

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA CÔNG NGHỆ PHẦN MỀM



ĐỒ ÁN MÔN HỌC
CÔNG NGHỆ WEB VÀ ỨNG DỤNG
THIẾT KẾ WEBSITE ĐỌC TRUYỆN

Giảng viên hướng dẫn : ThS. Trần Thị Hồng Yến

Sinh viên thực hiện : Phan Thị Linh

Mã sinh viên : 20520615

Lớp : SE347.O12

Tp HCM, tháng 01 năm 2024



NHIỆM VỤ ĐỒ ÁN MÔN HỌC

Họ và tên SVs: **Phan Thị Linh**..... MSSV: **20520615**.....

Lớp: **SE347.O12**

Tên đề tài: **THIẾT KẾ WEBSITE ĐỌC TRUYỆN**

Giảng viên giảng dạy: **ThS. Trần Thị Hồng Yến**.....

Thời gian thực hiện: **09/2023 đến 01/2024**

Nhiệm vụ đồ án môn học:

1. Xác định yêu cầu, thu thập thông tin và các dữ liệu liên quan cho việc thiết kế website.
2. Phân tích thiết kế và phác thảo giao diện các nhóm trang web.
3. Xử lý hình ảnh, tối ưu hóa ảnh cho web.
4. Thiết kế giao diện 1 Template.
5. Áp dụng 1 Template đã thiết kế để xây dựng website có 7 trang web.
6. Nộp file nén (*.rar) lưu sản phẩm đề tài bao gồm:
 - File báo cáo word (*.docx)
 - File thuyết trình (*.pptx)
 - Thư mục siteroot bao gồm: các file định dạng (*.css), template (*.dwt), các trang web (*.html), các file lập trình xử lý scripts nếu có (*.js), thư mục hình ảnh, âm thanh, videos, ...)

Tp.HCM, ngày 19 tháng 01 năm 2024

GIẢNG VIÊN GIẢNG DẠY

(Ký và ghi rõ họ tên)

ThS. Trần Thị Hồng Yến

LỜI CẢM ƠN

Trong không khí náo nức của những ngày kết thúc học kỳ, em, sinh viên lớp Công nghệ Web và Ứng dụng, xin phép dành lời cảm ơn sâu sắc và lòng biết ơn chân thành nhất đến cô Trần Thị Hồng Yến, người đã không chỉ là giảng viên dạy học, mà còn là nguồn cảm hứng và hỗ trợ tuyệt vời trong suốt quá trình học tập và thực hiện đồ án môn học tại Trường Đại học Công nghệ Thông tin - Đại học Quốc gia TP.HCM.

Em xin được tri ân cô đã luôn nhiệt tình truyền đạt kiến thức quý giá, chia sẻ kinh nghiệm thực tế, và hướng dẫn em một cách tận tâm trong suốt quá trình thực hiện đề tài “Xây dựng website đọc truyện”. Không chỉ giúp em nắm bắt được những kiến thức cốt lõi, cô còn hướng dẫn em cách ứng dụng chúng vào thực tế, từ việc thiết kế giao diện, quản lý sản phẩm, đến xử lý thanh toán và tối ưu hóa trải nghiệm người dùng, giúp đề tài của em không chỉ đáp ứng nhu cầu thực tế mà còn mang tính ứng dụng cao.

Bên cạnh đó, em cũng nhận thức rõ rằng, dù đã cố gắng hết sức nhưng không thể tránh khỏi những thiếu sót trong quá trình thực hiện. Em rất mong nhận được sự phản hồi và góp ý từ cô, để từ đó rút kinh nghiệm, hoàn thiện bản thân và nâng cao chất lượng công việc trong tương lai.

Xin gửi lời cảm ơn chân thành nhất đến cô Trần Thị Hồng Yến. Cô không chỉ là người giáo viên giảng dạy, mà còn là người hướng dẫn, người bạn đồng hành cùng em trong suốt quá trình học tập và phát triển. Những bài học và kinh nghiệm mà cô đã truyền đạt sẽ luôn là hành trang quý giá đối với em trên con đường tương lai.

Sinh viên thực hiện

PHAN THỊ LINH

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Tp.HCM, ngày ... tháng 01 năm 2024

GVHD

ThS. Trần Thị Hồng Yên

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Giới thiệu website đọc truyện:	1
1.2. Lý do chọn thiết kế Website:	2
1.3. Sơ đồ sitemap của Website:	3
1.4. Công nghệ sử dụng:	4
1.5. Môi trường thiết kế:	4
1.6. Công cụ hỗ trợ:.....	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	5
2.1. Ngôn ngữ HTML:	5
2.1.1. Tổng quan về HTML:	5
2.1.2. Cấu trúc cơ bản của một trang HTML:.....	5
2.1.3. Ưu điểm của HTML.....	5
2.1.4. Nhược điểm của HTML:.....	6
2.2. Định dạng Website với CSS:	7
2.2.1. Tổng quan về CSS.....	7
2.2.2. Cơ bản về cú pháp và cách sử dụng.....	7
2.2.3. Tính năng và ưu điểm của CSS:.....	8
2.2.4. Nhược điểm và hạn chế.....	8
2.3. ReactHooks	8
2.3.1. Mục đích và lợi ích	9
2.3.2. Các Hook cơ bản.....	9
2.3.3. Lưu ý khi sử dụng	10

2.4. Django Framework	10
2.4.1. Thành phần.....	11
2.4.2. Các phần dưới đây là các phần chính của một ứng dụng Django.....	12
2.4.3. Nhược điểm.....	13
2.5. SQLite.....	14
2.5.1. Tính năng	14
2.5.2. Ứng dụng chủ yếu	15
2.5.3. Điểm nổi bật.....	15
2.5.4. Nhược điểm.....	19
CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ	20
3.1. Thiết kế hệ thống.....	20
3.1.1. Sơ đồ usecase	20
3.1.2. Danh sách Actor.....	21
3.1.3. Danh sách Usecase.....	21
3.2. Thiết kế Cơ sở dữ liệu.....	22
3.2.1. Thiết kế dữ liệu	22
CHƯƠNG 4: XÂY DỰNG WEBSITE	23
4.1. Trang Home:	23
4.1.1. Giao diện:	23
4.1.2. Mô tả:	24
4.1.3. Kỹ thuật thiết kế:.....	24
4.1.4. Code thiết kế:	24
4.2. Trang Category:	27
4.2.1. Giao diện:	27

4.2.2. Mô tả:	27
4.2.3. Kỹ thuật thiết kế:.....	27
4.2.4. Code thiết kế:	28
4.3. Trang About:	32
4.3.1. Giao diện:	32
4.3.2. Mô tả:	33
4.3.3. Kỹ thuật thiết kế:.....	33
4.3.4. Code thiết kế:	33
4.4. Trang Story Detail:.....	38
4.4.1. Giao diện:	38
4.4.2. Mô tả:	39
4.4.3. Kỹ thuật thiết kế:.....	39
4.4.4. Code thiết kế:	39
4.5. Trang Chapter Detail:.....	42
4.5.1. Giao diện:	42
4.5.2. Mô tả:	43
4.5.3. Kỹ thuật thiết kế:.....	43
4.5.4. Code thiết kế:	43
4.6. Trang Profile:	45
4.6.1. Giao diện:	45
4.6.2. Mô tả:	46
4.6.3. Kỹ thuật thiết kế:.....	46
4.6.4. Code thiết kế:	46
4.7. Trang thêm Story:	51

4.7.1. Giao diện:	51
4.7.2. Mô tả:	52
4.7.3. Kỹ thuật thiết kế:	52
4.7.4. Code thiết kế:	52
4.8. Trang Đăng nhập, đăng ký:	56
4.8.1. Giao diện:	56
4.8.2. Mô tả:	56
4.8.3. Kỹ thuật thiết kế:	57
4.8.4. Code thiết kế:	57
CHƯƠNG 5: KẾT LUẬN	59
5.1. Ưu điểm của đồ án:	59
5.2. Hạn chế của đồ án:	59
5.3. Hướng phát triển của đồ án:	59
TÀI LIỆU THAM KHẢO.....	61

MỤC LỤC HÌNH ẢNH

Hình 1-1 Logo website.....	1
Hình 1-2 Sitemap của website	3
Hình 2-1 Logo HTML.....	5
Hình 2-2 Logo CSS	7
Hình 2-3 Ví dụ CSS	7
Hình 2-4 Logo ReactHooks	8
Hình 2-5 Ví dụ sử dụng ReactHooks	9
Hình 2-6 Logo Django	10
Hình 2-7 Các bước hoạt động của Django.....	11
Hình 2-8 Code file urls.py.....	12
Hình 2-9 Code file views.py	13
Hình 2-10 Logo SQL Lite	14
Hình 2-11 Code xóa cột trong SQLite.....	16
Hình 2-12 Code kích hoạt khoá ngoại SQLite	17
Hình 2-13 Code select Table SQLite.....	17
Hình 3-1 Sơ đồ Usecase.....	20
Hình 3-2 Sơ đồ quan hệ CSDL	22
Hình 4-1 Giao diện Trang Home	23
Hình 4-2 Giao diện Trang Category	27
Hình 4-3 Giao diện Trang About	32
Hình 4-4 Giao diện Trang Story Detail.....	38
Hình 4-5 Giao diện Chapter Detail	42

Hình 4-6 Giao diện Trang Profile	45
Hình 4-7 Giao diện Trang thsêm Story	51
Hình 4-8 Giao diện Trang Login.....	56
Hình 4-9 Giao diện Trang đăng ký	56

MỤC LỤC BẢNG BIỂU

Bảng 2-1 Cú pháp SQLite	18
Bảng 3-1 Danh sách Actor	21
Bảng 3-2 Danh sách Usecase	21

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu website đọc truyện:

- Tên website: WEBSITE ĐỌC TRUYỆN
- Logo:



Hình 1-1 Logo website

Website đọc truyện là trang web cho phép người dùng có thể tìm và đọc những cuốn truyện mình yêu thích. Ngoài ra website cũng cho phép người dùng có thể phát hành truyện và chia sẻ cho người khác cùng đọc. Website có những chức năng chính sau:

- Trang Chủ: Hiển thị danh sách các truyện mới nhất.
- Thẻ loại: Các truyện được gom nhóm theo từng thẻ loại riêng biệt.
- Trang: hiển thị thông tin website truyện.
- Trang cá nhân (Cần phải đăng nhập thông qua tính năng đăng nhập/đăng ký):
 - o Xem danh sách truyện đã lưu, đã đọc, truyện của user.
 - o Thêm truyện mới, sửa truyện đã có.

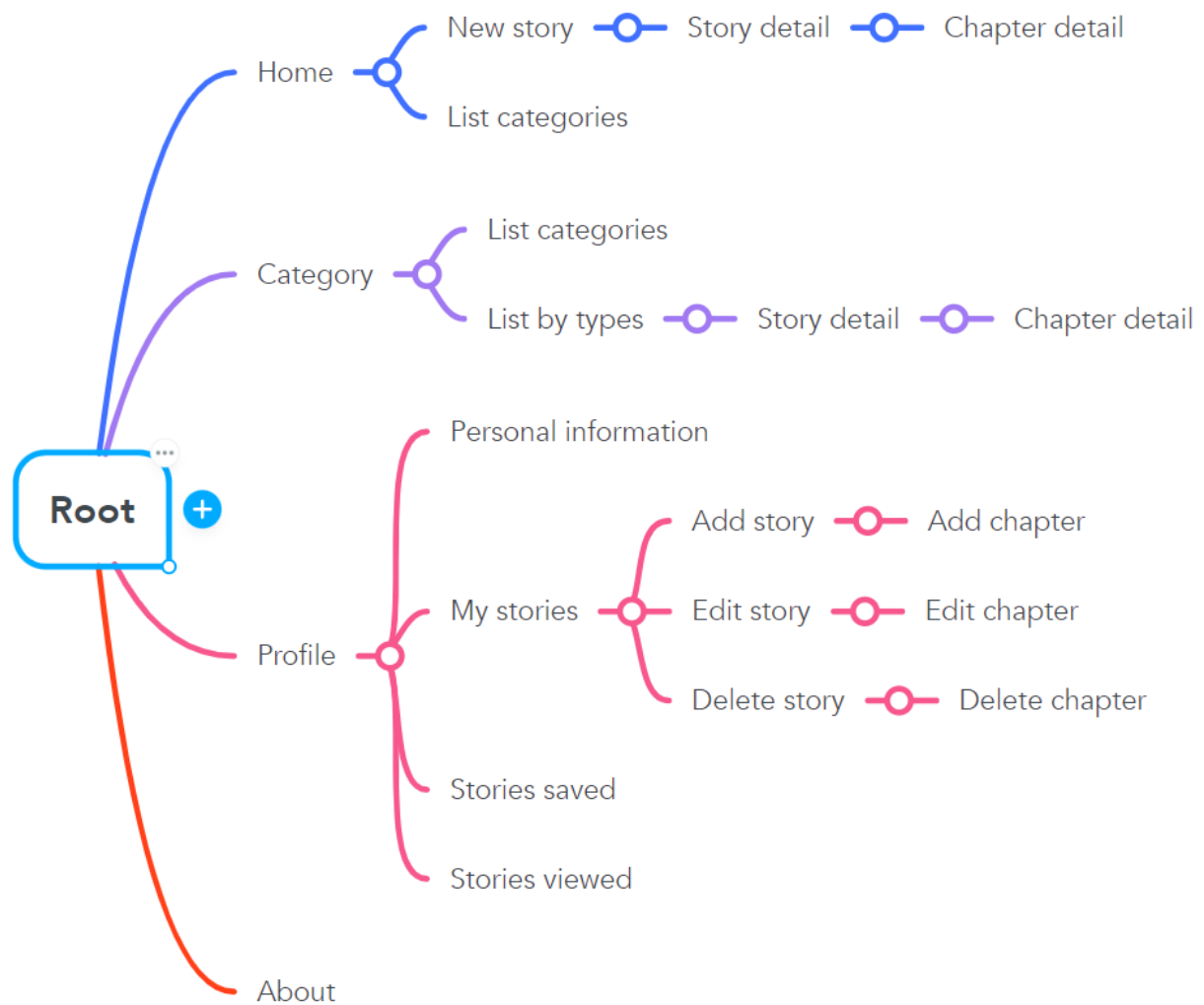
1.2. Lý do chọn thiết kế Website:

Truyện, với sức mạnh kỳ diệu của nó trong việc kể chuyện và truyền đạt văn hóa, luôn là một phần quan trọng của cuộc sống và tâm hồn mỗi người. Việc đọc và chia sẻ truyện không chỉ là niềm vui, mà còn là cách mà chúng ta kết nối, học hỏi, và phát triển tư duy. Trong thời đại công nghệ số ngày nay, đọc truyện trực tuyến đã trở thành một xu hướng phổ biến, không chỉ mang lại sự tiện lợi mà còn mở ra một thế giới văn học rộng lớn ngay tại ngón tay của chúng ta.

