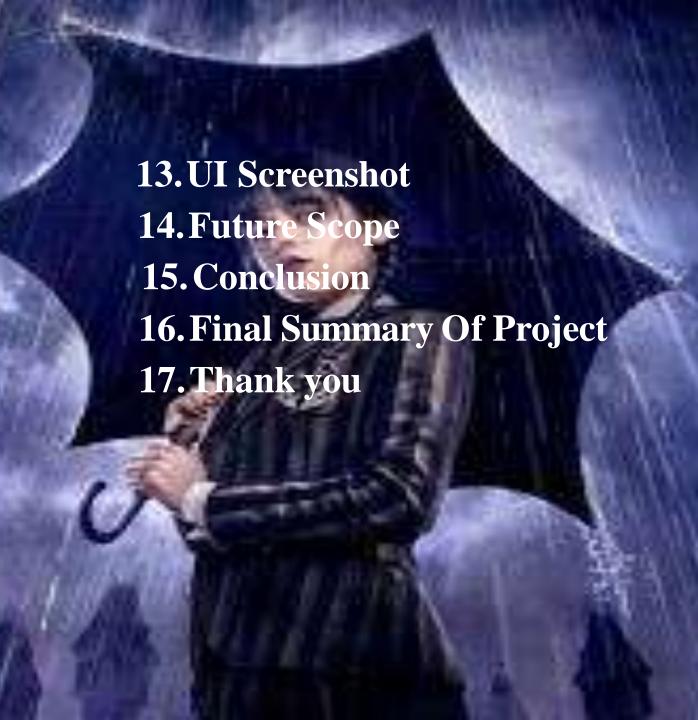


TOPIC OUTLINE

- 1. Introduction
- 2. Objective
- 3. Scope
- 4. Structure
- 5. System Requirement
- 6. Schema Diagram
- 7. Level 0 DFD
- 8. Level 1 DFD User
- 9. Lavel 1 DFD Admin
- 10. Sequence Diagram
- 11. Use Case Diagram



NETFLIX COLOE WEBSITE

A BREIF INTRODUCTION OF OUR WEBSITE

Our website is a OTT platform developed using modern web technologies. The frontend is built with React.js for a dynamic and Express.js, ensuring fast and scalable server-side operations. We use MongoDB as our database to efficiently manage user data, movies, and webseries information. This platform allows users to view the available content or videos about TV Showes, Movies, Web - series etc.



OBJECTIVE

The objective of building a OTT platform website using the MERN stack (MongoDB, Express.js, Node.js) is to create a dynamic, user-friendly web application that allows users to interact, purchase food key goals include:

- User Authentication: Secure sign-up, sign-in, and session management.
- Category selection: Users can choice any videos, movies, web-series and can also watch them.
- Real-Time Notifications: Users receive instant updates on activities like successful notifications, alert notification.
- Scalability and Performance: Handle growing user activity and data efficiently.

