

# MoviesHuB: An Ad-Free OTT Streaming Platform

(A Platform for Watching the Latest Movies and TV Shows in One Place)

**Author:**

Advait Jadhav

**Date:**

2/06/2024

**Affiliation:**

Rajiv Gandhi Institute of Technology College in Mumbai, Maharashtra

**GitHub Repository:**

<https://github.com/hunterhacker29/MoviesHuB>

## **Table of Contents**

1. **Abstract**
2. **Introduction**
  - 2.1. Background
  - 2.2. Objectives
3. **System Architecture**
  - 3.1. Technologies Used
  - 3.2. System Components
4. **Features**
  - 4.1. User Authentication
  - 4.2. Content Browsing
  - 4.3. Profile Management
  - 4.4. Administrative Controls
5. **Implementation**
  - 5.1. Frontend Development
  - 5.2. Backend Development
  - 5.3. Database Management
6. **User Interface Design**
  - 6.1. Login and Signup Pages
  - 6.2. Home Page
  - 6.3. Profile Pages
  - 6.4. Navigation Bar
7. **Database Structure**
8. **Security Considerations**
9. **Conclusion**
10. **References**

## **1. Abstract**

MoviesHuB is a modern web-based platform that provides users with access to a vast library of movies and TV shows from various OTT (Over-The-Top) services such as Netflix and Amazon Prime. Built using React, Firebase, and Node.js, MoviesHuB offers an ad-free streaming experience, enabling users to watch the latest releases without interruptions. This report outlines the architecture, features, implementation details, and security considerations of the MoviesHuB platform, demonstrating its potential to serve as a comprehensive entertainment hub.

## **2. Introduction**

### **2.1. Background**

The proliferation of OTT platforms has transformed the way consumers access and enjoy media content. With multiple services offering exclusive content, users often find it cumbersome to manage subscriptions across different platforms. MoviesHuB aims to simplify this by aggregating content from various OTT services into a single, user-friendly interface.

### **2.2. Objectives**

- **Unified Access:** Provide a centralized platform to access content from multiple OTT services.
- **Ad-Free Experience:** Ensure uninterrupted viewing by eliminating advertisements.
- **User-Friendly Interface:** Implement an intuitive design for easy navigation and content discovery.
- **Robust Authentication:** Secure user data through reliable authentication mechanisms.
- **Scalable Architecture:** Utilize technologies that support scalability and efficient content delivery.

## **3. System Architecture**

### **3.1. Technologies Used**

- **Frontend:** React.js
- **Backend:** Node.js
- **Database:** Firebase Realtime Database
- **Authentication:** Firebase Authentication (Email/Password and Google OAuth)
- **Hosting:** Firebase Hosting

### **3.2. System Components**

- **Client-Side Application:** Developed using React, responsible for rendering the user interface and handling user interactions.
- **Server-Side Application:** Built with Node.js, manages API requests, data processing, and business logic.
- **Database:** Firebase Realtime Database stores user data, content metadata, and other necessary information.
- **Authentication Module:** Manages user login, signup, and profile management through Firebase Authentication.

## **4. Features**

### **4.1. User Authentication**

MoviesHuB offers secure authentication options, allowing users to sign up and log in using email/password combinations or via their Google accounts. This ensures a seamless and secure access experience.

### **4.2. Content Browsing**

The home page features an infinite scroll mechanism that displays the latest uploaded content. Users can browse through a continuously loading list of movies and TV shows, sorted based on recency and popularity.

### **4.3. Profile Management**

Users can create and manage their profiles, enabling personalized content recommendations and maintaining watch histories. The profile pages are accessible upon clicking the profile icon in the navigation bar.

### **4.4. Administrative Controls**

As a superuser, administrators have exclusive access to add new content to the database. This ensures that only authorized personnel can manage the platform's content offerings.

## **5. Implementation**

### **5.1. Frontend Development**

The frontend is developed using React.js, leveraging its component-based architecture for building reusable UI elements. Key components include the login/signup forms, home page with infinite scrolling, navigation bar, and profile pages.

### **5.2. Backend Development**

The backend is powered by Node.js, handling API requests from the frontend. It manages data retrieval and storage operations with Firebase, ensuring efficient communication between the client and server.

