Department of Computer Science & Applications

Institute of Engineering & Technology

Project Report

On

PLAYSTREAM

A Video Sharing Platform

Submitted By:

Lokesh

Gauri Agrawal

Submitted To:

Mr. Bhanu Kapoor

(Technical Trainer)

CERTIFICATE

This is to certify that the Project Report entitled "**PLAYSTREAM**" is submitted to SUPERVISOR Mr. Bhanu Kapoor (Technical Trainee) of the department of Computer Science and Engineering, GLA University, Mathura in partial fulfillment for the award of the degree of bachelor of technology in computer science and engineering is a record of bona fide carried out by:

Lokesh (University Roll no: 201500371)

Gauri Agrawal (University Roll no: 201500252)

Signature of Authorized

Mr. Bhanu Kapoor

(Technical Trainer)

ABSTRACT

Our video sharing platform is a cutting-edge online community that allows users to create, upload, and share videos with ease. With an intuitive interface and powerful features, our platform provides a seamless experience for users to express themselves and connect with others through the power of video.

Users can upload videos of various genres, including entertainment, education, sports, news, and more. Our platform supports high-quality video playback, making it perfect for both amateur and professional videographers. Users can also interact with videos through comments, likes, and shares, fostering a vibrant and engaging community.

Our platform also offers advanced video management tools, including video analytics, privacy settings, and content moderation, giving users full control over their videos. Additionally, our platform is mobile-friendly, allowing users to access and share videos on the go.

Our video sharing platform is designed with user experience and community engagement in mind, providing a safe and inclusive space for users to express themselves, discover new content, and connect with like-minded individuals. Join our video sharing community and be part of the next generation of online video sharing.

Contents

INTRODUCTION

This platform boasts a sleek and modern interface, making it easy to navigate and explore the vast array of videos available. From engaging tutorials and entertaining vlogs to thought-provoking documentaries and captivating short films, our platform offers a diverse range of content that caters to a wide range of interests. This platform also provides robust search and recommendation features, helping users discover new videos based on their preferences and interests. Additionally, our responsive design ensures seamless viewing on various devices, including desktops, tablets, and mobile phones. The base technology used is based on REACT, HTML, CSS and front-end JavaScript framework. Back end and Database management of this system is developed with the help of Firebase database.

Development Tools:

Visual Studio Code: Used Visual Studio Code as working environment to run and test the code.

Visual Studio Installer: Used Visual Studio Installer as a setup tool to create installation programs and setup packages.

Framework Used: REACT, HTML, CSS, BOOTSTRAP, JavaScript and Firebase.

FRAMEWORK DESCRIPTION

React:

React is a popular open-source JavaScript library used for building user interfaces. Developed by Facebook, React allows developers to create reusable UI components that are efficient, declarative, and easily maintainable. With its virtual DOM and one-way data flow, React provides a performant way to build dynamic, interactive web applications. React's component-based architecture and extensive ecosystem of tools and libraries make it a preferred choice for modern web development, enabling developers to create highly interactive and responsive user interfaces.

JavaScript:

It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses JS to provide several forms of interactivity and simplicity. JavaScript is used to create interactive websites. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

Firebase:

Firebase is a product of Google which helps developers to build, manage, and grow their apps easily. It helps developers to build their apps faster and in a more secure way. No programming is required on the firebase side which makes it easy to use its features more efficiently. It provides services to android, iOS, web, and unity. It provides cloud storage. It uses NoSQL for the database for the storage of data.

CSS:

CSS allows you to apply styles to web pages. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser. CSS comprises style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule set consists of a selector and declaration block.

HTML:

HTML is used to create the structure of web pages that are displayed on the World Wide Web (www). It contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks. HTML is used to create the structure of web pages that are displayed on the World Wide Web (www). It contains Tags and Attributes that are used to design the web pages. Also, we can link multiple pages using Hyperlinks.

