Subject Name: **Front-End Engineering**

Subject Code: **CS186**

Cluster: **iGamma** Group:**19**

Department: **DCSE**

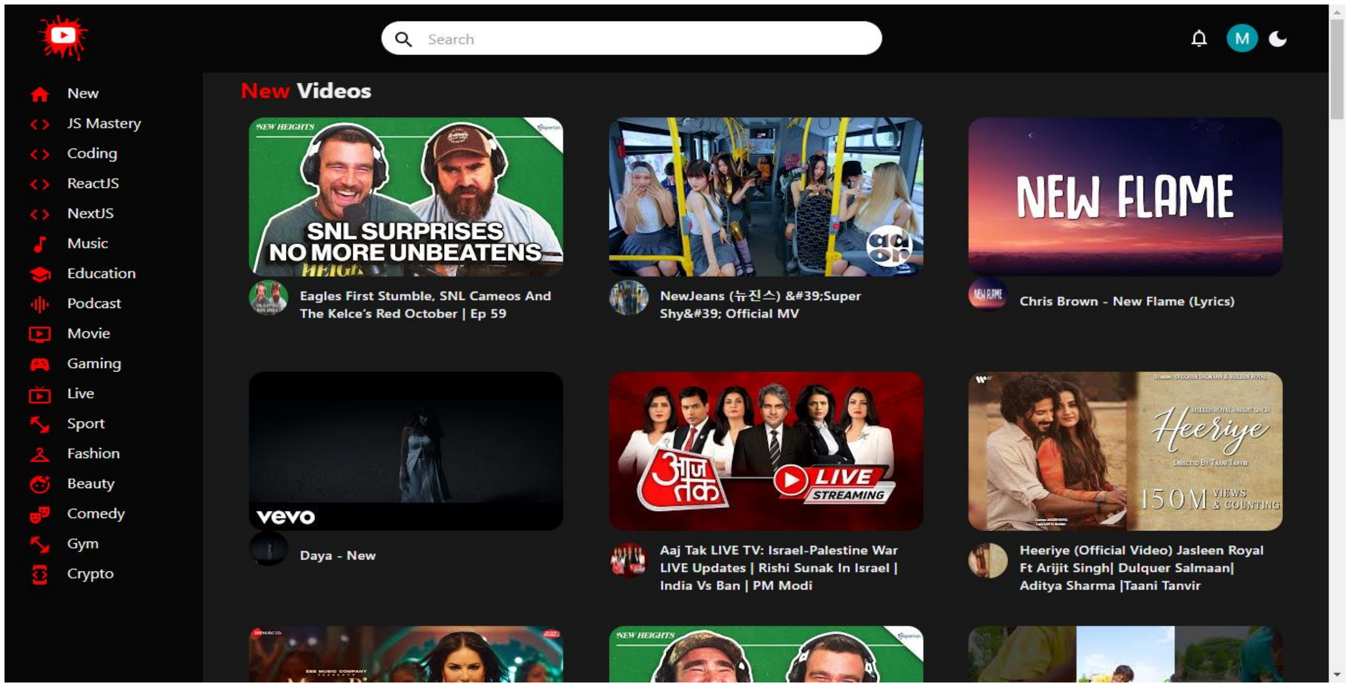
CHITKARA UNIVERSITY

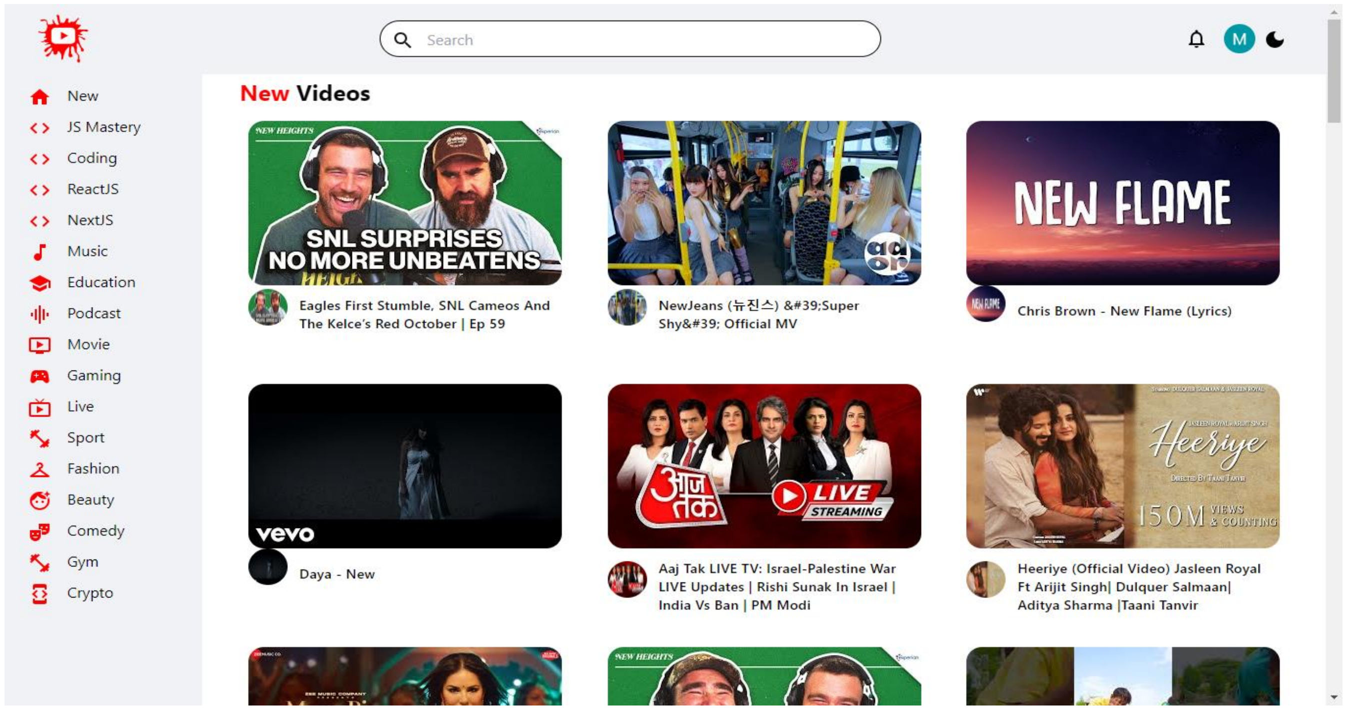
Submitted By:

Mayank Kumar Gupta

2110991946

PROECT: YT-CLONE


Search
New Videos
New
JS Mastery
Coding
ReactJS
NextJS
Music
Education
Podcast
Movie
Gaming
Live
Sport
Fashion
Beauty
Comedy
Gym
Crypto
M
Eagles First Stumble, SNL Cameos And The Kelce's Red October | Ep 59
NewJeans (뉴진스) &#39;Super Shy&#39; Official MV
Chris Brown - New Flame (Lyrics)
Daya - New
Aaj Tak LIVE TV: Israel-Palestine War LIVE Updates | Rishi Sunak In Israel | India Vs Ban | PM Modi
Heeriye (Official Video) Jasleen Royal Ft Arijit Singh| Dulquer Salmaan| Aditya Sharma |Taani Tanvir


Search
New Videos
New
JS Mastery
Coding
ReactJS
NextJS
Music
Education
Podcast
Movie
Gaming
Live
Sport
Fashion
Beauty
Comedy
Gym
Crypto
M
Eagles First Stumble, SNL Cameos And The Kelce's Red October | Ep 59
NewJeans (뉴진스) &#39;Super Shy&#39; Official MV
Chris Brown - New Flame (Lyrics)
Daya - New
Aaj Tak LIVE TV: Israel-Palestine War LIVE Updates | Rishi Sunak In Israel | India Vs Ban | PM Modi
Heeriye (Official Video) Jasleen Royal Ft Arijit Singh| Dulquer Salmaan| Aditya Sharma |Taani Tanvir

About The Project

Within this project, you'll find a comprehensive YouTube clone developed in React, featuring a rich set of functionalities. The application harnesses the power of the Context API for efficient state management, ensuring a seamless and responsive user experience. It utilizes the Fetch API to seamlessly retrieve video data from the server, empowering users to not only watch videos but also perform detailed searches, enriching their interaction with the platform. This YouTube clone is designed with the end user in mind, offering both dark and light mode themes to cater to individual preferences and enhance overall usability. The user interface is both visually appealing and fully responsive, meticulously crafted with the styling capabilities of Tailwind CSS. The outcome is a web application that not only mirrors the functionality of the popular video-sharing platform but also reflects a commitment to an enhanced user experience and aesthetic appeal.

