

Roteiro - A Bootstrap-Based Next.js Movie Discovery Platform

Abstract

Roteiro is a sophisticated web application that combines Next.js 15's modern architecture with Bootstrap 5's robust component system to create a feature-rich movie discovery platform. The application interfaces with The Movie Database (TMDB) API to provide comprehensive movie information, advanced search capabilities, and intuitive filtering options. Notable for its responsive design and dark theme implementation, Roteiro showcases the effective integration of Bootstrap's utility-first approach with Next.js's server-side capabilities, resulting in a performant and user-friendly movie exploration experience.

Introduction

Roteiro exemplifies modern web development practices by leveraging the strengths of both Bootstrap and Next.js frameworks. The project demonstrates several key architectural decisions:

Core Technologies

- **Next.js 15**: Utilized for its file-based routing, server components, and optimized image handling
- **Bootstrap 5**: Provides responsive design system and pre-built components
- **TypeScript**: Ensures type safety and better development experience
- **TMDB API**: Powers the movie data backend

Key Features

1. Search System

- Real-time search functionality
- Advanced filtering options
- Sort capabilities
- Responsive results display

2. Movie Information

- Detailed movie pages
- Rich metadata display
- Responsive image loading
- Financial information

3. User Interface

- Custom dark theme
- Responsive design
- Accessibility features
- Loading states

Bootstrap Usage - Component Analysis

1. Root Layout (`layout.tsx`)

The root layout serves as the application's foundation, implementing Bootstrap's core structure while maintaining Next.js's optimal performance characteristics.

```
import type { Metadata } from "next";
import "./globals.css";
import Navbar from "./navbar";
import Script from "next/script";

export const metadata: Metadata = {
  title: "Roteiro",
  description: "Know your movies.",
  icons: {
    icon: "/roteiro.svg",
  },
}
```

```

};

export default function RootLayout({
  children,
}: Readonly<{ children: React.ReactNode }>) {
  return (
    <html lang="en">
      <body>
        <Navbar />
        {children}
        <Script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" />
      </body>
    </html>
  );
}

```

Key Features:

- Metadata configuration for SEO
- Bootstrap JavaScript bundle loading optimization
- Global CSS imports
- Persistent navigation structure

Bootstrap Integration:

- Core Bootstrap bundle inclusion
- Global container structure
- Responsive viewport settings
- Theme variable accessibility

2. Navigation Bar (navbar.tsx)

The navigation component implements Bootstrap's navbar system with custom modifications for better integration with Next.js's Link component.

```

<nav className="navbar navbar-expand-lg">
  <div className="container">
    <Link href="/" className="navbar-brand d-flex align-items-center">
      <Image
        src="/roteiro.svg"
        alt="Logo"
        width="30"
        height="30"
        className="d-inline-block align-text-top me-2"
      />
      roteiro
    </Link>
    {/* Navbar content */}
  </div>
</nav>

```

Bootstrap Components Used:

- `navbar` and `navbar-expand-lg` for responsive navigation
- `container` for content width control
- `navbar-toggler` for mobile menu
- `collapse` and `navbar-collapse` for responsive behavior
- Bootstrap's flexbox utilities for alignment

Custom Enhancements:

- Integration with Next.js Image component
- Custom dark theme styling
- Responsive breakpoint adjustments
- Enhanced accessibility features

3. Home Page (page.tsx)

The main interface combines multiple Bootstrap components to create a cohesive search and browse experience.

Search Implementation:

```
<form onSubmit={searchMovies} className="mb-4">
  <div className="input-group">
    <input
      type="text"
      className="form-control"
      placeholder="Search for a movie..."
      value={query}
      onChange={(e) => setQuery(e.target.value)}
      aria-label="Search for a movie"
    />
    <button
      className="btn btn-primary d-flex align-items-center gap-2"
      type="submit"
      disabled={isLoading}
    >
      <Search size={18} />
      {isLoading ? "Searching..." : "Search"}
    </button>
  </div>
</form>
```

Filter System:

```
<div className="card mb-4">
  <div className="card-body">
    <h5 className="card-title mb-3">Filters</h5>
    <div className="mb-3">
      <label className="form-label">Genres:</label>
      <div className="d-flex flex-wrap gap-2">{/* Genre checkboxes */}</div>
    </div>
    {/* Additional filters */}
  </div>
</div>
```

Bootstrap Components Used:

- container for layout management
- row and col grid system
- card components
- form-control and input-group
- btn and btn-group
- alert components
- Utility classes for spacing and alignment

Advanced Features:

- Dynamic grid responsiveness
- Conditional rendering
- Loading states
- Error handling
- Filter persistence

4. Movie Details Page (movie/[id]/page.tsx)

The details page implements a sophisticated layout for displaying comprehensive movie information.

