng nhập:

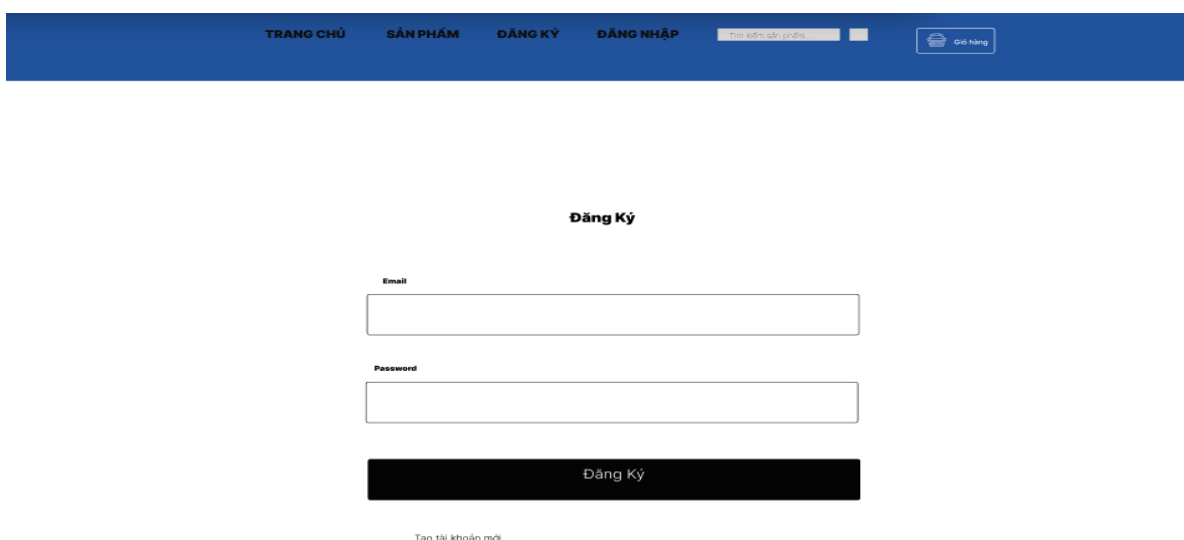
**Kết quả:**

- + Thiết kế giao diện tương ứng bằng Figma cho trang chủ của trang web



Hình 1. Giao diện Figma của trang chủ

- + Thiết kế giao diện tương ứng bằng Figma cho trang đăng ký



Hình 2. Giao diện Figma của trang đăng ký



- + Thiết kế giao diện tương ứng bằng Figma cho trang đăng nhập.