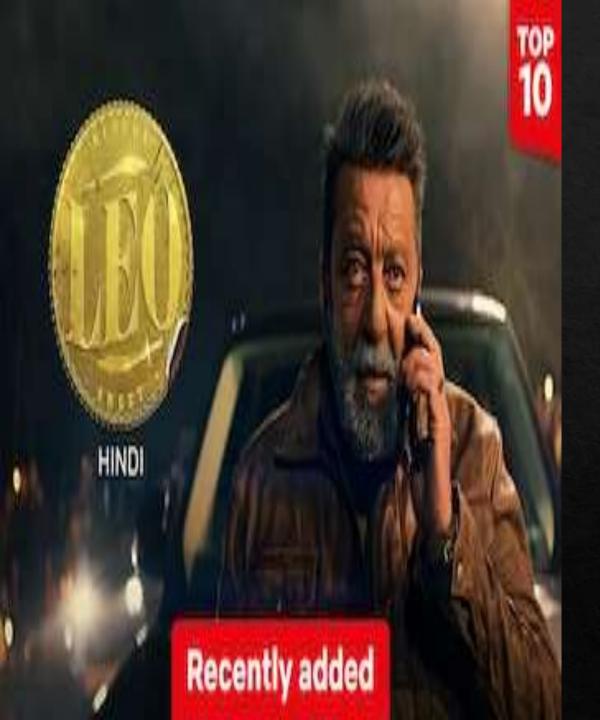


SCOPE

User Experience and Content

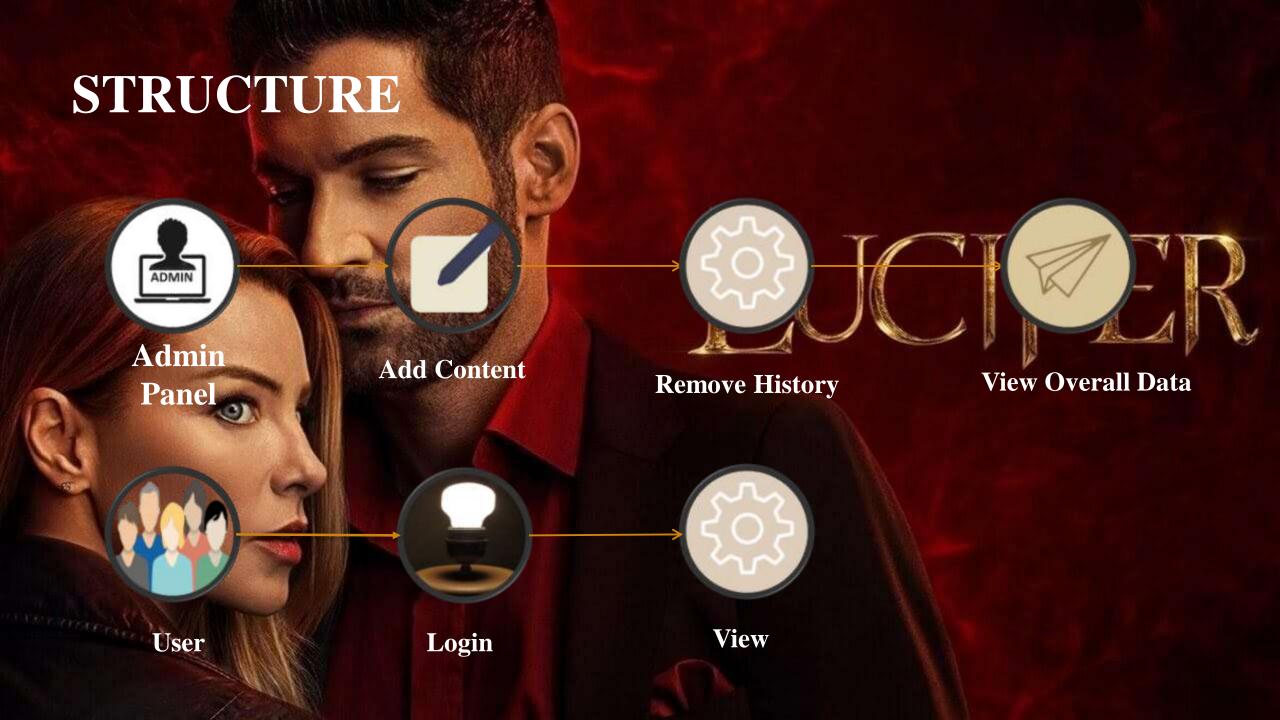
- 1. Content Library: The website serves as a gateway to Netflix's vast and diverse library of TV shows, movies, documentaries, and more.
- **2. Personalization:** A core function of the site is its recommendation algorithm. It analyzes a user's viewing history, ratings, and preferences to provide a personalized homepage with suggestions tailored to their interests.
- 3. User Accounts and Profiles: The website allows for the creation of multiple user profiles under a single account. Each profile has its own viewing history, recommendations, and settings. Parental controls can also be set up to restrict content for younger viewers.
- **4. Account Management**: Users can manage their subscription plans, payment methods, and account settings directly on the site. They can also change their password, email address, and other personal information.





Technical and Infrastructure

- 1. Accessibility: The Netflix website is designed to be accessible across a wide range of devices and web browsers, providing a consistent user experience whether on a desktop, laptop, or tablet.
- 2. Content Delivery: The website relies on a sophisticated content delivery network (CDN) called Open Connect. This network places content on servers geographically close to users, reducing latency and ensuring fast, smooth streaming with minimal buffering.
- 3. Performance and Optimization: The Netflix engineering team continuously works on optimizing the website's performance, using technologies like server-side rendering and prefetching to improve loading times and interactivity.



