SHORTFRAME: A WEB-BASED PLATFORM FOR SHORT FILM STREAMING AND MONETIZATION

RAVIKUMAR S
III BCA STUDENT
DEPARTMENT OF COMPUTER APPLICATIONS(UG),
SCHOOL OF COMPUTING SCIENCE,
VISTAS CHENNAI.
ksravikumar3331@gmail.com

DR. V. DIVYA
ASSISTANT PROFESSOR
DEPARTMENT OF COMPUTER APPLICATIONS(UG),
SCHOOL OF COMPUTING SCIENCE,
VISTAS CHENNAI.
divyavenkatraman1992@gmail.com

ABSTRACT

The increasing use of the internet has changed the way people watch videos, especially short-form content. However, most popular platforms mainly focus on entertainment and viral videos, making it difficult for short film creators to gain recognition and earn revenue. To solve this problem, a web-based platform named SHORTFRAME has been developed. The SHORTFRAME is designed specifically for hosting and sharing short films, documentaries, and create videos. It allows users to register, upload videos, watch content, and interact through likes and comments. The platform also provides a creator dashboard where users can tract views, engagement, and estimated earnings. The application is built using React for the frontend and firebase for backend services such as authentication and database management. Video files are stored and delivered using Cloudinary for better performance. Monetization is implemented using Google AdSense, enabling creators to earn from their content. The platform is available online at: http://shortframe-3c23f.web.app . Overall, SHORTFRAME provides a simple, efficient, and creator-focused solution for sharing and monetizing short video content.

KEYWORDS – Short Film Platform, Video Streaming, Web Application, Content Monetization, React, Firebase, Cloudinary, Google AdSense.

I. INTRODUCTON

A. BACKGROUND

The growth of internet technology and smartphones has made video content one of the most popular forms of digital media. Short-form videos, in particular, are widely consumed due to their quick and engaging nature. However, most existing platforms mainly promote entertainment and viral content, rather than meaningful short films or documentaries. This creates a need for a dedicated platform that supports creative storytelling and independent creators.

B. PROBLEM STATEMENT

Independent short film creators face several challenges on current platforms:

- Lack of visibility due to algorithm-based content promotion
- Difficulty in monetizing content without large audiences
- No dedicated platform for short films and creative videos
- Limited access to clear performance analytics

These issues make it difficult for new creators to grow and sustain their work.

C. OBJECTIVES

The main objectives of the SHORTFRAME project are:

- To develop a web-based platform for uploading and sharing short videos
- To allow users to like, comment, and search for videos
- To provide a creator dashboard with performance metrics
- To enable monetization through advertisements
- To ensure user authentication and management

D. SCOPE OF THE PROJECT

The project focuses on building a complete web application with the following features:

- User registration and login system
- Video upload and streaming
- Creator dashboard for analytics
- Admin panel for managing users and content

The system is designed as a responsive web application and does not include a mobile app at this stage.

II. LITERATURE REVIEW

Here, it provides an analysis of existing technologies, research, and platforms related to video streaming systems, cloud-based application development, and monetization strategies. The purpose of this review is to understand current industry practices and identify the limitations that affect independent creators. Based on these observations, the SHORTFRAME platform is designed to overcome these challenges by offering a focused and efficient solution.

A. VIDEO STREAMING PLATFORMS

Video streaming platforms have evolved rapidly over the last decade. Platforms such as YouTube, TikTok, and Vimeo allow users to upload and share video content globally. YouTube supports both long-form and short-form content, while TikTok focuses mainly on short, engaging videos.

However, these platforms rely heavily on recommendation algorithms that prioritize trending and highly engaging content. As a result, creative short films and documentaries often receive less visibility. Vimeo provides a better environment for creative professionals, but it follows a subscription-based model, which limits accessibility for many users. This analysis highlights the need for a specialized platform that focuses on short films and supports creators without strict monetization requirements.