CODES AND OUTPUTS:

Index.js

```
import ReactDOM from "react-dom";
import React from "react";
import App from './App';
import { Provider } from 'react-redux'
import 'bootstrap/dist/css/bootstrap.min.css'
import 'react-lazy-load-image-component/src/effects/blur.css'
import store from "./redux/store";
import { BrowserRouter as Router } from 'react-router-dom'



ReactDOM.render(
    <Provider store={store}>
        <Router>
            < App />
        </Router>
    </Provider>
    , document.getElementById('root'));
```

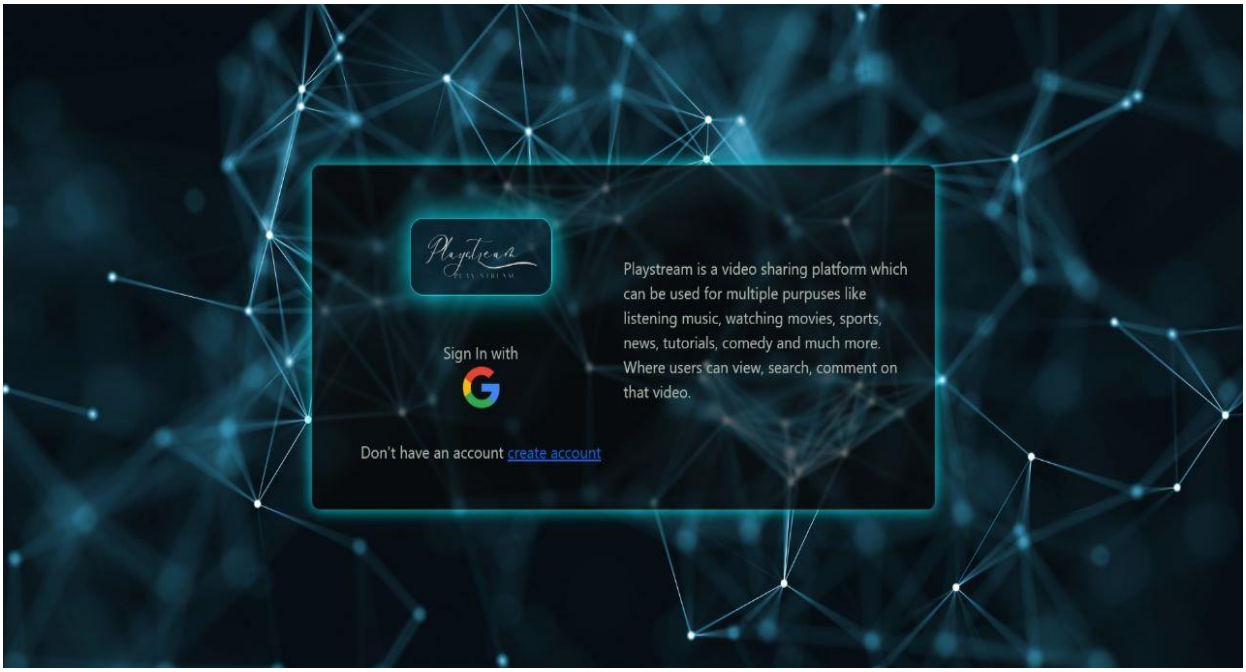
App.js

```
import React, { useEffect, useState } from 'react'
import { Container } from 'react-bootstrap'

import Header from './components/header/Header'
import Sidebar from './components/sidebar/SideBar'
import HomeScreen from './screens/homeScreen/HomeScreen'
import LoginScreen from './screens/loginScreen/LoginScreen'

import { Navigate, Route, Routes, useNavigate } from 'react-router-dom'

import './_app.scss'
import { useSelector } from 'react-redux'
import WatchScreen from './screens/watchScreen/WatchScreen'
import SearchScreen from './screens/searchScreen/SearchScreen'
import SubscriptionScreen from './screens/subscriptionScreen/SubscriptionScreen'
import ChannelScreen from './screens/channelScreen/ChannelScreen'




const Layout = ({ children }) => {
   const [sidebar, toggleSidebar] = useState(false)

   const handleToggleSidebar = () => toggleSidebar(value => !value)

   return (
      <>
         <Header handleToggleSidebar={handleToggleSidebar} />
         <div className='app_container'>
            <Sidebar
               sidebar={sidebar}
               handleToggleSidebar={handleToggleSidebar}
            />
            <Container fluid className='app_main '>
               {children}
            </Container>
         </div>
      </>
   )
}

const App = () => {



   const { accessToken, loading } = useSelector(state => state.auth)

   const navigate = useNavigate()


   useEffect(() => {
```

```
      if (!loading && !accessToken) {
         navigate('/auth')
      }

   }, [accessToken, loading, navigate])



   return (
      <Routes>

         <Route path='/' exact element={<Layout><HomeScreen /></Layout>} />


         <Route path='/auth' exact element={<LoginScreen />} />


         <Route path='/search/:query' exact element={<Layout><SearchScreen
/></Layout>} />

         <Route path='/watch/:id' exact element={<Layout><WatchScreen
/></Layout>} />

         <Route path='/feed/subscriptions' exact
element={<Layout><SubscriptionScreen /></Layout>} />

         <Route path='/channel/:channelId' exact element={<Layout><ChannelScreen
/></Layout>} />


         <Route exact element={<Navigate to='/' />} />
      </Routes>
   )
}

export default App
```

