**YouTube Clone**

A Project Progress Report

Submitted in partial fulfillment for the degree of

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

P BUJJI (N190422)

B SRUTHI (N190897)

R UMADEVI (N191106)

B PRAVALLIKA (N190396)

B NAGARANI (N190201)

*Under the Esteem Guidance of*

**Mrs. M. J. Blessy**



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**Rajiv Gandhi University of Knowledge Technologies – Nuzvid**

**Nuzvid, Krishna, Andhra Pradesh – 521202.**



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**Rajiv Gandhi University of Knowledge Technologies – Nuzvid**

**Nuzvid, Krishna, Andhra Pradesh – 521202.**

**CERTIFICATE OF COMPLETION**

This is to certify that the work entitled, **“YOUTUBE CLONE”** is the bonafied work of **P. BUJJI (ID No*:* N190422), B.SRUTHI (ID No*:* N190897)*,* R.UMADEVI *(*ID No*:N191106*)*,* B.PRAVALLIKA *(*ID No*:*N190396)*,* B.NAGARANI *(*IDNo:N190201)** carried out under my guidance and supervision for 3rd year project of **Bachelor of Technology** in the department of Computer Science and Engineering under RGUKT IIIT Nuzvid. This work is done during the academic session MAY 2024– JUNE 2024, under our guidance.

---------------------------------------- -------------------------------------

**Mrs.M.J.BLESSY Mrs.Nagarajuna Devi**

Assistant professor, Head of the department,

Department of CSE, Department of CSE,

RGUKT Nuzvid RGUKT Nuzvid



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**Rajiv Gandhi University of Knowledge Technologies – Nuzvid**

**Nuzvid, Krishna, Andhra Pradesh – 521202.**

**CERTIFICATE OF EXAMINATION**

This is to certify that the work entitled, **“YOUTUBE CLONE”** is the bonafied work of **P. BUJJI (ID No*:* N190422), B.SRUTHI (ID No*:* N190897)*,* R.UMADEVI *(*ID No*:N191106*)*,* B.PRAVALLIKA *(*ID No*:*N190396)*,* B.NAGARANI *(*IDNo:N190201)** and here by accord our approval of it as a study carried out and presented in a manner required for its acceptance in 3rd year of **Bachelor of Technology** for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn, as a recorded in this thesis. It only signifies the acceptance of this thesis for the purpose for which it has been submitted.

------------------------------------------- ------------------------------

**Mrs.M.J.Blessy Project Examiner**

Assistant Professor, RGUKT-Nuzvid

Department of CSE,

RGUKT-NUZVID



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**Rajiv Gandhi University of Knowledge Technologies – Nuzvid**

**Nuzvid, Krishna, Andhra Pradesh – 521202.**

**DECLARATION**

**P. BUJJI (ID No*:* N190422), B.SRUTHI (ID No*:* N190897)*,* R.UMADEVI *(*ID No*:N191106*)*,* B.PRAVALLIKA *(*ID No*:*N190396)*,* B.NAGARANI *(*IDNo:N190201)** hereby declare that the project report entitled **“YOUTUBE CLONE”** done by us under the guidance of Mrs.M.J.Blessy, Assistant Professor is submitted for the partial fulfillment for the award of degree of Bachelor of Technology in Computer Science and Engineering during the academic session May 2024-June 2024 at RGUKT-Nuzvid.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

**Date: 20-06-2024**

**Place: Nuzvid**

P BUJJI (N190422)

B SRUTHI (N190897)

R UMADEVI (N191106)

B PRAVALLIKA (N190396)

B NAGARANI (N190201)

**ACKNOWLEDGEMENT**

We would like to express our profound gratitude and deep regards to our guide **Mrs. M. J. Blessy** for his exemplary guidance, monitoring and constant encouragement to us throughout the B.Tech course. We shall always cherish the time spent with her during the course of this work due to the invaluable knowledge gained in the field of reliability engineering.

We are extremely grateful for the confidence bestowed in us and entrusting our project entitled **“YOUTUBE CLONE”**

We express gratitude to **Mrs. Nagarajuna devi mam (HOD of CSE)** and other faculty members for being source of inspiration and constant encouragement which helped us in completing the project successfully.

Our sincere thanks to all the batch mates of 2019 CSE, who have made our stay at RGUKT-NUZVID, a memorable one.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved God and parents for their blessings, our friends for their help and wishes for the successful completion of this project.

# 

# **ABSTRACT**

The YouTube Clone Project aims to develop a web-based platform that replicates the core functionalities of YouTube. This platform allows users to search videos and view them. It also includes features for video categorization, search functionality, and personalized recommendations.

Key components of the project include:

**Video Playback:** A robust video player supporting various formats and resolutions, with controls for play, pause, seek, volume adjustment, and fullscreen mode.

**Search and Discovery:** Advanced search capabilities with filters for video categories. Personalized recommendations based on user behavior and preferences.

**Responsive Design:** A user interface that adapts seamlessly to different devices and screen sizes, ensuring a consistent experience across desktops, tablets, and smartphones.

**Scalability and Performance:** Efficient handling of large volumes of data and concurrent users, with backend optimizations for performance and scalability.