Github Link : https://github.com/mkgupta01/yt-clone

Live Link: https://mkgupta.tech/yt-clone

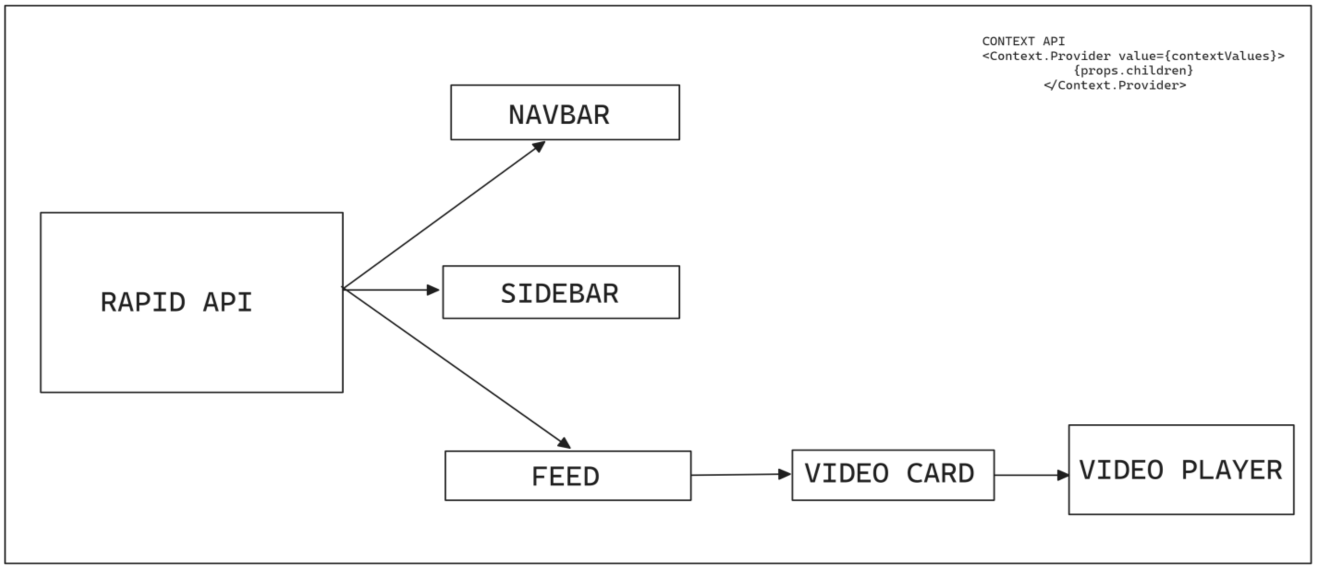
Tech Stack

- React JS
- Tailwind CSS
- Rapid API
- Context API
- React Video Player

Features

- Fully Functional
- Light/Dark mode
- No Adds
- Good UI

FLOWCHART



```
CONTEXT API
<Context.Provider value={contextValues}>
           {props.children}
        </Context.Provider>
```