SYSTEM REQUIREMENT

HARDWARE REQUIREMENT:

- PROCESSOR :- Dual core and above
- RAM / ROM :- 10 Gb of free disk space & minimum 8Gb

of RAM

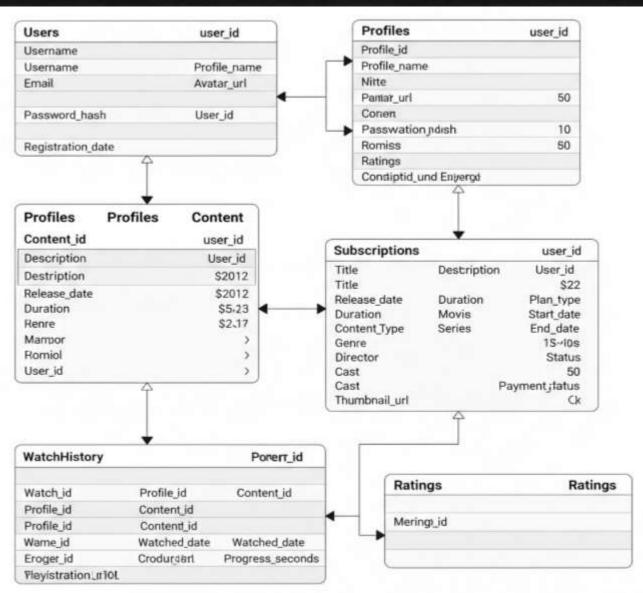
• MOTHERBOARD :- RYZEN , INTEL

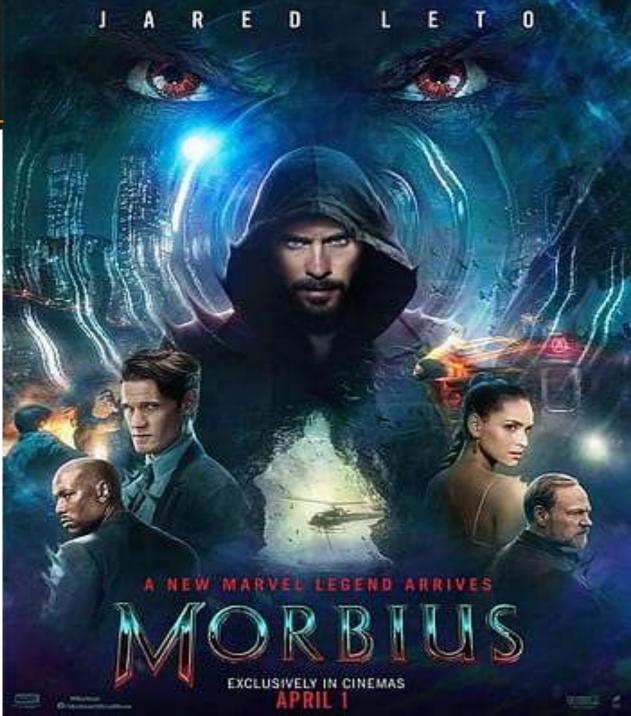
SOFTWARE REQUIREMENT:

- OPERATING SYSTEM :- Windows 10, 11
- PROG.LANGUAGE :- HTML, CSS, JS, React
- DATABASE :- MONGODB
- IDE USED :- VS Code

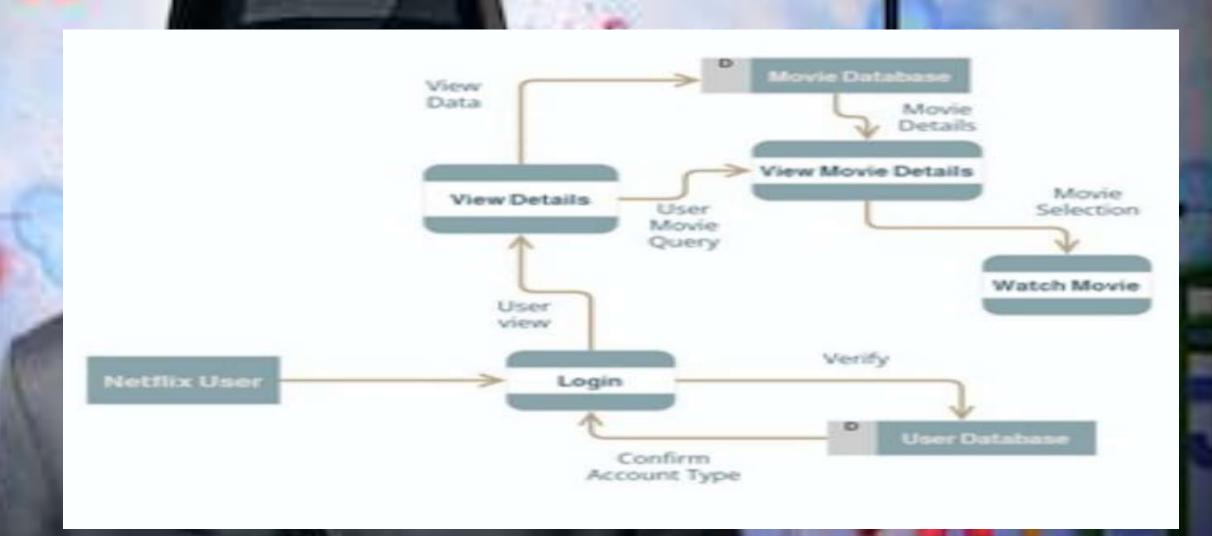


SCHEMA DIAGRAM



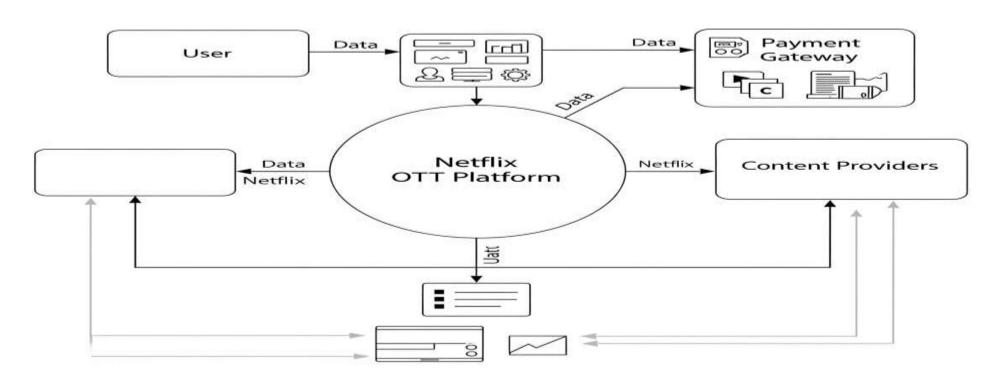


TEVEL O DFD STRUCTURE