The project utilizes modern web development technologies, including HTML, JavaScript, Material UI, React JS, React Player and rapidAPI Youtube v3.

Through this project, we aim to deliver a user-friendly and efficient front end design and React Router DOM for smooth navigation. With the help of the RapidAPI YouTube v3, we've integrated real-time data fetching from YouTube's vast library of videos, allowing users to seamlessly browse and watch their favorite content.

**Table of Contents**

**ABSTRACT** 6

**CHAPTER 1** 8

**INTRODUCTION** 8

**1.1 React JS**

**1.2 Material UI**

**1.3 React Router DOM**

**1.4 RapidAPI Youtube v3**

**1.5 React Player**

**CHAPTER 2** 9

**REQUIREMENTS AND ANALYSIS** 9

**2.1 Hardware components:** 9

**2.2 Software components:** 10

**2.3 Functional requirements** 10

**2.4 Non-Functional requirements** 10

**CHAPTER 3** 11

**3.1 Advantages and disadvantages** 11

3.1.1 Advantages 11

3.1.2 Disadvantages 11

**3.2 Applications** 12

**CHAPTER 4** 12

**IMPLEMENTATION** 12

**OUTPUT** 34

**CHAPTER 5** 32

**CONCLUSION** 35

**REFERENCES** 36­­­­

# **CHAPTER 1**

# **INTRODUCTION**

The YouTube Clone Project leverages modern web development technologies to achieve these goals. We utilize HTML, JavaScript, and Material UI for building a dynamic and intuitive front-end interface. React.js and React Player are employed for creating interactive components and video playback functionality, while React Router DOM facilitates smooth navigation within the platform.

Central to our implementation is the integration with RapidAPI's YouTube v3, which allows real-time data fetching from YouTube's extensive video library. This integration provides users with access to a vast array of videos, enhancing the overall user experience by enabling seamless browsing and watching of their favorite content.

Through this project, we aim to deliver an efficient and engaging video-sharing platform that showcases the capabilities of modern web technologies. Our focus on user experience, performance, and scalability ensures that the YouTube Clone Project not only replicates YouTube's core functionalities but also provides a robust foundation for future enhancements and innovations.

**React.js:**

React.js is a popular JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components, making it easier to manage complex applications. React's virtual DOM improves performance by updating only the parts of the page that have changed, rather than reloading the entire page. This efficiency, along with its component-based architecture, makes React.js a preferred choice for developing dynamic and responsive web applications.

**Material-UI:**

Material-UI is a comprehensive library of React components that implement Google's Material Design guidelines. It provides a wide range of pre-designed and customizable UI components, such as buttons, sliders, and grids, which help developers create visually appealing and consistent user interfaces. Material-UI enhances the development process by offering components that are both flexible and easy to integrate, ensuring a polished and professional look for the application.

**React Router DOM:**

React Router DOM is a routing library for React applications, enabling navigation between different components and views without reloading the page. It manages the application's URL and renders the corresponding components, allowing for smooth transitions and deep linking. React Router DOM supports nested routes, dynamic route matching, and easy access to route parameters, making it a powerful tool for building single-page applications with complex navigation requirements.

**RapidAPI YouTube v3:**

RapidAPI YouTube v3 is an API service that provides access to YouTube's extensive video library. It allows developers to fetch real-time data from YouTube, including video details, search results, and user information. By integrating RapidAPI YouTube v3 into the application, developers can offer users a rich video browsing experience, leveraging YouTube's vast content database. This API facilitates seamless data retrieval, ensuring that users can search for and watch videos effortlessly.

**React Player:**

React Player is a flexible media player component for React applications, supporting various media formats including YouTube, Vimeo, SoundCloud, and more. It provides a simple API for embedding and controlling media players within a React application. With features like play, pause, seek, volume control, and fullscreen mode, React Player ensures a smooth and customizable media playback experience. Its ease of integration and support for multiple media sources make it an ideal choice for building rich multimedia applications.

# **CHAPTER 2**

# **REQUIREMENTS AND ANALYSIS**

## **2.1 Hardware components:**

* Processor: 64-bit, quad-core, 2.5 GHz minimum per core
* RAM: 4 GB or more.
* HDD: 20 GB of available space or more.
* Display: Dual XGA (1024 x 768) or higher resolution monitors.
* Keyboard: A standard keyboard

## **2.2 Software components:**

* **Frontend Components**
  + HTML & CSS: Structure and styling of the web application.
  + JavaScript: Client-side scripting to create dynamic and interactive elements.
  + React.js: Framework for building the user interface.
  + Material-UI: Component library for implementing Material Design.
  + React Router DOM: Routing library for navigation between different views.
  + React Player: Media player component for video playback.
* **API Integration**
  + RapidAPI YouTube v3: API integration for fetching real-time data from YouTube.
* **Deployment**
  + Hosting Service: Netlify

## **2.3 Functional requirements**

## **Video Playback**

## A video player with controls for play, pause, seek, volume adjustment, and fullscreen mode.

## **Search and Discovery**

## Search functionality with advanced filtering options.Personalized recommendations based on user behavior and preferences.