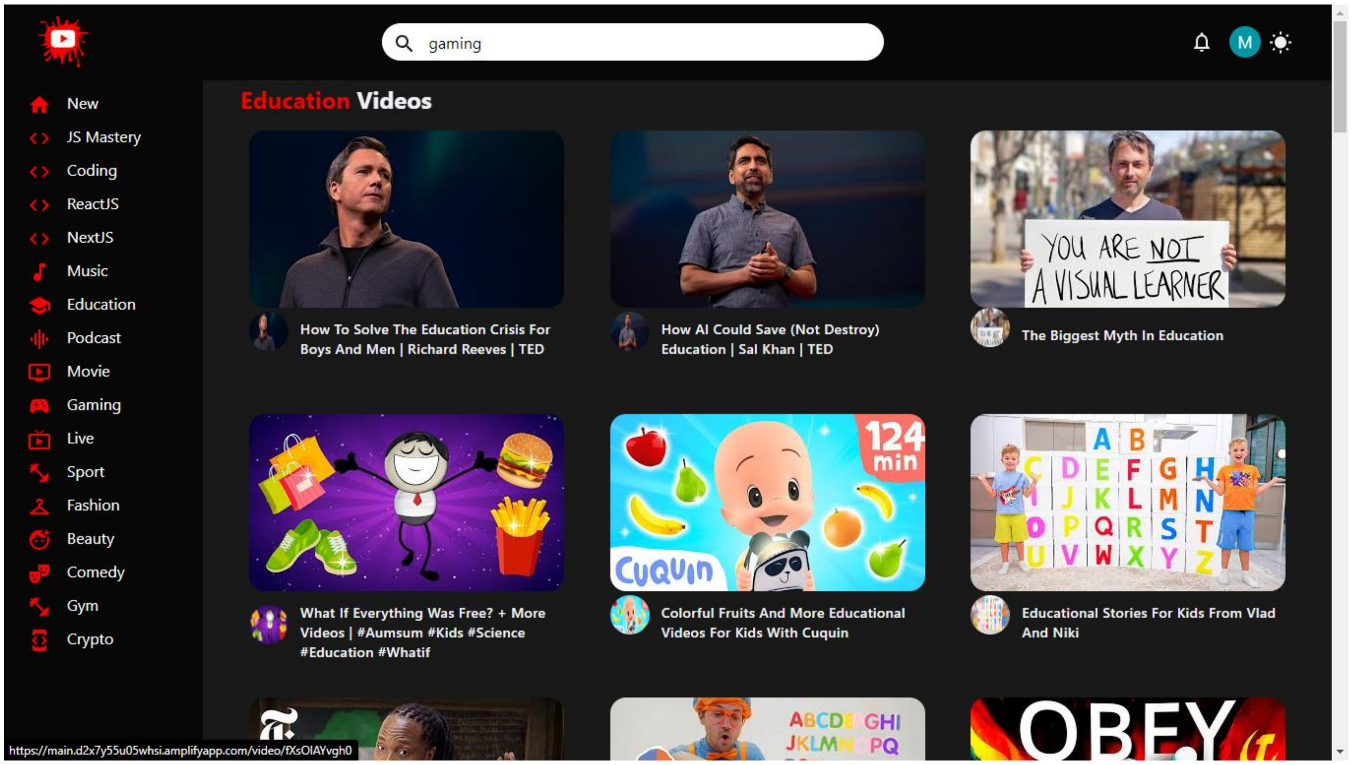
Why Context API ?

While React's state management is powerful and flexible, it can introduce certain challenges and problems if not used carefully. Some common issues with state in React include:

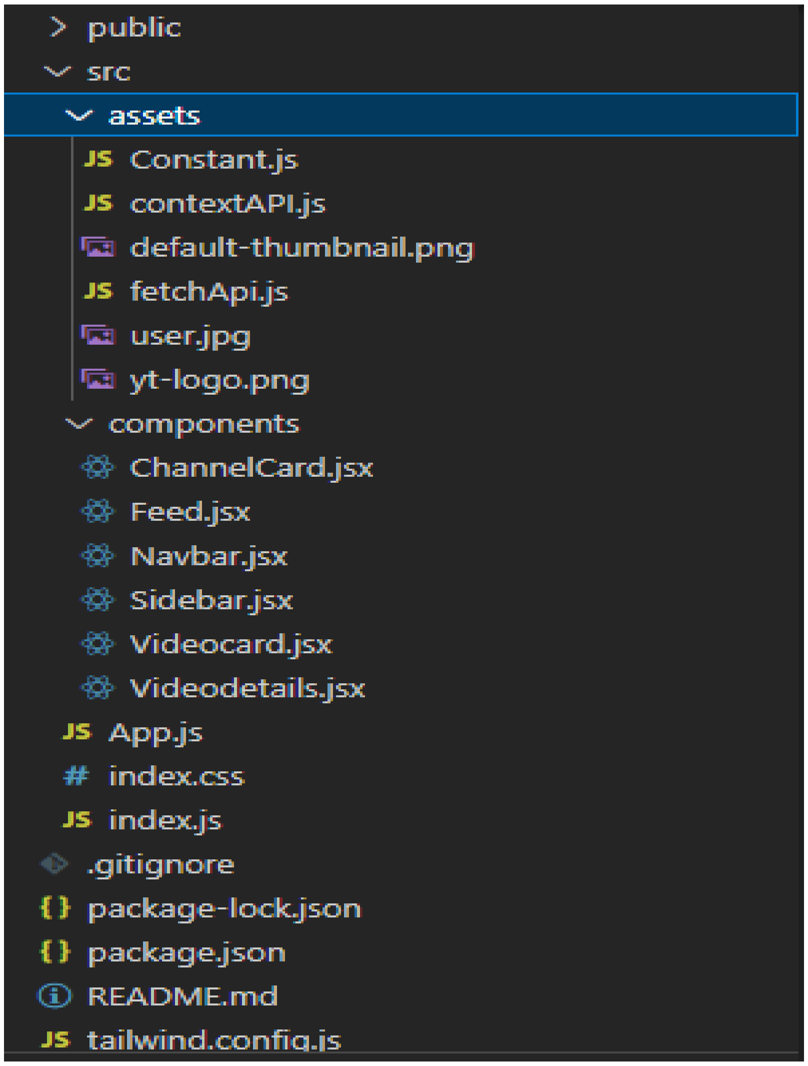
- **Complex State Management**: As a React application grows, managing state can become complex. You may have multiple components that need access to the same state, which can lead to prop drilling or overuse of context.

- **Prop Drilling**: When state needs to be passed down through many levels of nested components via props, it can make the code harder to maintain and understand. This is known as "prop drilling."

- **Component Re-renders**: Updating the state in React components can trigger re-renders. In some cases, this may cause unnecessary re-renders, impacting performance. Techniques like memoization and using the useMemo hook can help mitigate this issue


gaming
M
New
JS Mastery
Coding
ReactJS
NextJS
Music
Education
Podcast
Movie
Gaming
Live
Sport
Fashion
Beauty
Comedy
Gym
Crypto
Education Videos
How To Solve The Education Crisis For Boys And Men | Richard Reeves | TED
How AI Could Save (Not Destroy) Education | Sal Khan | TED
YOU ARE NOT A VISUAL LEARNER
The Biggest Myth In Education
124 min
CuQuin
What If Everything Was Free? + More Videos | #Aumsum #Kids #Science #Education #Whatif
Colorful Fruits And More Educational Videos For Kids With Cuquin
Educational Stories For Kids From Vlad And Niki
OBEY
https://main.d2x7y55u05whsi.amplifyapp.com/video/fXsOlAYvgh0

Folder Structure



```
> public
v src
  v assets
      Constant.js
      contextAPI.js
      default-thumbnail.png
      fetchApi.js
      user.jpg
      yt-logo.png
  v components
      ChannelCard.jsx
      Feed.jsx
      Navbar.jsx
      Sidebar.jsx
      Videocard.jsx
      Videodetails.jsx
    App.js
    index.css
    index.js
  .gitignore
  package-lock.json
  package.json
  README.md
  tailwind.config.js
```

The 'assets' directory contains the context provider, while the 'components' folder includes subfolders for images, and each subfolder within 'components' corresponds to its respective folder structure.