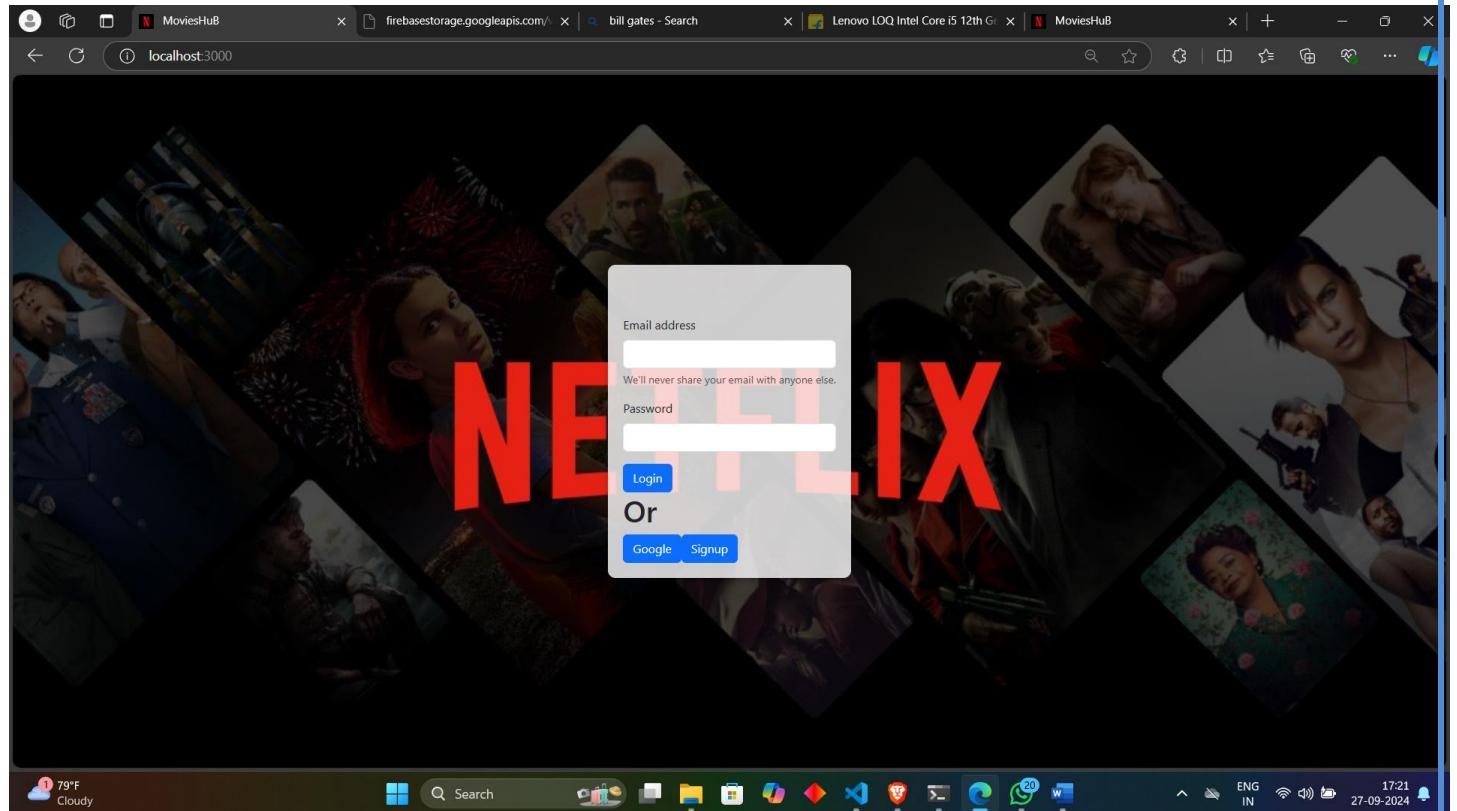
### **5.3. Database Management**

Firebase Realtime Database serves as the backbone for data storage, handling user information, content metadata, and other necessary data. Its real-time capabilities facilitate instant updates and synchronization across the platform.

