Internship Report

Submitted by: Diya Maheshwari
Internship Title: Front-end Web Development Intern
Organization: Nullclass Edtech Private Limited
Duration: 1 Month
Submission Date: 29/07/2025

---

1. Introduction

This report details the outcomes of an advanced frontend development internship project focused on building a high-performance, feature-rich, and responsive movie database website. The project was designed to replicate the core functionalities of a professional platform like IMDb, providing a seamless and visually engaging experience for users to discover, explore, and review movies and actors.
The primary goal was to architect and implement a sophisticated web application using a modern tech stack, with a strong emphasis on performance optimization, advanced **UI/UX** patterns, and real-time data handling. The application leverages **React** (within the Next.js framework), **Tailwind CSS**, and **Framer Motion** to deliver a polished and maintainable final product.

---

2. Background

This internship built upon a foundational understanding of React and web development principles. The project required moving beyond basic component creation and state management to tackle advanced, real-world development challenges. The tasks were specifically designed to introduce concepts such as server-side rendering (SSR), real-time database interaction, and advanced performance optimization techniques, preparing for the demands of modern web application development. The final deliverables were hosted on **GitHub** and deployed to a live URL for demonstration.

---

3. Learning Objectives

- To implement complex, animated, and **user-friendly** UI components, such as a draggable, autoplaying carousel.
- To master advanced data fetching and rendering patterns, including **Server-Side** Rendering **(SSR)** and Incremental Static Regeneration **(ISR)** with **Next.js** to enhance performance and SEO.
- To develop a real-time, full-featured interactive system (reviews) with functionalities like creating, editing, deleting, and voting.
- To utilize **Framer Motion** for creating sophisticated, fluid animations and microinteractions.

---

4. Tasks and Activities

Task1: Implement a horizontal carousel on the homepage

Objective: To create a visually appealing, draggable horizontal carousel on the homepage to display trending movies.

Approach:

1. **Component Design**:

   A reusable carousel component was built using **Swiper.js** for its robust touch and drag functionalities.

2. **Animation & Styling:**

   Framer Motion was integrated to create smooth, fluid transitions and animations for the carousel elements. **Tailwind CSS** was used to ensure the design was fully responsive, adjusting seamlessly across desktop and mobile devices.

3. **Functionality:**

   The carousel was configured with autoplay, looping, and navigation controls (arrows and dots) to provide a rich user experience.

Task 2: Design and Implement a Dynamic Actor Profile Page

Objective: To build a dynamic actor profile page that fetches and displays detailed information in real-time, with a strong focus on performance.

Approach:

1. **Data Fetching:**

   The page was designed to fetch actor data (name, biography, f ilmography) from a mock API (or Firestore).

2. **Performance Optimization:**

   To enhance load speed and ensure data was up-to-date, Next.js's Server-Side Rendering (SSR) was implemented. This allows the server to fetch the data and render the full HTML page for each request, providing fast initial load times and fresh content.

Task 3: Develop a Full-Featured Review System

Objective: To create a comprehensive review system where users can write, edit, and delete reviews, including advanced features like voting and dynamic sorting

Approach:

1. **Core Functionality:**

   A review component was developed allowing users to perform CRUD (Create, Read, Update, Delete) operations. Real-time validation was included to provide instant feedback on user input.

2. **Advanced Features:**

   An upvote/downvote system was implemented, along with dynamic sorting options

3. **Performance and UX :**

   To create a smooth, real-time feel, **optimistic UI** updates were used. This updates the interface immediately before the API call completes. API calls were also **debounced** to

prevent excessive network requests. **Framer Motion** was used to animate interactions like submitting or voting on a review.

---

5. Tools and Technologies Used

- **Core Framework**: React & Next.js
- **Styling**: Tailwind CSS
- **Animations**: Framer Motion
- **Carousels**: Swiper.js
- **Icons**: Lucide React
- **Version Control:** Git & GitHub [cite: 20]
- **Code Editor:** Visual Studio Code
- **API/Database**: Mock API / Firestore (for data fetching simulation)

---

6. Challenges and Problem Solving

- **Challenge**: Managing the complex state of the real-time review system, especially with optimistic UI updates.
  Solution: Utilized React's useReducer and useContext hooks to create a robust state management system.This allowed for predictable state transitions and prevented UI inconsistencies when handling background API requests.
- **Challenge**: Ensuring the actor profile pages were both fast and displayed up-to-date information.
  Solution: Implemented Server-Side Rendering (SSR) in Next.js. This approach balanced the need for performance with real-time data fetching, providing an excellent user experience while ensuring content was always current.
- **Challenge**: Implementing the upvote/downvote feature with dynamic sorting without causing excessive re-renders.
  Solution: The sorting logic was memoized using React's useMemo hook.This ensured that the list would only re-sort when the underlying data or the sort option actually changed, optimizing rendering performance.

---

7. Outcomes and Impact

The internship successfully culminated in the creation of a high-performance, fullfeatured movie database web application. The final product effectively demonstrates proficiency in advanced frontend development concepts and technologies.

The key outcomes include:

- A fully functional, responsive, and visually appealing web application deployed to a live URL.
- A well-documented and professional GitHub repository containing the complete source code.

---

8. Conclusion

This internship provided an invaluable opportunity to transition from foundational knowledge to the practical application of advanced frontend development techniques. The process of building this project from the ground up solidified my understanding of the entire development lifecycle, from initial design and architecture to performance optimization and deployment.

---

9. Key Deliveries

- **Live Website URL:** https://imdb-movie-project.netlify.app/
- **GitHub Repository**: https://github.com/diya5maheshwari/IMDBMovieProject

---