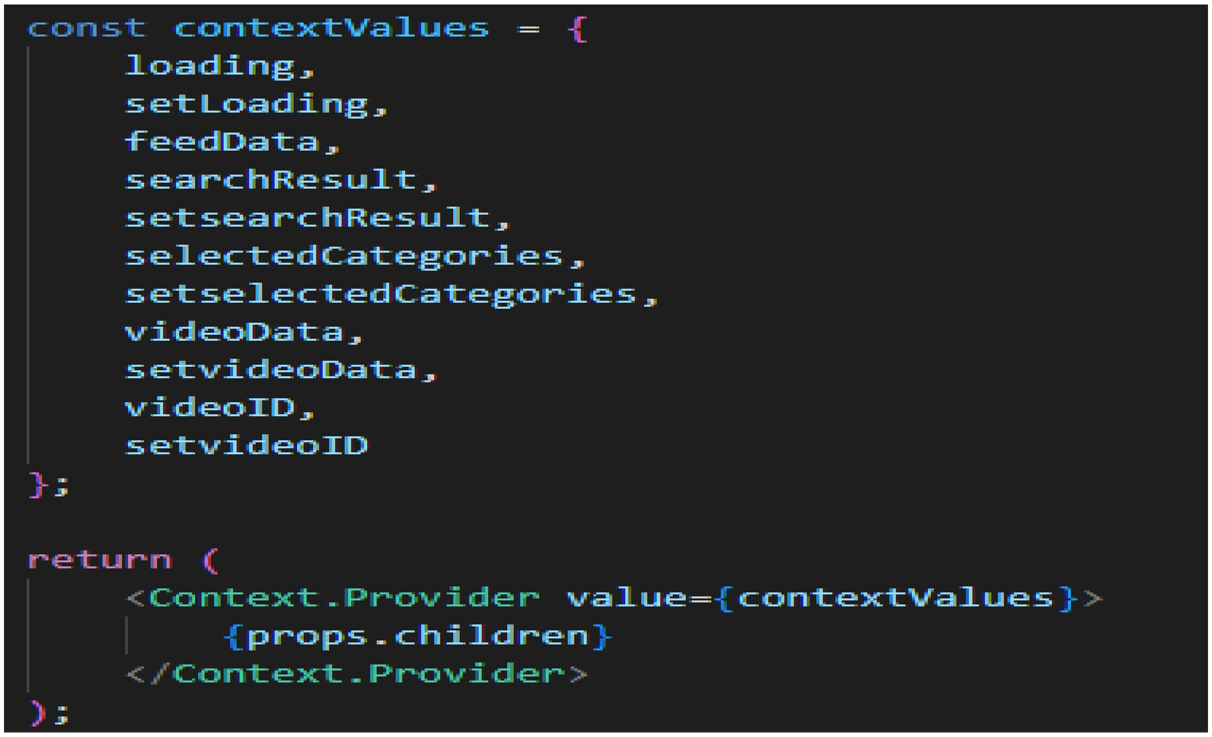
What is Context API ?

In React, the Context API is a built-in feature that provides a way to pass data through the component tree without having to manually pass props down the tree from parent to child components at every level. It is designed to solve the problem of "prop drilling," which occurs when you need to pass data from a top-level component down to deeply nested components.

The Context API consists of two main parts:

- **Context Provider**: The provider component is used to wrap a part of your component tree, typically in a higher-level component. It accepts a value prop, which can be an object, function, or any other data you want to share with the components within the tree. This value is then made available to all the components in the wrapped tree.

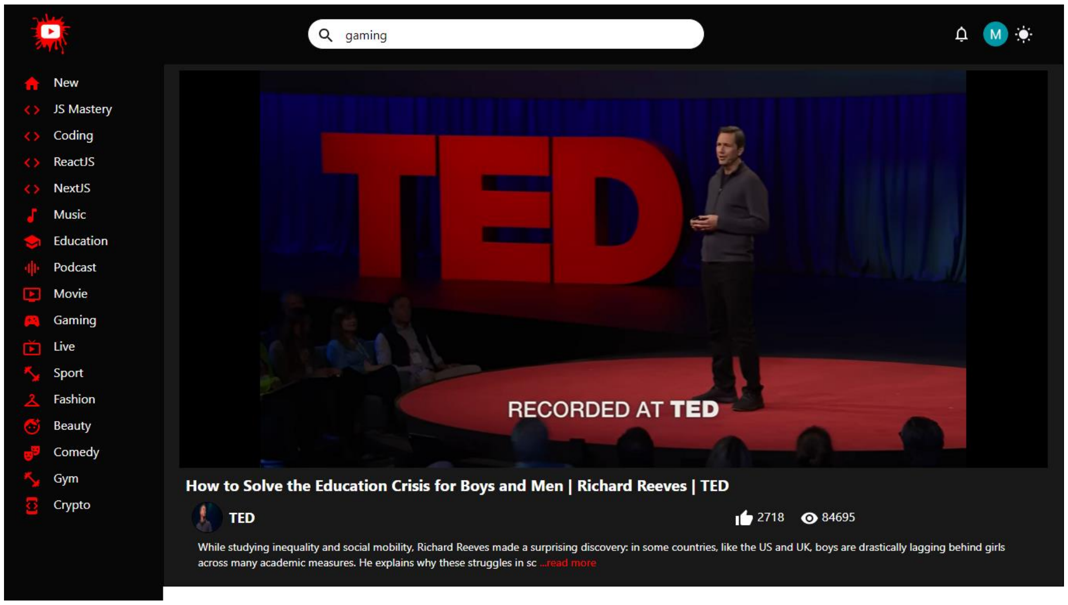
- **Context Consumer**: The consumer component is used in any component that needs access to the data provided by the context. It allows components to subscribe to the context and access the value prop provided by the nearest context provider in the component hierarchy.



```
const contextValues = {
    loading,
    setLoading,
    feedData,
    searchResult,
    setsearchResult,
    selectedCategories,
    setselectedCategories,
    videoData,
    setvideoData,
    videoID,
    setvideoID
};

return (
    <Context.Provider value={contextValues}>
        {props.children}
    </Context.Provider>
);
```