Trong bối cảnh số hóa hiện nay, công nghệ thông tin đang phát triển không ngừng và ngày càng trở thành một phần không thể tách rời của cuộc sống hàng ngày. Việc áp dụng công nghệ thông tin vào lĩnh vực văn hóa - nghệ thuật, đặc biệt là trong việc đọc và chia sẻ truyện, đã mở ra những cơ hội mới để tiếp cận với nguồn tri thức phong phú, giảm thiểu rào cản về không gian và thời gian, nâng cao trải nghiệm đọc sách và tạo điều kiện thuận lợi cho việc chia sẻ sáng tạo. Sự kết hợp giữa dữ liệu, trí tuệ nhân tạo và tích hợp trực tuyến có thể biến đổi cách thức chúng ta đọc và chia sẻ truyện, mang đến một trải nghiệm hoàn toàn mới mẻ và độc đáo.

Việc tạo ra một trang web đọc truyện trực tuyến sẽ không chỉ giúp những người yêu thích sách tăng cường tương tác với nhau, mà còn hoạt động liên tục 24/7, vượt qua mọi giới hạn về mặt thời gian và không gian. Điều này tạo điều kiện cho người đọc tiếp cận với một lượng lớn truyện từ khắp nơi trên thế giới, bất kể họ đang ở đâu và vào thời gian nào. Trang web này không chỉ là nơi để đọc truyện, mà còn là nơi để chia sẻ, sáng tạo, và tương tác với cộng đồng yêu sách, góp phần làm giàu thêm nền văn hóa đọc và sáng tạo trong xã hội hiện đại.

1.3. Sơ đồ sitemap của Website:



Hình 1-2 Sitemap của website

1.4. Công nghệ sử dụng:

- **Django Rest Framework:** Được sử dụng để xây dựng các API hiệu quả và linh hoạt cho các ứng dụng web^[1].
- **NextJS:** Framework dựa trên React, nhằm mục đích tạo ra các ứng dụng web hiện đại với khả năng server-side rendering và tối ưu hóa hiệu suất^[2].
- **React Hooks:** Tính năng trong React giúp sử dụng state và các chức năng React khác mà không cần dùng đến class components, làm cho việc viết mã trở nên gọn gàng và linh hoạt hơn^[3].
- **SQLite:** Một hệ quản trị cơ sở dữ liệu nhẹ, được sử dụng để lưu trữ dữ liệu một cách hiệu quả trong các ứng dụng di động, desktop và web^[4].

1.5. Môi trường thiết kế:

- **Figma:** Mục đích: Figma là một công cụ thiết kế đồ họa dựa trên nền tảng web và đám mây, chủ yếu được sử dụng để thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) cho các ứng dụng web và di động. Nó cung cấp tính năng hợp tác trực tuyến cho phép các nhóm thiết kế làm việc cùng nhau trong thời gian thực^[5].

1.6. Công cụ hỗ trợ:

- **Photoshop:** là một phần mềm chỉnh sửa ảnh và đồ họa raster chuyên nghiệp. Nó được sử dụng rộng rãi trong việc tạo và chỉnh sửa ảnh, thiết kế đồ họa, tạo ảnh ghép, và thậm chí là trong lĩnh vực nghệ thuật kỹ thuật số và animation. Photoshop cung cấp một loạt các công cụ và chức năng cho phép người dùng tạo ra hình ảnh chất lượng cao với sự chính xác và sáng tạo^[6].
- **DALLE:** là một mô hình trí tuệ nhân tạo phát triển bởi OpenAI, được thiết kế để tạo ra các hình ảnh từ mô tả văn bản. Nó cho phép người dùng tạo ra hình ảnh độc đáo và sáng tạo chỉ từ các mô tả bằng ngôn ngữ tự nhiên, mở ra khả năng mới cho việc tạo ra nội dung đồ họa, nghệ thuật, và thiết kế mà không cần đến kỹ năng vẽ hoặc thiết kế truyền thống^[7].

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Ngôn ngữ HTML:

2.1.1. Tổng quan về HTML:



Hình 2-1 Logo HTML

HTML (Hypertext Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để mô tả và xác định cấu trúc của nội dung trên web. HTML là một phần cơ bản của công nghệ web và hoạt động cùng với CSS (Cascading Style Sheets) và JavaScript để tạo nên những trang web đa dạng và tương tác.

HTML được cấu trúc dựa trên các thẻ (tags), với mỗi thẻ biểu diễn một phần tử cụ thể trên trang web. Ví dụ: thẻ `<h1>` đại diện cho tiêu đề cấp 1, trong khi thẻ `<p>` được sử dụng cho đoạn văn.

2.1.2. Cấu trúc cơ bản của một trang HTML:

Một trang web HTML cơ bản thường bao gồm các phần sau:

- `<!DOCTYPE html>`: Khai báo phiên bản HTML.
- `<html>`: Thẻ mở của trang.
- `<head>`: Chứa thông tin meta, tiêu đề và liên kết đến tài nguyên như CSS.
- `<body>`: Phần chính của trang, chứa nội dung hiển thị cho người dùng.

Bên trong thẻ `<body>`, chúng ta có thể sử dụng nhiều thẻ khác như `<h1>`, `<p>`, `<a>`, v.v. để tạo nên cấu trúc và nội dung của trang web.

2.1.3. Ưu điểm của HTML

- HTML có cú pháp rõ ràng và đơn giản, dễ dàng cho người mới học và bắt đầu sử dụng.

- HTML được hỗ trợ trên hầu hết các trình duyệt và thiết bị, từ máy tính để bàn đến điện thoại di động.
- HTML dễ dàng kết hợp với CSS để thiết kế giao diện và JavaScript để thêm tính năng tương tác.
- HTML là ngôn ngữ chuẩn của web, được sử dụng rộng rãi trên toàn thế giới.
- Với việc giới thiệu các phiên bản mới như HTML5, HTML không ngừng phát triển và thêm nhiều tính năng mạnh mẽ.
- HTML hỗ trợ nhiều loại nội dung, từ văn bản, hình ảnh đến video và âm thanh.

2.1.4. Nhược điểm của HTML:

- Mặc dù HTML có thể hiển thị nội dung, nhưng nó không thể thực hiện các chức năng phức tạp mà không có sự hỗ trợ từ ngôn ngữ lập trình khác như JavaScript.
- Có những thẻ và thuộc tính chỉ hoạt động trên một số phiên bản HTML hoặc trình duyệt cụ thể, gây ra vấn đề về tương thích.
- Nếu muốn cập nhật nội dung trên trang web HTML tĩnh, bạn cần phải chỉnh sửa và tải lên lại mã nguồn.
- Trang web chỉ sử dụng HTML tĩnh có thể khó khăn hơn trong việc tối ưu hóa cho công cụ tìm kiếm so với các trang web động.
- Đối với các ứng dụng web phức tạp, việc chỉ sử dụng HTML có thể không đủ và cần sự hỗ trợ từ các công nghệ khác.

2.2. Định dạng Website với CSS:

2.2.1. Tổng quan về CSS



Hình 2-2 Logo CSS

CSS (Cascading Style Sheets) là một ngôn ngữ định dạng giúp kiểm soát và tạo ra trải nghiệm trực quan cho nội dung web. Được giới thiệu sau HTML, CSS là cách mà các nhà phát triển web và thiết kế viên đưa ra để tách rời nội dung và trình bày, giúp trang web dễ dàng bảo trì và tăng khả năng tiếp cận.

2.2.2. Cơ bản về cú pháp và cách sử dụng

CSS được cấu trúc dựa trên quy tắc "selector" và "declaration block". Một selector xác định phần tử HTML nào sẽ được định dạng, và declaration block chứa một hoặc nhiều khai báo định dạng cho selector đó. Mỗi khai báo gồm một thuộc tính và một giá trị.

- Ví dụ:

```
CSS

p {
  color: red;
  font-size: 16px;
}
```

Hình 2-3 Ví dụ CSS

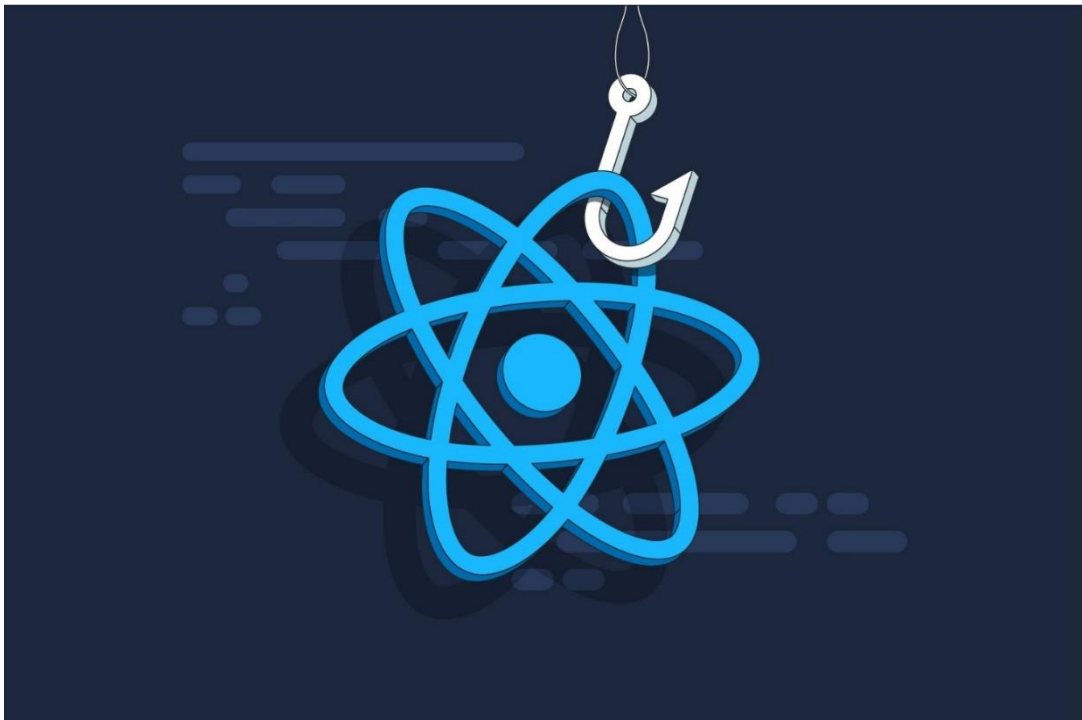
2.2.3. Tính năng và ưu điểm của CSS:

- CSS cho phép tạo ra trang web có nội dung và thiết kế tách biệt, giúp dễ dàng quản lý và cập nhật.
- Một tệp CSS có thể được áp dụng cho nhiều trang HTML, giúp đảm bảo tính nhất quán và giảm thiểu công sức lập trình.
- Với CSS, bạn có thể tạo ra các trang web đáp ứng phù hợp với mọi kích cỡ màn hình và thiết bị.
- CSS3 mang lại nhiều hiệu ứng và chuyển động mượt mà, giúp tăng cường trải nghiệm người dùng.

2.2.4. Nhược điểm và hạn chế

- Tương thích trình duyệt: Một số tính năng CSS có thể không hoạt động đồng nhất trên tất cả các trình duyệt.
- Phức tạp: Dù CSS cơ bản khá dễ hiểu, nhưng để trở nên chuyên nghiệp và tạo ra các trang web phức tạp, bạn cần phải hiểu rõ về các khái niệm như độ ưu tiên, kế thừa, và bố cục

2.3. ReactHooks



Hình 2-4 Logo ReactHooks

React Hooks là một tính năng mới được giới thiệu trong phiên bản React 16.8, mang lại một cách tiếp cận hoàn toàn mới trong việc sử dụng trạng thái và các tính năng khác của React mà không cần tạo ra một class component.

Để sử dụng React Hooks, chỉ cần import chúng từ thư viện React. Ví dụ:

```
javascript

import React, { useState, useEffect } from 'react';
```

Hình 2-5 Ví dụ sử dụng ReactHooks

2.3.1. Mục đích và lợi ích

Trước khi có React Hooks, việc sử dụng trạng thái và vòng đời component trong các function component không thể thực hiện được. Hooks giải quyết vấn đề này, giúp các lập trình viên có thể sử dụng tất cả các tính năng của React trong function component.

Lợi ích:

- Hooks giúp viết code trở nên gọn gàng và dễ đọc hơn.
- Hooks cho phép tái sử dụng logic trạng thái mà không làm thay đổi cấu trúc component.
- Có thể kết hợp nhiều hook trong một single component.
- Với sự ra mắt của Hooks, có rất nhiều thư viện và công cụ được phát triển dựa trên nó.

2.3.2. Các Hook cơ bản

Một số hook cơ bản và phổ biến trong React bao gồm:

- `useState`: Cho phép bạn thêm trạng thái vào function component.
- `useEffect`: Dùng để thực hiện side effects (như fetch dữ liệu, thao tác với DOM) trong function component.
- `useContext`: Giúp truy cập các giá trị của context cho component của bạn, giúp truyền dữ liệu một cách dễ dàng giữa các component.
- Ngoài ra, React cũng cung cấp nhiều hook khác như `useReducer`, `useCallback`, `useMemo`, và bạn cũng có thể tạo ra custom hooks riêng.