Layout Structure:

```
<main className="container py-4">
  <div className="row">
```

```

<div className="col-md-4 mb-4 mb-md-0">{/* Poster Image */}</div>
<div className="col-md-8">{/* Movie Details */}</div>
</div>
</main>

```

Bootstrap Components Used:

- Responsive grid system
- Badge components
- Image utilities
- Spacing utilities
- Flexbox utilities
- Typography classes

Enhanced Features:

- Responsive image handling
- Currency formatting
- Dynamic content sections
- Loading states
- Error boundaries

5. Styling Implementation (`globals.css`)

The styling system extends Bootstrap's theming capabilities with custom variables and modifications.

```

:root {
  --background: #09090b;
  --foreground: #fafafa;
  --card: #18181b;
  --card-foreground: #fafafa;
  --primary: #8b5cf6;
  /* Additional theme variables */
}

/* Component Overrides */
.navbar {
  background-color: var(--card) !important;
  border-bottom: 1px solid var(--border);
}

.btn-primary {
  background-color: var(--primary) !important;
  border-color: var(--primary) !important;
  color: var(--primary-foreground) !important;
}

```

Theme Customization:

- Custom color palette
- Dark theme implementation
- Component style overrides
- Typography system
- Spacing modifications

Bootstrap Extensions:

- Custom utility classes
- Modified component styles
- Enhanced form controls
- Dark mode optimizations

Technical Features

1. Responsive Design Implementation

- Bootstrap's grid system utilizing `col-*` classes
- Mobile-first approach with progressive enhancement
- Fluid typography using responsive units
- Dynamic image sizing and optimization
- Breakpoint-specific behavior
- Touch-friendly interactions

2. Filter System Architecture

- Real-time filter application
- Multiple filter criteria support
- Sort functionality
- State management
- Performance optimization
- Filter persistence

3. Accessibility Enhancements

- ARIA labels and roles
- Semantic HTML structure
- Keyboard navigation
- Screen reader support
- Focus management
- Loading indicators
- Error messaging

4. Performance Optimizations

- Image optimization with Next.js Image
- Lazy loading implementation
- Code splitting
- Bootstrap bundle optimization
- API request management
- State updates batching

Conclusion

Roteiro demonstrates the successful marriage of Bootstrap's comprehensive component library with Next.js's modern development features. The project showcases several key achievements:

1. Framework Integration

- Seamless Bootstrap integration with Next.js
- Optimized performance
- Enhanced development experience

2. User Experience

- Intuitive interface
- Responsive design
- Accessibility compliance
- Performance optimization

3. Development Efficiency

- Component reusability
- Maintainable architecture
- Type safety
- Scalable structure

