**Objective:-** Frontend: Using ReactJS, make Image Carousel, FAQ/Accordion and Shopping item list with filters.

**Code :-**

**App.jsx**

import Header from "./components/Header";

import Feed from "./components/Feed";

import SearchResult from "./components/SearchResult";

import VideoDetail from "./components/VideoDetail";

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

import { useEffect } from "react";

function App() {

  useEffect(() => {

    window.scrollTo(0, 0);

  }, []);

  return (

    <div className="flex flex-col h-full">

      <Header />

      <Routes>

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

        <Route path="/searchResult/:searchQuery" element={<SearchResult />} />

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

      </Routes>

    </div>

  );

}

export default App;

**App.css**

*#root {*

*max-width: 1280px;*

*margin: 0 auto;*

*padding: 2rem;*

*text-align: center;*

*}*

*.logo {*

*height: 6em;*

*padding: 1.5em;*

*will-change: filter;*

*transition: filter 300ms;*

*}*

*.logo:hover {*

*filter: drop-shadow(0 0 2em #646cffaa);*

*}*

*.logo.react:hover {*

*filter: drop-shadow(0 0 2em #61dafbaa);*

*}*

*@keyframes logo-spin {*

*from {*

*transform: rotate(0deg);*

*}*

*to {*

*transform: rotate(360deg);*

*}*

*}*

*@media (prefers-reduced-motion: no-preference) {*

*a:nth-of-type(2) .logo {*

*animation: logo-spin infinite 20s linear;*

*}*

*}*

*.card {*

*padding: 2em;*

*}*

*.read-the-docs {*

*color: #888;*

*}*

**index.css**

*@tailwind base;*

*@tailwind components;*

*@tailwind utilities;*

*body {*

*margin: 0;*

*font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",*

*"Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",*

*sans-serif;*

*-webkit-font-smoothing: antialiased;*

*-moz-osx-font-smoothing: grayscale;*

*}*

*#root {*

*height: 100vh;*

*}*

*code {*

*font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",*

*monospace;*

*}*

*.load-bar {*

*position: fixed;*

*top: 0;*

*left: 0;*

*width: 100%;*

*height: 3px;*

*background-color: red;*

*}*

*.bar {*

*content: "";*

*display: inline;*

*position: absolute;*

*width: 0;*

*height: 100%;*

*left: 50%;*

*text-align: center;*

*}*

*.bar:nth-child(1) {*

*background-color: #da4733;*

*animation: loading 3s linear infinite;*

*}*

*.bar:nth-child(2) {*

*background-color: #e7553b;*

*animation: loading 3s linear 1s infinite;*

*}*

*.bar:nth-child(3) {*

*background-color: #bd0d00;*

*animation: loading 3s linear 2s infinite;*

*}*

*@keyframes loading {*

*from {*

*left: 50%;*

*width: 0;*

*z-index: 100;*

*}*

*33.3333% {*

*left: 0;*

*width: 100%;*

*z-index: 10;*

*}*

*to {*

*left: 0;*

*width: 100%;*

*}*

*}*

*.custom-h {*

*height: auto !important;*

*}*

*@media screen and (min-width: 1024px) {*

*.custom-h {*

*height: 100vh !important;*

*}*

*}*

*.hide::-webkit-scrollbar {*

*display: none;*

*}*

**components/Feed.jsx**

import { useContext, useEffect } from "react";

import { Context } from "../context/ContextApi";

import LeftNav from "./LeftNav";

import VideoCard from "./VideoCard";

const Feed = () => {

  const { loading, searchResult } = useContext(Context);

  useEffect(() => {

    document.getElementById("root").classList.remove("custom-h");

    window.scrollTo(0, 0);

  }, []);

  return (

    <div className="flex flex-row h-[calc(100%-56px)]">

      <LeftNav />

      <div className="grow w-[calc(100%-240px)] h-full overflow-y-auto bg-white dark:bg-black">