## **6. User Interface Design**

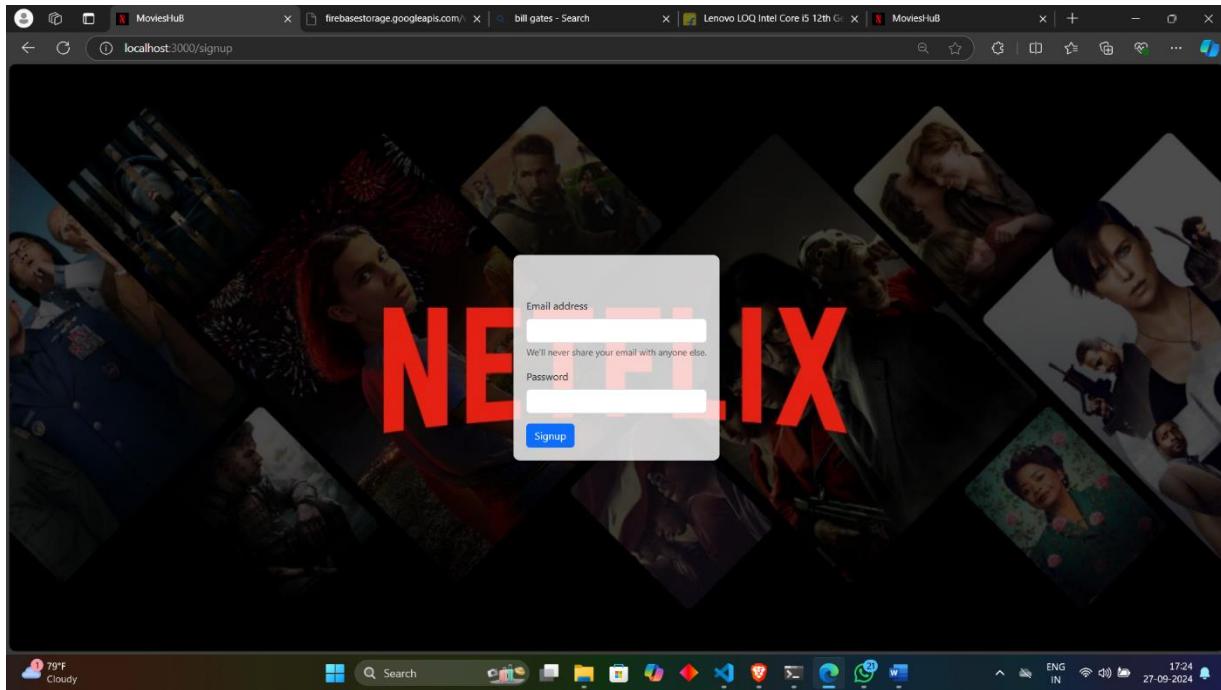
### **6.1. Login and Signup Pages**

*Figure 1: Login Page UI*



*Description: The user-friendly login interface where users can sign in using email/password or Google OAuth.*