## **Responsive Design**

## User interface adapts to various devices and screen sizes for a consistent experience.

## **2.4 Non-Functional requirements**

* Performance
* Scalability
* Maintainability
* Usability

# **CHAPTER 3**

## **3.1 Advantages and disadvantages**

### 3.1.1 Advantages

1. **User Experience:**
   * 1. Intuitive Interface: Leveraging React.js and Material-UI ensures a smooth, responsive, and aesthetically pleasing user interface, improving user satisfaction and engagement.
     2. Personalized Recommendations: Enhanced user experience through tailored video suggestions based on individual user behavior and preferences.
2. **Scalability and Performance:**
   * 1. Efficient Load Handling: The backend architecture is designed for scalability, allowing the system to handle large volumes of data and concurrent users efficiently.
     2. Fast Load Times: Optimized performance ensures quick load times and responsive interactions, essential for user retention.
3. **Real-Time Data Integration:**
   * 1. RapidAPI YouTube v3: Integration with YouTube’s extensive video library allows for real-time data fetching, ensuring users have access to the latest content.
4. **Cross-Platform Compatibility:**
   * 1. Responsive Design: The application adapts seamlessly to various devices and screen sizes, providing a consistent user experience across desktops, tablets, and smartphones.

### 3.1.2 Disadvantages

1. **Development Complexity:**
   * 1. Learning Curve: Technologies like React.js, Material-UI, and API integrations require a certain level of expertise, which can be challenging for new developers.
     2. Integration Challenges: Integrating multiple technologies (e.g., RapidAPI YouTube v3, cloud storage) can be complex and may require careful coordination and troubleshooting.
2. **Dependency on External APIs:**
   * 1. Reliance on YouTube API: The functionality and availability of the platform heavily depend on the YouTube API. Any changes or downtime in the API service could affect the platform’s performance and user experience.
     2. API Costs: Usage of external APIs like RapidAPI YouTube v3 may incur additional costs, particularly at higher usage levels.

## **3.2 Applications**

* Online Courses
* Skill Development
* Training Programs
* Entertainment and Media
* Remote Learning
* Continuing Education

# **CHAPTER 4**

# **IMPLEMENTATION**

**4.1 Building Model**

**App.Js:**

import { BrowserRouter, Routes, Route } from "react-router-dom"

import { Box } from "@mui/material"

import {ChannelDetail, Feed, Navbar, SearchFeed, VideoDetail} from './components'

const App = () => (

<BrowserRouter>

<Box sx={{backgroundColor: "#000"}}>

<Navbar/>

<Routes>

<Route exact path="/" element={<Feed/>}/>

<Route path="/video/:id" element={<VideoDetail/>}/>

<Route path="/channel/:id" element={<ChannelDetail/>}/>

<Route path="/search/:searchTerm" element={<SearchFeed/>}/>

</Routes>

</Box>

</BrowserRouter>

)

export default App

**Feed.jsx :**

import { Box, Stack, Typography } from "@mui/material";

import { useState, useEffect } from "react";

import Sidebar from "./Sidebar";

import Videos from './Videos';

import { fetchFromAPI } from "../utils/fetchFromAPI";

const Feed = () => {

const [selectedCategory, setSelectedCategory] = useState("New");

const [videos,setVideos]= useState([])

useEffect(() => {

fetchFromAPI(search?part=snippet&q=${selectedCategory})

.then((data)=>{

setVideos(data.items)

});

}, [selectedCategory]);

return (

<Stack sx={{ flexDirection: { sm: "column", md: "row" } }}>

<Box

sx={{

height: { sm: "auto", md: "92vh" },

borderRight: "1px solid #3d3d3d",

px: { sm: 0, md: 2 },

}}

>

<Sidebar

selectedCategory={selectedCategory}

setSelectedCategory={setSelectedCategory}

/>

<Typography

className="copyright"

variant="body2"

sx={{

mt: 1.5,

color: "#fff",

}}

>

Copyright 2022 JSM media

</Typography>

</Box>

<Box p={2} sx={{ overflowY: "auto", height: "90vh", flex: 2 }}>

<Typography

variant="h4"

fontWeight="bold"

mb={2}

sx={{ color: "white" }}

>

{selectedCategory}

<span style={{ color: "#F31503" }}> videos</span>

</Typography>

<Videos videos={videos} />

</Box>

</Stack>

);

};

export default Feed;

## **Sidebar.jsx :**

import { Stack } from "@mui/material";

import { categories } from "../utils/constants";

const Sidebar = ({selectedCategory, setSelectedCategory}) => (

<Stack

direction="row"

sx={{

overflowY: "auto",

height: { sm: "auto", md: "95%" },

flexDirection: { md: "column" },

}}

>

{categories.map((category) => (

<button

className="category-btn"

style={{

background: category.name === selectedCategory && "#FC1503",

color: "white",

}}

onClick={()=>setSelectedCategory(category.name)}

key={category.name}

>

<span

style={{

color: category.name === selectedCategory ? "white" : "red",

marginRight: "15px",

}}

>

{category.icon}

</span>

<span

style={{ opacity: category.name === selectedCategory ? "1" : "0.8" }}

>

{category.name}

</span>

</button>

))}

</Stack>

);

export default Sidebar;