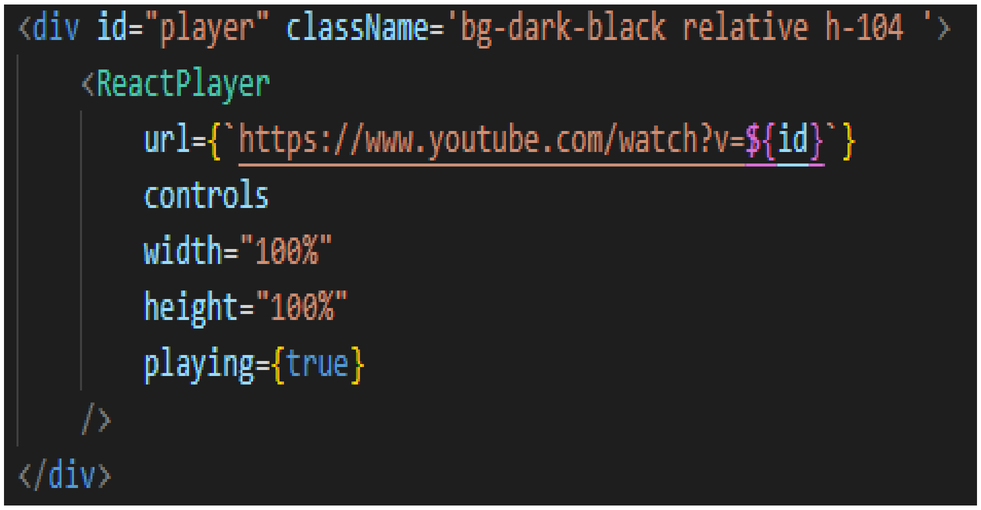
React Video Player



How to Solve the Education Crisis for Boys and Men | Richard Reeves | TED

While studying inequality and social mobility, Richard Reeves made a surprising discovery: in some countries, like the US and UK, boys are drastically lagging behind girls across many academic measures. He explains why these struggles in sc ...read more

A React video player is a user interface component or library designed to play and control video content within a React application. React video players are built on top of the React library and often use HTML5 <video> elements under the hood. They provide a set of features and controls for video playback, such as play, pause, volume control, fullscreen mode, and support for custom skins or themes.



```
<div id="player" className='bg-dark-black relative h-104 '>
    <ReactPlayer
        url={`https://www.youtube.com/watch?v=${id}`}
        controls
        width="100%"
        height="100%"
        playing={true}
    />
</div>
```

Conclusion

In conclusion, this project represents a comprehensive and feature-rich YouTube clone built with React. Leveraging the Context API for efficient state management, it offers a seamless user experience. The integration of the Fetch API allows for smooth data retrieval and search functionality, enabling users to access and enjoy their favorite video content. The thoughtful inclusion of both dark and light mode themes enhances the platform's personalization and usability. The user interface is not only visually appealing but also highly responsive, meticulously styled with Tailwind CSS. This YouTube clone serves as a testament to the developer's commitment to creating a user-friendly and visually engaging web application, delivering an experience that closely mirrors the original platform's functionality while introducing its own unique features.