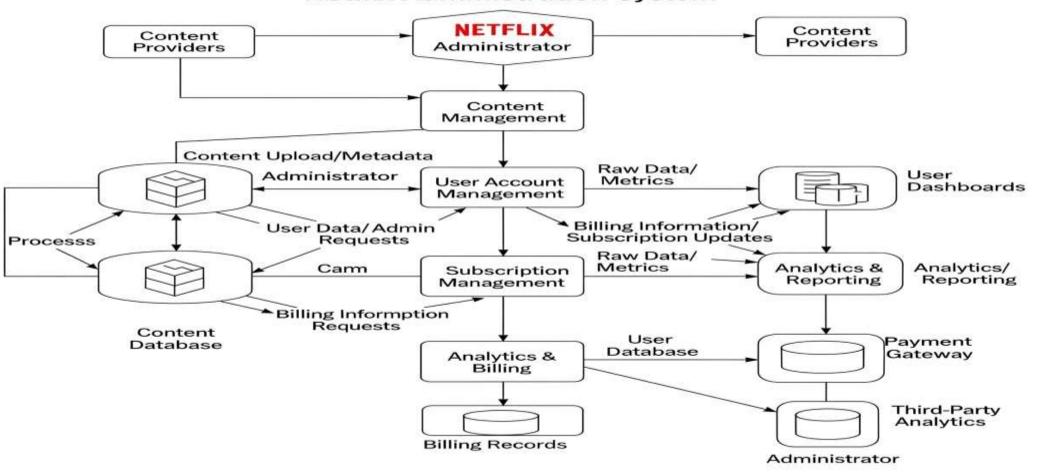
LEVEL 1 DFD STRUCTURE USER

Level 1 DFD Data Flow

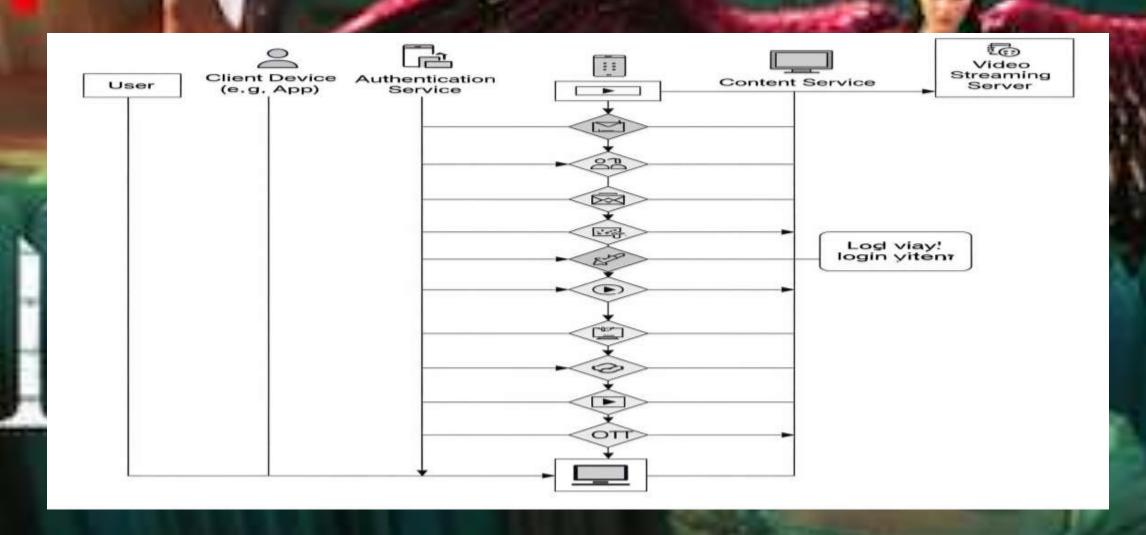


LEVEL 1 DFD STRUCTURE ADMIN

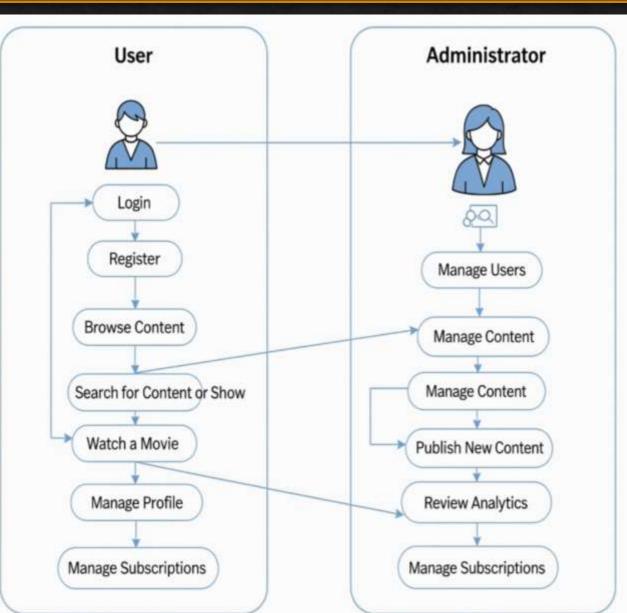




SEQUENCE DIAGRAM



USE CASE DIAGRAM