2.3.3. Lưu ý khi sử dụng

- Hooks chỉ nên được gọi ở mức top-level của function component. Không nên gọi hooks bên trong vòng lặp, điều kiện hoặc function con.
- Hooks chỉ nên được sử dụng trong function component hoặc custom hooks. Không sử dụng trong class component hoặc các function thông thường.

2.4. Django Framework



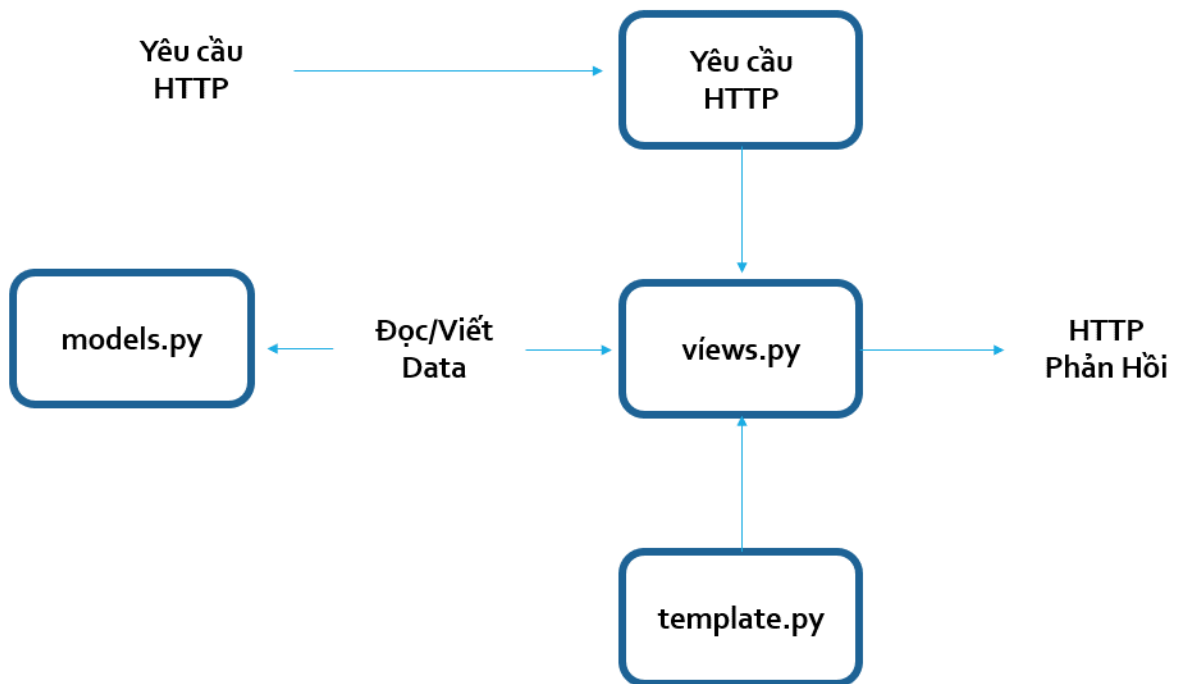
Hình 2-6 Logo Django

Django là một framework phát triển ứng dụng web mạnh mẽ và linh hoạt, được viết bằng ngôn ngữ lập trình Python. Được giới thiệu lần đầu vào năm 2005, Django tuân theo mô hình MVT (Model-View-Template) và đã trở thành một trong những framework web phổ biến nhất trong cộng đồng Python.

Django được thiết kế với mục tiêu giúp các lập trình viên xây dựng ứng dụng web nhanh chóng, dễ dàng và an toàn. Cùng với một loạt các công cụ và tiện ích được tích hợp, Django giúp giảm thiểu các công việc lặp đi lặp lại, cho phép lập trình viên tập trung vào việc phát triển các tính năng độc đáo của ứng dụng.

2.4.1. Thành phần

- Trong một trang web truyền tải dữ liệu truyền thống, một ứng dụng web chờ các yêu cầu HTTP từ trình duyệt web (hoặc ứng dụng khách khác). Khi nhận được yêu cầu, ứng dụng sẽ xác định những gì cần thiết dựa trên URL, có thể là thông tin trong dữ liệu POST hoặc dữ liệu GET. Tùy thuộc vào những gì được yêu cầu, nó có thể đọc hay ghi thông tin từ cơ sở dữ liệu hoặc thực hiện các tác vụ khác cần thiết để đáp ứng yêu cầu. Sau đó, ứng dụng sẽ gửi phản hồi về cho trình duyệt web. Thông thường, trang HTML sẽ được tự động tạo để trình duyệt hiển thị bằng cách chèn dữ liệu đã truy xuất vào trình giữ chỗ trong mẫu HTML.
- Các ứng dụng web Django thường nhóm mã xử lý cho từng bước này vào các tệp riêng biệt:



Hình 2-7 Các bước hoạt động của Django

- URLs: Mặc dù có thể xử lý các yêu cầu từ mọi URL thông qua một hàm duy nhất, nhưng việc viết một hàm xem (view) riêng để xử lý mỗi tài nguyên là cách quản lý tối ưu hơn. Một bộ URL được sử dụng để định hướng các yêu cầu HTTP đến chức năng view thích hợp dựa trên URL yêu cầu. Bộ URL cũng có thể phù hợp với một chuỗi ký tự hoặc số cụ thể có trong URL và chuyển chúng đến một view dựng sẵn dữ liệu.

- View: Một hàm xem (view) là một hàm xử lý yêu cầu. View nhận yêu cầu HTTP và trả về phản hồi HTTP. Hàm view truy cập dữ liệu cần thiết để đáp ứng các yêu cầu thông qua các mô hình (model) và cung cấp dữ liệu trả về cho các mẫu (template).
- Model: Mô hình (model) là các class trong Python dùng để tạo các cấu trúc dữ liệu của ứng dụng và cung cấp các phương pháp để quản lý (thêm, sửa, xóa) và truy cập vào cơ sở dữ liệu.
- Template: Mẫu (template) là một file giúp định dạng cấu trúc hoặc layout của file (ví dụ một trang HTML) với các vị trí ô đã được chỉ định để trình bày một nội dung cụ thể. Một hàm view có thể tự động tạo một trang HTML bằng cách sử dụng HTML template với dữ liệu được thêm vào từ model. Một template có thể sử dụng để tạo ra một cấu trúc cho bất kỳ loại file nào (không chỉ HTML).

2.4.2. Các phần dưới đây là các phần chính của một ứng dụng Django

2.4.2.1 Gửi yêu cầu đến một view phù hợp (urls.py)

- Một bộ URL thường được lưu trữ trong một tệp có tên là urls.py. Trong ví dụ dưới đây, urlpatterns xác định một danh sách các đường dẫn giữa các trang và các hàm view tương ứng. Nếu một yêu cầu HTTP được nhận có một URL khớp với một mẫu được chỉ định, thì hàm view tương ứng sẽ được gọi vào theo yêu cầu.

```
# filename: urls.py
urlpatterns = [
    path('admin/', admin.site.urls),
    path('book/<int:id>/', views.book_detail, name='book_detail'),
    path('catalog/', include('catalog.urls')),
    re_path(r'^([0-9]+)/$', views.best),
]
```

Hình 2-8 Code file urls.py

- urlpatterns là một danh sách các hàm path() hoặc re_path() (danh sách Python được định nghĩa bằng các dấu ngoặc vuông, trong đó các mục được phân tách bằng dấu phẩy và có thể có một dấu phẩy cuối tùy chọn. Ví dụ: [mục1, mục2, mục3,]).

- Đối số đầu tiên cho cả hai phương thức đều là một route (mẫu) sẽ được đối chiếu. Phương thức `path()` sử dụng ngoặc tròn để xác định các phần của URL. Nó sẽ được thu thập và truyền vào cho hàm view như các đối số. Hàm `re_path()` sử dụng một phương pháp khớp mẫu linh hoạt được gọi là biểu thức chính quy. Chúng ta sẽ nói về chúng trong một bài viết sau!
- Đối số thứ hai là một hàm khác sẽ được gọi khi mẫu được khớp. Ký hiệu `views.book_detail` cho thấy rằng hàm được gọi là `book_detail()` và có thể được tìm thấy trong một model được gọi là `views` (tức là bên trong một tệp được gọi là `views.py`)
- Xử lý yêu cầu (`views.py`): Các view là trái tim của ứng dụng web, nhận yêu cầu HTTP từ người dùng web và trả về phản hồi HTTP. Ngoài ra, chúng giải quyết các tài nguyên khác của framework để truy cập cơ sở dữ liệu, hiển thị các mẫu (template), vv.

Ví dụ dưới đây cho thấy một hàm view tối thiểu: `index()`. Nó có thể được gọi bởi bộ URL trong phần trước. Như tất cả các hàm view khác, nó nhận một đối tượng `HttpRequest` làm tham số (request) và trả về một đối tượng `HttpResponse`. Trong trường hợp này, chúng ta không làm gì với request và phản hồi của chúng ta trả về một chuỗi ký tự.

```
# filename: views.py
from django.http import HttpResponse
def index(request):
    return HttpResponse('Hello from Django!')
```

Hình 2-9 Code file `views.py`

2.4.3. Nhược điểm

- Django không có một bộ nguyên tắc để các nhà phát triển có thể tuân theo
- Không phù hợp với các dự án nhỏ
- Chỉ định URL bằng quy tắc biểu thức không dễ để thực hiện (đặc biệt là người mới bắt đầu). Nó cũng gây cảm giác cồng kềnh đối với các dự án nhỏ
- Django không thông báo lỗi trong mẫu. Nếu không biết, bạn sẽ mất rất nhiều thời gian để tìm ra vấn đề ở đâu
- Định tuyến tương đối phức tạp
- Không có khả năng thực hiện các yêu cầu khác nhau cùng lúc

2.5. SQLite



Hình 2-10 Logo SQLite

SQLite là hệ quản trị cơ sở dữ liệu (DBMS) quan hệ tương tự như Mysql, ... Đặc điểm nổi bật của SQLite so với các DBMS khác là gọn, nhẹ, đơn giản, đặc biệt không cần mô hình server-client, không cần cài đặt, cấu hình hay khởi động nên không có khái niệm user, password hay quyền hạn trong SQLite Database. Dữ liệu cũng được lưu ở một file duy nhất.

SQLite thường không được sử dụng với các hệ thống lớn nhưng với những hệ thống ở quy mô vừa và nhỏ thì SQLite không thua các DBMS khác về chức năng hay tốc độ. Vì không cần cài đặt hay cấu hình nên SQLite được sử dụng nhiều trong việc phát triển, thử nghiệm ... vì tránh được những rắc rối trong quá trình cài đặt.

2.5.1. Tính năng

- Giao dịch trong SQLite tuân thủ theo nguyên tắc (ACID) ngay cả sau khi hệ thống treo và mất điện.
- Không cấu hình:
- Không cần thiết lập hoặc quản trị
- SQLite hỗ trợ với đầy đủ tính năng với các khả năng nâng cao như các chỉ mục 1 phần, các chỉ mục về các biểu thức, JSON và các biểu thức bảng chung.
- Một sở dữ liệu hoàn chỉnh được lưu trữ trong một tệp đa nền tảng duy nhất. Phù hợp với sử dụng dưới dạng định dạng tệp ứng dụng
- Hỗ trợ các cơ sở dữ liệu có kích thước terabyte và các chuỗi có kích thước gigabyte.

- API
- Đơn giản dễ sử dụng
- Nhanh: Trong một số trường hợp, SQLite nhanh hơn hệ thống tệp tin trực tiếp I/O.
- Được viết bằng ANSI-C.
- Bindings cho hàng chục ngôn ngữ khác có sẵn 1 cách riêng biệt.
- Mã nguồn đầy, nguồn mở đủ có thể kiểm tra nhánh 100%.
- Nền tảng đa nền tảng:
- SQLite là có sẵn trên Android, *BSD, iOS, Linux, Mac, Solaris, Windows,.. Dễ dàng dịch chuyển sang các hệ thống khác.

2.5.2. Ứng dụng chủ yếu

- Cơ sở dữ liệu cho Internet Of Things.
 - SQLite là lựa chọn phổ biến cho các công cụ cơ sở dữ liệu trong điện thoại di động, PDA, máy nghe nhạc mp3, hộp set-top, và các tiện ích điện tử khác.
- Định dạng tệp ứng dụng.
 - Thay vì sử dụng fopen() để viết XML, JSON, CSV hoặc một số định dạng động quyền vào các tệp đĩa được ứng dụng của bạn sử dụng, hãy sử dụng SQLite.
- Cơ sở dữ liệu cho web.
 - Bởi vì SQLite không yêu cầu cấu hình và lưu trữ thông tin trong các tệp đĩa thông thường nên SQLite là lựa chọn phổ biến làm cơ sở dữ liệu để quay lại các trang web vừa và nhỏ.
- Stand-in cho một doanh nghiệp RDBMS:
 - SQLite được sử dụng như một RDBMS doanh nghiệp cho mục đích trình diễn hoặc để thử nghiệm vì SQLite nhanh và không yêu cầu thiết lập.

2.5.3. Điểm nổi bật

- AUTOINCREMENT
 - Một cột được khai báo INTEGER PRIMARY KEY sẽ tự động tăng thêm.
 - Khi chèn NULL vào cột này thì nó sẽ được tự động chuyển thành 1 số nguyên lớn nhất cột.