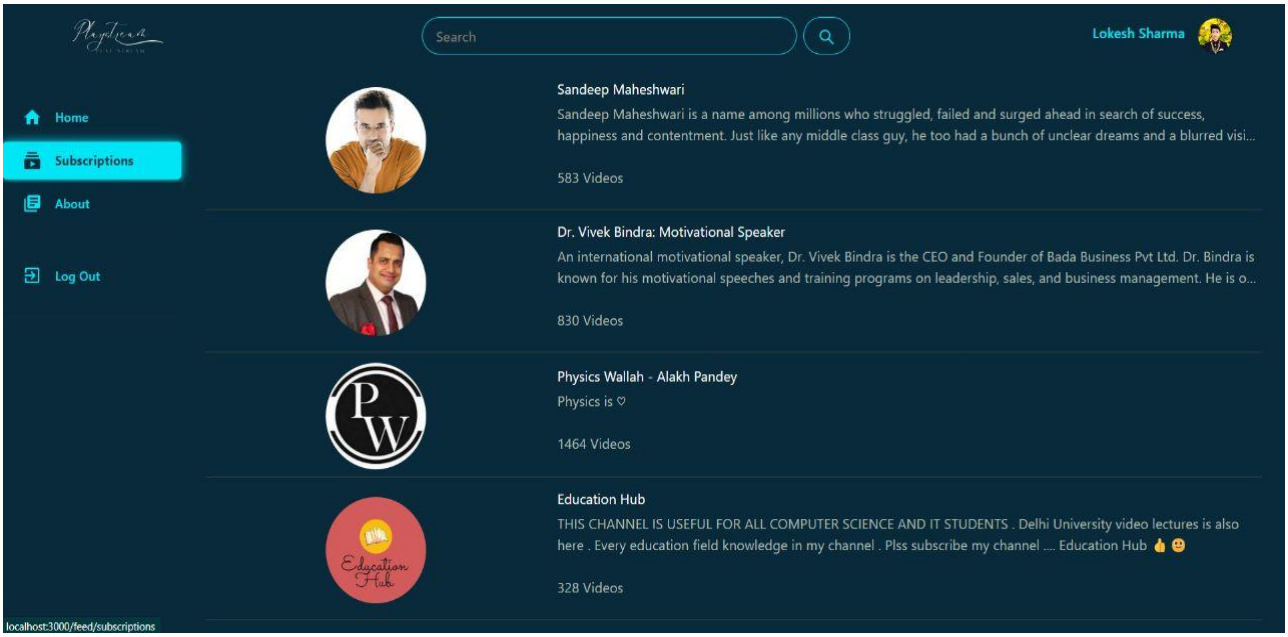
SCREENSHOTS :