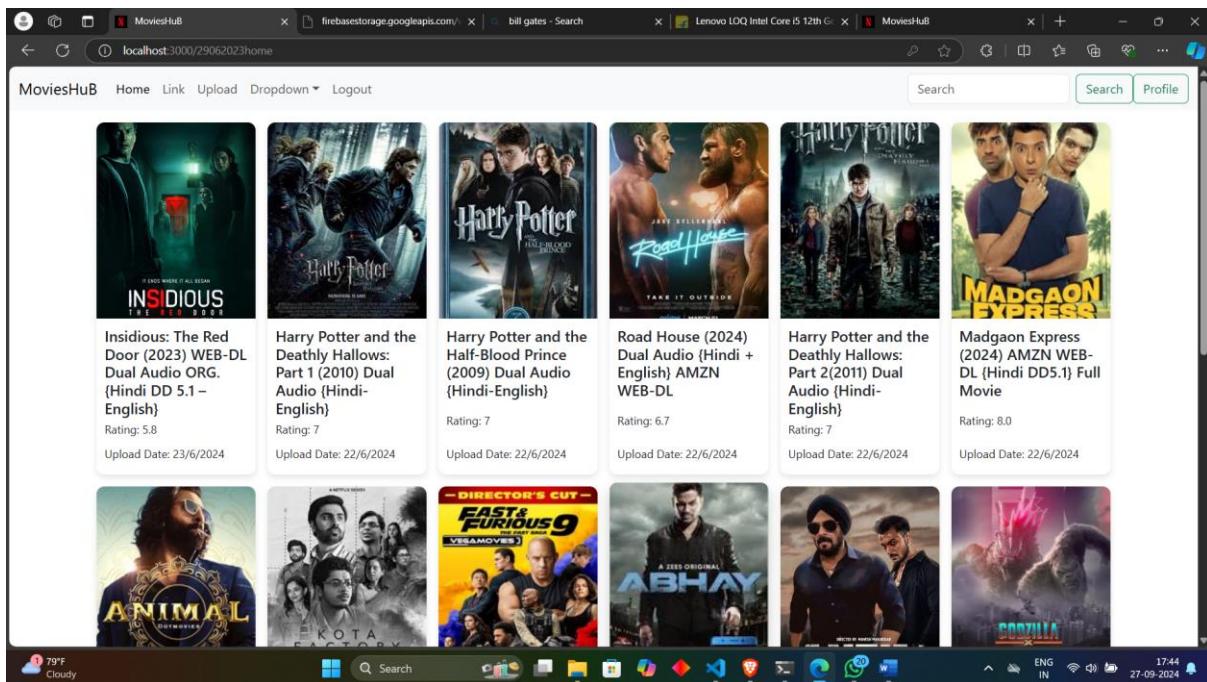
Figure 2: Signup Page UI



Description: The signup page that allows new users to create an account seamlessly.

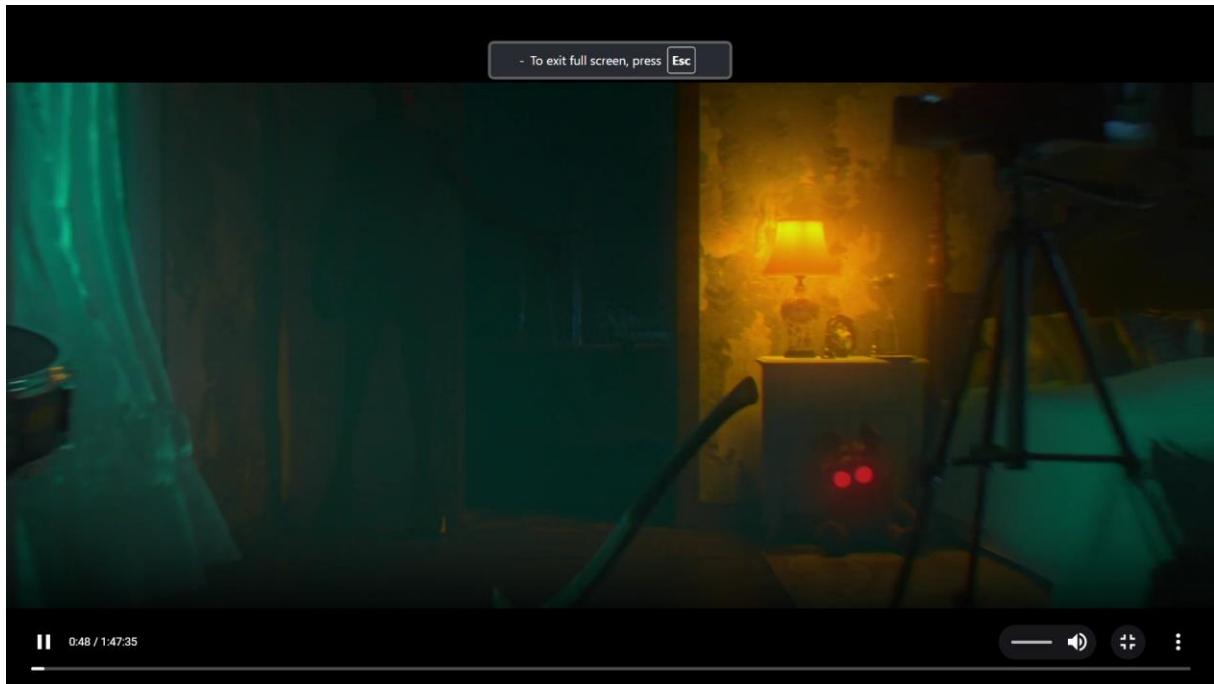
## 6.2. Home Page

Figure 3: Home Page UI



Description: The home page showcasing the infinite scroll feature for browsing the latest movies and TV shows.