        <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-4 p-5">

          {!loading &&

            searchResult.map((*item*, *index*) => {

              if (item.type !== "video") return **false**;

              return <VideoCard key={index} video={item?.video} />;

            })}

        </div>

      </div>

    </div>

  );

};

export default Feed;

**components/Header.jsx**

import { useContext, useState } from "react";

import { Link, useLocation, useNavigate } from "react-router-dom";

import ytlogo from "../assets/logo.png"; *// YouTube logo import*

import { SlMenu } from "react-icons/sl";

import { IoIosSearch } from "react-icons/io";

import { RiVideoAddLine } from "react-icons/ri";

import { FiBell } from "react-icons/fi";

import { CgClose } from "react-icons/cg";

import { Context } from "../context/ContextApi";

import Loader from "../shared/loader";

const Header = () => {

  const [searchQuery, setSearchQuery] = useState("");

  const { loading, mobileMenu, setMobileMenu } = useContext(Context);

  const navigate = useNavigate();

  const { pathname } = useLocation();

  const pageName = pathname?.split("/")?.filter(Boolean)?.[0];

  const searchQueryHandler = (*event*) => {

    if (

      (event?.key === "Enter" || event === "searchButton") &&

      searchQuery?.length > 0

    ) {

      navigate(`/searchResult/${searchQuery}`);

    }

  };

  const mobileMenuToggle = () => {

    setMobileMenu(!mobileMenu);

  };

  return (

    <div className="sticky top-0 z-20 flex flex-row items-center justify-between min-h-20 px-4 md:px-5 bg-white dark:bg-black">

      {loading && <Loader />}

      <div className="flex items-center">

        {pageName !== "video" && (

          <div

            className="flex md:hidden md:mr-6 mr-4 cursor-pointer items-center justify-center h-10 w-10 rounded-full hover:bg-[#8888]/[0.6]"

            onClick={mobileMenuToggle}

          >

            {mobileMenu ? (

              <CgClose className="dark:text-white text-black text-xl" />

            ) : (

              <SlMenu className="dark:text-white text-black text-xl" />

            )}

          </div>

        )}

        {*/\* YouTube logo \*/*}

        <Link to="/" className="flex items-center">

          <img src={ytlogo} alt="YouTube Logo" className="h-10" />

        </Link>

      </div>

      <div className="group flex items-center my-1 mr-7">

        <div className="flex h-8 md:h-10 md:ml-10 md:pl-5 border border-[#888888] rounded-l-3xl group-focus-within:border-blue-500 md:group-focus-within:ml-5 md:group-focus-within:pl-0">

          <div className="w-10 items-center justify-center hidden group-focus-within:md:flex">

            <IoIosSearch className="text-black dark:text-white text-xl" />

          </div>

          <input

            type="text"

            className="bg-transparent outline-none text-black dark:text-white pr-5 pl-5  md:pl-0 w-44 md:group-focus-within:pl-0 md:w-64 lg:w-[500px]"

            onChange={(*e*) => setSearchQuery(e.target.value)}

            onKeyUp={searchQueryHandler}

            placeholder="Search"

            value={searchQuery}

          />

        </div>

        <button

          className="w-[40px] md:w-[60px] h-8 md:h-10 flex items-center justify-center border border-l-0 border-[#888888] rounded-r-3xl bg-white/[0.1] dark:bg-black/[0.1]"

          onClick={() => searchQueryHandler("searchButton")}

        >

          <IoIosSearch className="text-black dark:text-white text-xl" />

        </button>

      </div>

      <div className="flex items-center">

        <div className="hidden md:flex">

          <div className="flex items-center justify-center h-10 w-10 rounded-full hover:bg-[#303030]/[0.6]">

            <RiVideoAddLine className="text-black dark:text-white text-xl cursor-pointer" />

          </div>

          <div className="flex items-center justify-center ml-2 h-10 w-10 rounded-full hover:bg-[#303030]/[0.6]">