B. BACKEND-AS-A-SERVICE (BAAS)

Backend-as-a-Service (BaaS) platforms simplify application development by providing ready-to-use backend functionalities. Firebase is one of the most popular BaaS platforms and offers services such as authentication, database management, cloud hosting, and real-time updates. Using Firebase eliminates the need for building and maintaining a custom backend server. It also reduces development time and allows developers to focus more on frontend features and user experience. The Fire store database used in Firebase is scalable and suitable for applications that require frequent read and write operations, such as social media platforms.

C. REAL-TIME DATABASES

Traditional databases require manual refresh or polling to update data, which increases latency. In contrast, Fire store provides real-time data synchronization using listeners. This allows applications to automatically update content when changes occur in the database.In the SHORTFRAME platform, this feature is used to implement real-time comments. When a user posts a comment, it instantly appears for all other users viewing the same video. This improves user engagement and creates an interactive environment.

D. CLOUD MEDIA DELIVERY

Efficient media storage and delivery are essential for video-based platforms. Cloudinary is a cloud-based media management service that provides storage, optimization, and delivery through a Content Delivery Network (CDN). Compared to traditional storage solutions, Cloudinary ensures faster loading times and reduces buffering during video playback. It also supports various file formats and automatic optimization, which enhances performance across different devices and network conditions.

E. WEB MONETIZATION

Monetization is a key aspect of any content platform. Google AdSense is widely used for displaying advertisements on websites. It allows content creators to earn revenue based on impressions and clicks. Ad-based monetization is simple to integrate and does not require users to pay for accessing content. This makes it suitable for platforms like SHORTFRAME, where the goal is to support independent creators and provide free access to viewers.

F. SINGLE PAGE APPLICATIONS

Single Page Applications (SPAs) are modern web applications that load a single HTML page and dynamically update content without reloading the entire page. React is one of the most popular frameworks used to build SPAs. React improves performance by using a virtual DOM, which updates only the required parts of the interface. This results in faster loading times and a smoother user experience. In SHORTFRAME, React is used to create a responsive and interactive interface for users.

G. SECURITY IN WEB APPLICATIONS

Security is a critical factor in any web application. Firebase Security Rules are used to control access to the database and ensure that only authorized users can perform certain actions. For example, users can only modify their own data, while admin users have higher privileges to manage content and users. This approach ensures data integrity and protects sensitive information such as user details and earnings data.

III. SYSTEM DESIGN AND ARCHITECTURE

A. DESIGN PHILOSOPHY

The SHORTFRAME platform is designed based on three main principles: **simplicity**, **accessibility**, and **transparency**. Simplicity ensures that users can easily navigate and use the platform. Accessibility means the platform is available to all users through a web browser without any cost. Transparency ensures that creators can clearly view their performance metrics and earnings without hidden conditions.

B. SYSTEM ARCHITECTURE

The system follows a three-tier architecture:

- **Presentation Layer**: Built using React, responsible for user interface and user interaction
- **Application Layer**: Handles business logic such as authentication, video handling, and data processing
- **Data Layer**: Uses Firebase Fire store for storing data and Cloudinary for storing media files