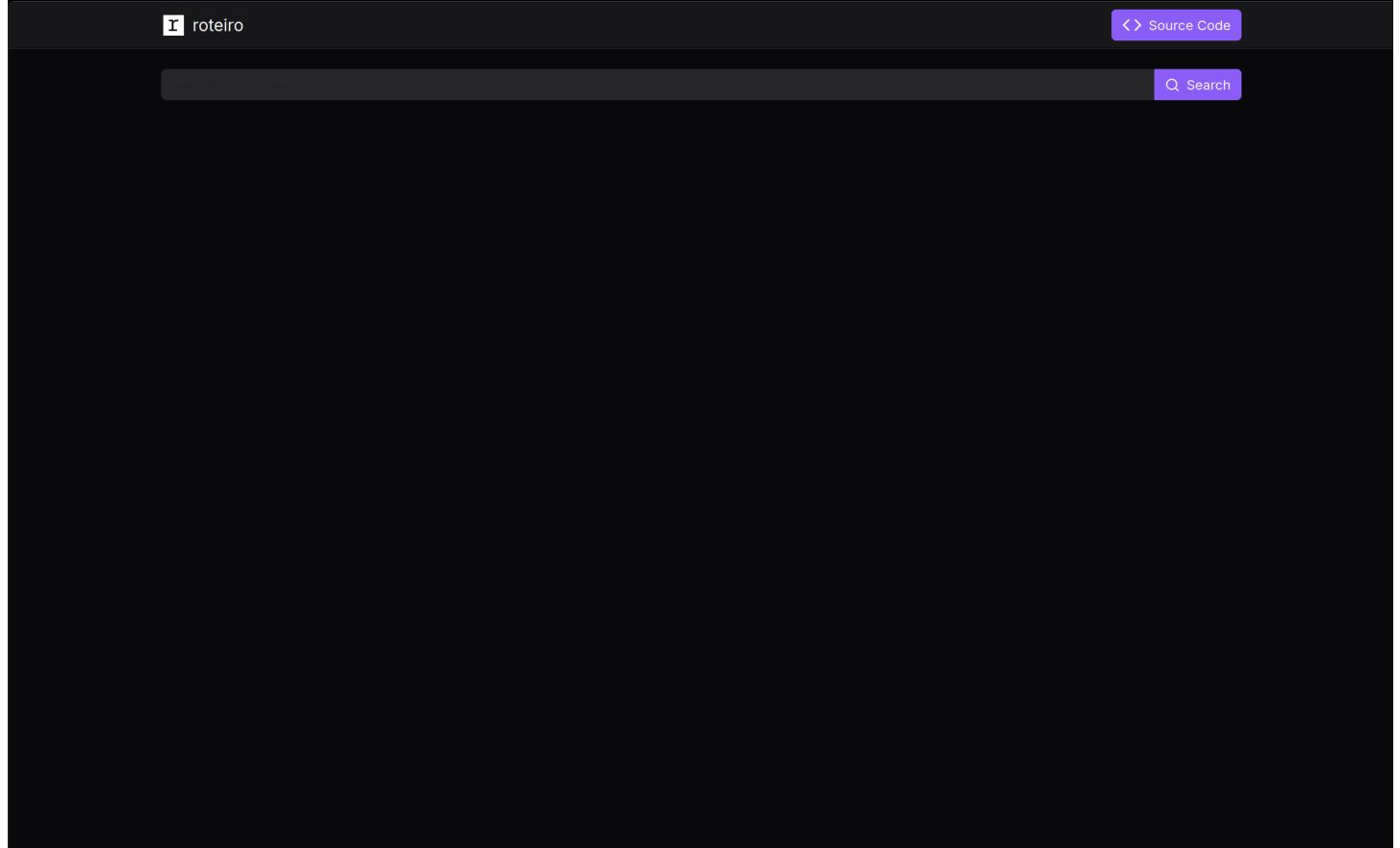
The combination of Bootstrap's utility-first approach and Next.js's powerful features results in a highly functional, maintainable, and user-friendly application that serves as an excellent example of modern web development practices.

Here's how you can include the images in the report under the heading **Visual Overview of Roteiro**, along with brief descriptions:

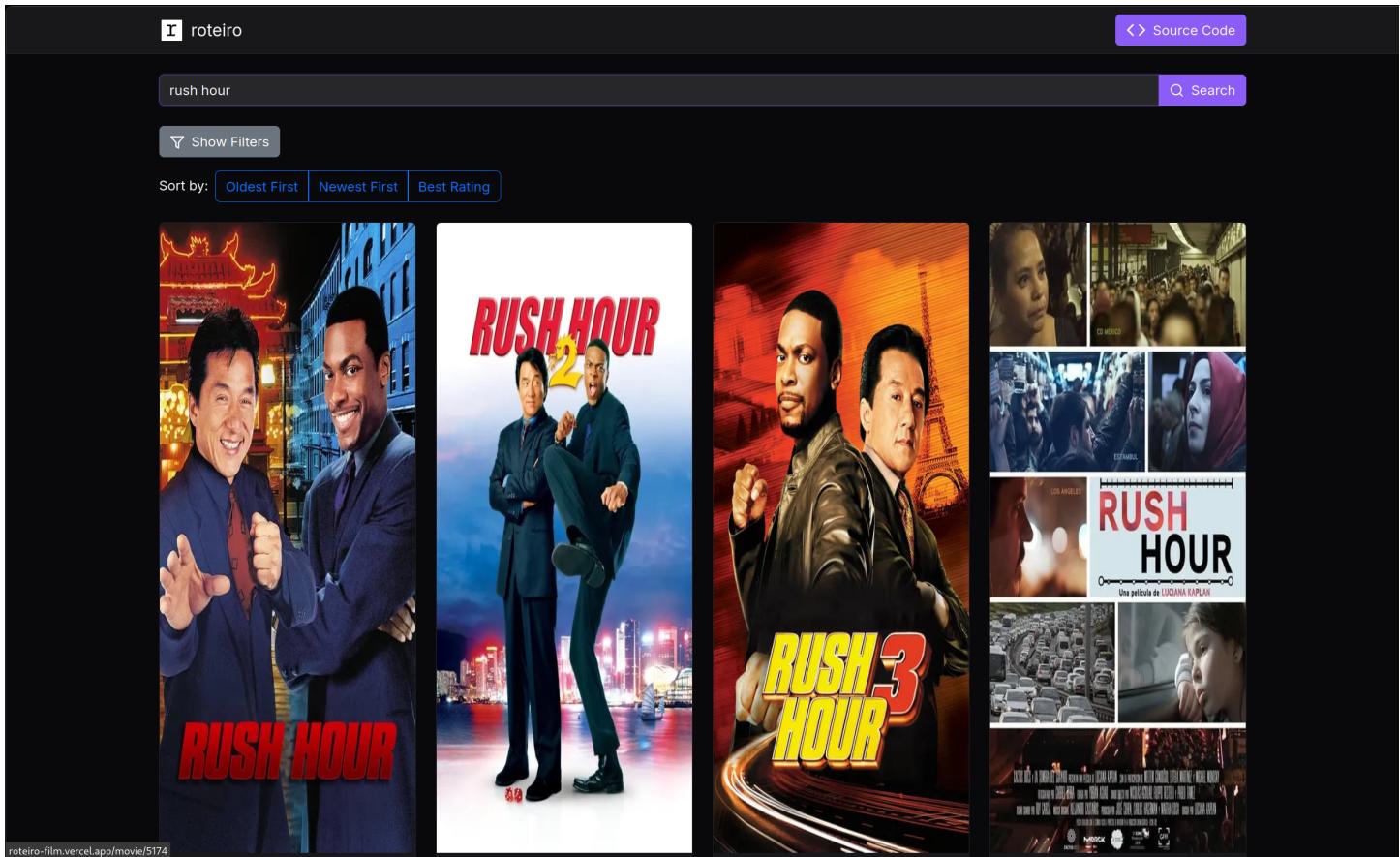
Visual Overview of Roteiro

Below are screenshots highlighting Roteiro's user interface and features:

- 1. Homepage** The homepage provides users with an intuitive starting point to explore movies, featuring a search bar and popular movie suggestions.



- 2. Search Results** The search results page showcases responsive and dynamic movie search functionality, complete with sorting and filtering options.



3. Filters Section The filters section allows users to refine their movie searches based on genres and other criteria, enhancing discoverability.

This screenshot shows the same dark-themed interface as above, but with the 'Filters' section open. The 'Filters' section includes fields for 'Genres', 'Release Year Range', and 'Minimum Vote Count'. It also features a 'Sort by' dropdown at the bottom. Below the filters, the same four movie posters are displayed: 'EL PLANETA VIVO' (2009), 'RUSH HOUR' (1998), 'RUSH HOUR 2' (2001), and 'GODZINA SZCZYTU' (2011).

Open Source Project

Roteiro is an open-source initiative, making its development and functionality accessible to developers worldwide. The platform welcomes contributions, issue reporting, and discussions to enhance the application.

- **Source Code:** The complete source code for Roteiro is available on GitHub at <https://github.com/themohitnair/roteiro>.
- **License:** Roteiro is licensed under the **MIT License**, allowing freedom to use, modify, and distribute the software while ensuring proper attribution to the original creators.

Roteiro - Hosted on Vercel

Roteiro is deployed and hosted on Vercel, providing a seamless and performant user experience.

- **Live Application:** Explore Roteiro live at roteiro-film.vercel.app.

Vercel's deployment capabilities ensure fast response times, automatic scaling, and a robust backend for the platform.

Conclusion

Roteiro exemplifies the integration of modern web development frameworks, combining the versatility of Bootstrap with the advanced capabilities of Next.js. The project highlights the effectiveness of utility-based styling, responsive design, and performance optimization, resulting in a highly functional and user-friendly movie discovery platform.

Being open-source and hosted on Vercel, Roteiro encourages collaboration and accessibility for developers worldwide. The platform's MIT License ensures that its contributions can drive innovation while adhering to the principles of open-source development.

By exploring Roteiro, users experience a seamless, feature-rich interface designed for efficient and enjoyable movie exploration, serving as a benchmark for combining aesthetics and functionality in modern web applications.