Login Screen

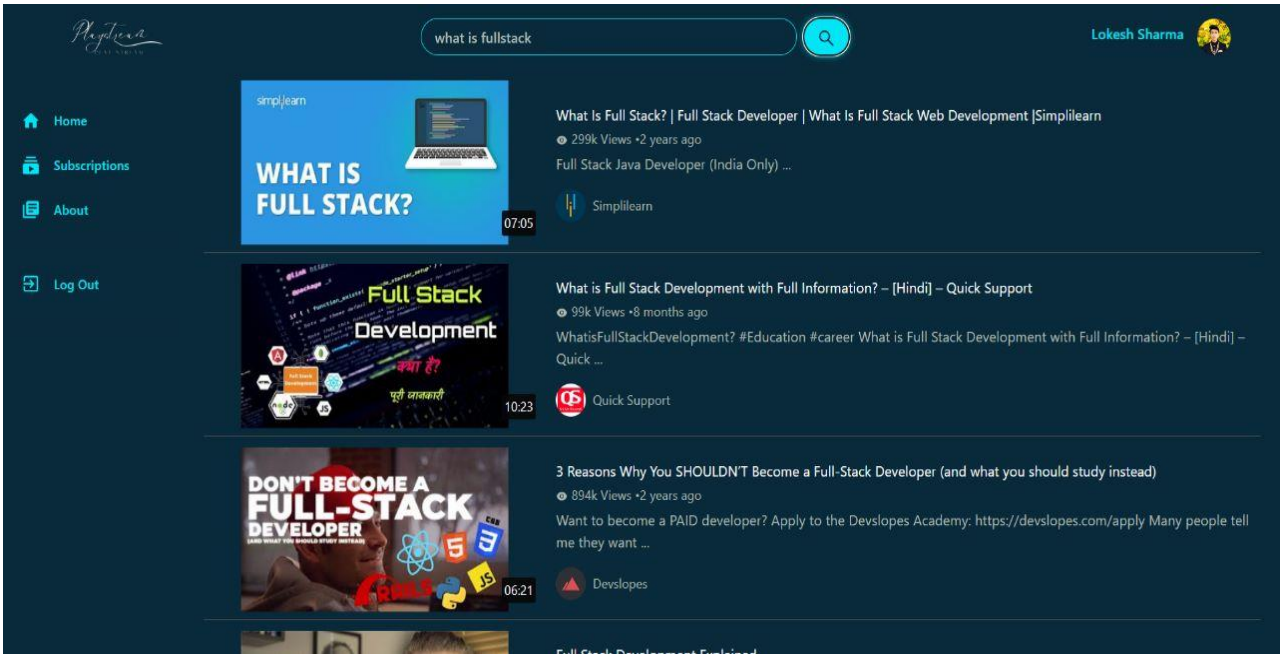


Home Screen

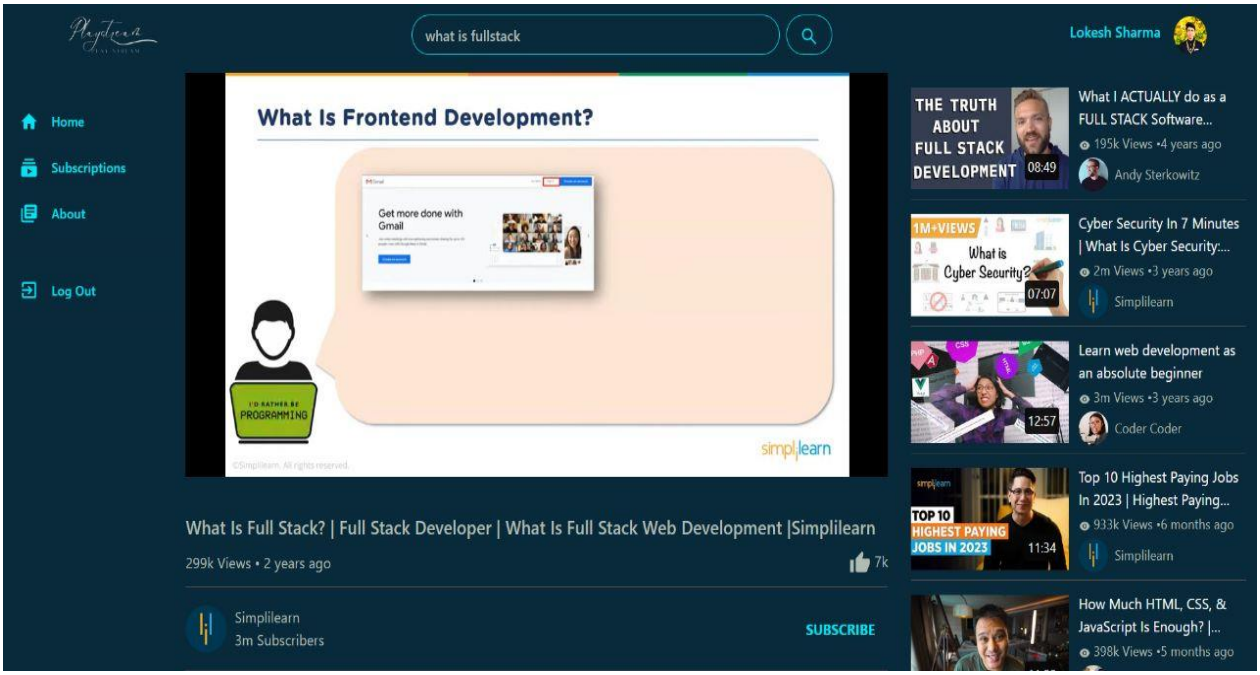
Subscription Screen



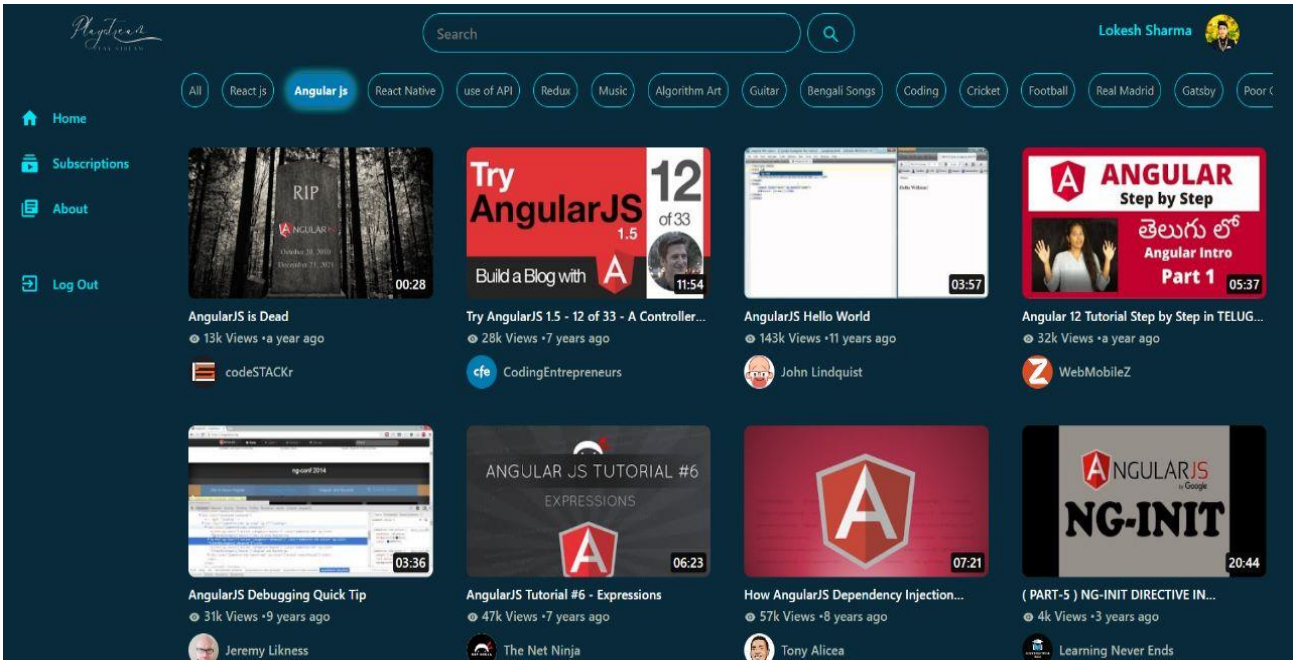
Search Bar



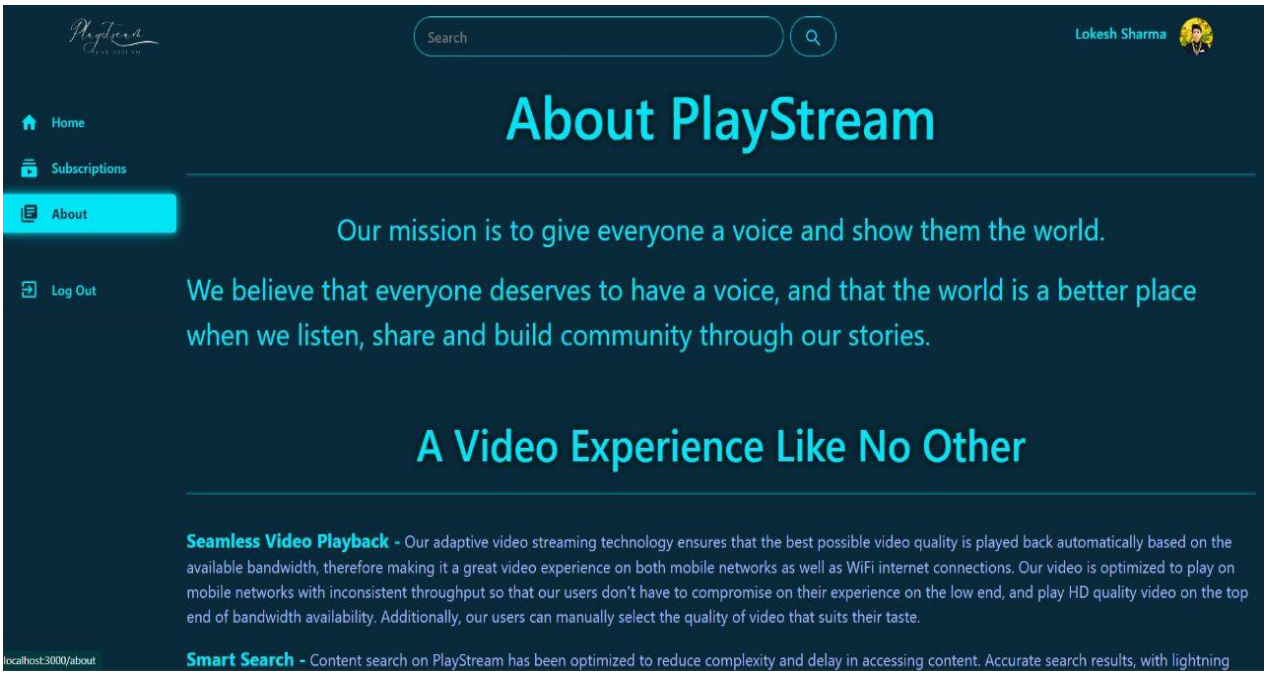
Selected Video Screen



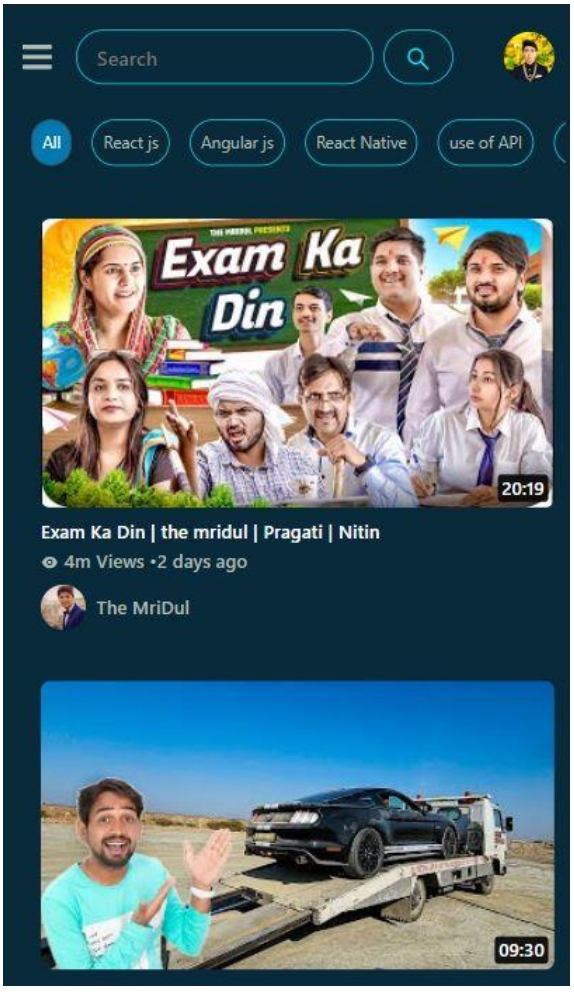
Categories Bar



About Screen



Responsiveness



CONCLUSION AND SCOPE

We aim to provide a platform for content creators to showcase their talent, share their stories, and connect with audiences globally. Video sharing platforms have also become a powerful tool for businesses, marketers, and educators to communicate their messages and engage