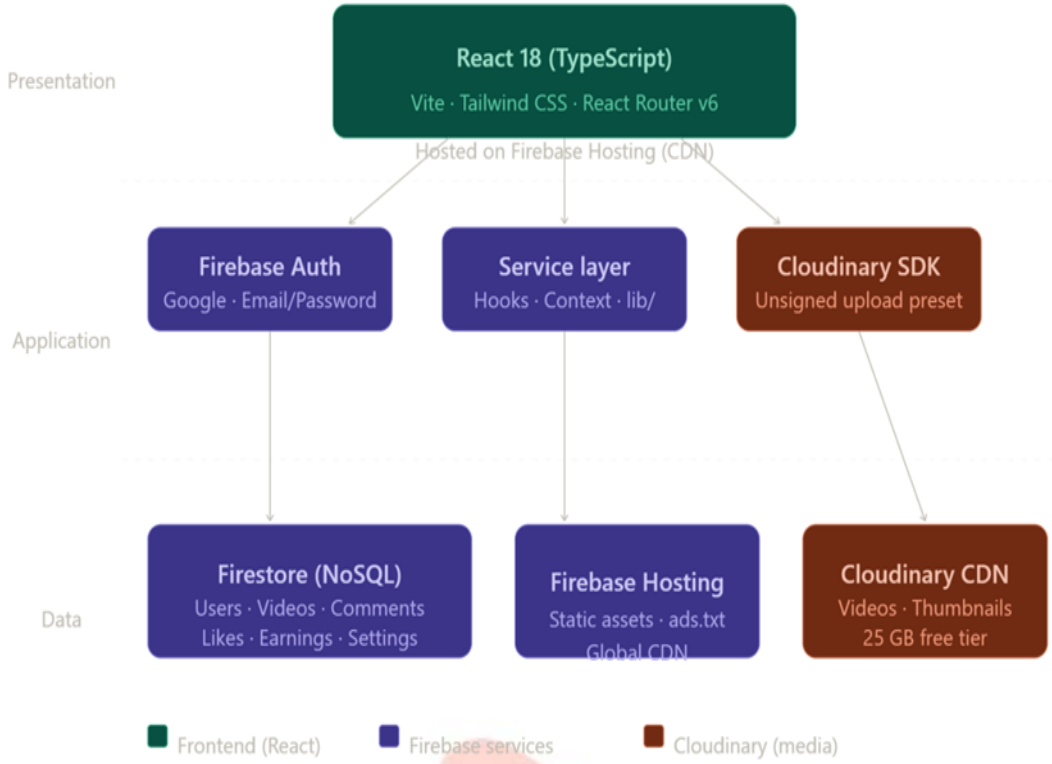

Fig.3.1 Architecture of three-tier

C. DATA MODEL DESIGN

The database is designed using Fire store collections for efficient data management:

- **Users Collection** → stores user details and roles
- **Videos Collection** → stores video information such as title, description, URL, and views
- **Comments Collection** → stores user comments for each video
- **Likes Collection** → tracks user likes for videos
- **Earnings Collection** → stores estimated earnings per video

This structure allows fast data retrieval and easy scalability

D. AUTHENTICATION FLOW

User authentication is handled using Firebase Authentication.

- Users can sign in using Google or email/password
- On successful login, user data is stored in Fire store
- Authentication state is maintained across sessions