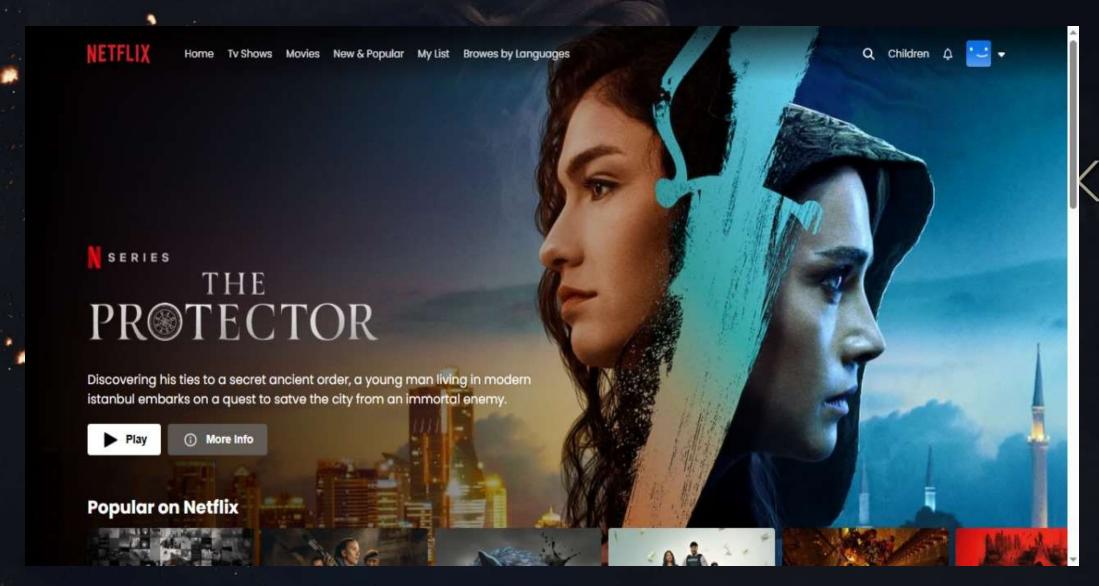


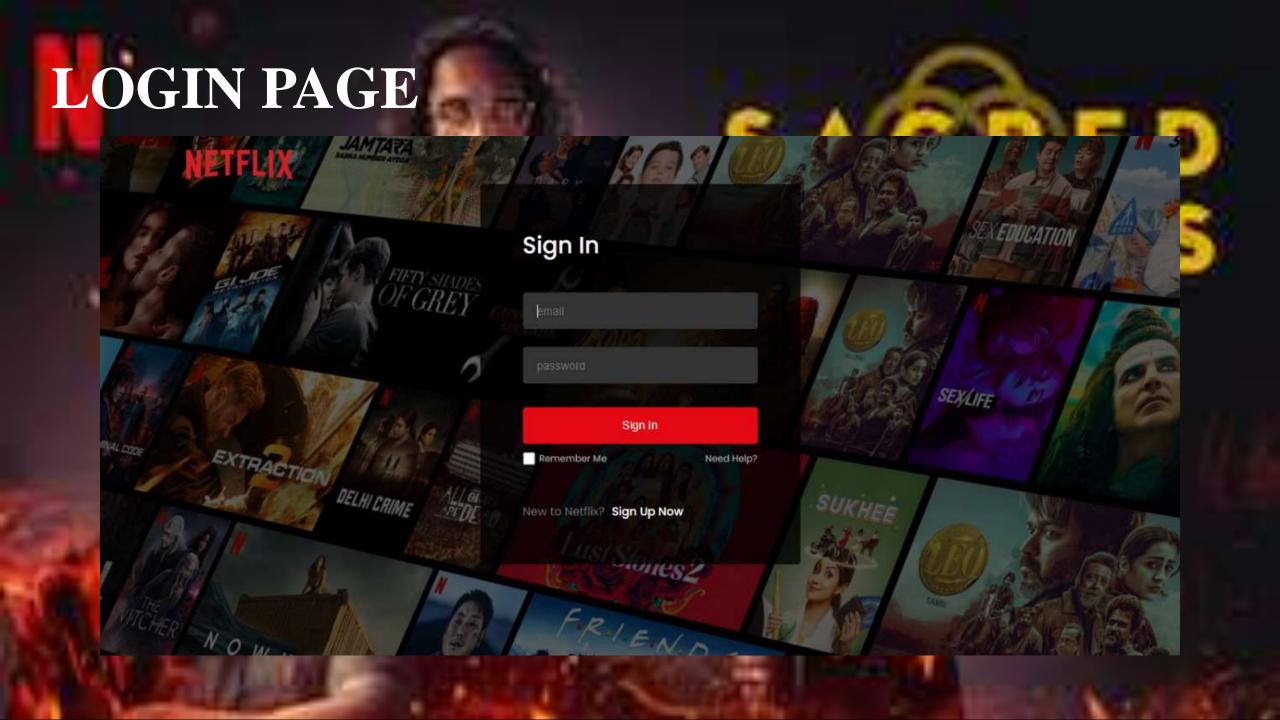


PROJECT'S UI

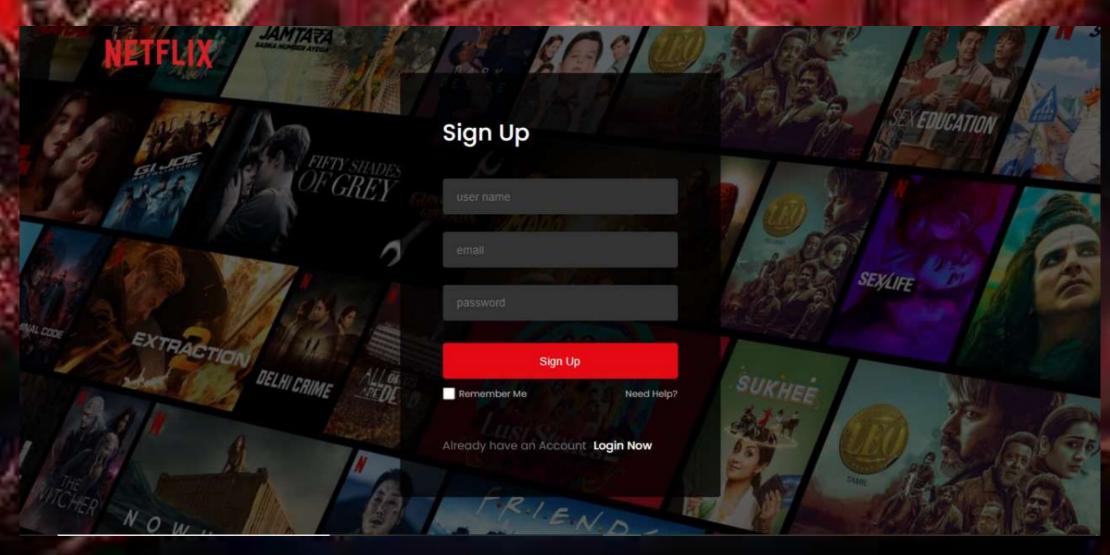


HOME PAGE





SIGN-UP PAGE



EXPLORE MOVIE PAGE



Home Tv Shows Movies New & Popular My List Browes by Languages

Q Children △ 🐸 🕶