**videos.jsx:**

import { Stack, Box } from "@mui/material";

import { VideoCard, ChannelCard } from "./";

const Videos = ({ videos, direction }) => {

if (!videos?.length) return "Loading..."

return (

<Stack direction={direction ||"row"} flexWrap="wrap" justifyContent="start" gap={2}>

{videos.map((item, index) => (

<Box key={index}>

{item.id.videoId && <VideoCard video={item} />}

{item.id.channelId && <ChannelCard channelDetail={item} />}

</Box>

))}

</Stack>

);

};

export default Videos;

**videoCard.jsx :**

import { Link } from "react-router-dom";

import { Typography, Card, CardContent, CardMedia } from "@mui/material";

import { CheckCircle } from "@mui/icons-material";

import {

demoVideoUrl,

demoVideoTitle,

demoChannelTitle,

demoChannelUrl,

} from "../utils/constants";

const VideoCard = ({

video: {

id: { videoId },

snippet,

},

}) => {

console.log(snippet, videoId);

return (

<Card

sx={{

width: { xs: "100%",md: "320px", sm:"358px" },

boxShadow: "none",

borderRadius: 0,

}}

>

<Link to={videoId ? /video/${videoId} : demoVideoUrl}>

<CardMedia

image={snippet?.thumbnails?.high?.url}

alt={snippet?.title}

sx={{ width: {xs:"100%", sm:"358px", md:"320px"}, height: 180 }}

/>

</Link>

<CardContent sx={{ backgroundColor: "#1e1e1e", height: "106px" }}>

<Link to={videoId ? /video/${videoId} : demoVideoUrl}>

<Typography variant="subtitle1" fontWeight="bold" color="#FFF">

{snippet?.title.slice(0, 60) || demoVideoTitle.slice(0, 60)}

</Typography>

</Link>

<Link

to={

snippet?.channelId

? /channel/${snippet?.channelId}

: demoChannelUrl

}

>

<Typography variant="subtitle2" fontWeight="bold" color="gray">

{snippet?.channelTitle.slice(0, 60) ||

demoChannelTitle.slice(0, 60)}

<CheckCircle sx={{ fontSize: 12, color: "gray", ml: "5px" }} />

</Typography>

</Link>

</CardContent>

</Card>

);

};

export default VideoCard;

**VideoDetails.jsx:**

import {useState, useEffect} from 'react'

import {Link, useParams } from 'react-router-dom'

import ReactPlayer from 'react-player'

import { Typography, Box, Stack } from '@mui/material'

import { CheckCircle } from '@mui/icons-material'

import {Videos} from './'

import { fetchFromAPI } from '../utils/fetchFromAPI'

const VideoDetail = () => {

const [videoDetail, setVideoDetail]= useState(null);

const [videos, setVideos] = useState(null);

const {id}= useParams()

useEffect(()=>{

fetchFromAPI(videos?part=snippet,statistics&id=${id})

.then((data)=>setVideoDetail(data.items[0]))

fetchFromAPI(search?part=snippet&relatedToVidoeId=${id}&type=video)

.then((data)=>setVideos(data.items))

}, [id])

if (!videoDetail?.snippet) return "Loading...."

const {snippet:{title, channelId, channelTitle}, statistics: {viewCount, likeCount}}= videoDetail;

return (

<Box minHeight="95vh">

<Stack direction={{xs: "column", md:"row"}}>

<Box flex={1}>

<Box sx={{width:"100%", position:"sticky", top:"86px"}}>

<ReactPlayer url={https://www.youtube.com/watch?v=${id}}

className="react-player" controls

/>

<Typography color="#fff" variant='h5' fontWeight="bold" p={2}>

{title}

</Typography>

<Stack direction="row" justifyContent="space-between" sx={{color: "#fff"}} py={1} px={2}>

<Link to={/channel/${channelId}}>

<Typography variant={{sm:"subTitle1", md:"h6"}} color="#Fff">{channelTitle}

<CheckCircle sx={{fontSize:"12px", color:"gray", ml:"5px"}}/>

</Typography>

</Link>

<Stack direction="row" gap="20px">

<Typography variant='body1'

sx={{opacity: 0.7}}>

{parseInt(viewCount).toLocaleString()} views

</Typography>

<Typography variant='body1'

sx={{opacity: 0.7}}>

{parseInt(likeCount).toLocaleString()} likes

</Typography>

</Stack>

</Stack>

</Box>

</Box>

<Box px={2} py={{md:1, xs:5}} justifyContent="center" alignItems="center">

<Videos videos={videos} direction="column"/>

</Box>

</Stack>

</Box>

)

