

HENRIQUE
LOPES

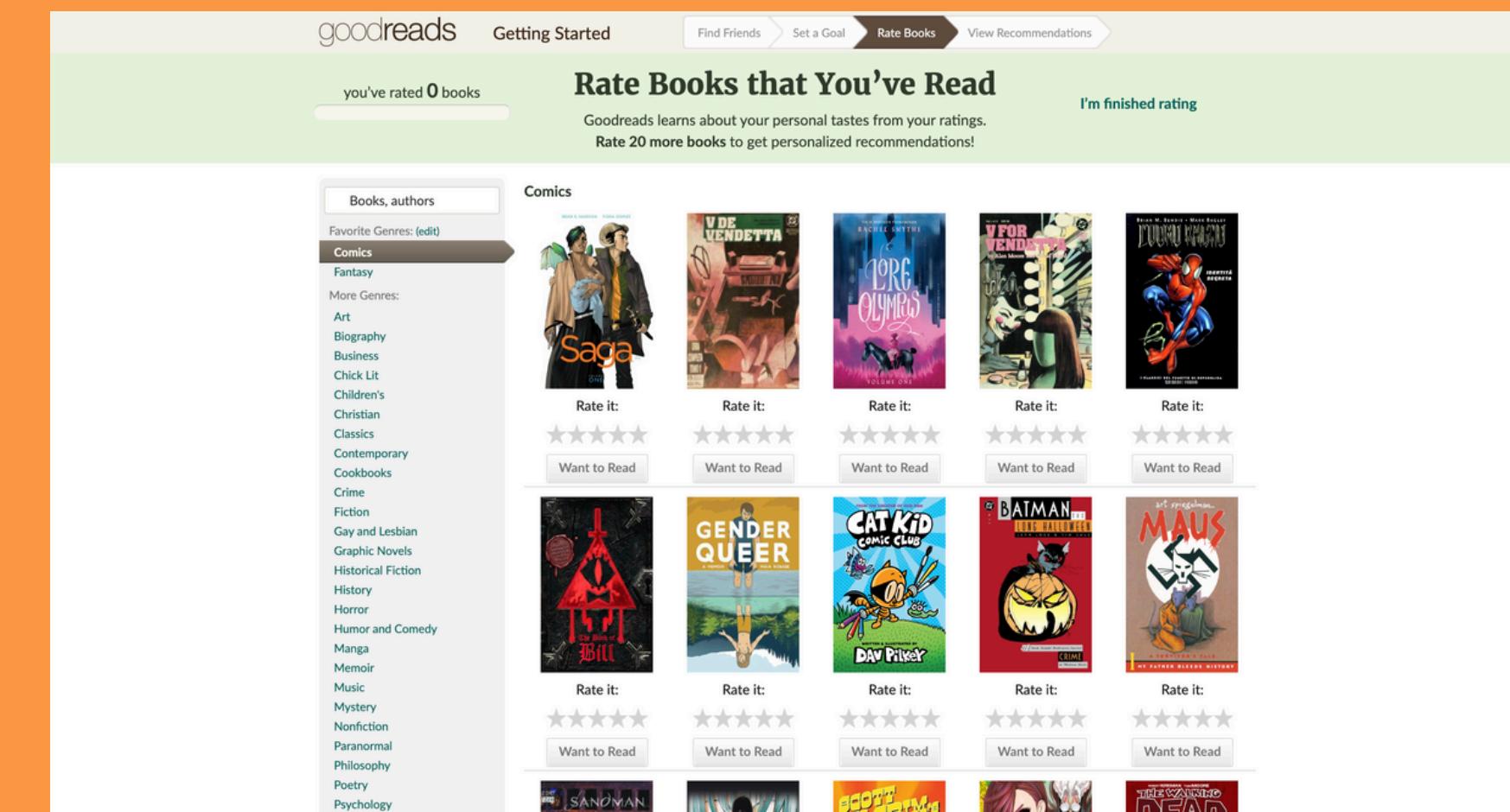
