আন্তর্জাতিক ইসলামী বিশ্ববিদ্যালয় চট্টগ্রাম
الجامعة الإسلامية العالمية شيتاغونغ
International Islamic University Chittagong

Project Title: Movie Server

Course Code : CSE - 3532

Course Title : Tools and Technologies for Internet Programming

Submission Date : 09.07.25

Submitted To:

Sara Karim,
Adjunct Lecturer,
Dept. of CSE, IIUC.

Submitted By:

Name : Nazia Afrin Tanna
ID : C231539
Name: Jannatul Maowa Prome
ID: C231514
Semester : 5th
Section : 5DF
Department : CSE

Remarks:

Signature:

MOVIE SERVER

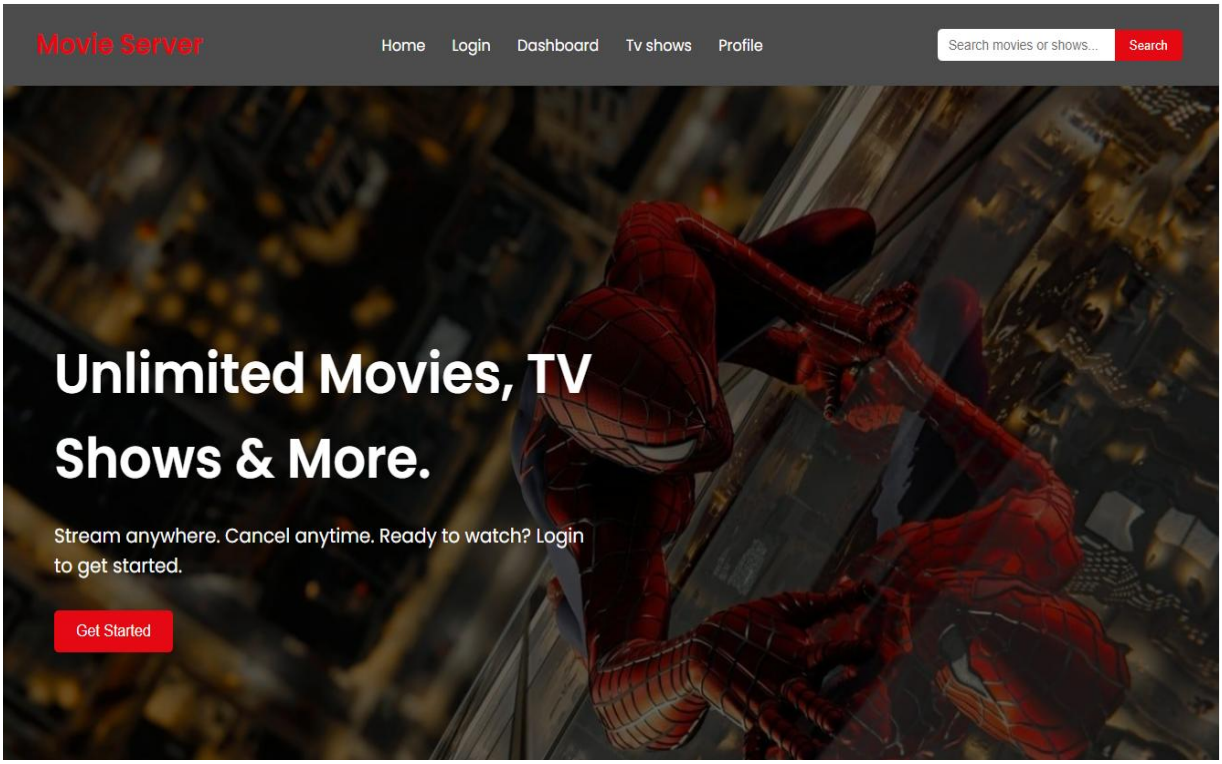


Table of Contents

Abstract

The **Movie Server** is a visually rich and interactive web-based movie and TV show streaming platform designed to simulate the experience of popular services like Netflix. Built using HTML, CSS, and JavaScript, PHP this project offers users a seamless interface to browse trending movies, explore top-rated films, and view popular TV shows — all within a responsive and user-friendly environment. With an attractive home page, engaging hero section, and clean navigation bar, users can easily access different sections of the site, including login, dashboard, profile, and a dedicated TV shows page.