- Các kiểu dữ liệu
 - SQLite sử dụng một hệ thống kiểu động. Trong SQLite, kiểu dữ liệu là một giá trị được liên kết với chính giá trị đó, không liên kết với Container.
 - Các kiểu dữ liệu Integer, Real, Text, Blob, Null
 - SQLite không thực thi ràng buộc dữ liệu.
 - Dữ liệu loại nào cũng có thể chèn vào bất kì cột. Nhưng Integer primary key chỉ có thể chèn được số nguyên 64bit.
 - SQLite sử dụng khai báo dữ liệu của một cột làm gợi ý định dạng dữ liệu.
 - VD: Với một cột được khai báo integer bạn chèn kiểu chuỗi vào thì nó sẽ cố chuyển chuỗi thành số nguyên nếu có thể sẽ chèn số nguyên đó thay cho chuỗi.
=> Loại mối quan hệ.
- Xóa cột trong bảng
 - SQLite có hỗ trợ ALTER TABLE nhưng rất hạn chế chỉ có thể thêm cột và thay đổi tên.
 - Nếu muốn xóa cột thì chúng ta thực hiện các bước sau:
 - Tạo bảng mới có các cột cần thiết
 - Sao chép dữ liệu từ bảng cũ vào
 - Xóa bảng cũ
 - Tạo lại bảng với tên bảng cũ
 - Sao chép dữ liệu từ bảng tạm vào

```

TRANSACTION;
CREATE TEMPRARY TABLE new_backup (a,b)
INSERT INTO new_backup SELECT a,b FORM old;
DROP TABLE old;
CREATE TABLE old(a,b);
INSERT INTO SELECT a,b FORM new_backup;
DROP TABLE new_backup;
COMMIT;

```

Hình 2-11 Code xóa cột trong SQLite

- Khóa ngoài
 - o SQLite hỗ trợ khóa ngoài nhưng theo mặc định thì khóa ngoài sẽ tắt.
 - o Để cho phép thực thi khóa ngoài ta sử dụng câu lệnh

```
PRAGMA foreign keys = ON

# hoặc biên dịch với
DSQLITE DEFAULT FOREIGN KEYS = 1
```

Hình 2-12 Code kích hoạt khoá ngoài SQLite

Dấu nháy: SQLite phân biệt cách sử dụng dấu nháy

- o Nháy đôi cho tên bảng, cột
- o Đơn cho giá trị
- Mệnh đề GLOB
 - o So khớp giá trị với các giá trị tương tự bởi sử dụng các toán tử wildcard.
 - o Không giống LIKE, GLOB phân biệt kiểu chữ và nó theo cú pháp của UNIX để xác định các toán tử Wildcard sau:
 - '*' : số 0,1 hoặc nhiều số hoặc kí tự (tương tự như %)
 - '?' : 1 số hoặc 1 kí tự đơn (tương tự như _)

```
SELECT FROM table_name
WHERE column GLOB 'XXXX*'
```

Hình 2-13 Code select Table SQLite

- Cú pháp

- SQLite hỗ trợ gần như đầy đủ các cú pháp trong chuẩn SQL92.
- Dưới đây là một số câu lệnh thường được sử dụng

Bảng 2-1 Cú pháp SQLite

STT	Cú pháp	Ý nghĩa
1	sqlite3 <name.db>	Tạo database
2	ATTACH DATABASE '<databasename>' As '<alias-name>';	Sử dụng database, có thể đặt alias cho database và sử dụng như tên của database, mỗi một lần gọi lệnh sử dụng thì ta có thể sử dụng tên alias khác nhau
3	DETACH DATABASE '<name-name>';	Xóa cơ sở dữ liệu sử dụng với tên alias
4	CREATE TABLE <databasename.tablename>();	Tạo bảng
5	DROP TABLE database_name.table_name;	Xóa bảng
6	INSERT INTO table_name [(column1, column2,...)] VALUES (value1, value2,...);	Thêm dữ liệu vào bảng
7	INSERT INTO table1 [(column...)] SELECT column FROM table2 [WHERE];	Chèn dữ liệu vào bảng từ một bảng khác
8	SELECT sql FROM table;	Hiển thị thông tin bảng
9	SELECT (12+8) AS ADDITION; #20	Thực hiện biểu thức số học
10	SELECT COUNT(*) AS "RECORDS" FROM table;	đếm bảng ghi trong bảng
11	SELECT CURRENT_TIMESTAMP;	Hiển thị thời gian hệ thống
12	UPDATE table_name SET column1 = value,... WHERE ..;	Update dữ liệu bảng

STT	Cú pháp	Ý nghĩa
13	DELETE FROM table_name WHERE ...;	Xóa bản ghi
14	PRAGMA pragma_name;	Điều khiển các biến môi trường và các flag trạng thái đa dạng
15	PRAGMA pragma_name = value;	Thiết lập giá trị
16	SELECT ... FROM table1 CROSS JOIN table2 ...	CROSS JOIN: kết nối mọi hàng của bảng đầu tiên với mỗi hàng của bảng thứ hai
17	SELECT ... FROM table1 [INNER] JOIN table2 ON conditional_expression ...	INNER JOIN
18	SELECT ... FROM table1 LEFT OUTER JOIN table2 ON conditional_expression ...	OUTER JOIN: chỉ hỗ trợ LEFT JOIN

2.5.4. Nhược điểm

- Một số tính năng của SQL92 không được hỗ trợ trong SQLite như ALTER DROP COLUMN, ADD CONSTRAINT không được hỗ trợ; RIGHT JOIN; TRIGGER; phân quyền GRANT và REVOKE.
- Vì SQLite không cần cấu hình, cài đặt, không hỗ trợ GRANT và REVOKE nên việc phân quyền truy cập cơ sở dữ liệu chỉ có thể là quyền truy cập file của hệ thống.
- SQLite sử dụng cơ chế coarse-grained locking nên trong cùng một thời điểm có thể hỗ trợ nhiều người đọc dữ liệu, nhưng chỉ có 1 người có thể ghi dữ liệu.
- SQLite không phù hợp với các hệ thống có nhu cầu xử lý trên một khối lượng dữ liệu lớn, phát sinh liên tục.

3.1.2. Danh sách Actor

Bảng 3-1 Danh sách Actor

STT	Tên Actor	Ý nghĩa
1	User	Người dùng có thể vào để đọc truyện, tạo tài khoản upload truyện

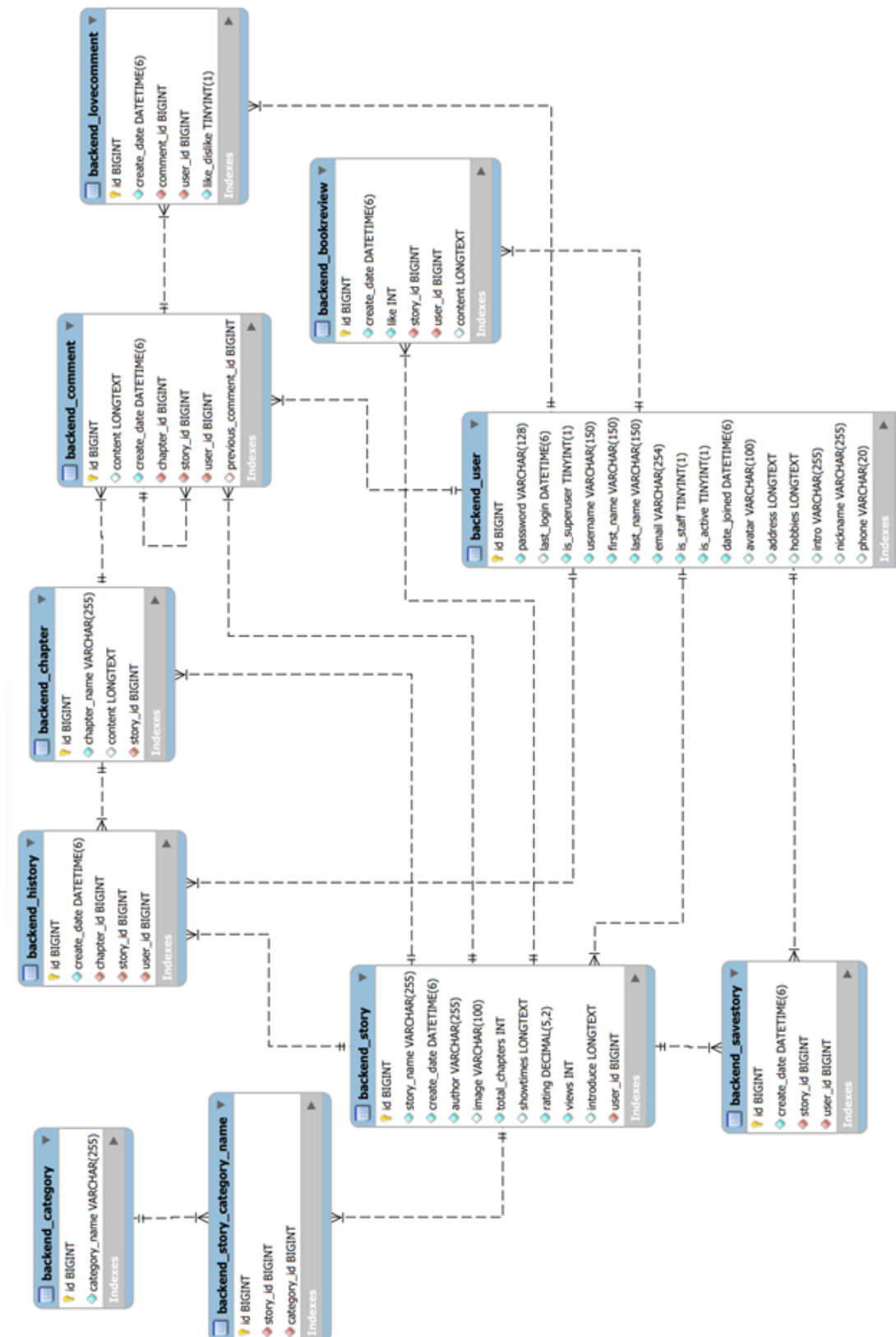
3.1.3. Danh sách Usecase

Bảng 3-2 Danh sách Usecase

STT	Tên Usecase	Ý nghĩa
1	Đọc truyện	Duyệt qua các trang truyện để đọc
2	Quản lý truyện	Bao gồm các thao tác Thêm/Xoá/Sửa
3	Lưu truyện	Lưu lại những cuốn truyện yêu thích
4	Đăng nhập	Đăng nhập vào hệ thống
5	Đăng ký	Đăng ký tài khoản mới

3.2. Thiết kế Cơ sở dữ liệu

3.2.1. Thiết kế dữ liệu

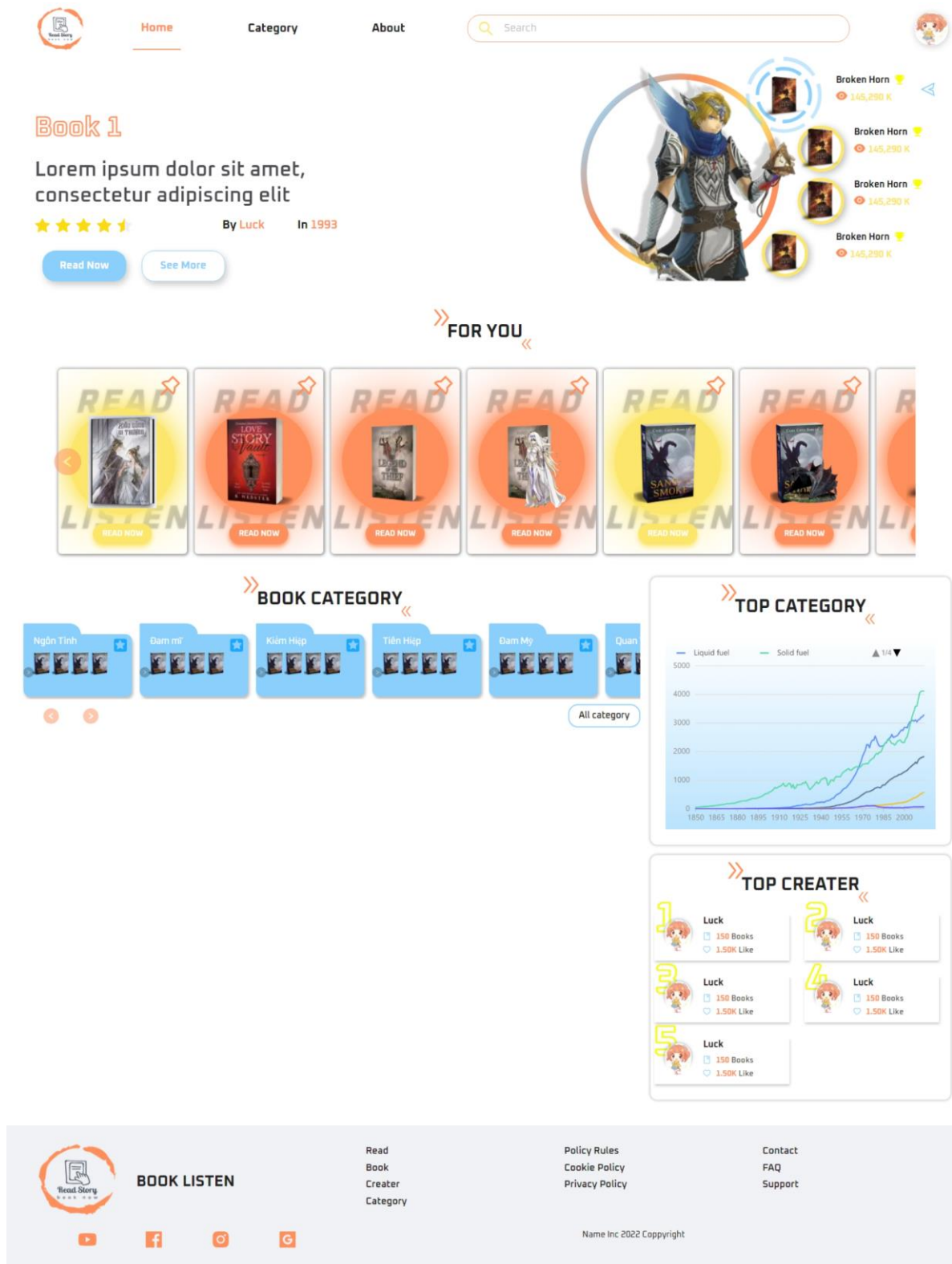


Hình 3-2 Sơ đồ quan hệ CSDL

CHƯƠNG 4: XÂY DỰNG WEBSITE

4.1. Trang Home:

4.1.1. Giao diện:



Hình 4-1 Giao diện Trang Home

4.1.2. Mô tả:

- Trang chủ hiển thị các truyện mới nhất và các thể loại truyện hiện có.

