

KTube App-YouTube2.0 -> My replica to YouTube App, one of the largest online video streaming and sharing social media platforms owned by Google. KTube App in React.js using the newest version of Material UI, which contains a stunning new video section, custom categories responsive channel, and video cards channel pages. The user can play the video straight from the App, browse related videos, and search intended videos. Along with building this App, I have learned a lot on API fetching, dependencies and packages management, React functional components and their reusability, React file and folder structure, Vscode extension usage, fetching media queries for satisfactory responsiveness on various devices using RapidAPI to fetch data from unlimited sources.

Developing Languages, Tools, and Techniques Needed:

Vscode 1.74 https://code.visualstudio.com/updates/v1_74

ReactJS <https://reactjs.org/>

RapidAPI Vscode Client <https://rapidapi.com/guides/categories/rapidapi-client-vscode>

Node.JS npm/npx <https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

Yarn <https://classic.yarnpkg.com/lang/en/docs/install/#mac-stable>

Material UI <https://mui.com/>

JavaScript <https://www.javascript.com/>

ES7+ React/Redux/React-Native snippets Vscode Extension

<https://marketplace.visualstudio.com/items?itemName=dsznajder.es7-react-js-snippets>

npm Intellisense Vscode Extension

<https://marketplace.visualstudio.com/items?itemName=christian-kohler.npm-intellisense>

RapidAPI YouTubeV3 API Key <https://rapidapi.com/ytdlfree/api/youtube-v31>

Prerequisites & Setups:

Create React app in Vscode Terminal:

```
npx create-react-app ./
```

Check who own the node_modules:

```
ls -la /usr/local/lib/node_modules
```

Change the ownership to my own username:

```
sudo chown -R MYUSERNAME: /usr/local/lib/node_modules
```

Install needed dependencies in Vscode Terminal:

```
yarn add @emotion/react @emotion/styled @mui/icons-material  
@mui/material axios react-router-dom
```

Audit all dependencies to the latest version:

```
npm install --legacy-peer-deps
```

crbugs deprecated npm start not working error. DEBUGGING: Install React-Player running:

```
yarn add react-player.
```

Install needed font at <https://mui.com/material-ui/getting-started/installation/>

Synchronous Developing Note:

Navbar Customization:

Customize search bar text input and icon in SearchBar.jsx:

```
<input  
  className="search-bar"  
  placeholder="Search for videos on KTube..."
```

```

        value=""
        onChange={() => {}}
      />
      <IconButton type="submit" sx={{ p: '10px', color: 'red' }}>
        <Search />
      </IconButton>
    </Paper>
  )
}
...

```

Navbar is done customizing.

Sidebar Customization:

Customize Sidebar by fetching sidebar sections from constants.js in Sidebar.jsx:

```

    {categories.map((category) => (
      <button
        className="category-btn"
        style={{
          background: category.name ===
            selectedCategory && '#FC1503',
          color: 'white'
        }}
        key={category.name}
      >
        <span style={{
          color: category.name ===
            selectedCategory ? 'white' : 'red',
          marginRight: '15px'
        }}
        ...

```

sidebar done customizing.PNG

Fetch API from RapidAPI, and design VideoCard.jsx:

```

const VideoCard = ({ video: { id: { videoId }, snippet } }) => (
  <Card sx={{ width: { xs: '100%', sm: '358px', md: "320px", },
    boxShadow: "none", borderRadius: 0 }}>
    <Link to={videoId ? `/video/${videoId}` :
      `/video/cV2gBU6hKfY`} >
      <CardMedia image={snippet?.thumbnails?.high?.url ||
        demoThumbnailUrl} alt={snippet?.title}
    ...

```

Design ChannelCard.jsx:

```

    sx={{
      boxShadow: 'none',
      borderRadius: '20px',
      display: 'flex',

```

```

justifyContent: 'center',
alignItems: 'center',
width: { xs: '356px', md: '320px' },
height: '326px',
margin: 'auto',
marginTop,

```

Now the video page fetched based on categories successfully!

videos of categories fetched.PNG

ChannelDetails Customization:

Fetch channel details and API data in ChannelDetail.jsx:

```

useEffect(() => {
  const fetchResults = async () => {
    const data = await
fetchFromAPI(`channels?part=snippet&id=${id}`);
    setChannelDetail(data?.items[0]);
    const videosData = await
fetchFromAPI(`search?channelId=${id}&part=snippet%2Cid&order=date`);
    setVideos(videosData?.items);
  };

```

channel details fetched.PNG

Search feed customization:

Customize feed searching option in SearchFeed.jsx:

```

const SearchFeed = () => {
  const [videos, setVideos] = useState(null);
  const { searchTerm } = useParams();
  useEffect(() => {
    fetchFromAPI(`search?part=snippet&q=${searchTerm}`)
      .then((data) => setVideos(data.items))
  }, [searchTerm]);
  return (
    <Box p={2} minHeight="95vh">
      <Typography variant="h4" fontWeight={900} color="white" mb={3}
ml={{ sm: "100px" }}>
        Search Results for <span style={{ color: "#FC1503"
}}>{searchTerm}</span> videos
      </Typography>
      <Box display="flex">
...

```

searching feed works, showing results.PNG

To make the videos playable, fetch React Player in VideoDetails.jsx:

```

<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}>
```

...

videos are playable.PNG