A search feature is implemented to allow real-time filtering of movie titles, and each movie card includes essential details like age rating and a download button to simulate real-world interaction. The login page features a basic authentication mechanism using hardcoded credentials, providing a foundation for future backend development. The project also includes a profile section where user details are displayed, adding a personalized touch to the platform.

Introduction

In the modern digital era, online streaming platforms have revolutionized how people consume entertainment. With increasing demand for instant access to movies and TV shows, platforms like Netflix, Amazon Prime, and Disney+ have set new standards in how media is delivered and experienced. Inspired by this trend, the **Movie Server** project aims to replicate the core features of such services through a simple yet engaging front-end web application.

This project is developed using **HTML, CSS, and JavaScript** focusing on creating a visually appealing and responsive user interface. Users can navigate through various sections of the platform including Home, Login, Profile, Dashboard, and a TV Shows page. The website is designed to offer a realistic experience by showcasing movie thumbnails, age ratings, interactive buttons, and a functional search bar that filters content in real time.

Although the current version uses **static data and hardcoded login credentials**, it lays the foundation for future integration with a backend system using PHP and MySQL. The project reflects a strong understanding of web layout design, user experience principles, and client-side interactivity — making it a solid step toward building full-stack web applications in the future.

Objectives

The main objective of the **Movie Server** project is to create an elegant and responsive front-end platform that mimics the user experience of real-world movie streaming websites. It aims to help users browse trending and top-rated content through an intuitive layout, navigate between pages effortlessly, and interact with components such as search bars, profile sections, and login forms. The project also sets the groundwork for connecting to a backend database using PHP and XAMPP, allowing future enhancements such as dynamic content loading and user authentication.

Beyond functionality, the project encourages clean UI/UX practices, organized code structure, and responsive design — enabling a consistent viewing experience across different devices including desktops, tablets, and smartphones.

Key Features

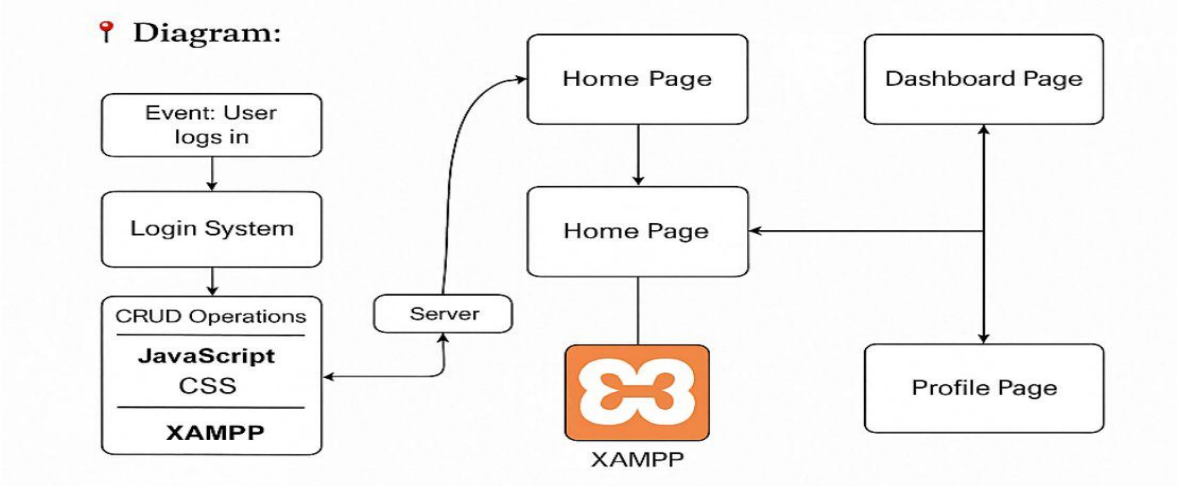
- A **fully responsive homepage** with a hero section and category-based movie cards.
- A **login page** with email-password validation using hardcoded credentials.
- A **search functionality** to filter movies or TV shows by title.
- Interactive **download buttons** that simulate a download action.
- A personal **profile section** showing user information and profile picture.
- A clean and modern layout designed using custom CSS, Poppins fonts, and styled components.
- Navigation between pages such as Home, Login, Dashboard, TV Shows, and Profile.