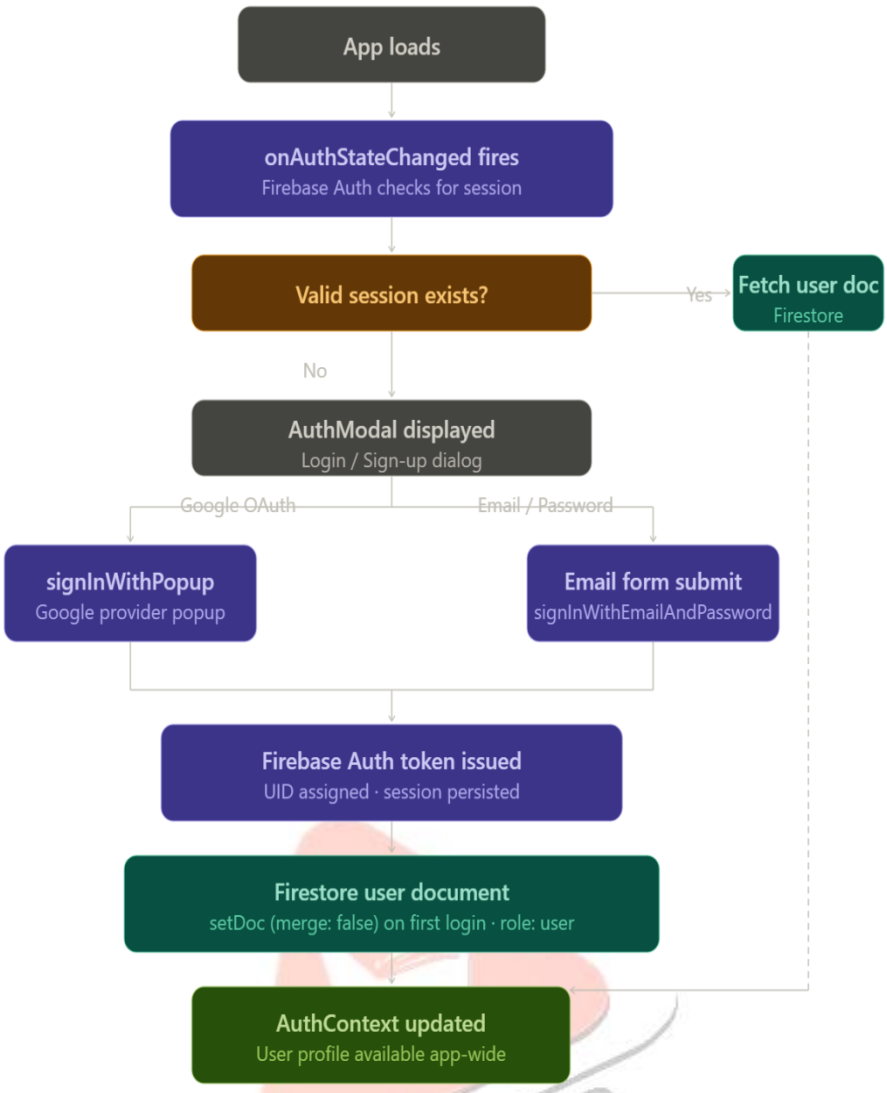

Fig.3.2 Authentication Flowchart

E. VIDEO UPLOAD FLOW

The video upload process follows a structured flow:

- User selects video and thumbnail
- Files are uploaded to Cloudinary
- URLs are returned after upload
- Video details are stored in Fire store