### 6.3. Video stream



## 6.4. Profile Pages

Figure 4: Create Profile Page UI

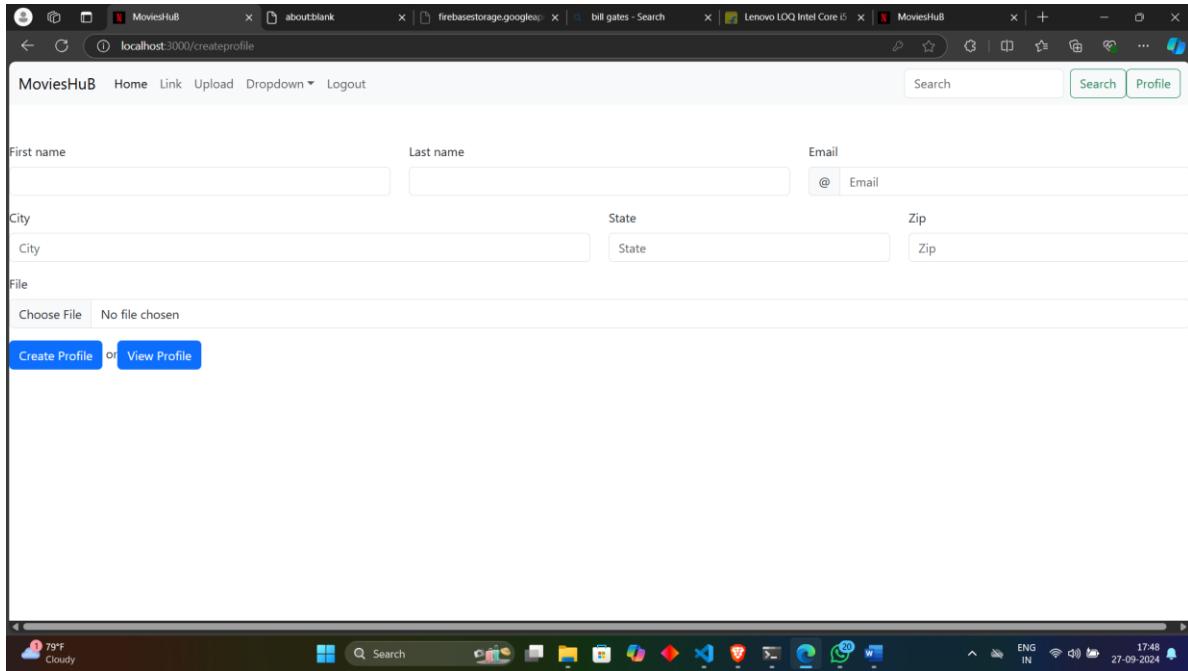
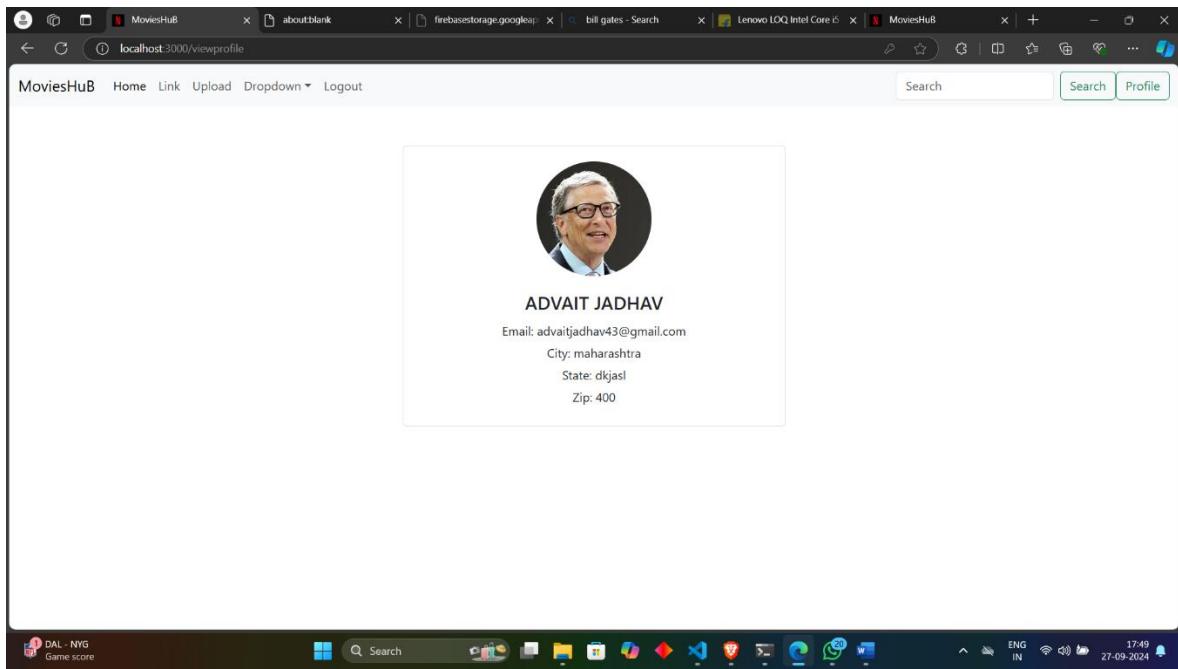