}

export default VideoDetail

**ChannelCard.jsx :**

import { Box, Typography, CardContent, CardMedia } from "@mui/material";

import { CheckCircle } from "@mui/icons-material";

import { Link } from "react-router-dom";

import { demoProfilePicture } from "../utils/constants";

const ChannelCard = ({ channelDetail, marginTop }) => {

return (

<Box

sx={{

boxShadow: "none",

borderRadius: "20px",

display: "flex",

justifyContent: "center",

alignItems: "center",

width: { xs: "356px", md: "320px" },

height:"326px",

margin:"auto",

marginTop:{marginTop}

}}

>

<Link to={/channel/${channelDetail?.id?.channelId}}>

<CardContent

sx={{

display: "flex",

flexDirection: "column",

justifyContent: "center",

textAlign: "center",

color: "#fff",

}}

>

<CardMedia

image={

channelDetail?.snippet?.thumbnails?.high?.url ||

demoProfilePicture

}

alt={channelDetail?.snippet?.title}

sx={{

borderRadius: "50%",

height: "180px",

width: "180px",

mb: 2,

border: "1px solid #e3e3e3",

}}

/>

<Typography variant="h6">

{channelDetail?.snippet?.title}

<CheckCircle sx={{ fontSize: 12, color: "gray", ml: "5px" }} />

</Typography>

{channelDetail?.statistics?.subscriberCount && (

<Typography>

{parseInt(

channelDetail?.statistics?.subscriberCount

).toLocaleString()}{" "}

Subscribers

</Typography>

)}

</CardContent>

</Link>

</Box>

);

};

export default ChannelCard;

**ChannelDetails.jsx:**

import { Box } from "@mui/material";

import { useState, useEffect } from "react";

import { useParams } from "react-router-dom";

import { Videos, ChannelCard } from "./";

import { fetchFromAPI } from "../utils/fetchFromAPI";

const ChannelDetail = () => {

const [channelDetail, setChannelDetail] = useState(null);

const [videos, setVideos] = useState([]);

const { id } = useParams();

console.log(channelDetail, videos);

useEffect(() => {

fetchFromAPI(channels?part=snippet&id=${id}).then((data) =>

setChannelDetail(data?.items[0])

);

fetchFromAPI(search?channelId=${id}&part=snippet&order=date).then(

(data) => setVideos(data?.items)

);

}, [id]);

return (

<Box minHeight="95vh">

<Box>

<div

style={{

background:

"linear-gradient(90deg, rgba(0, 238, 247, 1) 0%, rgba(206, 3, 184, 1) 100%, rgba(0, 212, 255,1) 100%)",

zIndex: 10,

height: "300px",

}}

/>

<ChannelCard channelDetail={channelDetail} marginTop="-110px" />

</Box>

<Box display="flex" p="2">

<Box sx={{ mr: { sm: "100px" } }} />

<Videos videos={videos} />

</Box>

</Box>

);

};

export default ChannelDetail;

**Navbar.jsx:**

import { Stack } from "@mui/material";

import { Link } from "react-router-dom";

import { logo } from "../utils/constants";

import SearchBar from "./SearchBar";

const Navbar = () => (

<Stack

direction="row"

alignItems="center"

p={2}

sx={{

position: "sticky",

background: "#000",

top: 0,

justifyContent: "space-between",

}}

>

<Link to="/" style={{ display: "flex", alignItems: "center" }}>

<img src={logo} alt="logo" height={45} />

</Link>

<SearchBar />

</Stack>

);

export default Navbar;

**Searchbar.jsx:**

import { Stack } from "@mui/material";

import { Link } from "react-router-dom";

import { logo } from "../utils/constants";

import SearchBar from "./SearchBar";

const Navbar = () => (

<Stack

direction="row"

alignItems="center"

p={2}

sx={{

position: "sticky",

background: "#000",

top: 0,

justifyContent: "space-between",

}}

>

<Link to="/" style={{ display: "flex", alignItems: "center" }}>

<img src={logo} alt="logo" height={45} />

</Link>

<SearchBar />

</Stack>

);

export default Navbar;

**Searchbar.jsx:**

import { useState } from "react";

import { useNavigate } from "react-router-dom";

import { Paper, IconButton } from "@mui/material";

import { Search } from "@mui/icons-material";

const SearchBar = () => {

const [searchTerm, setsearchTerm] = useState("");

const navigate=useNavigate()

const handleSubmit=(e)=>{

e.preventDefault();

if(searchTerm){

navigate(/search/${searchTerm})

}

}

return(

<Paper

component="form"

sx={{

borderRadius: 20,

border: "1px solid #e3e3e3",

pl: 2,

boxShadow: "none",

mr: { sm: 5 },

}}

onSubmit={handleSubmit}

>

<input

className="search-bar"

placeholder="search..."

value={searchTerm}

onChange={(e) => {setsearchTerm(e.target.value)}}

/>

<IconButton type="submit" sx={{p:"10px", color:"red"}}>

<Search/>

</IconButton>

</Paper>

);

}

export default SearchBar;

**SearchFeed.jsx:**

import { Box, Typography } from "@mui/material";

import { useState, useEffect } from "react";

import Videos from './Videos'

import { useParams } from "react-router-dom";

import { fetchFromAPI } from "../utils/fetchFromAPI";

const SearchFeed = () => {

const [videos,setVideos]= useState([])

const {searchTerm}= useParams()

useEffect(() => {

fetchFromAPI(search?part=snippet&q=${searchTerm})

.then((data)=>{

setVideos(data.items)

});