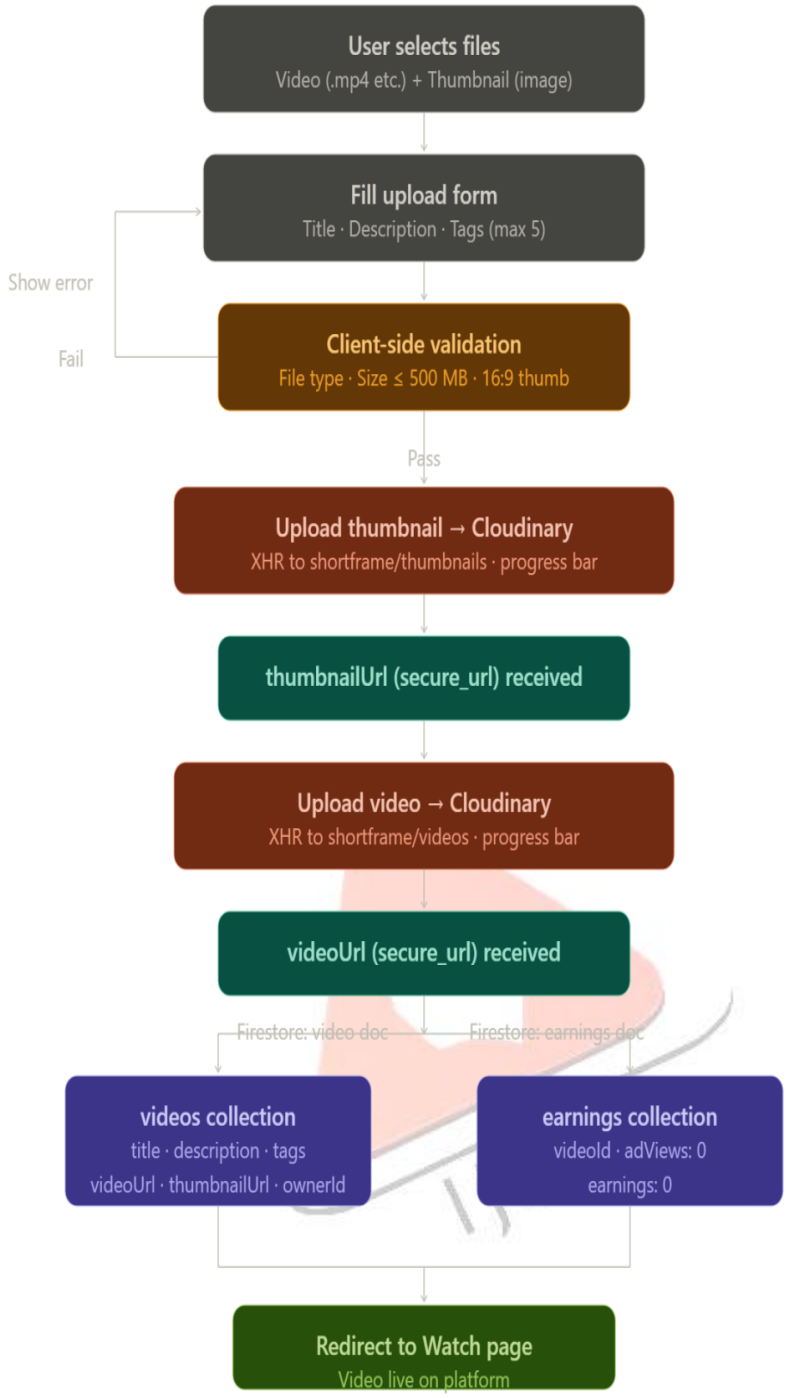

Fig.3.3 Video Upload Flowchart

IV. IMPLIMENTATION

A. TECHNOLOGY STACK

The technologies used in this project are:

- **Frontend**: React with TypeScript
- **Styling**: Tailwind CSS
- **Backend Services**: Firebase (Authentication, Fire store, Hosting)
- **Media Storage**: Cloudinary
- **Routing**: React Router
- **Monetization**: Google AdSense

B. FIREBASE PROJECT SETUP

Firebase is used as the backend service for authentication, database, and hosting. The setup includes:

- Creating a Firebase project

- Enabling Authentication (Google and Email/Password)
- Setting up Fire store Database for storing users, videos, comments, and other data
- Configuring Firebase Hosting to deploy the web application

Firebase provides a secure and scalable backend without requiring a custom server.

C. CLOUDINARY SETUP

Cloudinary is used for storing and delivering video and image files.

- A Cloudinary account is created
- Upload presets are configured for secure uploads
- Videos and thumbnails are uploaded directly from the frontend
- This reduces storage cost and improves media delivery performance using CDN.

D. GOOGLE ADSENSE SETUP

Google AdSense is integrated to enable monetization on the platform.

- The website is registered in AdSense
- Ad units are created for displaying ads
- Publisher ID and ad slot IDs are configured in the project

This allows creators to earn revenue based on video views and user engagement.

E. ENVIRONMENT VARIABLE

Sensitive data is stored using environment variables in a .env file.

Examples include:

- Firebase configuration keys
- Cloudinary credentials
- AdSense client and slot IDs

This ensures that important information is kept secure and not exposed in the source code.

F. VIEW AND EARNINGS TRACKING

The system tracks user engagement through views and interactions.

- A view is counted after a short duration of video playback
- Earnings are calculated based on ad impressions
- Data is updated in real time in the database

This provides creators with a clear understanding of their content performance.

G. REAL-TIME COMMENTS

The comment system uses Fire store real-time listeners.

- When a user posts a comment, it is instantly updated for all viewers
- No page refresh is required
- Improves user engagement and interaction

This creates a dynamic and interactive user experience.

H. SEARCH IMPLEMENTATION

The search feature allows users to find videos using keywords.

- Searches are performed based on title, description, and tags
- Filtering is done on the client side
- Results are displayed instantly

Although simple, this method is effective for small to medium datasets.

I. SECURITY RULES DESIGN

Security is implemented using Firebase Security Rules.

- Only authenticated users can perform actions like upload and comment
- Users can modify only their own data
- Admin users have full control over the system

This ensures data protection and prevents unauthorized access.