TRANG CHỦSẢN PHẨMĐĂNG KÝĐĂNG NHẬP

Tìm kiếm sản phẩm

Giỏ hàng

Đăng Nhập

Email

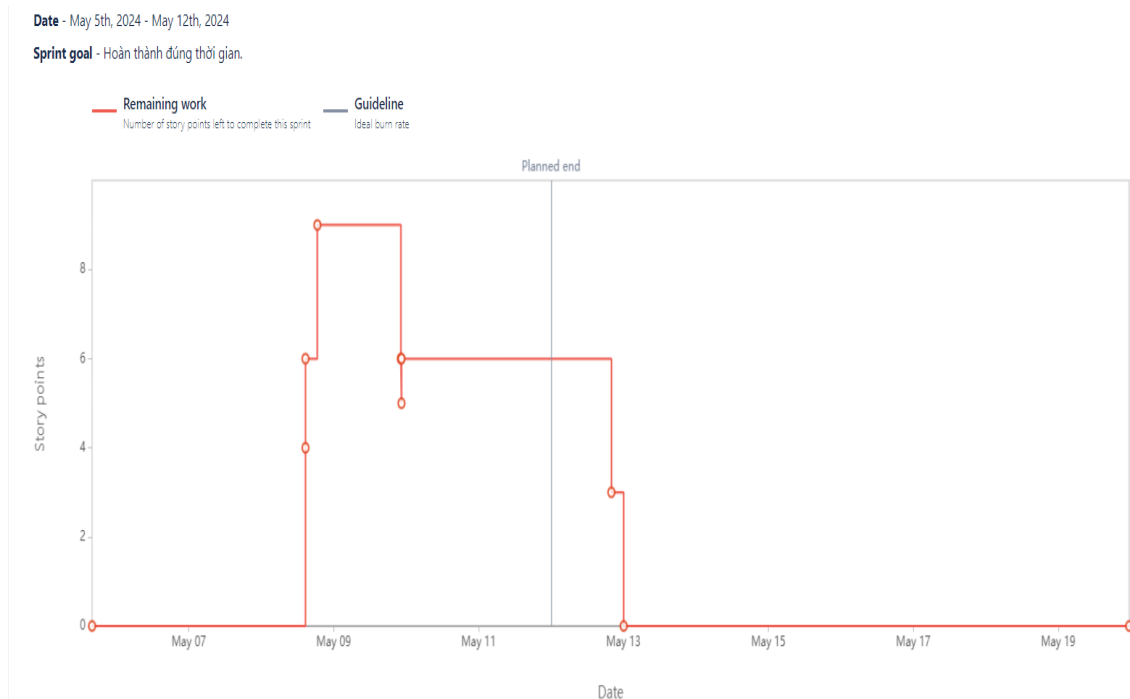
Password

Đăng Nhập

Tạo tài khoản mới

Hình 3. Giao diện Figma của trang đăng nhập

- + Burndown chart cho thấy tiến độ hoàn thành các công việc trong Sprint 1, với tất cả các nhiệm vụ được hoàn thành đúng hạn.