}, [searchTerm]);

return (

<Box p={2} sx={{ overflowY: "auto", height: "90vh", flex: 2 }}>

<Typography

variant="h4"

fontWeight="bold"

mb={2}

sx={{ color: "white" }}

>

Search Results for:

<span style={{ color: "#F31503" }}> {searchTerm}</span> videos

</Typography>

<Videos videos={videos} />

</Box>

);

};

export default SearchFeed;

**Constants.jsx:**

import MusicNoteIcon from '@mui/icons-material/MusicNote';

import HomeIcon from '@mui/icons-material/Home';

import CodeIcon from '@mui/icons-material/Code';

import OndemandVideoIcon from '@mui/icons-material/OndemandVideo';

import SportsEsportsIcon from '@mui/icons-material/SportsEsports';

import LiveTvIcon from '@mui/icons-material/LiveTv';

import SchoolIcon from '@mui/icons-material/School';

import FaceRetouchingNaturalIcon from '@mui/icons-material/FaceRetouchingNatural';

import CheckroomIcon from '@mui/icons-material/Checkroom';

import GraphicEqIcon from '@mui/icons-material/GraphicEq';

import TheaterComedyIcon from '@mui/icons-material/TheaterComedy';

import FitnessCenterIcon from '@mui/icons-material/FitnessCenter';

import DeveloperModeIcon from '@mui/icons-material/DeveloperMode';

export const logo = 'https://i.ibb.co/s9Qys2j/logo.png';

export const categories = [

{ name: 'New', icon: <HomeIcon />, },

{ name: 'JS Mastery', icon: <CodeIcon />, },

{ name: 'Coding', icon: <CodeIcon />, },

{ name: 'ReactJS', icon: <CodeIcon />, },

{ name: 'NextJS', icon: <CodeIcon />, },

{ name: 'Music', icon: <MusicNoteIcon /> },

{ name: 'Education', icon: <SchoolIcon />, },

{ name: 'Podcast', icon: <GraphicEqIcon />, },

{ name: 'Movie', icon: <OndemandVideoIcon />, },

{ name: 'Gaming', icon: <SportsEsportsIcon />, },

{ name: 'Live', icon: <LiveTvIcon />, },

{ name: 'Sport', icon: <FitnessCenterIcon />, },

{ name: 'Fashion', icon: <CheckroomIcon />, },

{ name: 'Beauty', icon: <FaceRetouchingNaturalIcon />, },

{ name: 'Comedy', icon: <TheaterComedyIcon />, },

{ name: 'Gym', icon: <FitnessCenterIcon />, },

{ name: 'Crypto', icon: <DeveloperModeIcon />, },

];

export const demoThumbnailUrl = 'https://i.ibb.co/G2L2Gwp/API-Course.png';

export const demoChannelUrl = '/channel/UCmXmlB4-HJytD7wek0Uo97A';

export const demoVideoUrl = '/video/GDa8kZLNhJ4';

export const demoChannelTitle = 'JavaScript Mastery';

export const demoVideoTitle = 'Build and Deploy 5 JavaScript & React API Projects in 10 Hours - Full Course | RapidAPI';

export const demoProfilePicture = 'http://dergipark.org.tr/assets/app/images/buddy\_sample.png'

**fetchFromAPI.js:**

import axios from "axios";

const BASE\_URL='https://youtube-v31.p.rapidapi.com'

const options = {

params: {

maxResults: '50',

},

headers: {

'X-RapidAPI-Key': '030b297dc5msh29e5e624a98a481p175343jsn643c69a00675',

'X-RapidAPI-Host': 'youtube-v31.p.rapidapi.com'