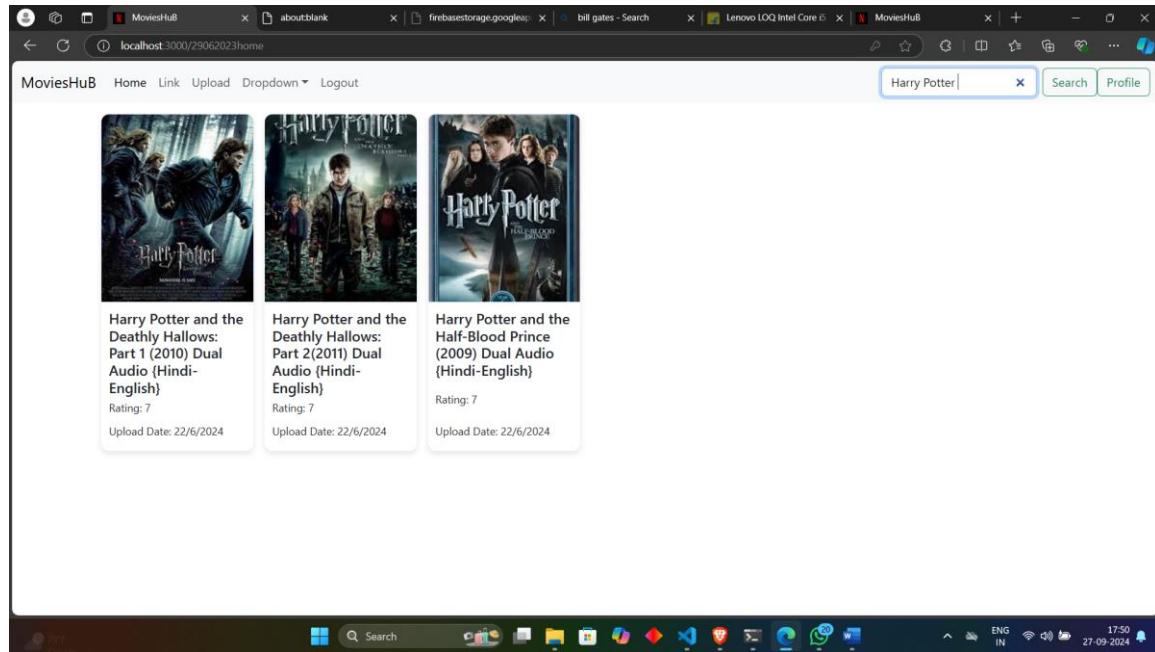
Figure 5: Profile Page UI



Description: The profile page where users can view and manage their profiles, including watch history and preferences.