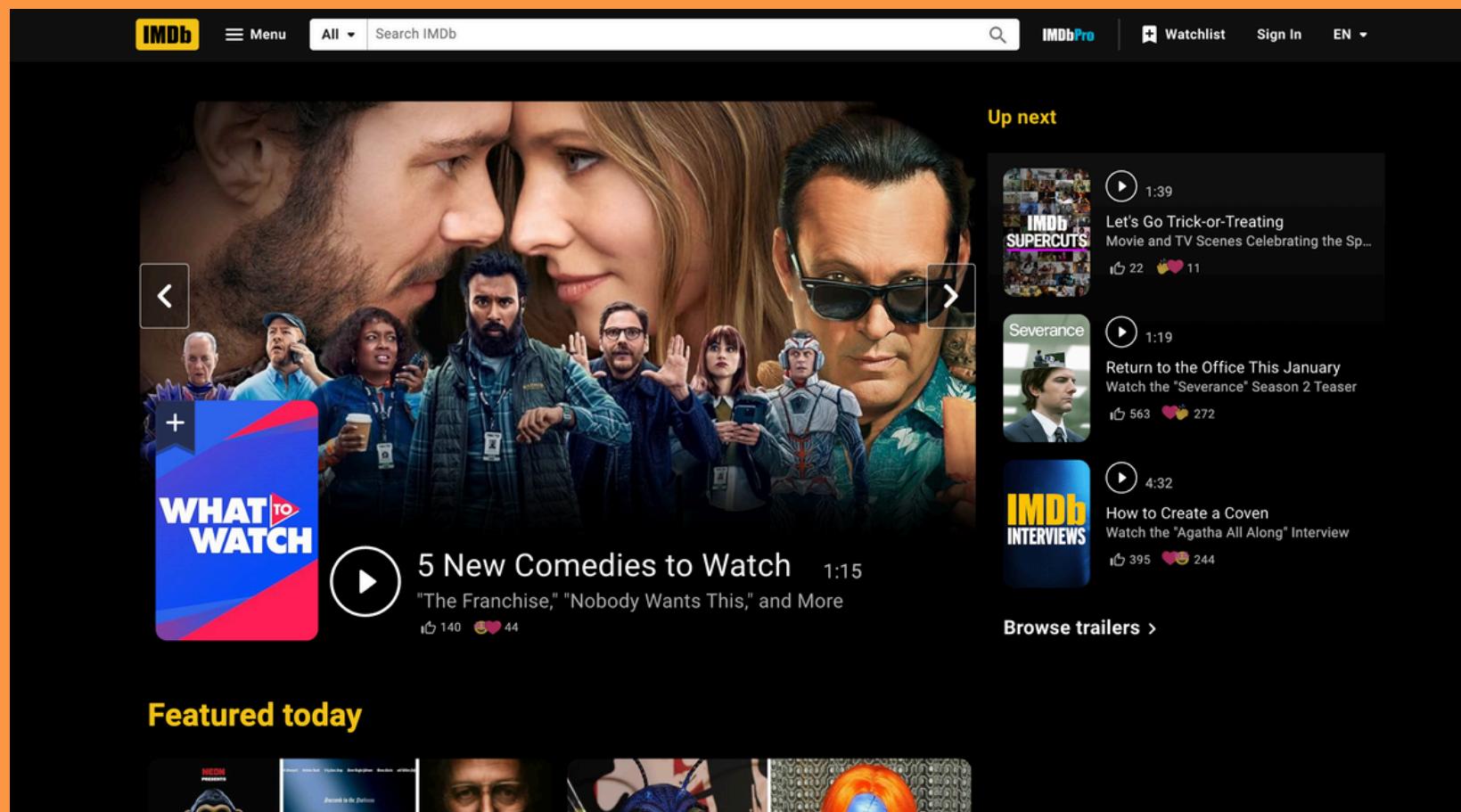
FEUP