4.1.3. Kỹ thuật thiết kế:

- Phân chia các thành phần của website thành các component nhỏ hơn để tăng tính linh hoạt và khả năng tái sử dụng ở những trang khác.

4.1.4. Code thiết kế:

```
<Main>
  <Baner />
  <TextHeading >For You</TextHeading>
  <div className='home__silde'>
    <div className='home__next__back--slide'>
      <LeftCircleFilled className='btn__slide btn__back' />
      <RightCircleFilled className='btn__slide btn__next' />
    </div>
    <div className='home__silde__container' >

      {
        story.map((story) => {
          return (
            <Card index={story.id} story_name={story.story_name} image={story.image} total_chapter
s={story.total_chapters} introduce={story.introduce} owner={story.user} />)
          })
        }
      </div>
    </div>
    <div className='home__contaner__content'>
      <Row >
        <Col lg={16} sm={24} xs={24}>
          <TextHeading>Book Category</TextHeading>
          <div>
            <div className='home__contaner__content--listfile' onClick={getCaStory}>
              {
                category.map((categori) => {
                  return (
                    <File index={categori.id} name={categori.category_name} />)
                  })
                }
              </div>
            <Row align='middle'>
```

```

<Col lg={4} sm={6} xs={12}>
  <div className='home__next__back--slide slide__small'>
    <LeftCircleFilled className='btn__slide btn__back' />
    <RightCircleFilled className='btn__slide btn__next' />
  </div>
</Col>
<Col lg={14} sm={12} xs={0}>
  <div className='slide__dot__bottom'>

  </div>
</Col>
<Col lg={6} sm={6} xs={12}>
  <div>
    <Button className='btn__all__category'>All category</Button>
  </div>
</Col>
</Row>
<Row>
  {
    caStory.map((caStory) => {
      return (
        <Card index={caStory.id} story_name={caStory.story_name} image={caStory.image} total_chapters={caStory.total_chapters} introduce={caStory.introduce} owner={caStory.user} />
      )
    })
  }
</Row>
</div>
</Col>
<Col lg={8} sm={24} xs={24}>
  <Row>
    <Col lg={24} sm={24} xs={0}>
      <div className='home__bar__line'>
        <TextHeading>Top Category</TextHeading>
        <Bar />
      </div>

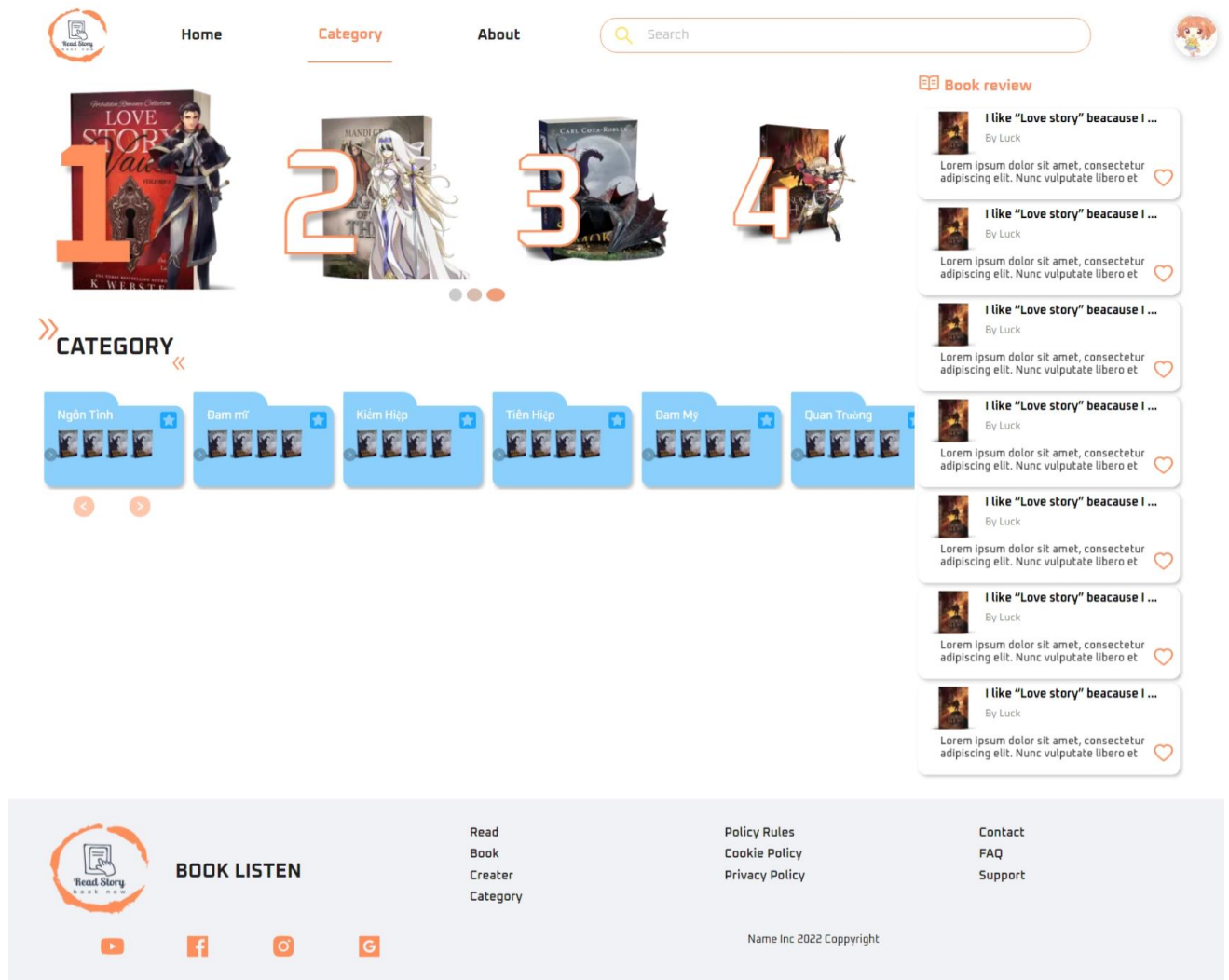
    </Col>
  </Row>
  <Row>
    <Col lg={24} sm={24} xs={24}>
      <div className='home__bar__line'>
        <TextHeading>Top Creator</TextHeading>
      </div>
    </Col>
  </Row>

```

```
<Col lg={12} sm={6} xs={12} >
  <CardCreator {...{ number: '1' }} />
</Col>
<Col lg={12} sm={6} xs={12} >
  <CardCreator {...{ number: '2' }} />
</Col>
<Col lg={12} sm={6} xs={12} >
  <CardCreator {...{ number: '3' }} />
</Col>
<Col lg={12} sm={6} xs={12} >
  <CardCreator {...{ number: '4' }} />
</Col>
<Col lg={12} sm={6} xs={12} >
  <CardCreator {...{ number: '5' }} />
</Col>
</Row>
</div>
</Col>
</Row>
</Col>
</Row>
</div>
</Main>
```

4.2. Trang Category:

4.2.1. Giao diện:



Hình 4-2 Giao diện Trang Category

4.2.2. Mô tả:

- Trang hiển thị các truyện theo thể loại.

4.2.3. Kỹ thuật thiết kế:

- Sử dụng lại các component như SilderBook, Card.

4.2.4. Code thiết kế:

```
<Main>
  <div className="category__silder">
    <Row>
      <Col lg={18} sm={24} xs={0}>
        <Carousel autoplay dots={{ className: 'dot__silder__category' }}>
          <div>
            <SilderBook />
          </div>
          <div>
            <SilderBook />
          </div>
          <div>
            <SilderBook />
          </div>
        </Carousel>
        <div>
          <div className="category__silder__title">
            <TextHeading >Category</TextHeading>
          </div>
          <div className='category__contaner__content--listfile' onClick={getCaStory}>
            {
              category.map((categori) => {
                // console.log(categori);
                return (
                  <File index={categori.id} name={categori.category_name} />
                )
              })
            }
          </div>
          <Row align='middle'>
            <Col lg={4} sm={6} xs={12}>
              <div className='home__next__back--slide slide_small'>
                <LeftCircleFilled className='btn__slide btn__back' />
                <RightCircleFilled className='btn__slide btn__next' />
              </div>
            </Col>
            <Col lg={14} sm={12} xs={0}>
              <div className='silde__dot__bottom'>
                </div>
            </Col>
          </Row>
        </div>
      </Col>
    </Row>
  </div>
</Main>
```

```

        </Row>
      </div>
      <div className="category__list__book">
        <Row>
          {
            caStory.map((caStory) => {
              return (
                <Card index={caStory.id} story_name={caStory.story_name} image
={caStory.image} total_chapters={caStory.total_chapters} introduce={caStory.introduce} owner={caSt
ory.user} />)
              })
          }
        </Row>
      </div>

    </Col>
    <Col lg={0} sm={0} xs={24}>
      <Carousel autoplay dots={{ className: 'dot__silder__category' }}>
        <div>
          <SilderOneBook />
        </div>
        <div>
          <SilderOneBook />
        </div>
        <div>
          <SilderOneBook />
        </div>
      </Carousel>
      <div>
        <div className="category__silder__title">
          <TextHeading >Category</TextHeading>
        </div>
        <div className='category__contaner__content--listfile'>

        </div>
        <Row align='middle'>
          <Col lg={4} sm={6} xs={12}>
            <div className='home__next__back--slide slide_small'>
              <LeftCircleFilled className='btn_slide btn_back' />
              <RightCircleFilled className='btn_slide btn_next' />
            </div>
          </Col>

```



```

        <Col lg={14} sm={12} xs={0}>
            <div className='slide_dot_bottom'>

            </div>
        </Col>

    </Row>
</div>
<div className="category__list__book">
    <Row>
        <Col lg={6} sm={8} xs={12}>
            <Card index={1} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={2} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={3} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={4} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={5} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={6} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={7} />
        </Col>
        <Col lg={6} sm={8} xs={12}>
            <Card index={8} />
        </Col>
    </Row>
</div>

</Col>
<Col lg={6} sm={24} xs={24}>
    <div className="category__navbar__review">
        <div className="category__navbar__review--title">
            <ReadOutlined />

```

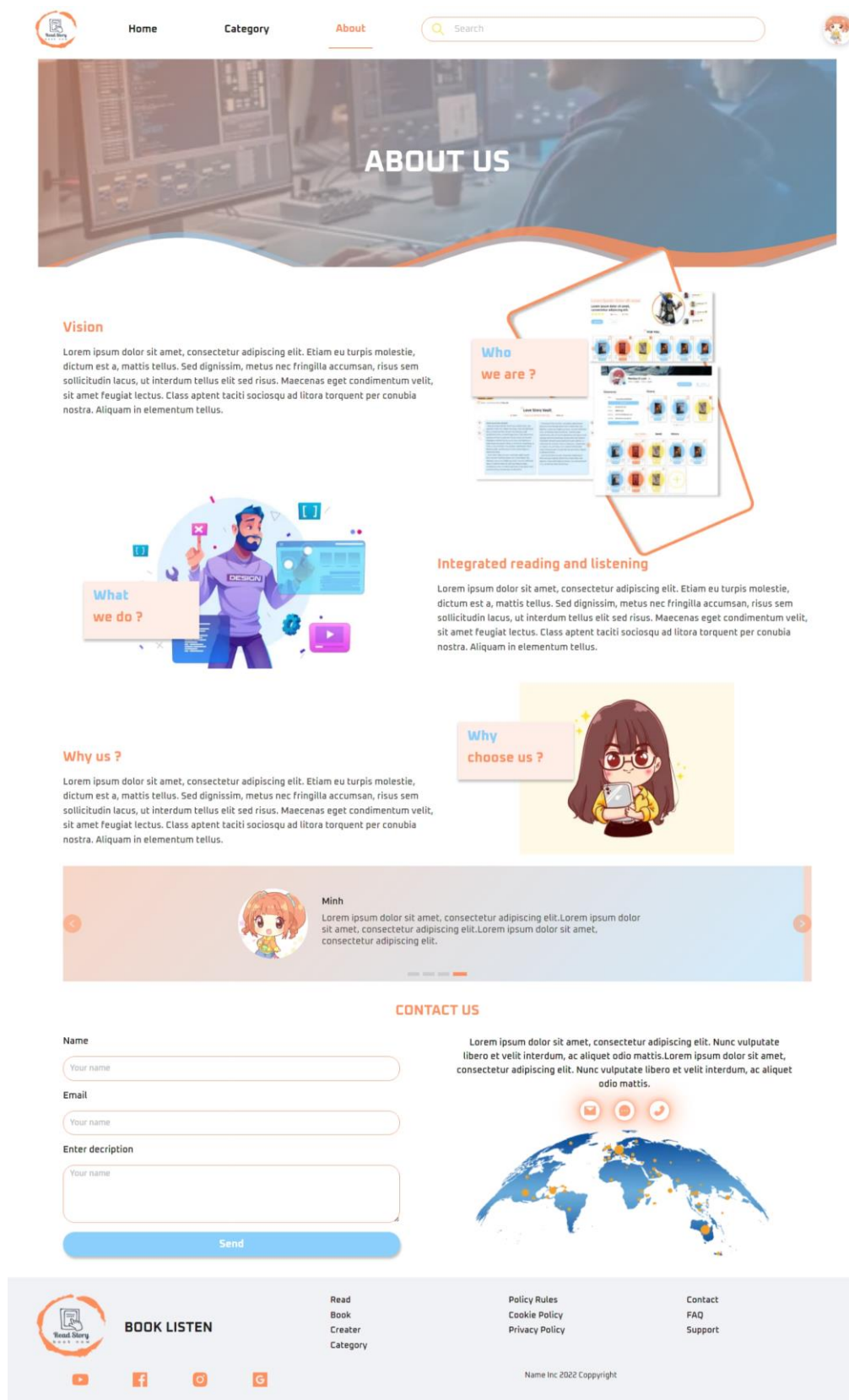
```

        <span>Book review</span>
    </div>
    <Row>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
        <Col lg={24} sm={12} xs={24}>
            <ItemReview></ItemReview>
        </Col>
    </Row>
</div>
</Col>
</Row>
</div>
</Main>

```

4.3. Trang About:

4.3.1. Giao diện:



Hình 4-3 Giao diện Trang About

4.3.2. Mô tả:

- Trang thông tin giới thiệu về website.