## 6.5. Navigation Bar

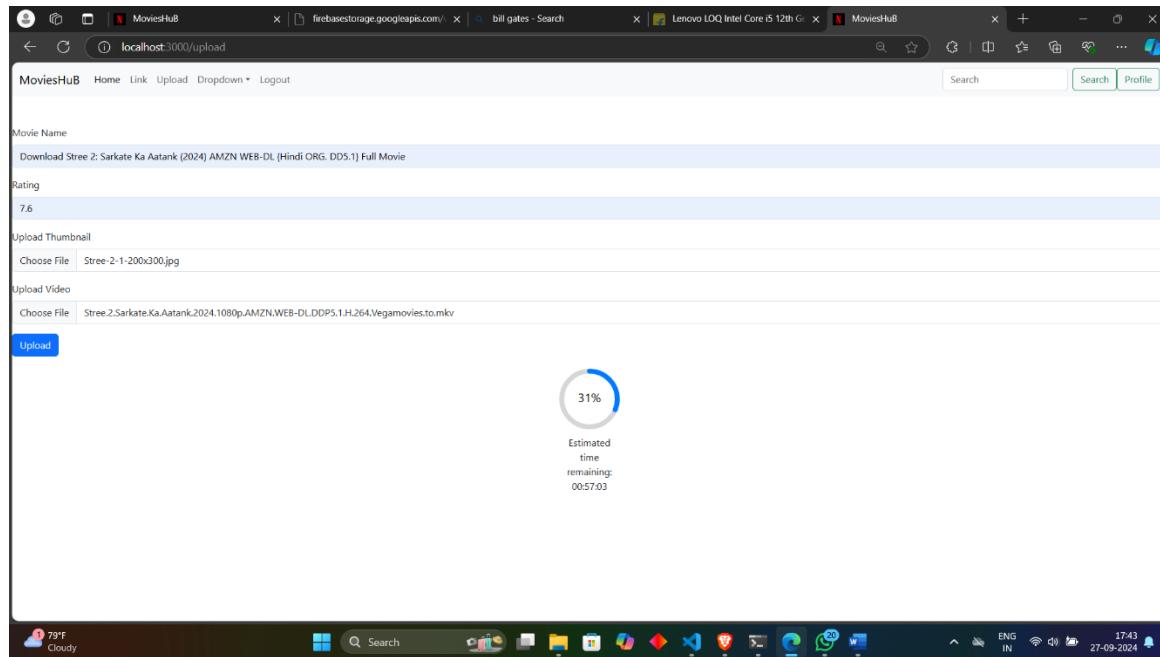
*Figure 6: Navigation Bar with search engine*



*Description: The navigation bar that facilitates easy access to various sections of the platform.*

## 7. Upload content

*Figure 7: upload content page*



## 7. Database Structure

Figure 7: Firebase Database Structure

The image displays two screenshots of the Firebase console interface, illustrating the database structure for a project named "movieweb".

**Storage Section:** This screenshot shows the "Storage" tab under the "movieweb" project. The left sidebar lists various services, with "Storage" selected. The main area shows a list of files in the "gs://movieweb-f2893.appspot.com/movies" bucket. The files listed include various movie thumbnails and videos, such as "2k\_video\_1719601478572.thumbnail", "2k\_video\_1719601478572\_video", and several large video files ranging from 2.23 GB to 23.44 GB.

Name	Size	Type	Last modified
2k_video_1719601478572.thumbnail	212.9 KB	image/png	29 Jun 2024
2k_video_1719601478572_video	17.69 MB	video/mp4	29 Jun 2024
Achay_2020_Season_1_Hindi_Complete_Zee5_Original_WEB_Series_1718872258495_video	23.44 GB	video/mp4	20 Jun 2024
Animal_2023_Hindi_Full_Movie_NF_WEB_DL_1719038128101_thumbnail	28.04 KB	image/jpeg	22 Jun 2024
Animal_2023_Hindi_Full_Movie_NF_WEB_DL_1719038128101_video	2.23 GB	video/x-matroska	22 Jun 2024
Antim_The_Final_Truth_2021_Hindi_1080p_WEB_DL_1718795069813_thumbnail	287.06 KB	image/png	19 Jun 2024
Antim_The_Final_Truth_2021_Hindi_1080p_WEB_DL_1718795069813_video	3.15 GB	video/x-matroska	19 Jun 2024
F9_The_Fast_Saga_2021_BluRay_Director_s_Cut_Dual_Audio_Hindi_English_1718876777500_thumbnail	19.97 KB	image/jpeg	20 Jun 2024
F9_The_Fast_Saga_2021_BluRay_Director_s_Cut_Dual_Audio_Hindi_English_1718876777500_video	3.17 GB	video/x-matroska	20 Jun 2024

**Authentication Section:** This screenshot shows the "Authentication" tab under the "movieweb" project. The left sidebar lists various services, with "Authentication" selected. The main area shows a list of users. The users listed are "advait@somaiya.edu", "sharvili123@gmail.com", "xyz@gmail.com", "advait@advav43@gmail.com", and "luciferhunter43@gmail.com". Each user entry includes their identifier, provider (Google), creation date, sign-in date, and user ID.