}

}

export const fetchFromAPI= async(url) => {

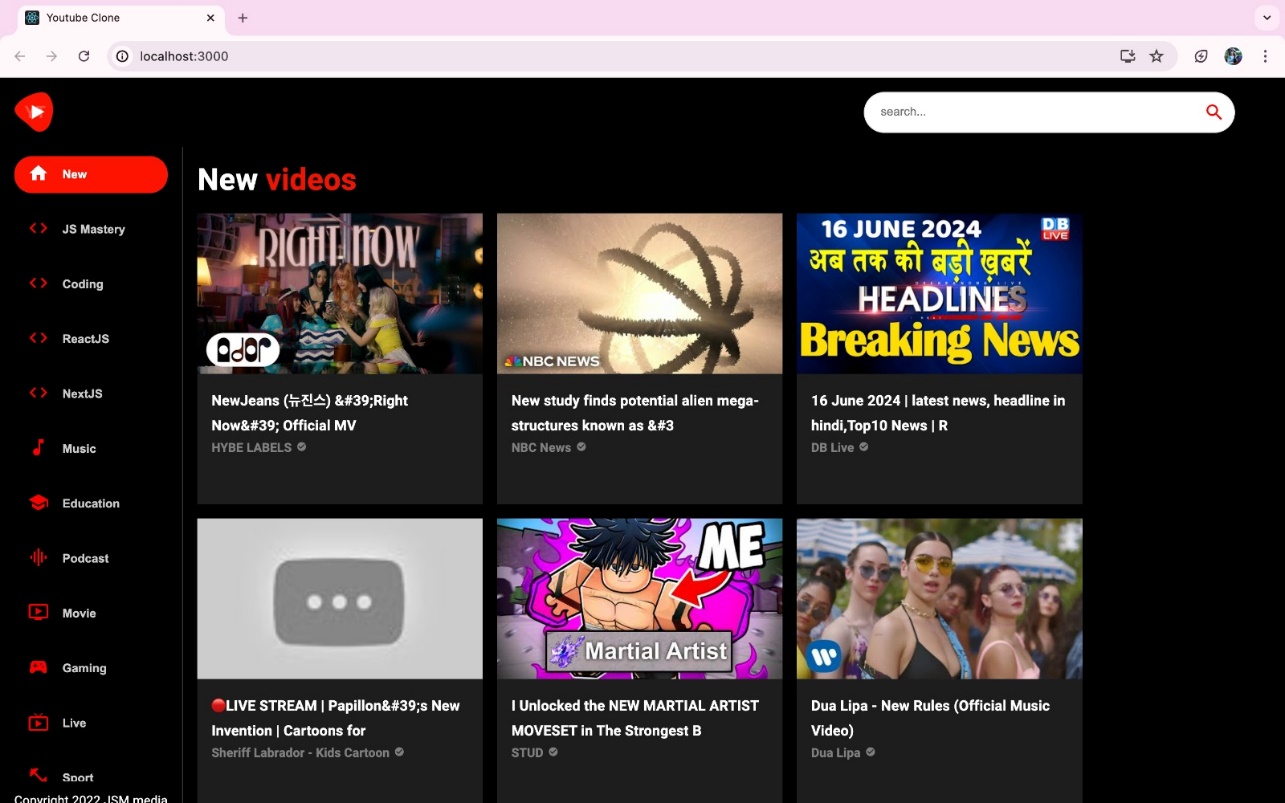
const {data }= await axios.get(${BASE\_URL}/${url}, options);

return data;