Blockbuster Movies













Only on Netflix







Jurassic World Rebirth









RS FILM

A RUSSO

Upcoming















FOOTER



Top Pics for you





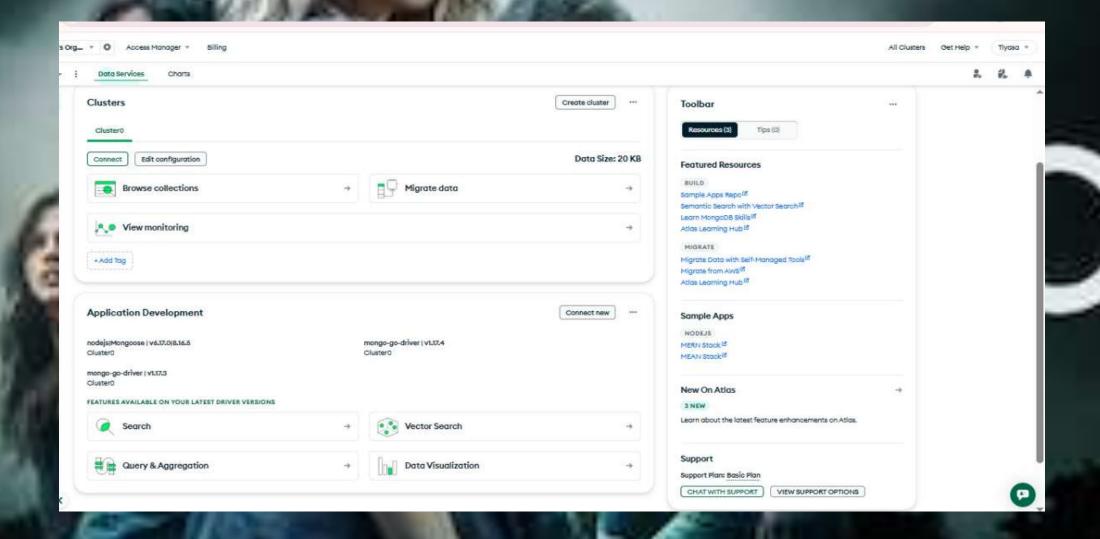
Audio Description Help Centre Gift Cards Media Centre