Methodology/System Design

The development of the *Movie Server* project follows a modular and user-centered design approach. At its core, the system consists of multiple interconnected web pages (like Home, Login, Dashboard, TV Shows, and Profile), styled using CSS and made interactive with JavaScript. The backend, built using PHP, manages the login logic and will later support dynamic content such as user profiles and downloadable media.

The user initiates their interaction through the login page, which authenticates credentials via hardcoded logic in PHP. Once authenticated, users are redirected to the homepage, where they can browse trending movies, top-rated shows, and device compatibility options. Each section is structured in a grid layout for ease of viewing and navigation.

The backend structure, powered by XAMPP (Apache + PHP), allows local testing and future scalability with database integration (like MySQL). The overall system is designed to be visually appealing, responsive across devices, and easy to understand, making it suitable for both academic demonstration and future development.

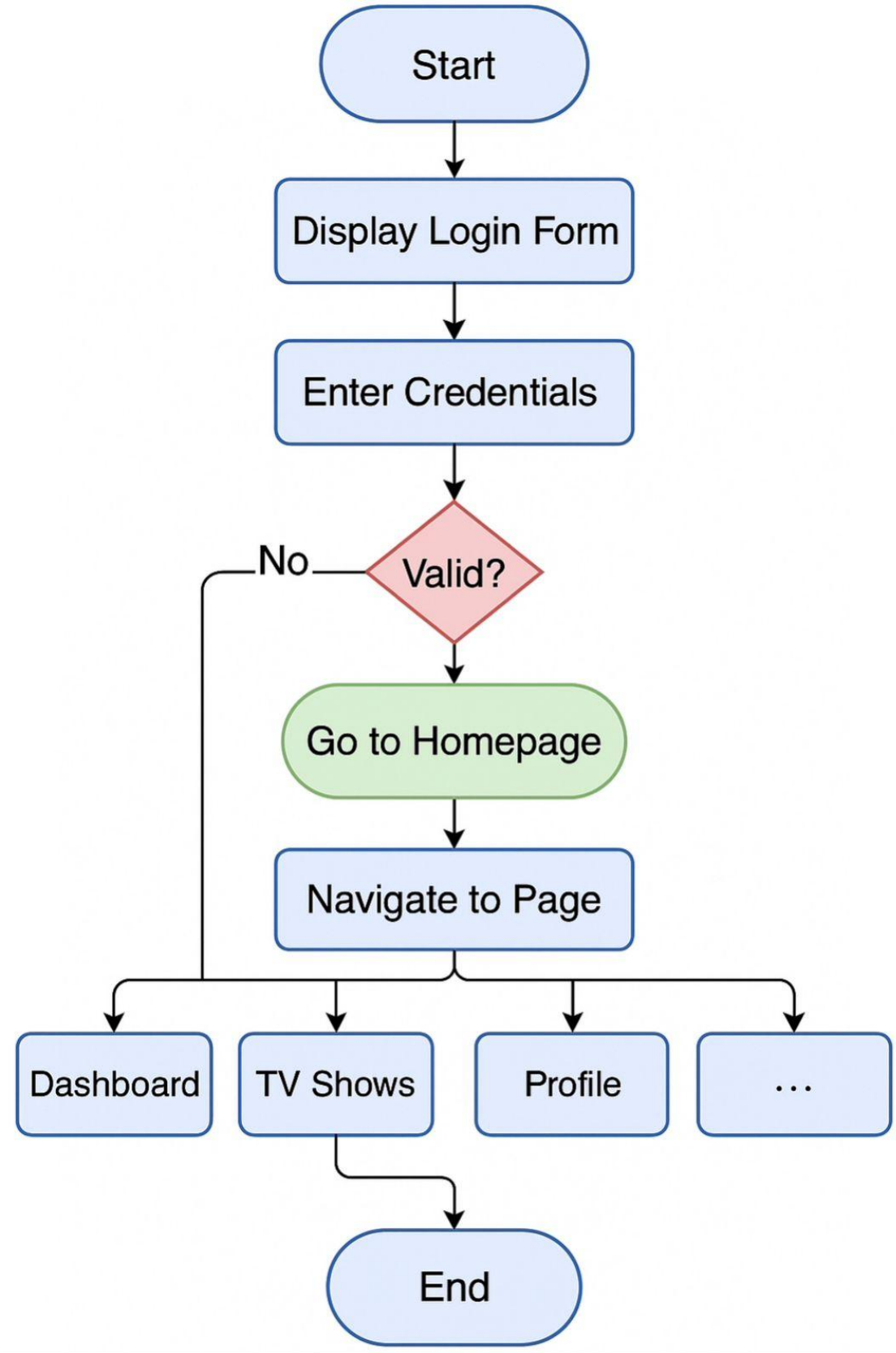


Technologies Used

The project is built using the following core technologies:

1. **HTML5** — For structuring the content and creating the layout of all web pages.
2. **CSS3** — For styling and responsiveness, including layouts, buttons, and animations.
3. **JavaScript (Vanilla JS)** — For client-side interactivity like search functionality and download alerts.
4. **PHP** — For handling basic backend operations such as login authentication.
5. **XAMPP** — As the local server environment (Apache + PHP) to run and test backend logic.
6. **VS Code** — The primary code editor used for writing HTML, CSS, JavaScript, and PHP files.
7. **Google Fonts (Poppins)** — For a modern and clean typography.
8. **File Structure (Assets Folder)** — For organizing images like movie posters and device mockups.

Flowchart


Start
Display Login Form
Enter Credentials
Valid?
No
Go to Homepage
Navigate to Page
Dashboard
TV Shows
Profile
...
End

Implementation

1. HTML Snippet – Hero Section

```
html
CopyEdit
<section class="hero">
  <div class="hero-content">
    <h1>Unlimited Movies, TV Shows & More.</h1>
    <p>Stream anywhere. Cancel anytime. Ready to watch? Login to get started.</p>
    <a href="login.html"><button class="button">Get Started</button></a>
  </div></section>
```

2. **CSS Snippet – Movie Card Design**

```
css
CopyEdit
.movie-card {
  background: #222;
  padding: 15px;
  border-radius: 10px;
  text-align: center;
  color: #fff;
  transition: transform 0.3s ease;
}
.movie-card img {
  width: 100%;
  border-radius: 5px;
  margin-bottom: 10px;
}
.movie-card:hover {
  transform: scale(1.05);
}
```

3. JavaScript Snippet – Search Functionality

```
javascript
CopyEdit
document.getElementById('searchButton').addEventListener('click', function() {
  const input = document.getElementById('searchInput').value.toLowerCase();
  const movies = document.querySelectorAll('.movie-card');

  movies.forEach(movie => {
    const title = movie.querySelector('h3').innerText.toLowerCase();
    movie.style.display = title.includes(input) ? 'block' : 'none';
  });
```

4. JavaScript Snippet – Download Button

```
javascript
CopyEdit
const downloadBtns = document.querySelectorAll('.download-btn');
downloadBtns.forEach(btn => {
  btn.addEventListener('click', () => {
    alert('Download started (Fake)');
  });
});
```

5. Login Functionality (Hardcoded)

```
javascript
CopyEdit
document.getElementById('login-form').addEventListener('submit', function(e) {
  e.preventDefault();
  const email = document.getElementById('login-email').value.trim();
  const password = document.getElementById('login-password').value.trim();

  const fakeUser = {
    email: 'tannaprome@gmail.com',
    password: '123456'
  };

  if (email === User.email && password === User.password) {
    alert('Login successful!');
    window.location.href = 'index.html';
  } else {
    alert('Invalid email or password.');
  }
});
```

6. **PHP Snippet – Connect to Backend (Optional)**

```
php
CopyEdit
<?php$servername = "localhost";$username = "root";$password = "";$database =
"movie_server";
// Create connection$conn = new mysqli($servername, $username, $password,
$database);
// Check connectionif ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}?>
```

