YOUTUBE CLONE SRS

1. INTRODUCTION:

I. PURPOSE:

YouTube Clone helps to show the recent feed, search result videos, related videos to that search, channel pages, channel details, channel video details, likes and views on the video and helps user to set the sound, quality of the video, subtitles of the video, playback speed and the user can also share a video and can add that video to watch later.

II. SCOPE:

The scope of the YouTube Clone is to provide search results to all the people over the world who are accessing the website.

III. OVERVIEW:

The overview of the YouTube Clone is that it can be used by people to watch their needed videos without ads and can get the results within a shorter period of time.

2. GENERAL DESCRIPTION:

YouTube allows users to search, view channel page, like, share, add to watch later, report, and change the quality of videos. It offers a wide variety of user-generated and corporate media videos. Available content includes video clips, TV show clips, music videos and documentary films, movie trailers, live streams, and other content such as video blogging, short original videos, and educational videos.

3.FUNCTIONAL REQUIREMENT:

R:1: App( )

DESCRIPTION

This a centralised functional component from where all the other components are routed to different endpoints and it also consist of those components which remains fixed throughout the whole application.

R:2: fetchFromApi( )

DESCRIPTION

This is a async function which takes a URL as Input and then fetch function is applied on that which takes that URL along with some options like how many results will be displayed and the Rapid API key and Rapid API Host and then it tries to get some response then it returns the data that it got from the response.

INPUT

https://youtube-v31.p.rapidapi.com/

OUTPUT

Data will be generated in Object Format consisting of key and values.

R:3: SearchBar( )

DESCRIPTION

Search bar is a functional component which is present inside the Navbar component and the work of the search bar component is to handle the user typed search term and try to manage the state variable by updating the state and once the user types something and clicks on enter button or click on the search icon then use navigate hook is fired which helps to reload the page based on the recent search term and then it clears the input field so that any other searches can be done on that input field.

INPUT

Coding

OUTPUT

https://abit-youtube.vercel.app/search/coding

The above link will be loaded consisting of the videos related to coding only and the search field will be empty.

R:4: Feed( )

DESCRIPTION

Feed is a functional component which is connected with two other functional components that is sidebar component and video component. Feed function first uses its use Effect hooks to load the videos based on the selected category state then update the video value and then that video value is used to pass to videos component as props and inside side bar component we pass selected category and set selected category as props.

R:5: SideBar( )

DESCRIPTION

All the categories that is present inside the utils constant file is mapped and it returns a button and some span and the button has on click event which is used to set the value by using prop drilling and passing that set function.

R:6: Videos( )

DESCRIPTION

Video component includes three different components that is video card, channel card and loader. Whenever we search a term inside the input field if that term related to the channel name, then it loads the channel card and some videos which are related to that channel tag line and if that search term is related to a video query then video component is loaded and in that mean time of loading a loader component is executed so we can see a loader at that time.

R:7: VideoCard( )

DESCRIPTION

Video card functional component consist of props out of which we only need the video id and snippet then we have stored that using object destructuring and the values of this snippet is used inside the JSX to load the video card.

R:8: VideoDetails( )

DESCRIPTION

Video Details card uses fetchFromApi function to get the video details and also uses useParams hook to get the value of id that is present in the URL. Then uses this id to get that particular video details and use that data to update two states that is video details state and related video state so that video details can be used to show the video and related video details is used to display the related video to that video.

R:9: ChannelCard( )

DESCRIPTION

Channel Card is a functional component which have some props and out of this prop some of the values like channel name, channel image, channel id and channel subscribers are taken which is needed to load the JSX of the channel card and from react router dom Link is taken which is used to redirect to that channel URL when someone click on the channel image or title and until unless the data is loaded the demo title, demo image of the channel is used to fill initially but once our data is loaded it changes suddenly the view.

R:10: ChannelDetails( )

DESCRIPTION

Channel Details component includes two other component that is Videos component and channel card component and uses useParams hook to get the id value then use that id value to fetch the data related to that channel id then the two states are updated first is the channel details state and the second one is the video details state then this data is used to load the JSX of channel details. At the top of this component the channel card is displayed along with the subscribers and channel titles and under this component all the videos that are uploaded by this channel is loaded and displayed by using useEffect hook and also fetchFromApi function is also used to load and update the states.

4. INTERFACE REQUIREMENT:

DESCRIPTION

The connectivity of the complete web application from will be done using react router dom and the fetching of updated data for feed and results on search is done by using Rapid API and Deployment is done on Netlify along with the domain name is also taken from the Netlify.

5. PERFORMANCE REQUIREMENT:

I. Time – 1 month

II. Accuracy – 100%

III. Memory Needed at the Time of Development – 720 MB

IV. Memory Needed for Hosting – 2 MB (Optimised Production Version)

6. DESIGN CONSTRAINTS:

The YouTube Clone is made using the Tech Stack: React Js, Material UI, JSX, CSS, Vanilla JavaScript and some of the optimal algorithms.

7. NON-FUNCTIONAL REQUIREMENTS:

I. Security – OK

II. Portability – yes

III. Reusability – yes

IV. Reliability – applied

8. PRIMARY STRUCTURE BUDGET:

Schedule: 1 group [1 team leader]

Budget: None

Group consist of 4 members

Abhishek Nayak

Kshitun Harina Dehury

Laxmipriya Pradhan

Alok Kumar Swain