Investor Relations Jobs Terms of use Privacy

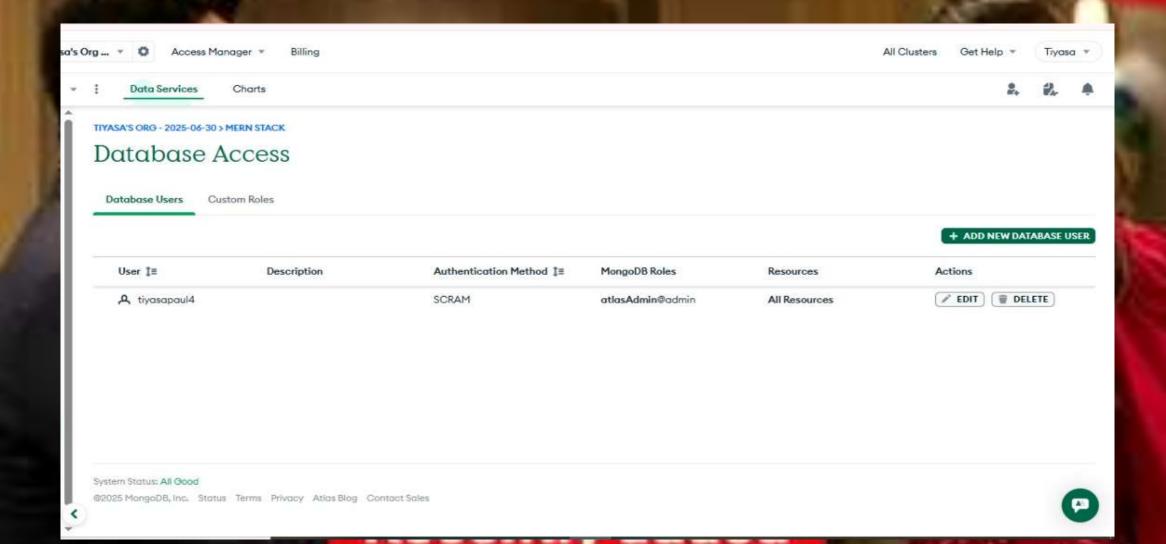
Legal Notices Cookie Preferences Corporate Information Contact Us

@ 1997-2025 Netflix, Inc.

DATABASE



USER DATABASE



CONCLUSION

The process of building a Netflix clone with the MERN stack demonstrates the seamless integration and power of these technologies. Here are the key conclusions:

- Full-Stack Proficiency: The project solidifies a developer's understanding of full-stack development. You learn how to connect the front-end (often built with a framework like React, although not a core part of the MERN acronym, it's a common pairing) with the back-end, handle data flow, and manage a database.
- Backend and API Development: Express.js and Node.js are crucial for creating a robust backend. Developers gain experience in building RESTful APIs to handle requests for movie data, user authentication, and other key features. The asynchronous and event-driven nature of Node.js makes it ideal for handling the high volume of requests typical of a streaming service.
- Database Management: MongoDB's document-oriented database structure is well-suited for storing diverse and unstructured data like movie details, user profiles, and watchlists. The project teaches developers how to perform CRUD operations (Create, Read, Update, Delete) and design a schema that is flexible and scalable.
- User Authentication: Implementing secure user authentication is a fundamental aspect of this project. It involves using technologies like JSON Web Tokens (JWT) to manage user sessions and protect routes, which is a critical skill for building any secure web application.
- Scalability and Performance: While a clone project may not operate at the scale of Netflix, it introduces the architectural concepts necessary for building scalable applications. You learn about non-blocking I/O and how to design a system that can handle multiple concurrent users without performance degradation.

In conclusion, a MERN stack Netflix clone project is more than just a copy of a popular website. It's a deep dive into the architecture of modern web applications, providing developers with the practical skills needed to build dynamic, data-driven, and scalable platforms.