STORYSHOW

INTRODUÇÃO

O **StoryShow** é uma plataforma de catálogo interativo para filmes, séries e livros. O objetivo é permitir que os usuários explorem conteúdos organizados por género, e que tenham acesso a recomendações baseadas nas avaliações de especialistas e leiam sinopses de uma forma clara e intuitiva. Este relatório cobre o desenvolvimento da arquitetura de informação, o design das páginas principais, fluxo de navegação e elementos importantes implementados no código de programação deste website.

ANÁLISE DE PROJETOS SIMILARES



IMDb: Focado em filmes e séries, oferece sinopses, trailers e avaliações. A interface é direta, com elementos visuais que atraem o usuário para o conteúdo.

Goodreads: Orientado a livros, o Goodreads permite que os usuários avaliem e organizem suas leituras. Possui uma navegação intuitiva e filtros para explorar gêneros variados.

PERSONAS

Para definir o público-alvo, criamos duas personas principais para o **StoryShow**:

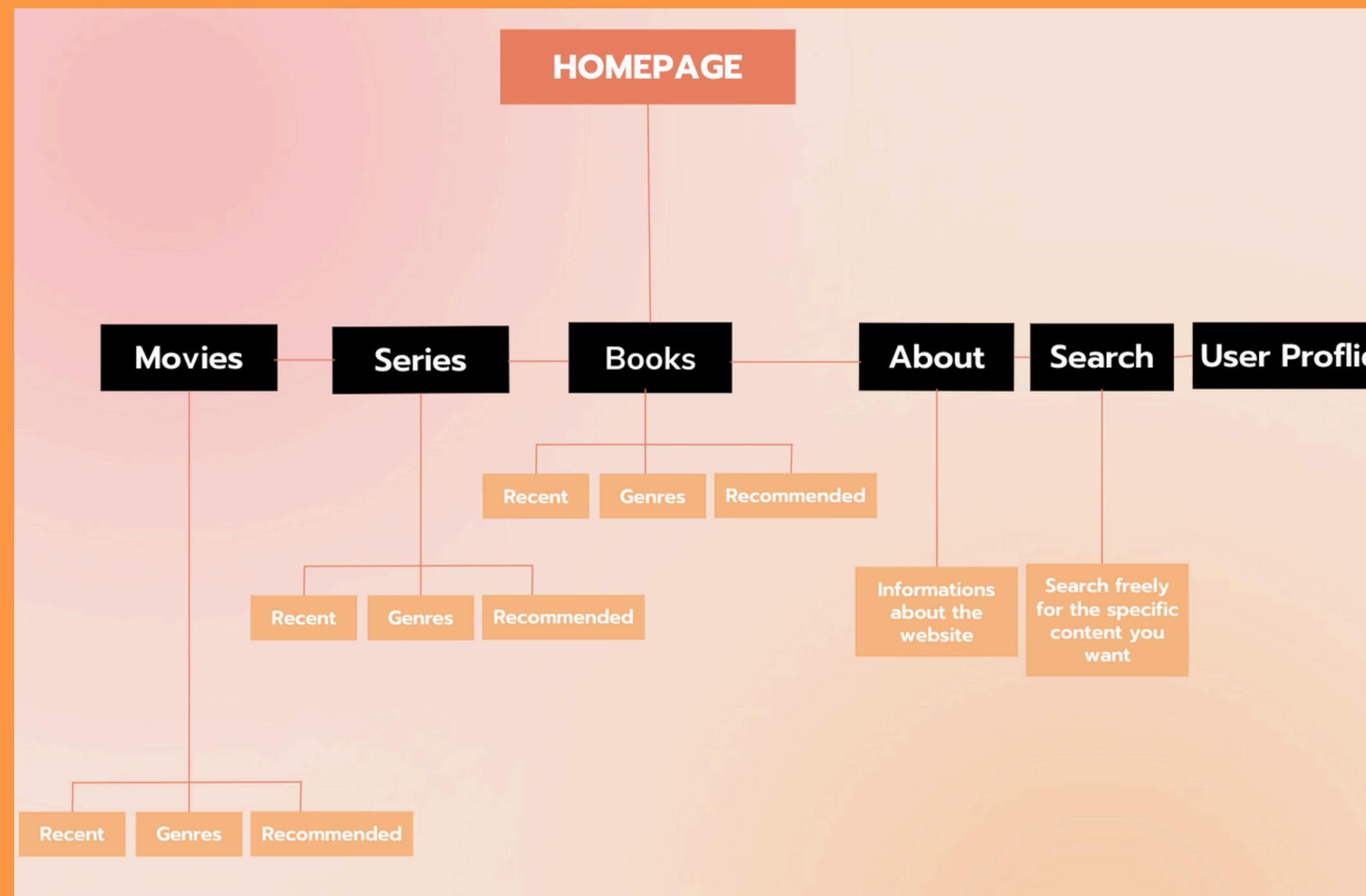
Lucas, 30 anos, fã de ficção científica e fantasia

- **Objetivos:** Explorar filmes, séries e livros em seu género favorito, com recomendações personalizadas.
- **Comportamento no Site:** Consulta semanalmente, navega pelos géneros.
- **Necessidades:** Interface intuitiva para explorar avaliações e listas de conteúdos favoritos.