Identifier	Providers	Created	Signed In	User UID
advait@somaiya.edu	G	27 Sept 2024	27 Sept 2024	hdDjYENLmfcm2CsrofJLQ...
sharvili123@gmail.com	E	29 Jun 2024	29 Jun 2024	fwycghIR1XTQbbmOkQFBHD...
xyz@gmail.com	E	22 Jun 2024	22 Jun 2024	QzHuNaZz5XZ9TWcOHr973...
advait@advav43@gmail.com	G	20 Jun 2024	21 Sept 2024	FQSvZ2xdLca3pbf0Fmu14sR...
luciferhunter43@gmail.com	G	20 Jun 2024	12 Jul 2024	iTHB6es6XdfNMSShasC43G...

The screenshot shows the Firebase Cloud Firestore interface for a project named "movieweb". The left sidebar lists various services: Generative AI, Build with Gemini, Authentication, Firestore Database (selected), Functions, Storage, Hosting, Data Connect, App Check, Crashlytics, Machine Learning, Performance, Analytics Dashboard, and a placeholder for "New View". Below this, it says "Spark No cost (\$0/month)" and "Upgrade". The main area displays a hierarchical view of collections: "home" > "movie" > "movies". A modal window titled "movies" is open, showing a list of document names: "Fast...\_Furious\_8...\_The\_Fate\_of\_the\_Furious...\_2017...\_Dual...\_Godzilla\_x\_Kong\_The\_New\_Empire\_2024\_BluRay\_1080p\_Hindi...\_Harry\_Potter\_and\_the\_Deathly\_Hallows\_Part\_1\_2010\_D...\_Harry\_Potter\_and\_the\_Deathly\_Hallows\_Part\_2\_2011\_D...\_Harry\_Potter\_and\_the\_Half\_Blood\_Prince\_2009\_Dual\_Au...\_Insidious\_The\_Red\_Door\_2023\_WEB\_DL\_Dual\_Audio\_ORG...\_Jurassic\_World\_CBF0\_U\_A\_2015\_Action\_Sci\_fi\_2h\_4m...\_Kota\_Factory\_Season\_3\_Hindi\_Complete\_Netflix\_Original...\_Madgaon\_Express\_2024\_AMZN\_WEB\_DL\_Hindi\_DDS\_1\_Full...\_Maharshi\_2019\_Dual\_Audio\_Hindi\_ORG\_2\_0\_Telugu...\_Mirzapur\_Season\_3\_Official\_Teaser\_Pankaj\_Tripathi...\_Road\_House\_2024\_Dual\_Audio\_Hindi\_English\_AMZN\_W...\_Swatantra\_Veer\_Savarkar\_2024\_WebRip\_1080p\_Hindi\_DD\_5...\_Wild\_Wild\_Punjab\_Movie\_Netflix\_Full\_Cast\_and\_Crew\_Wik...". A message at the bottom of the modal states: "This document does not exist. It will not appear in queries or snapshots. Learn more". The status bar at the bottom shows "Database location: nam5", the date "27-09-2024", and the time "17:14".

*Description: A diagram illustrating the structure of the Firebase Realtime Database used in MoviesHub, showing collections for user data and content.*

## **8. Security Considerations**

Security is paramount in MoviesHuB, especially concerning user authentication and data protection. Firebase Authentication ensures secure login mechanisms, while Firebase Realtime Database employs robust security rules to safeguard user data. Additionally, role-based access control restricts administrative functionalities to superusers only.

## **9. Conclusion**

MoviesHuB presents a comprehensive solution for users seeking a unified, ad-free streaming experience across multiple OTT platforms. By leveraging modern technologies such as React, Firebase, and Node.js, MoviesHuB delivers a scalable, secure, and user-friendly platform. Future enhancements may include advanced recommendation algorithms, social features, and expanded content libraries to further enrich the user experience.

## **10. References**

- React.js Documentation
- Firebase Documentation
- Node.js Documentation
- OAuth 2.0 and OpenID Connect
- IEEE Citation Guidelines