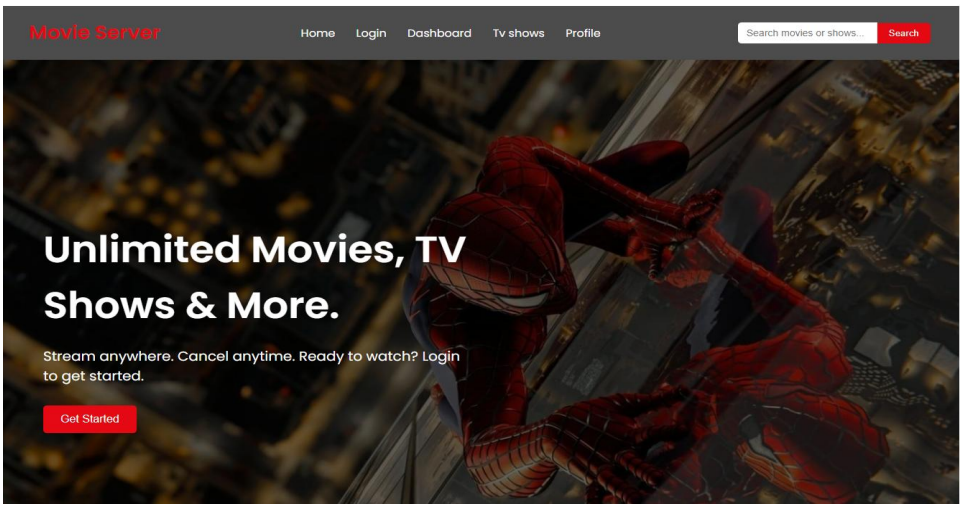
7. Profile HTML Snippet

```
html
CopyEdit
<div class="profile-card">
  <img src="profile-pic.jpeg" alt="Profile Picture" class="profile-pic">
  <h2>[User's Name]</h2>
  <p>Email: [user@example.com]</p>
  <p>Location: [City, Country]</p>
  <p>Joined: [Join Date]</p>
  <button class="button">Edit Profile</button></div>
```

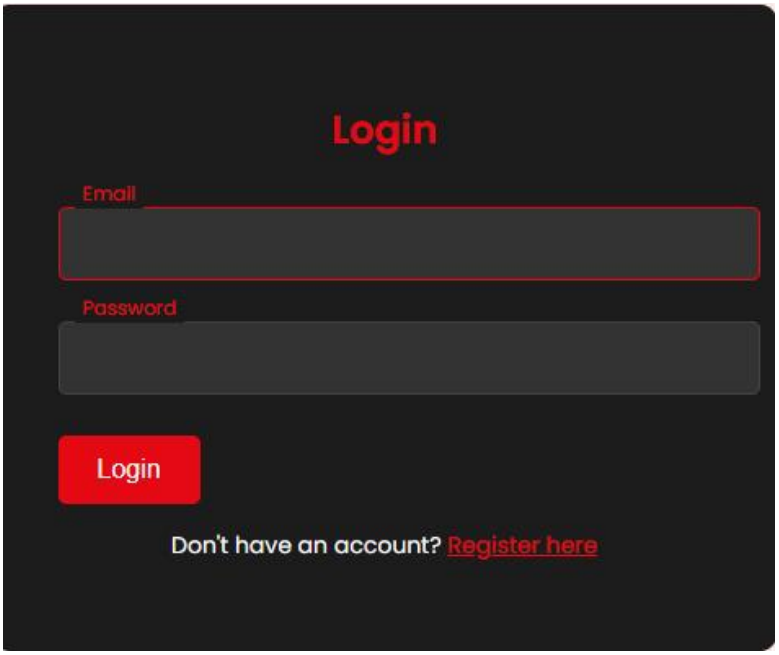
Result & Analysis

The "Movie Server" project successfully demonstrates the core features of a basic movie streaming web application. The frontend design and JavaScript functionalities were implemented effectively using HTML, CSS, and JS. The site simulates user login, movie browsing, downloading (fake), and profile management. Although no real-time backend/database was used, the project lays a solid foundation for a functional system.