Ana, 42 anos, exploradora de novos autores e diretores independentes

- **Objetivos:** Descobrir filmes e livros independentes e ver sugestões de novos talentos.
- **Comportamento no Site:** Explora recomendações e utiliza filtros de género.
- **Necessidades:** Sistema de recomendação focado em títulos menos populares e informações detalhadas sobre criadores.

MAPA DO SITE



IDENTIFICAÇÃO DE CONTEÚDO

| Tipo de Conteúdo | Título | Ano | Gênero | Sinopse | Diretores | Atores/Atrizes | Avaliação IMDb | Trailer URL | Autor(es) |
|------------------|-----------------------|------|-------------------------|--|-------------------|---|----------------|-----------------------------------|---------------|
| Filme | Interstellar | 2014 | Ficção Científica | Em uma Terra morrendo, uma equipe busca outro planeta. | Christopher Nolan | Matthew McConaughey, Anne Hathaway | 8.6 | link para trailer | - |
| Série | Stranger Things | 2016 | Fantasia, Suspense | Um garoto desaparece, revelando um mundo paranormal. | Irmãos Duffer | Winona Ryder, David Harbour | 8.7 | link para trailer | - |
| Livro | 1984 | 1949 | Distopia | Em uma sociedade opressiva, Winston tenta resistir. | - | - | - | - | George Orwell |
| Filme | Inception | 2010 | Ação, Ficção Científica | Um ladrão invade sonhos para plantar uma ideia. | Christopher Nolan | Leonardo DiCaprio, Joseph Gordon-Levitt | 8.8 | link para trailer | - |
| Livro | Orgulho e Preconceito | 1813 | Romance | Uma jovem enfrenta pressões sociais para se casar. | - | - | - | - | Jane Austen |

ESTRUTURA DE CONTEÚDO

O **StoryShow** utiliza uma estrutura **hierárquica** para organizar o conteúdo, o que facilita a navegação entre as diferentes seções. A hierarquia é simples e intuitiva, oferecendo ao usuário um acesso organizado e lógico às principais áreas do site.

- Nível 1 (Home Page): A Home Page serve como ponto de partida, com links diretos para as áreas principais.
- Nível 2 (Principais Categorias): Inclui as seções Filmes, Séries, Livros, Search, User Profile e About.
- Nível 3 (Subcategorias): Dentro de cada categoria principal, o usuário pode escolher Géneros (Genre Page) específicos (Ação, Romance, etc.), vizualizar os conteúdos mais recentes (Recent Page) e mais recomendados (Recommended Page), explorar sinopses e trailers.

Esta estrutura facilita a exploração do conteúdo, organizando os itens em níveis lógicos para otimizar a experiência do usuário.

WIREFRAMES E DESIGN DAS PÁGINAS

StoryShow

Movies

Inception The Matrix Avengers: Endgame Parasite



StoryShow

Movies

Inception The Matrix Avengers: Endgame Parasite



StoryShow

PARASITE

RATING: 8.6/10

A DARK SOCIAL SATIRE ABOUT CLASS DISPARITY...

THE TRUMAN SHOW

RATING: 8.2/10

A MAN DISCOVERS HIS LIFE IS A REALITY TV SHOW...

THE WOLF OF WALL STREET

RATING: 8.2/10

THE RISE AND FALL OF STOCKBROKER JORDAN BELFORT...

LA LA LAND

RATING: 8.0/10

A JAZZ MUSICIAN AND AN ASPIRING ACTRESS FALL IN LOVE IN LOS ANGELES...

THE BIG LEBOWSKI

RATING: 8.1/10

JEFF 'THE DUDE' LEBOWSKI IS MISTAKEN FOR A MILLIONAIRE OF THE SAME NAME...