}

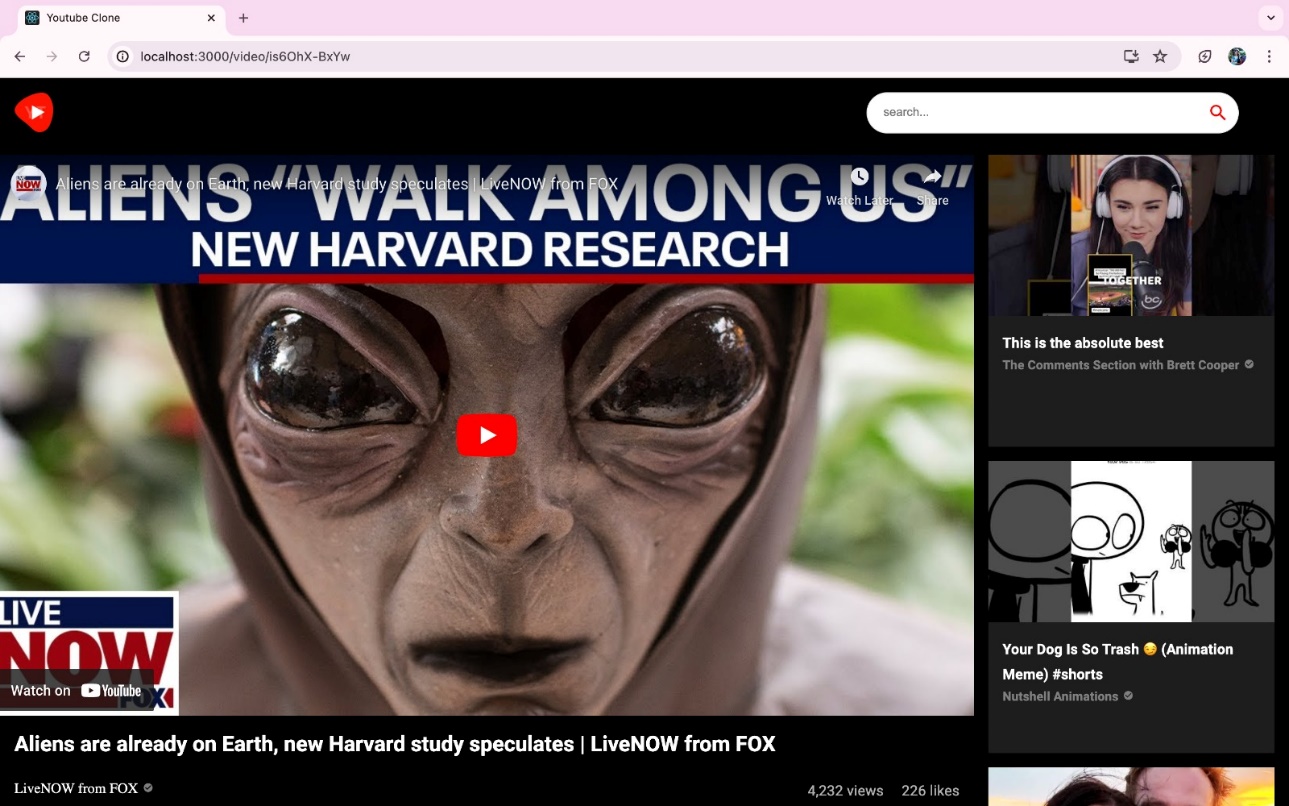
# 

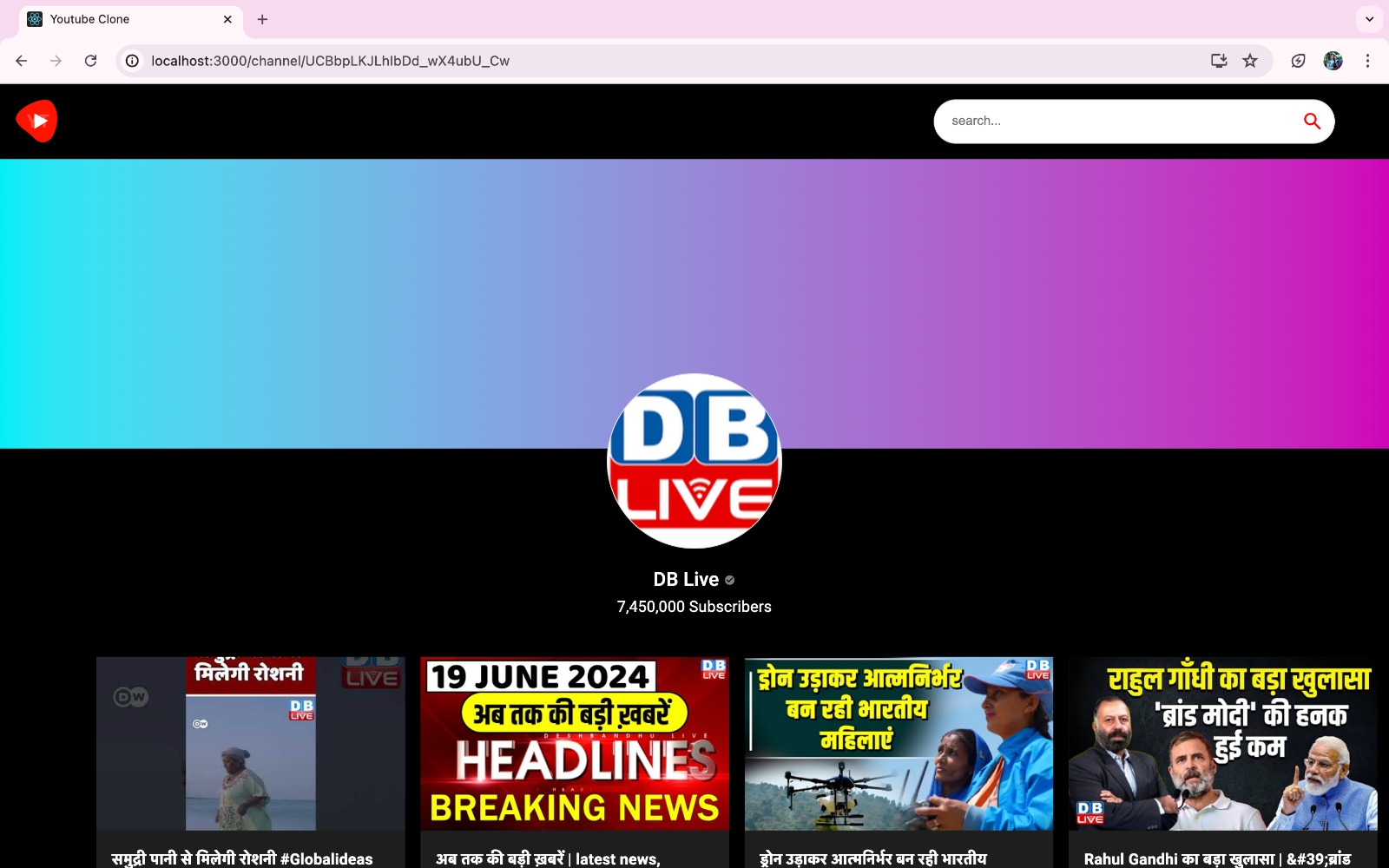
# **OUTPUT**



(d)

(c)





(b)

(a)

# **CHAPTER 5**

# **CONCLUSION**

The YouTube Clone Project represents a comprehensive effort to replicate and enhance the core functionalities of YouTube through modern web development technologies. By leveraging HTML, JavaScript, Material UI, React.js, and the RapidAPI YouTube v3, the project aims to deliver a robust platform with a user-friendly front end design and seamless navigation using React Router DOM.

Key components such as the robust video player supporting various formats and resolutions, advanced search capabilities with personalized recommendations, and responsive design ensure a consistent and engaging user experience across different devices. The emphasis on scalability and performance optimizations ensures efficient handling of large datasets and concurrent users, making the platform capable of meeting growing demands.

Ultimately, the project strives to offer users a familiar yet enhanced experience akin to YouTube, enabling seamless video playback, efficient video discovery through advanced search and recommendation features, all while maintaining high standards of performance and scalability.

# **REFERENCES**

1. Youtube
2. Github