J. DEPLOYMENT STRATEGY

The application is deployed using Firebase Hosting.

- The project is built using Vite
- Files are deployed to Firebase CDN
- The platform is accessible globally

Live Application:

https://shortframe-3c23f.web.app

This deployment method ensures fast loading and high availability.

VI. RESULTS AND DISCUSSION

A. FUNCTIONAL TESTING RESULTS

The SHORTFRAME platform was tested to ensure all features work correctly. Testing was performed on different browsers such as Chrome and Edge.

The following functionalities were tested:

- User registration and login
- Video upload and playback
- Like and comment system
- Search functionality
- Creator dashboard
- Admin controls

All major features worked successfully without critical errors. The system was able to handle user interactions smoothly and maintain data consistency.

Table. 5.1 Testing Result

S.No	Feature	Test Description	Expected Result	Actual Result	Status
1	User Registration	Register with email and password	User account created successfully	Account created successfully	Pass
2	User Login	Login using valid credentials	User logged in and redirected to homepage	Login successful	Pass
3	Video Upload	Upload video with thumbnail	Video uploaded and displayed	Upload successful	Pass
4	Video Playback	Play uploaded video	Video should play smoothly	Video played without issues	Pass

S.No	Feature	Test Description	Expected Result	Actual Result	Status
5	Like System	Like and unlike a video	Like count updates correctly	Like system working correctly	Pass
6	Comment System	Add a comment on a video	Comment appears instantly	Comment displayed in real-time	Pass
7	Search Function	Search using title or tag	Relevant videos should be displayed	Search results displayed correctly	Pass
8	Dashboard	View creator statistics	Shows views, likes, and earnings	Dashboard displayed correct data	Pass
9	Admin Control	Delete a video as admin	Video removed from platform	Video deleted successfully	Pass
10	Security	Access restricted page without login	Access should be denied	Redirected to login page	Pass

B. PERFORMANCE ANALYSIS

The performance of the application was evaluated based on loading speed and responsiveness.

- The homepage loads quickly due to optimized frontend design
- Videos stream smoothly using Cloudinary CDN
- Real-time updates (comments, likes) occur with minimal delay
- Navigation between pages is fast due to single-page application design

Overall, the platform provides a smooth and responsive user experience.

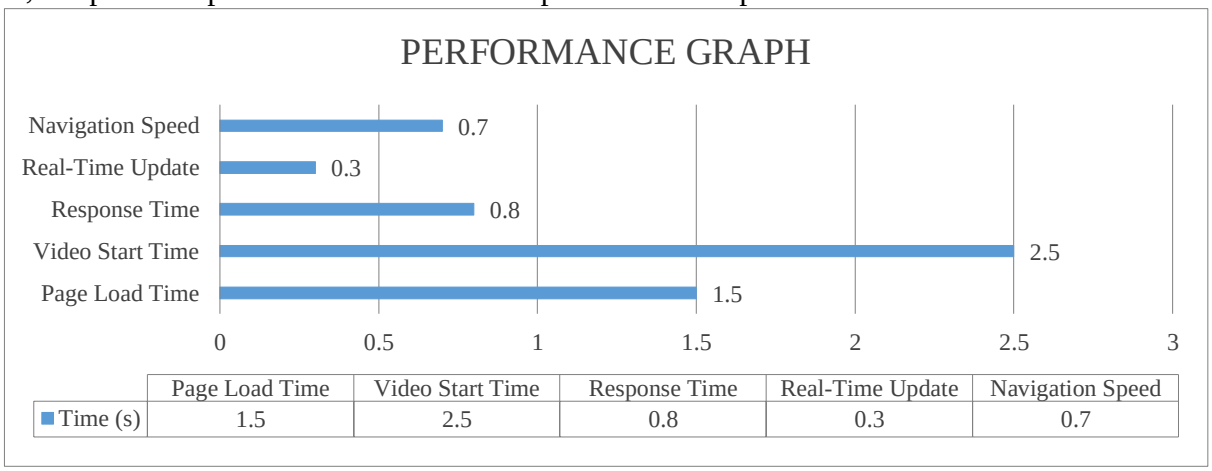

Fig. 5.1 Performance Graph

C. CREATOR DASHBOARD

The dashboard provides creators with insights about their content performance. It displays total views, likes, and earnings in a clear format. Each uploaded video is also listed with individual statistics.