4.3.3. Kỹ thuật thiết kế:

- Phân chia website thành 2 cột.

4.3.4. Code thiết kế:

```
<Main>
  <div className='container__abouts'>
    <div className='container__abouts__header'>
      <div className='container__abouts__header--img '>
        <Image src={bg} layout='fill' objectFit='cover' />
      </div>
      <div>
        <div className='wave__img'>
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path fill="#fff" fillOpacity="1" d="M0,256L48,234.7C96,213,192,171,288,181.3C384,192,480,256,576,256C672,256,768,192,864,186.7C960,181,1056,235,1152,245.3C1248,256,1344,224,1392,208L1440,192L1440,320L1392,320C1344,320,1248,320,1152,320C1056,320,960,320,864,320C768,320,672,320,576,320C480,320,384,320,288,320C192,320,96,320,48,320L0,320Z"></path></svg>
        </div>
        <div className='wave__img'>
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path fill="#FF8F5C" fillOpacity="0.8" d="M0,192L48,176C96,160,192,128,288,138.7C384,149,480,203,576,197.3C672,192,768,128,864,117.3C960,107,1056,149,1152,160C1248,171,1344,149,1392,138.7L1440,128L1440,320L1392,320C1344,320,1248,320,1152,320C1056,320,960,320,864,320C768,320,672,320,576,320C480,320,384,320,288,320C192,320,96,320,48,320L0,320Z"></path></svg>
        </div>
        <div className='wave__img'>
          <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 1440 320"><path fill="#8BD0FC" fillOpacity="0.5" d="M0,256L48,234.7C96,213,192,171,288,181.3C384,192,480,256,576,256C672,256,768,192,864,186.7C960,181,1056,235,1152,245.3C1248,256,1344,224,1392,208L1440,192L1440,320L1392,320C1344,320,1248,320,1152,320C1056,320,960,320,864,320C768,320,672,320,576,320C480,320,384,320,288,320C192,320,96,320,48,320L0,320Z"></path></svg>
        </div>
      </div>
    </div>
    <div className='container__abouts__content'>
      <Row>
        <Col lg={12} sm={10} xs={24}>
          <div style={{marginTop: '62px'}}>
            <p className='container__abouts__content--text-title'>Vision</p>
            <p className='container__abouts__content--text-detail'>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, u
```

```

t interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra. Aliquam in elementum tellus.</p>

    </div>
  </Col>
  <Col lg={12} sm={14} xs={0}>
    <div style={{padding: '0 24px',minHeight: '320px'}}>
      <div className='container__abouts__content--bg-text'>
        <p className='container__abouts__content--text-
title'><b style={{color: '#8BD0FC'}}>Who</b><br><br> we are ?</p>
      </div>
      <div className='container__abouts__content--img' >
        <Image src={about1} />
      </div>
    </div>
  </Col>
  <Col lg={12} sm={14} xs={0}>
    <div style={{marginTop: '64px'}}>
      <div className='container__abouts__content--bg-text'>
        <p className='container__abouts__content--text-
title'><b style={{color: '#8BD0FC'}}>What</b><br><br> we do ?</p>
      </div>
      <div className='container__abouts__content--img'>
        <Image src={about2} />
      </div>
    </div>
  </Col>
  <Col lg={12} sm={10} xs={24}>
    <div style={{marginTop: '52px'}}>
      <p className='container__abouts__content--text-
title'>Integrated reading and listening</p>
      <p className='container__abouts__content--text-
detail'>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum
est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, u
t interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra. Aliquam in elementum tellus.</p>
    </div>
  </Col>
  <Col lg={12} sm={10} xs={24}>
    <div style={{marginTop: '142px'}} className='about__why'>
      <p className='container__abouts__content--text-title'>Why us ?</p>
      <p className='container__abouts__content--text-
detail'>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam eu turpis molestie, dictum
est a, mattis tellus. Sed dignissim, metus nec fringilla accumsan, risus sem sollicitudin lacus, u
t interdum tellus elit sed risus. Maecenas eget condimentum velit, sit amet feugiat lectus. Class
aptent taciti sociosqu ad litora torquent per conubia nostra. Aliquam in elementum tellus.</p>
    </div>
  </Col>

```

```

        <Col lg={12} sm={14} xs={0}>
            <div style={{marginTop: '64px'}}>
                <div className='container__abouts__content--bg-text'>
                    <p className='container__abouts__content--text-
title'><b style={{color: '#8BD0FC'}}>Why</b><br></br> choose us ?</p>
                </div>
                <div className='container__abouts__content--
img' style={{top: '32px',left: '142px'}}>
                    <Image src={about3} />
                </div>
            </div>
        </Col>
    </Row>
    <div className='abouts__silder'>
        <div className='abouts__silder__icon'>
            <LeftCircleFilled />
            <RightCircleFilled />
        </div>
        <Carousel autoplay dots={{className:'abouts__silder__dots'}} >
            <div className='abouts__silder__content'>
                <div className='abouts__silder__content--avatar' >
                    <Image src={avatar1} objectFit='cover' />
                </div>
                <div className='abouts__silder__text' style={{maxWidth:'45%'}}>
                    <p className='abouts__silder__content--name'>Luck</p>
                    <p className='abouts__silder__content--
info'>Lorem ipsum dolor sit amet, consectetur adipiscing elit.Lorem ipsum dolor sit amet, consecte
tur adipiscing elit.Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
                </div>
            </div>
            <div className='abouts__silder__content'>
                <div className='abouts__silder__content--avatar'>
                    <Image src={avatar1} objectFit='cover' />
                </div>
                <div className='abouts__silder__text' style={{maxWidth:'45%'}}>
                    <p className='abouts__silder__content--name'>Linh</p>
                    <p className='abouts__silder__content--
info'>Lorem ipsum dolor sit amet, consectetur adipiscing elit.Lorem ipsum dolor sit amet, consecte
tur adipiscing elit.Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
                </div>
            </div>
            <div className='abouts__silder__content'>
                <div className='abouts__silder__content--avatar'>
                    <Image src={avatar1} objectFit='cover' />
                </div>
            </div>
        </Carousel>
    </div>

```

```

        </div>
        <div className='about__silder__text' style={{maxWidth:'45%'}}>
            <p className='abouts__silder__content--name'>Khoa</p>
            <p className='abouts__silder__content--
info'>Lorem ipsum dolor sit amet, consectetur adipiscing elit.Lorem ipsum dolor sit amet, consecte
tur adipiscing elit.Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        </div>
    </div>

    <div className='abouts__silder__content'>
        <div className='abouts__silder__content--avatar'>
            <Image src={avatar1} objectFit='cover' />
        </div>
        <div className='about__silder__text' style={{maxWidth:'45%'}}>
            <p className='abouts__silder__content--name'>Minh</p>
            <p className='abouts__silder__content--
info'>Lorem ipsum dolor sit amet, consectetur adipiscing elit.Lorem ipsum dolor sit amet, consecte
tur adipiscing elit.Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
        </div>
    </div>

</Carousel>
</div>

<div className='abouts__contact'>
    <p className='abouts__contact__title'>Contact us</p>
    <Row>
        <Col lg={12} sm={12} xs={24}>
            <div>
                <p className='abouts__contact__text'>Name</p>
                <Input className='abouts__contact__input' placeholder="Your name" />
            </div>
            <div>
                <p className='abouts__contact__text'>Email</p>
                <Input className='abouts__contact__input' placeholder="Your name" />
            </div>
            <div>
                <p className='abouts__contact__text'>Enter decription</p>
                <TextArea className='abouts__contact__input abouts__contact__area' placeholder
="Your name" />
            </div>
            <Button className='btn__send__abouts'>Send</Button>
        </Col>
        <Col lg={12} sm={12} xs={24}>
            <div className='abouts__contact__information'>

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc vulputate libero et velit interdum, ac aliquet odio mattis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc vulputate libero et velit interdum, ac aliquet odio mattis.</p>

<div className='abouts__contact__listicon'>

<div className='abouts__contact__icon--contact'><MailFilled /></div>

<div className='abouts__contact__icon--contact'><MessageFilled /></div>

<div className='abouts__contact__icon--contact'><PhoneFilled /></div>

</div>

<div className='abouts__contact__information--img'>

<Image src={world} objectFit='cover' />

</div>

</div>

</Col>

</Row>

</div>


</div>


</div>

</Main>


4.4. Trang Story Detail:

4.4.1. Giao diện:

[Home](#)[Category](#)[About](#)



Book / KIỀU SÙNG VI THƯỢNG



KIỀU SÙNG VI THƯỢNG

★★★★★

Main content

Đôi trước, Thẩm Xu bị bắt đi hòa thân ở Hung Nô xa xôi; vài năm sau hai bên giao chiến, nàng vốn là công chúa không được sung ái hoàn toàn trở về hi sinh. Khi mọi ý niệm đã hóa thành tro tàn, nàng tận mắt thấy nam nhân máu lạnh vô tình trong lời đồn vì nàng mà chết trên nơi sa trường. Nàng trúng sinh vào một buổi tối trước ngày hòa thân, nghiêng nghiêng ngả ngả lao đao chạy ra khỏi tâm điện, trực tiếp đâm vào lồng ngực Bùi Văn Khiêm. Thân sắc nam nhân hoàng hốt trong nháy mắt, ánh mắt thiếu nữ trong ngực long lạnh như nước quyên rũ động lòng: "Ta không muốn đi hòa thân, tướng quân có nguyên ý cười ta không?". Mọi người đều biết, Linh An công chúa vì tránh né hòa thân mà gả cho gian thần. Bùi Văn Khiêm hi vọng vô thường, hung ác nham hiểm, chỉ sợ không bao lâu sau, công chúa sẽ hương tiêu ngọc vãn.

New chap

Chap 2: Tở Hôn

Chap 4: Sợ Ta

Chap 3: Đại Hôn

Author: 4

Category: Ngôn Tình

Total chap: 46

» ALL CHAP «

Chap 1: Trùng Sinh

Chap 2: Tở Hôn


Chap 4: Sợ Ta

Chap 3: Đại Hôn

» COMMENT «


Add Comment

1 replies


 Han Solo · 8 hours ago

We supply a series of design principles, practical patterns and high quality design resources (Sketch and Axure), to help people create their product prototypes beautifully and efficiently.


» SIMILAR BOOKS «




READ NOW




READ NOW







READ NOW



READ NOW



BOOK LISTEN



Read Book Creator Category

[Policy Rules](#)[Cookie Policy](#)[Privacy Policy](#)

Contact FAQ Support

Name Inc 2022 Copyright

Hình 4-4 Giao diện Trang Story Detail

4.4.2. Mô tả:

- Trang hiển thị chi tiết nội dung của một cuốn truyện.

4.4.3. Kỹ thuật thiết kế:

- Tiếp tục chia và tạo ra các component nhỏ hơn.

4.4.4. Code thiết kế:

```
<Main>
  <div className='book__detail'>
    <Breadcrumb className='book__detail__breadcurmb'>
      <Breadcrumb.Item href='/ '>
        <ReadOutlined className='icon__book' />
        <span>Book</span>
      </Breadcrumb.Item>
      <Breadcrumb.Item>{book?.story_name}</Breadcrumb.Item>
    </Breadcrumb>
    <div className='book__detail__content'>
      <Row>
        <Col span={18}>
          <div>
            <BookDetail
              book={book}
              newChapterList={chapterList.slice(
                chapterList.length - 3,
                chapterList.length
              )}
            ></BookDetail>
          </div>
          <div className='book__detail__content__listchap'>
            <Row>
              <Col span={24}>
                <div className='book__detail__content__listchap--title'>
                  <TextHeading>All chap</TextHeading>
                </div>
              </Col>
              {chapterList.map((chap, index) => (
                <Col span={12} key={index}>
                  <Chap chap={chap} book={book} />
                </Col>
              ))}
            </Row>
          </div>
        </Col>
      </Row>
    </div>
  </div>
</Main>
```

```

</div>
<div className='book__detail__content__comment'>
  <Row>
    <Col span={24}>
      <div className='book__detail__content__listchap--title'>
        <TextHeading>Comment</TextHeading>
      </div>
    </Col>
    <div className='book__detail__content__comment--comment'>
      <CommentBook></CommentBook>
    </div>
  </Row>
</div>
<div className='book__detail__content__similar'>
  <Row>
    <Col span={24}>
      <div className='book__detail__content__listchap--title'>
        <TextHeading>Similar books</TextHeading>
      </div>
    </Col>
    <div>
      <Row>
        <Col span={6}>
          <Card index={1}></Card>
        </Col>
        <Col span={6}>
          <Card index={2}></Card>
        </Col>
        <Col span={6}>
          <Card index={3}></Card>
        </Col>
        <Col span={6}>
          <Card index={4}></Card>
        </Col>
      </Row>
    </div>
  </Row>
</div>
<Col span={6}>
  <div className='book__detail__navbar__review--title'>
    <ReadOutlined />
  </div>
</Col>

```

```
        <span>Book review</span>
    </div>

    <Row>
        {bookReviewList.map((bookReview, index) => (
            <Col span={24} key={index}>
                <ItemReview bookReview={bookReview} book={book} />
            </Col>
        ))}
    </Row>
</Col>
</Row>
</div>
</div>
</Main>
```

4.5. Trang Chapter Detail:

4.5.1. Giao diện:



Hình 4-5 Giao diện Chapter Detail

4.5.2. Mô tả:

- Trang hiển thị chi tiết nội dung của một tập

4.5.3. Kỹ thuật thiết kế:

- Sử dụng 2 column để hiển thị nội dung.

4.5.4. Code thiết kế:

```
<Main>
  <div className="readbook__content">
    <Breadcrumb className="readbook__detail__breadcurmb">
      <Breadcrumb.Item href="/">
        <ReadOutlined className="icon__book" />
        <span>Book</span>
      </Breadcrumb.Item>
      <Breadcrumb.Item onClick={() => router.push(`/books/detail/${hello}`)}>
        <span>Love story valut</span>
      </Breadcrumb.Item>
      <Breadcrumb.Item>Chap {iChap}</Breadcrumb.Item>
    </Breadcrumb>
    <div className="readbook__content__main">
      <TextHeading>{sName}</TextHeading>
      <div className="readbook__content__main--title">
        <Row justify="center" align='middle' >
          <Col lg={5} sm={2} xs={0}></Col>
          <Col lg={4} sm={6} xs={7} >
            <Button onClick={btnPre}>
              <BackwardOutlined className="icon__nextback" />
            </Button>
          </Col>
          <Col lg={6} sm={8} xs={10}>
            <p className="readbook__content__main--title--chap">Chap {iChap}: {nChap}</p>
          </Col>
          <Col lg={4} sm={6} xs={7} >
            <Button onClick={btnNex}>
              <ForwardOutlined className="icon__nextback" />
            </Button>
          </Col>
          <Col lg={5} sm={2} xs={0}></Col>
        </Row>
      </div>
    </div>
  </div>
```

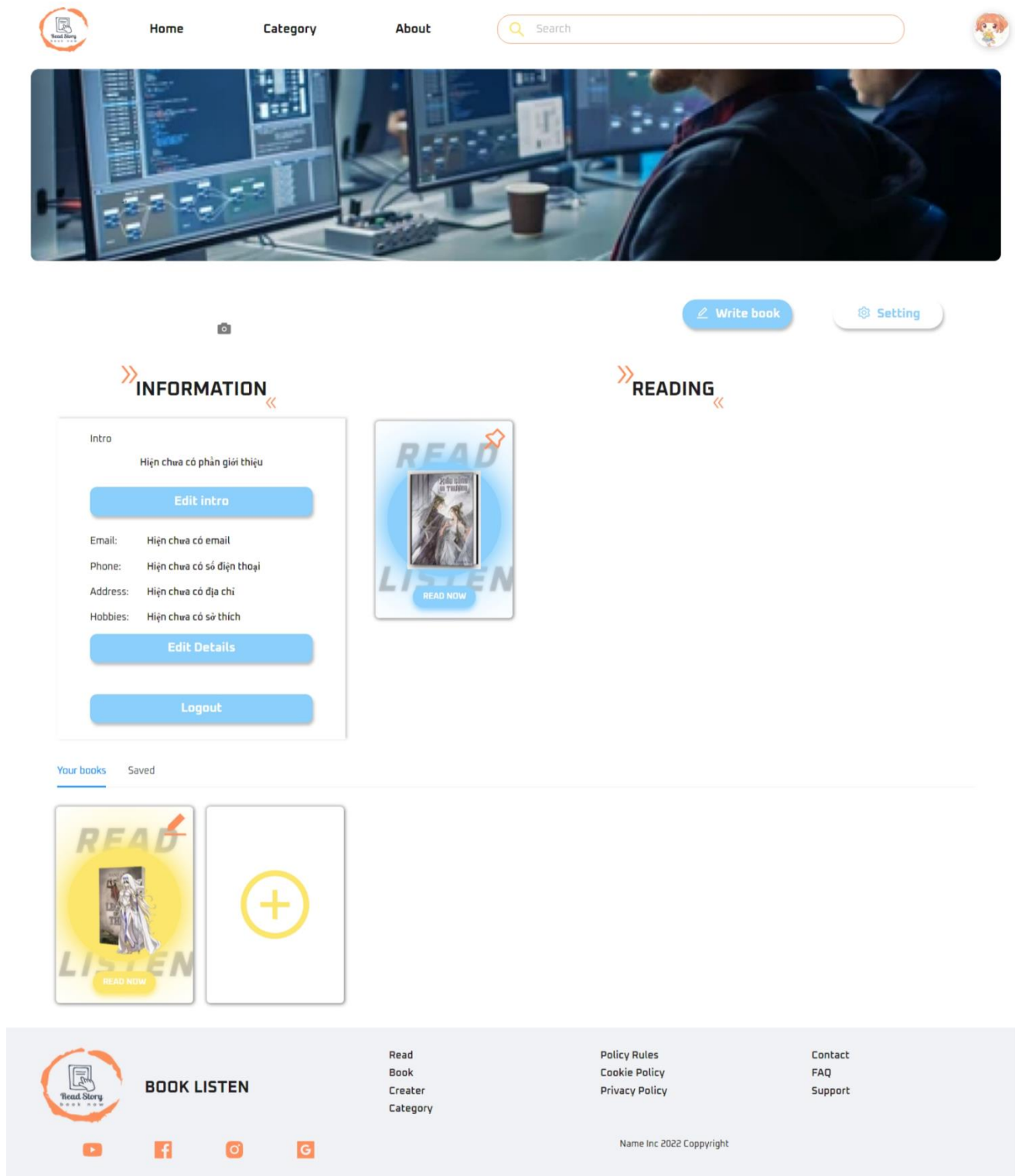
```

        </Row>
    </div>
    <input type='checkbox' id='change__mode' className="change__mode__book" onClick={() =>
    { setCheck(!check) }} style={{ display: 'none' }} />
    <div className="readbook__content__main--content">
        <div className="readbook__content__main--groundicon">
            <div className="icon__light__darkmode" >
                <label style={{ height: '36px', cursor: 'pointer' }} htmlFor="change__mode
">
                    {check ? <Image src={IconDark} /> : <Image src={IconLight} />}
                </label>
            </div>
        </div>
    </div>
    <Book check={check} content={cChap}></Book>
</div>
</div>
</Main>

```

4.6. Trang Profile:

4.6.1. Giao diện:



Hình 4-6 Giao diện Trang Profile

Hình 4.3.

4.6.2. Mô tả:

- Trang hiển thị thông tin cá nhân, truyện của tôi, truyện đã đọc, truyện đã lưu.

4.6.3. Kỹ thuật thiết kế:

- Sử dụng DOM để chèn có popup.

4.6.4. Code thiết kế:

```
<Main>
  <div>
    <div className='user-section'>
      <div
        className='banner-background'
        style={{
          backgroundImage: `url(${bannerUrl.src})`,
        }}
      >
        <div className='user-info'>
          <div>
            <div className='user-avatar'>
              <Image
                src={user?.avatar ? `${user?.avatar}` : ''}
                loader={myLoader}
                alt='Avatar'
                layout='fill'
                className='user-avatar-img'
              />
              <span className='user-change-avatar'>
                <CameraFilled />
              </span>
            </div>
          </div>
          <div className='user-info-options'>
            <div className='write-book-btn'>
              <Button
                className='btn__create__book'
                onClick={() => router.push(`books/create/${name}`)}
              >
                <span className='option-icon'>
                  <EditOutlined />
                </span>
            </div>
          </div>
        </div>
      </div>
    </div>
  </Main>
```

```

        <span className='option-content'>Write book</span>
      </Button>
    </div>
    <div className='setting-btn'>
      <Button className='btn__setting__user' disabled>
        <span className='option-icon'>
          <SettingOutlined />
        </span>
        <span className='option-content'>Setting</span>
      </Button>
    </div>
    <div className='write-book-btn-mb'>
      <Button
        className='btn__create__book'
        onClick={() => router.push(`books/create/${name}`)}
      >
        <span className='option-icon'>
          <EditOutlined />
        </span>
      </Button>
    </div>
    <div className='setting-btn-mb'>
      <Button className='btn__setting__user' disabled>
        <span className='option-icon'>
          <SettingOutlined />
        </span>
      </Button>
    </div>
  </div>
</div>
<div className='main-section-profile'>
  <Row gutter={38}>
    <Col lg={8} sm={10} xs={24}>
      <TextHeading className='text-heading-section'>
        Information
      </TextHeading>
      <div className='user-introduce'>
        <div className='user-introduce-item user-intro'>
          <span className='user-introduce-item-title'>Intro</span>
          <div className='user-intro-text'>

```

```

        <span>
            {user?.intro || 'Hiện chưa có phần giới thiệu'}
        </span>
    </div>
    <div>
        <Button className='edit__intro__user'>Edit intro</Button>
    </div>
</div>
<div className='user-introduce-item'>
    <span className='user-introduce-item-title'>Email:</span>
    <span className='user-introduce-item-content'>
        {user?.email || 'Hiện chưa có email'}
    </span>
</div>
<div className='user-introduce-item'>
    <span className='user-introduce-item-title'>Phone:</span>
    <span className='user-introduce-item-content'>
        {user?.phone || 'Hiện chưa có số điện thoại'}
    </span>
</div>
<div className='user-introduce-item'>
    <span className='user-introduce-item-title'>Address:</span>
    <span className='user-introduce-item-content'>
        {user?.address || 'Hiện chưa có địa chỉ'}
    </span>
</div>
<div className='user-introduce-item'>
    <span className='user-introduce-item-title'>Hobbies:</span>
    <span className='user-introduce-item-content'>
        {user?.hobbies || 'Hiện chưa có sở thích'}
    </span>
</div>
<div>
    <Button className='edit__intro__user'>Edit Details</Button>
</div>
<br></br>
<div className='item__card__book--btn'>
    <Link href='/'>
        <Button
            className='edit__intro__user'
            onClick={() => {
                logOut();
            }}
        >

```

```

        }}
    >
    Logout
  </Button>
</Link>
</div>
</div>
</Col>
<Col lg={16} sm={14} xs={24}>
  <TextHeading className='text-heading-section'>
    Reading
  </TextHeading>
  <Row>
    {
      hiStory.map((hiStory) => {
        return (
          <Card index={hiStory.id} story_name={hiStory.story_name} image={hiStory.image} total_chapters={hiStory.total_chapters} introduce={hiStory.introduce} owner={hiStory.user} />
        )
      })
    }
  </Row>

</Col>
</Row>
</div>
<div className='profile-tabs'>
  <div className='tab_book_profile'>
    <Tabs defaultActiveKey="1">
      <Tabs.TabPane tab="Your books" key="1">
        <div>
          <Row justify="start" gutter={16}>
            {
              myStory.map((myStory) => {
                return (
                  <Card index={myStory.id} story_name={myStory.story_name} image={myStory.image} total_chapters={myStory.total_chapters} introduce={myStory.introduce} owner={myStory.user} />
                )
              })
            }
          <Col flex={2} sm={8} xs={12}>
            <div className='create_book--item' onClick={() => router.push(`books/create/${name}`)}>
              <PlusCircleOutlined />

```



```

        </div>
      </Col>
    </Row>
  </div>
</Tabs.TabPane>
<Tabs.TabPane tab="Saved" key="2">
  <div>
    <Row justify="start" gutter={16}>
      {
        mySave.map((mySave) => {
          return (
            <Card index={mySave.id} story_name={mySave.story_name} image
={mySave.image} total_chapters={mySave.total_chapters} introduce={mySave.introduce} owner={mySave.
user} />
          )
        })
      }
    </Row>
  </div>
</Tabs.TabPane>
</Tabs>
</div>
</div>
</div>
</Main>


```


4.7. Trang thêm Story:

4.7.1. Giao diện:

[Home](#)[Category](#)[About](#)

[Book](#) / [Create & Edit](#)



Main content 

Name book :

Category

Ngôn Tình

▼

Chap estimate :

Add story

»

EDIT & ADD CHAP

«

Select chap

Select Chapter

▼

Name chap

Content



BOOK LISTEN



Read
Book
Creator
Category

Policy Rules
Cookie Policy
Privacy Policy

Contact
FAQ
Support

Name Inc 2022 Copyright

Hình 4-7 Giao diện Trang thêm Story

4.7.2. Mô tả:

- Trang web dùng để tạo truyện mới và thêm các chương cho truyện.

4.7.3. Kỹ thuật thiết kế:

- Thêm, Xóa, Sửa sử dụng chung một component.

4.7.4. Code thiết kế:

```
<Main>
  <div className="book__edit_book">
    <Breadcrumb className="book__detail__breadcurmb">
      <Breadcrumb.Item href="/">
        <ReadOutlined className="icon__book" />
        <span>Book</span>
      </Breadcrumb.Item >
      <Breadcrumb.Item > Create & Edit</Breadcrumb.Item>
    </Breadcrumb>
    <Row>
      <Col lg={58} sm={24} xs={24}>
        <div className="component__detail__edit">
          <Row>
            <Col lg={8} sm={12} xs={24}>
              <input type={'file'} style={{ display: 'none' }} id='update__baner__book' onChange={updateImage} />
              {
                imgBook ? (
                  <div className="component__detail__edit__img">
                    <Image src={imgBook} layout='fill' alt='hình bia sach' height={310} />
                    <label htmlFor="update__baner__book"><div className="component__detail__edit__content--icon icon__edit" >
                      <EditFilled />
                    </div></label>
                  </div>
                ) : (
                  <div>
                    <label htmlFor="update__baner__book"><div className="upload__img__book">
                      <PlusCircleOutlined />
                    </div></label>
                  </div>
                )
              }
            </Col>
          </Row>
        </div>
      </Col>
    </Row>
  </div>
</Main>
```

```

<div className="component__detail__edit__info">
  <Row>
    <Col span={24}>
      <div className="component__detail__edit__information">
        <div className="input__name__book">
          <p className="title__input__name">Name book :</p>
          <Input value={nameBook} id="nStory" className="input__name__book--
input" placeholder="Your book name" onChange={event => setNameBook(event.target.value)} />
          <p id="cStory" className="title__input__name">Category</p>
          <Select value={parseInt(cata)} className="select__chap__edit" placeholder="Choose your category" onChange={handleChange} options={category}></Select>
          <br><br><br></br>
          <span className="title__input__name">Chap estimate :</span>
          <br><br><br></br>
          <Input value={estimate} placeholder="Chap estimate" id="iStory" className="input__name__book--
input" type="number" min="0" onChange={event => setEstimate(event.target.value)} />
        </div>
      </div>
    </Col>
  </Row>
</div>
</Col>
<Col lg={16} sm={12} xs={24}>
  <div className="component__detail__edit__content">
    <h3 className="component__detail__edit__content--title">Main content</h3>
    <span className="component__detail__edit__content--
icon" onClick={() => { setAttribute(false) }} >
      <EditFilled />
    </span>
    <TextArea value={mainContent} id="mStory" onClick={() => { setAttribute(true) }} className="component__detail__edit__content--
detail" defaultValue={mainContent} allowClear readOnly onChange={event => setMainContent(event.target.value)} />
  </div>
  <div className="component__detail__edit__content--btn">
    <Button hidden id="btnDelStory" className="btn__cancel__edit" onClick={() => { delStory() }} >Delete</Button>
    <Button id="btnAddStory" className="btn__save__add" onClick={() => { addStory() }} >Add story</Button>
    <Button hidden id="btnEditStory" className="btn__save__edit" onClick={() => { editStory() }} >Edit story</Button>
    <Button hidden id="btnSaveStory" className="btn__save__edit" onClick={() => { saveStory() }} >Save</Button>
  </div>
</Col>

```