StoryShow

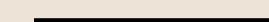
Movies Genres

| | | | | | | | |
|----------|-----------|--------------------|-------------|-----------|---------|----------|------------|
| Action | Animation | Adventure | Comedy | Crime | Drama | Fantasy | Horror |
| Mystery | Romance | Science Fiction | Sci-Fi | Biography | History | Thriller | True Story |
| Classics | Dystopian | Historical Fiction | Non-Fiction | Self-Help | Science | Musical | Western |

WIREFRAMES E DESIGN DAS PÁGINAS

Movies

Inception The Matrix Avengers: Endgame Parasite



Search for Movies, Series, or Books

the SEARCH

the

THE MATRIX THE DARK KNIGHT THE SHAWSHANK THE GODFATHER THE GODFATHER: PART II

Login

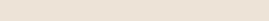
Username: henrique9

Password:

Login

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

Or [ENTER AS GUEST](#)



Welcome, henrique9!

This is your user profile. You can view and update your account details here.

Content Suggestions

Do you have any suggestions for content you'd like us to add to the website? Feel free to let us know below!

Enter your suggestion or feedback here...

Submit Feedback

LOGOUT

ESTRUTURA DE NAVEGAÇÃO

Descrição do Wireflow

1. Log In → Home Page

- Usuário faz login e é redirecionado para a Home Page, embora exista a opção de entrar como convidado.

2. Home Page → Principais Seções

- Filmes, Séries, Livros, About, Search e User Profile estão no mesmo nível e acessíveis diretamente.
- Recomendações: Acessível por secções (Livros, Séries e Filmes)

3. Retorno à Home

- Cada seção contém um caminho claro de retorno à Home Page, pois basta carregar no logótipo do website.

TÉCNICAS EM CSS

Flexbox:

Utilizado para estruturas de disposição como barras de navegação, contentores de pesquisa e grelhas de itens (por exemplo, display: flex em .navbar ou .items-grid). O Flexbox facilita o alinhamento e a distribuição de itens em linhas ou colunas, garantindo um espaçamento consistente e um comportamento reativo.

Layout de grade:

Utilizado para disposições estruturadas de conteúdo (por exemplo, géneros ou cartões de itens). A grelha é ideal para criar disposições uniformes com controlo preciso sobre linhas e colunas.

Seletores CSS:

Nomeação de classes: Nomes de classe no estilo BEM (Block Element Modifier) para maior clareza e reutilização, por exemplo, navbar, content-container. Os estilos definidos são definidos em cascata de forma lógica, garantindo consistência e evitando conflitos. Os estilos em linha são evitados para aproveitar a cascata da folha de estilo.

Estratégia de design responsivo:

Ajustar os layouts para ecrãs mais pequenos. Utilizar larguras percentuais e ajuste automático ou preenchimento automático na grelha para uma colocação flexível dos itens. Assegurar que o sítio se adapta de forma harmoniosa a vários tamanhos de ecrã.

CÓDIGO EM CSS

Design Responsivo (.item-card)

```
@media (max-width: 768px) {  
    .recommended-page h1,  
    .genre-page h1,  
    .recent-page h1 {  
        font-size: 2rem;  
    }  
  
    .item-card {  
        width: 100%;  
        max-width: 100%;  
    }  
}
```

Animações (fadeIn)

```
@keyframes fadeIn {  
    0% {  
        opacity: 0;  
    }  
    100% {  
        opacity: 1;  
    }  
}
```

Layout em grade

```
.grid {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 20px;  
    justify-content: space-around;  
    padding-top: 100px; /* Ensure the grid is below the navbar */  
}
```

Decoração do Navbar (background-color, box-shadow)

```
.navbar {  
    position: fixed;  
    top: 0;  
    left: 0;  
    right: 0;  
    display: flex;  
    justify-content: flex-start; /* Align all elements to the left */  
    align-items: center;  
    background-color: #ed9121;  
    width: 100%;  
    height: 90px;  
    padding: 0 20px; /* Add padding for spacing */  
    z-index: 1000;  
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);  
}
```

Elementos Interativos (hover)

```
.nav-button:hover {  
    color: #fff;  
    background-color: #000;  
}
```