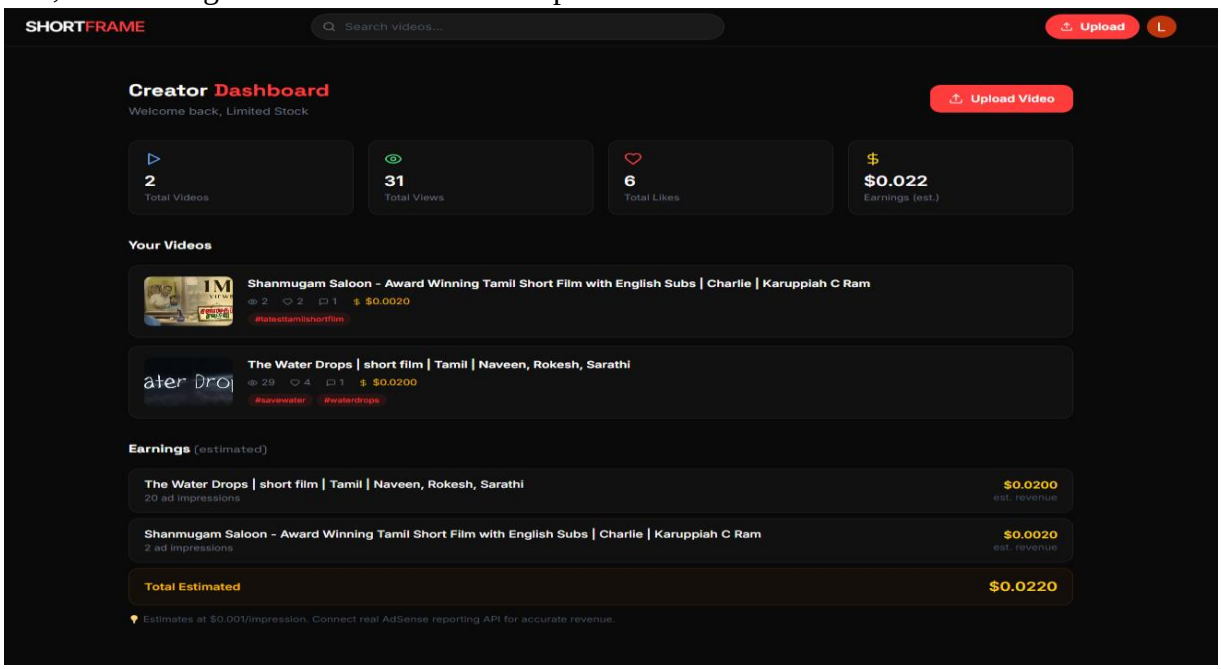

Fig.5.2 Dashboard Page

D. SECURITY EVALUATION

Security testing was conducted to ensure data protection.

- Unauthorized users cannot access restricted features
- Users can modify only their own data
- Admin-only actions are properly restricted
- Sensitive data is protected using Firebase Security Rules

The system successfully prevents unauthorized access and maintains data integrity.

E. LIMITATIONS

Despite successful implementation, some limitations were identified:

- Search functionality is not scalable for large datasets
- Earnings are currently estimated and not real-time
- No mobile application version available
- No advanced recommendation system

These limitations can be improved in future updates.

VI. CONCLUSTION & FUTURE-WORK

A. CONCLUTION

The SHORTFRAME project was developed to provide a dedicated platform for short films and creative video content. The system successfully allows users to upload, watch, like, comment, and search videos in a simple and structured environment. It also provides a creator dashboard where users can track their video performance and estimated earnings. The platform is built using modern technologies such as React for the frontend and Firebase for backend services, ensuring scalability and ease of development. Cloudinary is used for efficient media storage and delivery, improving video streaming performance. The application is deployed and accessible online at: https://shortframe-3c23f.web.app. Overall, the project meets its main objectives and provides a functional solution for independent content creators.