with their target audiences. The rapid growth and popularity of video sharing platforms have transformed the media landscape, creating new opportunities and challenges for users, content creators, and platform operators alike.

The scope of video sharing platforms is vast and ever-evolving. With advancements in technology and changing user behaviors, the scope of video sharing platforms continues to expand. Some key areas within the scope of video sharing platforms include:

1. **Content creation**: Video sharing platforms enable users to create a wide variety of content, including vlogs, tutorials, reviews, entertainment, and more. The scope of content creation on video sharing platforms is vast and caters to diverse interests and niches.

2. **Content consumption**: Video sharing platforms allow users to consume videos on-demand, providing a vast library of content for users to watch, engage with, and share. The scope of content consumption includes entertainment, education, news, and information across various genres and languages.

3. **Social interaction**: Video sharing platforms offer social features that allow users to engage with content creators, other users, and communities. The scope of social interaction includes likes, comments, shares, collaborations, and live streaming, fostering online communities and interactions.

4. **Technology and innovation**: Video sharing platforms are constantly evolving, driven by advancements in technology and changing user expectations. The scope of technology and innovation includes features such as video quality, streaming, virtual reality, augmented reality, and other emerging technologies.

Resources :-

1. https://developers.google.com/youtube/v3/docs
2. https://developers.google.com/youtube/v3/guides/implementation
3. https://console.cloud.google.com/cloud-resource-manager

Github Link :-

https://github.com/Lokeshsharma1703/video-streaming-app