ESTRUTURA EM ÁRVORE (HTML)

```
<html>
  └─ <head>
    |   └─ <meta> (Character encoding, viewport)
    |   └─ <title> (Dynamic Page Titles: "Search", "Item Details", etc.)
    |   └─ <link> (Stylesheets: project.css, Google Fonts)
    |   └─ <script> (JavaScript Files: script.js, data.js)
    |
    └─ <body>
      └─ <nav class="navbar"> (Navigation Bar)
        └─ <a class="logo">
          └─ <img> (Logo Icon)
          └─ "StoryShow" (Website Name)
        └─ <ul> (Navigation Links – Optional)

      └─ <main> (Main Content Section)
        └─ <section id="content" class="content-container">
          └─ Dynamic Content (Rendered by JavaScript, e.g., item details)
          └─ <img> (Item Images)
          └─ <div class="details">
            └─ <h1> (Item Title)
            └─ <p> (Rating, Release Year, Genres)
            └─ <p> (Description)
            └─ <iframe> (Embedded Trailer)

        └─ <section id="search-results" class="items-grid"> (Search Results)
          └─ <a class="item-card">
            └─ <img> (Item Thumbnail)
            └─ <div class="item-info">
              └─ <h3> (Title)
              └─ <p> (Snippet Description)

        └─ <div class="search-container"> (Search Section)
          └─ <input id="search-bar"> (Search Query Field)
          └─ <button id="search-button"> (Trigger Search)
```

CÓDIGO EM HTML

Define um link para a Home Page em todas as páginas

```
<div class="navbar">
    <a href="index.php" class="logo">
         StoryShow
    </a>
</div>
```

Captura dados do usuário para login

```
<form method="POST" action="login.php">
    <div class="input-group">
        <label for="username">Username</label>
        <input type="text" id="username" name="username" required>
    </div>

    <div class="input-group">
        <label for="password">Password</label>
        <input type="password" id="password" name="password" required>
    </div>
```

CÓDIGO EM HTML

Estruturação de menus para facilitar a navegação, facilita a interação e a acessibilidade no site (classe dropdown, hoverable menu)

Exibição Dinâmica de Detalhes

```
<!-- Main Content Area -->
<main>
|   <section id="content" class="content-container">
|       <!-- Content will be dynamically injected here -->
|   </section>
| </main>
```

```
<div class="navbar">
|   <a href="index.php" class="logo">
|        StoryShow
|   </a>
|   <ul class="nav">
|       <li class="dropdown">
|           <a href="#movies" class="nav-button">Movies</a>
|           <div class="dropdown-content">
|               <a href="template.html?item=recent-movies">Recent</a>
|               <a href="template.html?item=genres-movies">Genres</a>
|               <a href="template.html?item=recommended-movies">Recommended</a>
|           </div>
|       </li>
|       <li class="dropdown">
|           <a href="#series" class="nav-button">Series</a>
|           <div class="dropdown-content">
|               <a href="template.html?item=recent-series">Recent</a>
|               <a href="template.html?item=genres-series">Genres</a>
|               <a href="template.html?item=recommended-series">Recommended</a>
|           </div>
|       </li>
|       <li class="dropdown">
|           <a href="#books" class="nav-button">Books</a>
|           <div class="dropdown-content">
|               <a href="template.html?item=recent-books">Recent</a>
|               <a href="template.html?item=genres-books">Genres</a>
|               <a href="template.html?item=recommended-books">Recommended</a>
|           </div>
|       </li>
|       <li><a href="#about" class="nav-button">About</a></li>
|       <li><a href="search.html" class="nav-button">Search</a></li>
|       <li><a href="profile.php" class="nav-button">User Profile</a></li>
|   </ul>
| </div>
```

TÉCNICAS EM JAVASCRIPT

Renderização de conteúdo dinâmico:

Funções utilizadas para gerar e apresentar conteúdo dinamicamente com base na interação do utilizador ou em parâmetros de URL. Reduz a redundância de HTML estático e permite a reutilização de modelos para itens semelhantes.

Tratamento de eventos:

Utilizado para interações do utilizador, como a funcionalidade de pesquisa ou a filtragem de géneros. Melhora a interatividade e a capacidade de resposta.

Debouncing:

Evita chamadas API excessivas durante a introdução de pesquisa. Melhora o desempenho e evita atrasos.