- KEY CONTRIBUTIONS: The project makes the following important contributions:
  - Development of a dedicated platform for short film creators
  - Integration of video upload, streaming, and user interaction features
  - Implementation of a creator dashboard with performance tracking
  - Use of cloud-based services to reduce infrastructure complexity
  - Secure data management using Firebase Authentication and Fire store

  These contributions highlight the practical value of the project in real-world apps.
- LESSONS LEARNED: During the development of this project, several important lessons were learned:
  - Cloud-based services like Firebase simplify backend development
  - Proper database design is important for performance and scalability
  - Real-time features improve user engagement significantly
  - Security rules must be carefully implemented to protect data
  - Planning for scalability is important for future growth

  These learnings will be helpful for future projects and improvements

B. FUTUREWORK

The SHORTFRAME platform has been successfully developed with core features, but several improvements can be made in future to enhance its functionality and scalability.

- Advanced Search System:
  Implement a full-text search system using tools like Alogia or Type-sense to handle large volumes of data efficiently.
- Real-Time Revenue Integration:
  Integrate official Google AdSense reporting APIs to display accurate and real-time earnings instead of estimated values.
- Mobile Application Development:
  Develop a mobile app using React Native to provide a better user experience on smartphones.
- AI-Based Recommendation System:
  Implement machine learning algorithms to suggest personalized video content based on user behaviour and watch history.
- Video Optimization and Transcoding:
  Add support for multiple video resolutions (e.g., 720p, 480p) to improve streaming performance across different network conditions.
- Notification System:
  Introduce email or push notifications for user interactions such as comments, likes, and new uploads.
- Creator Support Features:
  Add features like user subscriptions, follow system, and direct support options (payments or tips).
- Content Moderation System:
  Implement automated content moderation using AI tools to detect inappropriate content.

These improvements will enhance the platform's usability, scalability, and overall user experience in future versions.

REFERANCE

[1] **Lobato, R. (2016).** The cultural logic of digital intermediaries: YouTube and the future of media distribution. Journal of Media Studies.

[2] **Cunningham, S., & Craig, D. (2019).** Social Media Entertainment: The New Intersection of Hollywood and Silicon Valley. NYU Press**.**

[3] **Smith, J., Brown, L., & Taylor, K. (2021).** Video streaming trends and user engagement analysis. International Journal of Digital Media.
[4] **Poell, T., Nieborg, D., & Duffy, B. (2022).** Platforms and Cultural Production. Polity Press.
[5] **Kyncl, R., & Peyvan, M. (2017).** Streampunks: YouTube and the Rebels Remaking Media. Harper Business.
[6] **Sehl, K. (2023).** Digital video marketing trends and monetization strategies. Hootsuite Research Report.
[7] **Cheng, X., Liu, J., & Dale, C. (2013).** Understanding the characteristics of internet short video sharing: A YouTube-based measurement study. IEEE Transactions.
[8] **Duffy, B. E. (2020).** Algorithmic precarity in social media platforms. Communication Research Journal.
[9] **Firebase Documentation. (2024).** Firebase Authentication and firestone Guide. Available at: https://firebase.google.com/docs
[10] **React Documentation. (2024).** React Official Guide. Available at: https://react.dev
[11] **Cloudinary Documentation. (2024).** Media Management and Delivery. Available at: https://cloudinary.com/documentation
[12] **Google AdSense Help. (2024).** Ad Monetization for Websites. Available at: https://support.google.com/adsense
[13] **Vite Documentation. (2024).** Frontend Build Tool Guide. Available at: https://vitejs.dev
[14] **Tailwind CSS Documentation. (2024).** Utility-first CSS Framework. Available at: https://tailwindcss.com/docs
[15] **Statista. (2023).** Growth of short-form video consumption worldwide. Statista Research Department.