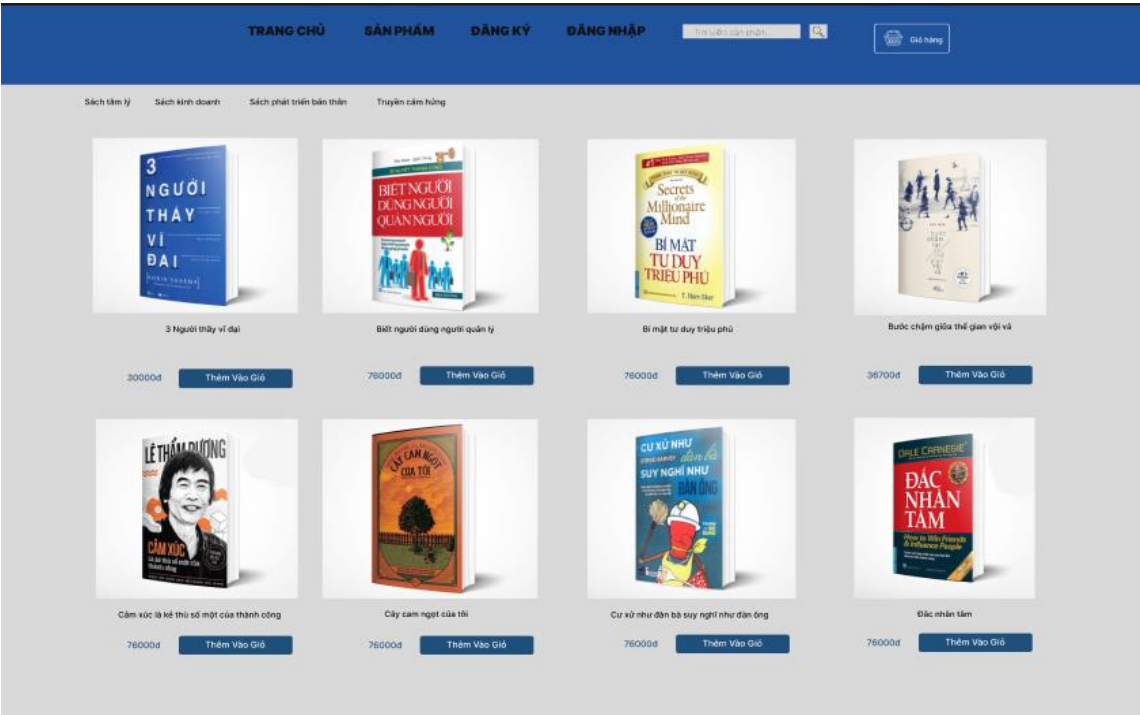


Hình 4. Burndown chart tiến độ của Sprint 1

**Sprint 2. Xây dựng các chức năng và giao diện:**

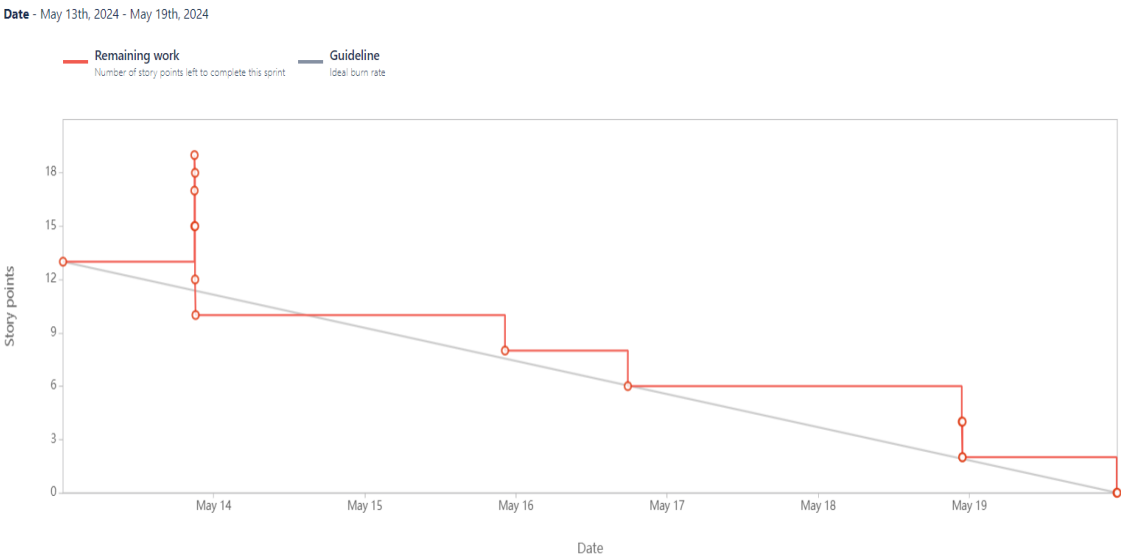
**Kết quả:**

- + Thiết kế giao diện tương ứng bằng Figma cho trang sản phẩm



*Hình 5. Giao diện Figma của trang sản phẩm*

- + Burndown chart cho thấy tiến độ hoàn thành các công việc trong Sprint 2, với tất cả các nhiệm vụ được hoàn thành đúng hạn.