Filtragem dinâmica:

Utilizada para filtragem baseada no género e na classificação. Permite actualizações em tempo real sem recarregar a página.

CÓDIGO EM JAVASCRIPT

Manipulação de Parâmetros de URL

```
document.addEventListener('DOMContentLoaded', () => [
  console.log('Page loaded and script running');
  const params = new URLSearchParams(window.location.search);
  const itemId = params.get('item');
```

Renderização Dinâmica de Itens

```
// Render item details dynamically
function renderItemDetails(itemData) {
  const content = document.getElementById('content');
  content.className = 'details-page'; // Add a class for styling
```

Busca com Filtro

```
function performSearch() {
  const query = document.getElementById('search-bar').value.toLowerCase().trim();
  const resultsContainer = document.getElementById('search-results');
  resultsContainer.innerHTML = '';
```

CÓDIGO EM JAVASCRIPT

Debouncing (Evitar Requisições Excessivas)

```
function debounce(func, delay) {
  return function () {
    const context = this;
    const args = arguments;
    clearTimeout(debounceTimer);
    debounceTimer = setTimeout(() => func.apply(context, args), delay);
  };
}
```

Criação Dinâmica de Elementos

```
// Utility: Create item card HTML
function createItemCard(item) {
  return `
    <a href="template.html?item=${item.id}" class="item-card">
      
      <div class="item-info">
        <h3>${item.title}</h3>
        <p><strong>Rating:</strong> ${item.rating}</p>
        <p>${item.description.substring(0, 100)}...</p>
      </div>
    </a>
  `;
}
```

CÓDIGO EM PHP

Modelos dinâmicos:

Os modelos geram páginas dinamicamente para uma estrutura consistente. Simplifica as actualizações e layouts consistentes.

Validação e segurança:

Valida a entrada do utilizador e evita a injeção de SQL com instruções preparadas. Garante a segurança e a fiabilidade.

Gestão de sessões:

Acompanha a autenticação do utilizador e o envio de feedback. Melhora a experiência do utilizador através da manutenção dos estados de início de sessão.

Conteúdo dinâmico de bases de dados:

Utilizado para armazenamento e recuperação de dados de utilizadores e comentários. Integra actualizações de dados em tempo real.

CÓDIGO EM PHP

- Uso de `password_hash` para salvar senhas de forma segura.
- Prevenção de SQL Injection com prepared statements (`$stmt->bind_param()`).
- Validação no servidor, como verificar se o nome de usuário já existe e se as senhas coincidem.

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get form data
    $username = $_POST['username'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];

    // Check if username already exists
    $sql = "SELECT id FROM users WHERE username = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        // If username already exists
        $error = "Username is already taken. Please choose another.";
    } elseif ($password !== $confirm_password) {
        // Check if passwords do not match
        $error = "Passwords do not match.";
    } else {
        // If username is unique and passwords match
        $hashed_password = password_hash($password, PASSWORD_DEFAULT);

        $sql = "INSERT INTO users (username, password) VALUES (?, ?)";
        $stmt = $conn->prepare($sql);
        $stmt->bind_param("ss", $username, $hashed_password);

        if ($stmt->execute()) {
            $_SESSION['username'] = $username;
            $_SESSION['user_id'] = $conn->insert_id; // Store user ID
            header('Location: index.php');
            exit();
        } else {
            $error = "Error registering user. Please try again.";
        }
    }
}
```

- Criação de uma conexão segura e confiável com a base de dados MySQL usando `mysqli`.
- Verificação de erros na conexão, garantindo que o sistema não continue caso haja falha.

```
<?php
$servername = "localhost";
$username = "root";
$password = ""; // Leave empty if no password is set
$dbname = "my_website"; // Your database name

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname)