```

        </div>
      </Col>
    <Col span={24}>
      <Row>
        <Col span={24}>
          <div className="component__detail__edit--addchap">
            <TextHeading>Edit & add chap</TextHeading>
            </div>
          </Col>
        </Row>
        <div className="component__detail__edit--choosechap">
          <h3 className="component__detail__edit__content--title" style={{ display: 'block' }}>Select chap</h3>
          <Select
            id="inChap"
            value={chapterIsCho}
            onChange={nameChapter}
            disabled={btnSelect}
            className=" select__chap__edit"
            placeholder={showIndexChap}
            popupClassName='select__chap__edit--dropdown'
            dropdownRender={(menu) => (
              <>
                {menu}
                <Input placeholder="Chapter index" id="id_chapter" className="input__name__book--input" type="number" min="0" onClick={nameChapter} onChange={event => setIndexChap(event.target.value)} />
              </>
            )}
            <Button
              id="btnInsertChap"
              type="text" icon={<PlusOutlined />}
              onClick={addItem}>
                Add Chap
              </Button>
            </>
          </Select>
          <br></br><br></br>
          <h3 className="component__detail__edit__content--title">Name chap</h3>

```

```

        <Input disabled={select} id="nStory" className="input__name__book--
input" placeholder="Your chap name" value={nameChap} onChange={event => setNameChap(event.target.v
alue)} />

        <h3 className="component__detail__edit__content--title">Content</h3>

        <TextArea disabled={select} className="component__detail__edit__content--
detail component__detail__edit__content--
content" value={content} allowClear onChange={event => setContent(event.target.value)} />
    </div>

    <div className="component__detail__edit__content--btn">

        <Button hidden disabled={btnCancel} id="btnCancel" className="btn__cancel__edit" o
nClick={delChapter}>Delete</Button>

        <Button hidden disabled={btnAdd} id="btnAdd" className="btn__save__add" onClick={a
ddChapter}>Add</Button>

        <Button hidden disabled={btnSave} id="btnSave" className="btn__save__edit" onClick
={editChapter}>Save</Button>

    </div>

</Col>

</Row>

</div>

    </Col>

</Row>

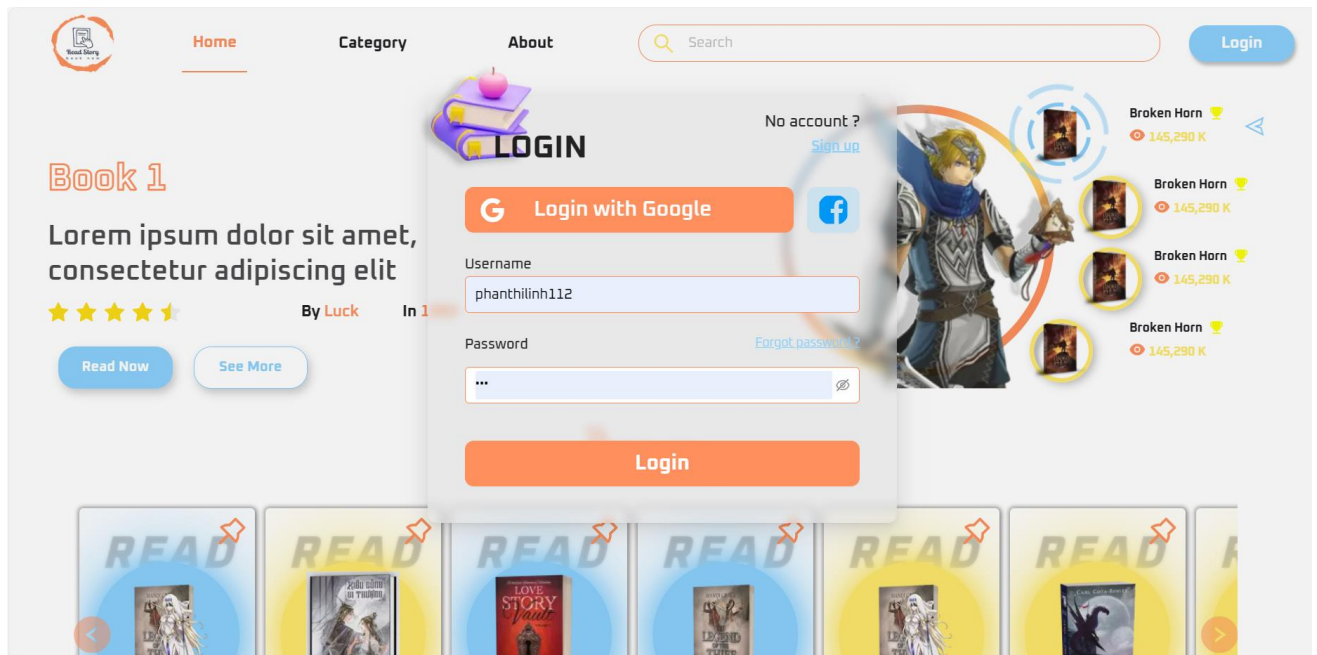
</div>

</Main>

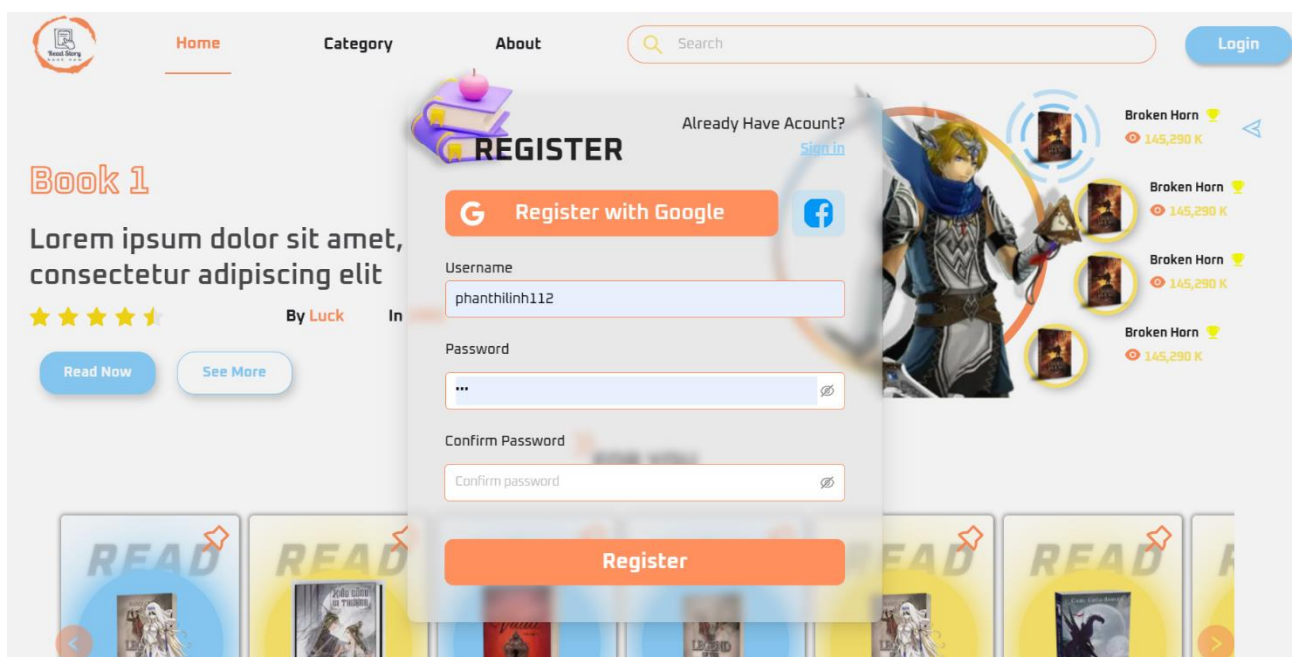
```

4.8. Trang Đăng nhập, đăng ký:

4.8.1. Giao diện:



Hình 4-8 Giao diện Trang Login



Hình 4-9 Giao diện Trang đăng ký

4.8.2. Mô tả:

- Popup dùng để đăng nhập, đăng ký.

4.8.3. Kỹ thuật thiết kế:

- Tái sử dụng popup đăng nhập cho đăng ký.

4.8.4. Code thiết kế:

```
<div className='container__modal_login'>
  <Row>
    <Col span={20}>
      <Button
        className='btn__container_loginGG'
        icon={<GoogleOutlined />}
      >
        {isModalLoginOpen
          ? 'Login with Google'
          : 'Register with Google'}
      </Button>
    </Col>
    <Col span={4}>
      <Button
        className='btn__container_loginFB'
        icon={<FacebookFilled />}
      ></Button>
    </Col>
  </Row>
  <div className='container__modal_login-input'>
    <label className='container__modal_login-input-text'>Username</label>
    <Input
      id="username"
      className='input-login'
      placeholder='Username'
      // field={username}
      // change={event => setUsername(event.target.value)}
    />
  </div>
  <div className='container__modal_login-input'>
    <div className='container__modal_login-forgot'>
      <label className='container__modal_login-input-text'>
        Password
      </label>
      {isModalLoginOpen && <span>Forgot password ?</span>}
    </div>
  </div>
</div>
```

```

<Input.Password
  id='password'
  className='input-login'
  placeholder='Your password'
/>
</div>
{!isModalLoginOpen && (
  <div className='container__modal__login-input'>
    <div className='container__modal__login-forgot'>
      <label className='container__modal__login-input-text'>
        Confirm Password
      </label>
    </div>
    <Input.Password
      id='re_password'
      className='input-login'
      placeholder='Confirm password'
    />
  </div>
)}

<Button className='btn__modal__login' onClick={() => { checkLogin() }}>

  {isModalLoginOpen ? 'Login' : 'Register'}
</Button>
</div>

```

CHƯƠNG 5: KẾT LUẬN

5.1. Ưu điểm của đồ án:

- Giao Diện Thân Thiện và Dễ Sử Dụng: Thiết kế giao diện đơn giản, trực quan giúp người dùng dễ dàng tìm kiếm và đọc truyện.
- Cung cấp chứng năng tạo truyện giúp ai cũng có thể tạo và chia sẻ truyện của mình.
- Cung cấp các trải nghiệm cá nhân như lưu lịch sử đọc, lưu truyện yêu thích giúp tối ưu hoá trải nghiệm người đọc.
- Giao diện linh hoạt, có thể hoạt động trên nhiều kích thước màn hình khác nhau.

5.2. Hạn chế của đồ án:

- Khả Năng Tải Trang Chậm: Tốc độ tải trang còn tương đối chậm.
- Giới Hạn Tính Năng Tùy Chỉnh: Người dùng có thể cảm thấy hạn chế về các tùy chọn cá nhân hóa giao diện hoặc trải nghiệm đọc.
- Giới Hạn về Nội Dung: Website chưa có kho truyện nhiều và đa dạng.
- Tính Năng Tương Tác chưa cao: Người dùng chưa có thể bình luận, đánh giá truyện, để tạo cộng đồng đọc giả sôi nổi.

5.3. Hướng phát triển của đồ án:

- Bổ sung các chức năng còn thiếu như bình luận truyện, review truyện nhằm tăng tương tác với người dùng.
- Bổ sung thêm nhiều Template để thay đổi phù hợp với sở thích của mỗi người, phù hợp với chủ đề.
- Bổ sung thêm tính năng đọc truyện thành tiếng với nhiều giọng đọc khác nhau.
- Nâng Cao Chất Lượng Nội Dung: Thiết lập hệ thống kiểm soát nội dung giúp đảm bảo chất lượng truyện.
- Phát triển thêm App di động để tiếp cận nhiều người đọc hơn.
- Xây dựng công cụ để tự động lấy thêm truyện từ các nguồn miễn phí hoặc xin phép để tạo tính đa dạng cho trang truyện.
- Cải Thiện Hiệu Năng Website: Tối ưu hóa mã nguồn để giảm thời gian tải trang.
- Phát Triển Tính Năng Đọc Offline: Cho phép người dùng tải truyện và đọc ngoại tuyến.

- Mở Rộng Hệ Thống Đề Xuất Truyện: Áp dụng AI để gợi ý truyện dựa trên sở thích và lịch sử đọc của người dùng.

TÀI LIỆU THAM KHẢO

- [1] “Home - Django REST framework”. Truy cập: 8 Tháng Giêng 2024. [Online]. Available at: <https://www.django-rest-framework.org/>
- [2] “Next.js by Vercel - The React Framework”. Truy cập: 20 Tháng Giêng 2024. [Online]. Available at: <https://nextjs.org/>
- [3] “Introducing Hooks – React”. Truy cập: 5 Tháng Giêng 2024. [Online]. Available at: <https://legacy.reactjs.org/docs/hooks-intro.html>
- [4] “SQLite Documentation”. Truy cập: 20 Tháng Giêng 2024. [Online]. Available at: <https://www.sqlite.org/docs.html>
- [5] “Free Design Tool for Websites, Product Design & More”, Figma. Truy cập: 4 Tháng Giêng 2024. [Online]. Available at: <https://www.figma.com/design/>
- [6] “Các tính năng của Photoshop | Adobe”. Truy cập: 11 Tháng Giêng 2024. [Online]. Available at: https://www.adobe.com/vn_vi/products/photoshop/features.html
- [7] “DALL·E 2”. Truy cập: 1 Tháng Giêng 2024. [Online]. Available at: <https://openai.com/dall-e-2>