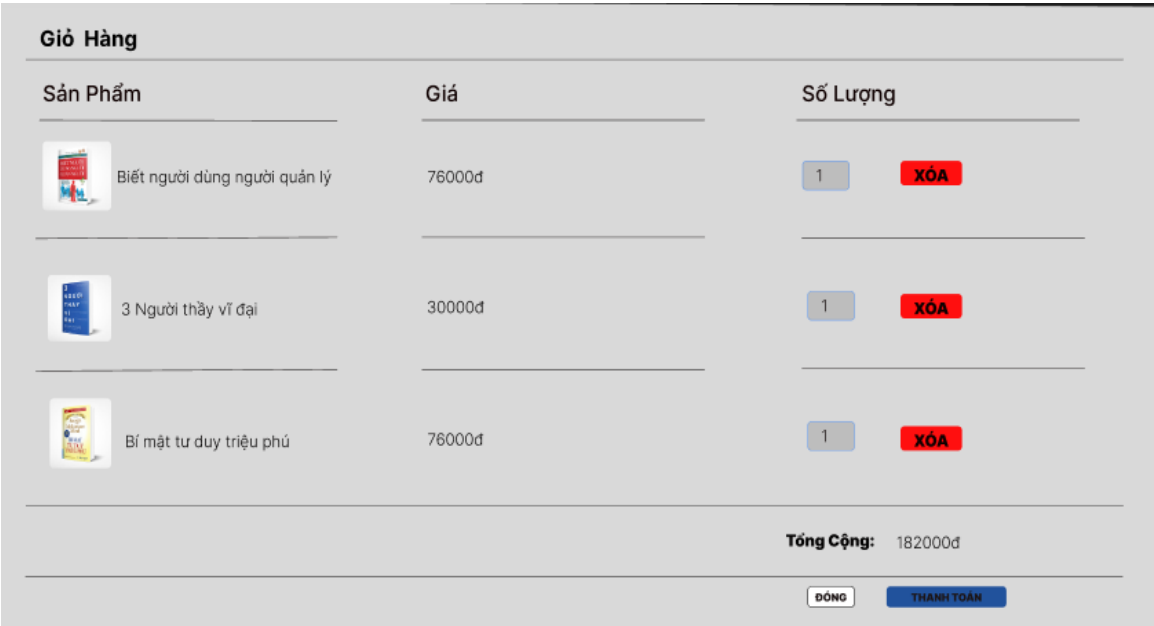


*Hình 6. Burndown chart tiến độ của Sprint 2*

**Sprint 3. Xây dựng các chức năng và giao diện khác:**

**Kết quả:**

- + Thiết kế giao diện tương ứng bằng Figma cho giỏ hàng



*Hình 7. Giao diện Figma của trang giỏ hàng*

- + Burndown chart cho thấy tiến độ hoàn thành các công việc trong Sprint 3, với tất cả các nhiệm vụ được hoàn thành đúng hạn.



*Hình 8. Burndown chart tiến độ của Sprint 3*

---

**Sprint 4.** Triển khai và kiểm thử và hoàn thành dự án:

**Kết quả:**

- + Tiến hành kiểm thử toàn diện.
- + Tối ưu hóa giao diện cho các thiết bị khác nhau.
- + Triển khai ứng dụng lên môi trường Docker.

**Docker file**

```
FROM node:16

WORKDIR /app

COPY package*.json ./

RUN npm install -g ember-cli && npm install

COPY . .

EXPOSE 4200

CMD ["ember", "serve"]
```

**Docker-compose**

```
FROM node:16

WORKDIR /app

COPY package*.json ./

RUN npm install -g ember-cli && npm install

COPY . .

EXPOSE 4200

CMD ["ember", "serve"]
```

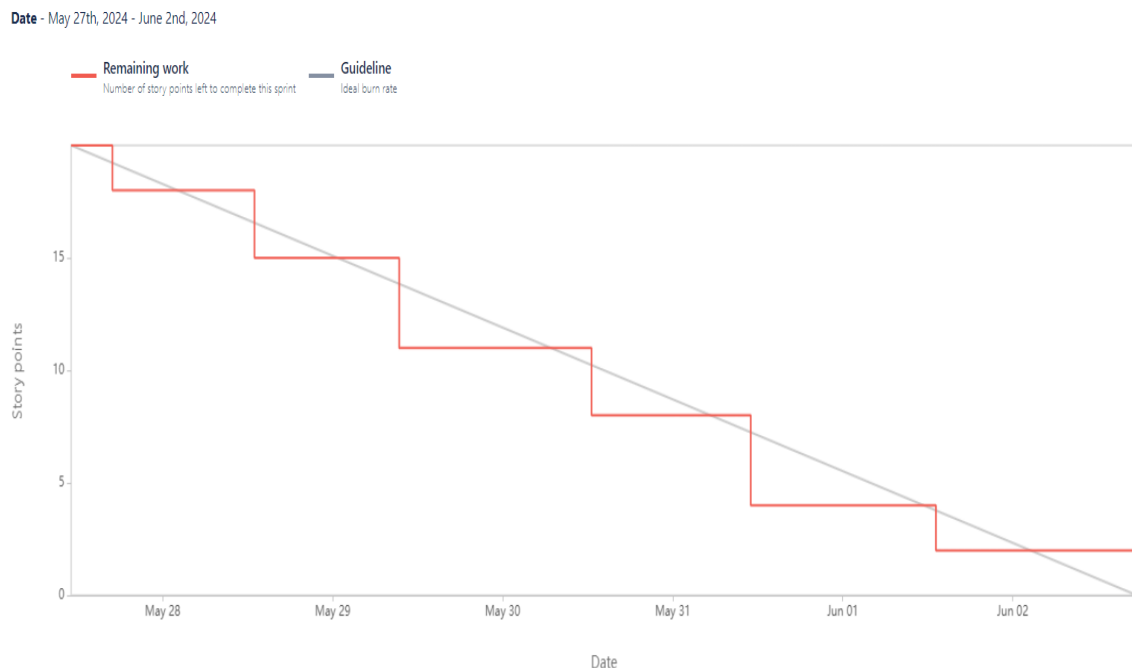
```

PS D:\project-cnpm\csn-da21ttb-nguyenducthinh-webbansach-emberjs\src> docker-compose up
Starting src_ember-app_1 ... done
Attaching to src_ember-app_1
ember-app_1 | Browserslist: caniuse-lite is outdated. Please run:
ember-app_1 |   npx update-browserslist-db@latest
ember-app_1 |   Why you should do it regularly: https://github.com/browserslist/update-db#readme
ember-app_1 | - building...
ember-app_1 | Build successful (10161ms) - Serving on http://localhost:4200/
ember-app_1 | Slowest Nodes (totalTime >= 5%) | Total (avg)
ember-app_1 | +-
ember-app_1 | ember-auto-import-webpack (1) | 3058ms
ember-app_1 | Babel: @ember/test-helpers (1) | 2018ms
ember-app_1 |