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
?>
```

CÓDIGO EM PHP

- Registo do feedback enviado por usuários (Guests não podem submeter feedback)
- Verificação de campos vazios para evitar envio de dados incompletos.
- Uso do NOW() para salvar automaticamente a data/hora de criação.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['feedback'])) {  
    // Check if the user is logged in (has user_id) or is a guest  
    if (isset($_SESSION['user_id'])) {  
        $user_id = $_SESSION['user_id']; // Get user ID for logged-in users  
    } else {  
  
        // For guests, handle them separately  
        $user_id = null; // Optionally, handle guest feedback differently, or set a default ID  
    }  
  
    $feedback = trim($_POST['feedback']); // Get the feedback from the form  
  
    // Ensure feedback is not empty  
    if (!empty($feedback)) {  
        // Insert feedback into the database  
        $sql = "INSERT INTO feedback (user_id, feedback_text, created_at) VALUES (?, ?, NOW())";  
        $stmt = $conn->prepare($sql);  
        $stmt->bind_param("is", $user_id, $feedback);  
        $stmt->execute();  
  
        // Redirect back to profile page with success message  
        $_SESSION['feedback_success'] = "Thank you for your feedback!";  
        header('Location: profile.php');  
        exit();  
    } else {  
        // If feedback is empty  
        $_SESSION['feedback_error'] = "Please provide some feedback!";  
        header('Location: profile.php');  
        exit();  
    }  
}
```

- Autenticação de usuários utilizando password_verify para verificar senhas hash.
- Manipulação de sessões (\$_SESSION) para manter o usuário logado.

```
$sql = "SELECT * FROM users WHERE username = ?";  
$stmt = $conn->prepare($sql);  
$stmt->bind_param("s", $username);  
$stmt->execute();  
$result = $stmt->get_result();  
  
if ($result->num_rows > 0) {  
    $user = $result->fetch_assoc();  
    if (password_verify($password, $user['password'])) {  
        $_SESSION['username'] = $username;  
        $_SESSION['user_id'] = $user['id']; // Store user_id in session  
  
        // Redirect to the main page  
        header('Location: index.php');  
        exit();  
    } else {  
        $error = "Invalid password.";  
    }  
} else {  
    $error = "User not found.";  
}
```

BASE DE DADOS (MYSQL)

Tabela dos usuários

```
CREATE TABLE `users` (
  `id` int NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(255) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `username` (`username`)
```

Tabela de feedbacks dos usuários

```
CREATE TABLE `feedback` (
  `id` int NOT NULL AUTO_INCREMENT,
  `user_id` int DEFAULT NULL,
  `feedback_text` text NOT NULL,
  `created_at` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `feedback_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`id`)
```

BASE DE DADOS DE CONTEÚDOS

Criada em JavaScript com Identificadores únicos e campos detalhados

```
const data = {
  movies: [
    {
      id: "movie1",
      title: "Inception",
      description: "A mind-bending thriller about dream manipulation.",
      image: "imagens/inception.jpg",
      rating: "9/10",
      releaseYear: 2010,
      directors: ["Christopher Nolan"],
      actors: ["Leonardo DiCaprio", "Joseph Gordon-Levitt", "Elliot Page"],
      imdbEvaluation: "8.8/10",
      genres: ["Action", "Sci-Fi", "Thriller"],
      trailer: "https://www.youtube.com/watch?v=YoHD9XEInc0"
    },
    {
      id: "series1",
      title: "Stranger Things",
      description: "A supernatural mystery in a small town.",
      image: "imagens/stranger-things.jpg",
      rating: "8.7/10",
      releaseYear: 2016,
      directors: ["The Duffer Brothers"],
      actors: ["Millie Bobby Brown", "Finn Wolfhard", "Winona Ryder"],
      imdbEvaluation: "8.7/10",
      genres: ["Drama", "Fantasy", "Horror"],
      trailer: "https://www.youtube.com/watch?v=b9EkMc79ZSU"
    },
    {
      id: "book1",
      title: "The Hobbit",
      description: "Bilbo Baggins' epic adventure.",
      image: "imagens/hobbit.jpg",
      rating: "4.28/5",
      releaseYear: 1937,
      authors: ["J.R.R. Tolkien"],
      genres: ["Fantasy", "Adventure"]
    }
  ]
}
```

CONCLUSÃO

O StoryShow foi projetado para oferecer uma experiência de navegação organizada e intuitiva, com foco na descoberta de novos conteúdos. A estrutura hierárquica, combinada com recomendações personalizadas e filtros de género, permite uma experiência de exploração otimizada, atendendo às necessidades de diversos tipos de usuários.

Este design proporciona flexibilidade e clareza, criando uma plataforma onde o usuário pode encontrar e ver avaliação de conteúdos de filmes, séries e livros de forma prática.