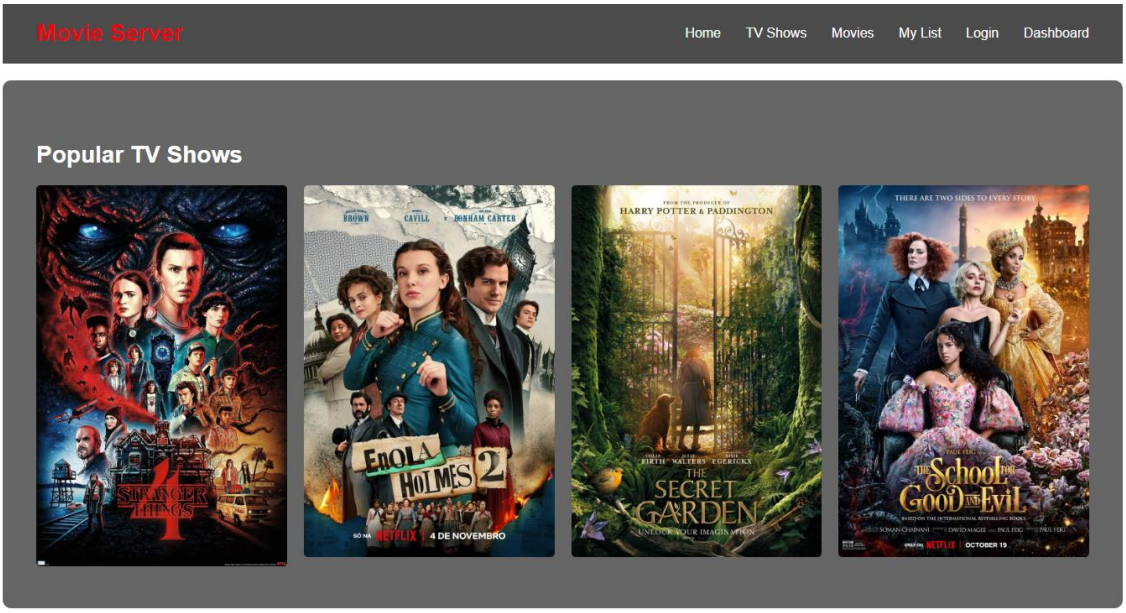
Homepage



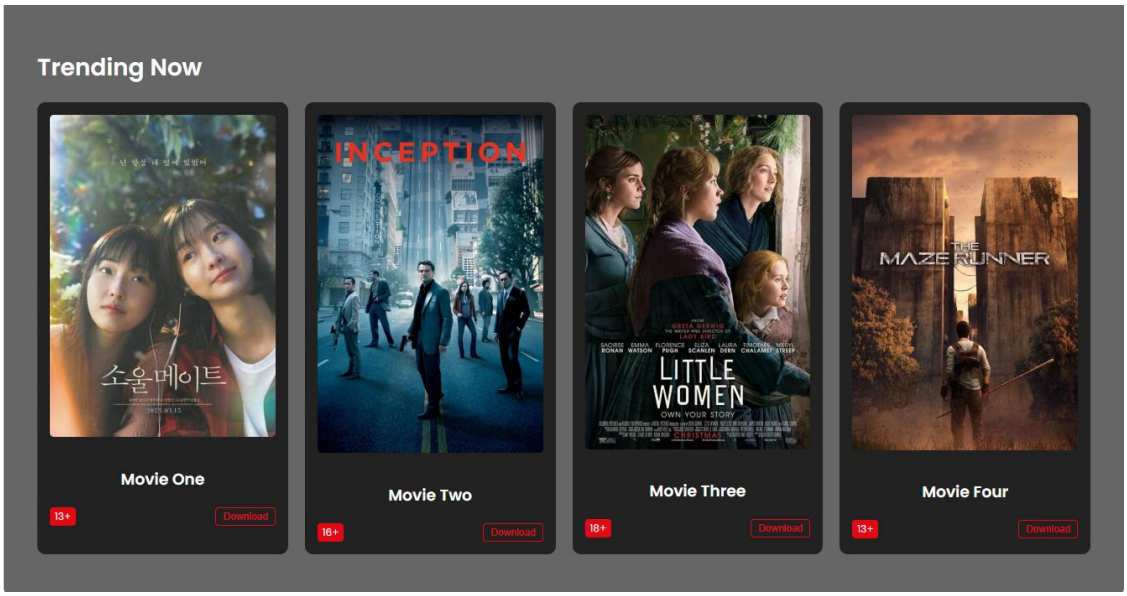
Login Page



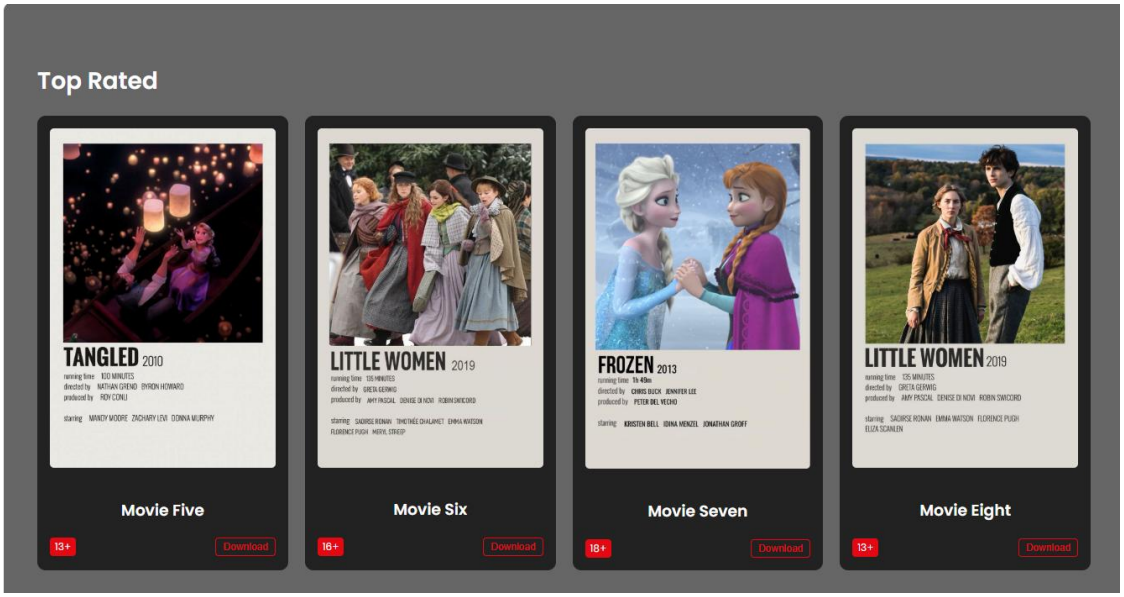
Popular Tv Shows

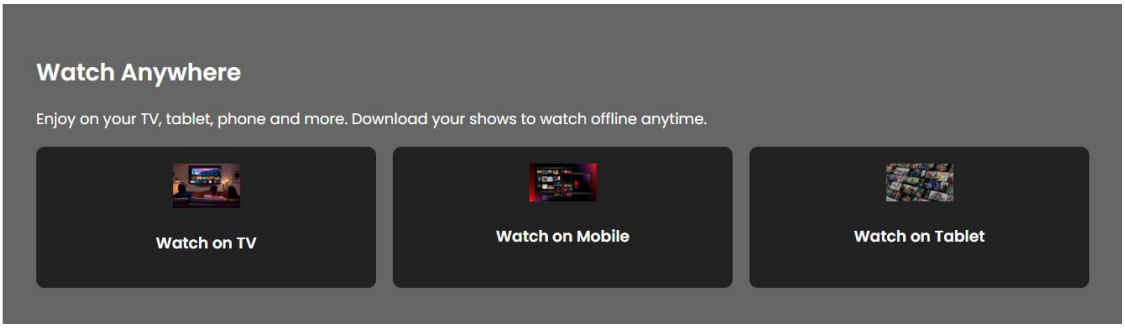


Trending Movies



Top Rated Movies





Challenges

During the development of the Movie Server project, we encountered several challenges. First, managing image and asset paths was confusing at times, especially when organizing files in different folders. Ensuring responsiveness across devices like mobile and desktop also required extra attention to media queries and layout adjustments. Implementing a basic login system without a backend was tricky, as we had to simulate validation using JavaScript only. Developing the search functionality to filter movie cards dynamically was challenging and needed multiple revisions. Finally, maintaining a consistent UI design across all pages—like Home, Login, Profile, and TV Shows—required continuous refinement and careful CSS structuring.

Conclusion

The Movie Server project successfully demonstrates how a responsive, user-friendly movie streaming interface can be built using HTML, CSS, and JavaScript. Despite the absence of a real backend, the project simulates login validation, dynamic search, and interactive UI behavior effectively. Through this project, we gained valuable hands-on experience in frontend development, layout design, file management, and user interaction. It has enhanced our understanding of how real-world streaming platforms work and provided a solid foundation for future integration with backend technologies.

References

1.W3Schools – HTML, CSS, JavaScript Tutorials
*(*https://www.w3schools.com*)*

2.MDN Web Docs – JavaScript, DOM, and Web APIs
*(*https://developer.mozilla.org*)*

3.Google Fonts – Poppins Font
(https://fonts.google.com/specimen/Poppins)

4.YouTube – Frontend design inspiration and walkthroughs
*(*https://www.youtube.com*)*

5.GitHub – Sample project structures and UI ideas
*(*https://github.com*)*