```

Hình 9. Kết quả triển khai ứng dụng lên docker

- + Burndown chart cho thấy tiến độ hoàn thành các công việc trong Sprint 4, với tất cả các nhiệm vụ được hoàn thành đúng hạn.



Hình 10. Burndown chart tiến độ của Sprint 4

---

## CHƯƠNG 6: KẾT LUẬN

### 6.1. Kết luận:

#### **EmberJS:**

Phân loại sách: Cho phép phân chia sản phẩm ra nhiều nhóm phù hợp giúp người dùng dễ dàng tìm kiếm sách cần mua.

Tìm kiếm sách: Cung cấp ô tìm kiếm để khách hàng nhập tên hoặc từ khóa liên quan để nhanh chóng tìm thấy sách cần mua.

Đăng ký/Đăng nhập: Cho phép khách hàng tạo tài khoản cá nhân để quản lý thông tin, lịch sử đơn hàng, tiện thanh toán sau này.

Giỏ hàng: Lưu trữ tạm thời các sách đã chọn mua, cập nhật số lượng, tổng tiền trước khi thanh toán.

Những chức năng này đem đến cho người dùng trải nghiệm mua sách thuận tiện, nhanh chóng từ tìm kiếm, lựa chọn đến thanh toán, góp phần mang đến cho người dùng website.

#### **Jira:**

Jira là một công cụ quản lý dự án mạnh mẽ và toàn diện, cung cấp nhiều tính năng và công cụ hữu ích để giúp các nhóm và tổ chức quản lý công việc, theo dõi vấn đề, hợp tác và cải thiện quy trình làm việc. Với khả năng tùy chỉnh cao, hỗ trợ nhiều loại phương pháp phát triển phần mềm và linh hoạt trong việc mở rộng chức năng, ưu hóa quá trình làm việc.

#### **Hướng phát triển:**

Hệ thống đánh giá, bình luận sách: Người dùng đánh giá sách, đọc bình luận của người khác để có cái nhìn toàn diện hơn.

Quản lý danh sách mong muốn: Lưu trữ danh sách sách mong muốn mua sau, khi có ưu đãi sẽ thông báo.

So sánh sản phẩm: Chọn 2-3 sách cùng loại để so sánh thông số kỹ thuật, đánh giá người dùng.

Đặt hàng và thanh toán: Các tùy chọn thanh toán như thẻ tín dụng, chuyển khoản ngân hàng...

---

Hệ thống ưu đãi, khuyến mãi: Mã giảm giá, tặng quà khi đạt mức đơn hàng.

Thông tin sự kiện: Thông báo chương trình khuyến mãi, triển lãm sách sắp diễn ra.

Hệ thống tích điểm khách hàng thân thiết.

---

## DANH MỤC TÀI LIỆU THAM KHẢO

[1] W3School “W3.CSS Tutorial” W3Schools Online WebTutorials ,15/11/2023

URL: [http://www.en.w3academy.com/emberjs/emberjs\\_model.htm](http://www.en.w3academy.com/emberjs/emberjs_model.htm)

[2] Đoàn Phước Miên, Phạm Thị Trúc Mai (2013), Tài liệu giảng dạy môn Thiết kế web, Bộ môn Công nghệ Thông tin, Khoa Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh.

[3] Base.vn Blog, "Agile là gì? Tìm hiểu về phương pháp Agile trong quản lý dự án,"

URL: [https://base.vn/blog/agile-la-gi/#1\\_Tong\\_quan\\_ve\\_Agile](https://base.vn/blog/agile-la-gi/#1_Tong_quan_ve_Agile).

[4] Viblo, "Các thuật ngữ trong Scrum chính xác là như thế nào,"

URL: <https://viblo.asia/p/cac-thuat-ngu-trong-scrum-chinh-xac-la-nhu-the-nao-jvEla0y6Zkw>.