            <FiBell className="text-black dark:text-white text-xl cursor-pointer" />

          </div>

        </div>

        <span className="bg-gradient-to-r from-blue-400 to-blue-600 text-white flex h-10 w-10 justify-center items-center font-bold overflow-hidden rounded-full md:ml-4 mx-1 shadow-lg">

          AL

        </span>

      </div>

    </div>

  );

};

export default Header;

**components/LeftNav.jsx**

import React, { useContext } from "react";

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

import LeftNavMenuItem from "./LeftNavMenuItem";

import { Context } from "../context/ContextApi";

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

function LeftNav() {

  const { selectcategories, setSelectCategories, mobileMenu } =

    useContext(Context);

  const navigate = useNavigate();

  const clickHandle = (*name*, *type*) => {

    switch (type) {

      case "category":

        return setSelectCategories(name);

      case "home":

        return setSelectCategories(name);

      case "menu":

        return **false**;

      default:

        break;

    }

  };

  return (

    <div

      className={`md:block w-[240px] overflow-y-auto h-full py-4 bg-white dark:bg-black absolute md:relative z-10 hide translate-x-[-240px] md:translate-x-0 transition-all ${mobileMenu ? "translate-x-[0px]" : ""

        }`}

    >

      <div className="flex px-5 flex-col">

        {categories?.map((*item*) => {

          return (

            <React.Fragment key={item.name}>

              <LeftNavMenuItem

                key={item.type}

                text={item.type === "home" ? "Home" : item.name}

                icon={item.icon}

                action={() => {

                  clickHandle(item.name, item.type);

                  navigate("/");

                }}

                className={`${selectcategories === item.name

                  ? "bg-black/[0.15] dark:bg-white/[0.15]"

                  : ""

                  }`}

              />

              {item.divider && (

                <hr className="my-5 border-black[0.2] dark:border-white[0.2]" />

              )}

            </React.Fragment>

          );

        })}

        <hr className="my-5 border-white[0.2]" />

      </div>

    </div>

  );

}

export default LeftNav;

**components/LeftNavMenuItem.jsx**

function LeftNavMenuItem({ *text*, *icon*, *className*, *action* }) {

  return (

    <div

      className={

        "text-black dark:text-white text-sm cursor-pointer h-10 flex items-center px-3 mb-[4px] rounded-lg hover:dark:bg-white/[0.15] hover:dark:bg-white[0.15] " +

        className

      }

      onClick={action}

    >

      <span className="text-xl mr-5">{icon}</span>

      {text}

    </div>

  );

}

export default LeftNavMenuItem;

**components/SearchResult.jsx**

import { useState, useEffect, useContext } from "react";

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

import { fetchDataFromApi } from "../utils/Api";

import { Context } from "../context/ContextApi";

import LeftNav from "./LeftNav";

import SearchResultVideoCard from "./SearchResultVideoCard";

function SearchResult() {

  const [result, setResult] = useState();

  const { searchQuery } = useParams();

  const { setLoading } = useContext(Context);

  useEffect(() => {

    document.getElementById("root").classList.remove("custom-h");

    fetchSearchResult();

    window.scrollTo(0, 0);

  }, [searchQuery]);

  const fetchSearchResult = () => {

    setLoading(**true**);

    fetchDataFromApi(`search/?q=${searchQuery}`).then((*res*) => {

      setResult(res.contents);

      setLoading(**false**);

    });

  };

  return (

    <div className="flex flex-row  h-[calc(100%-56px)]">

      <LeftNav />

      <div className="grow w-[calc(100%-240px)] h-full overflow-y-auto bg-white dark:bg-black">

        <div className="grid grid-cols-1 gap-2 p-5">

          {result?.map((*item*) => {

            if (item.type !== "video") return **false**;

            return (

              <SearchResultVideoCard

                key={item?.video?.videoId}

                video={item?.video}

              />

            );

          })}

        </div>

      </div>

    </div>

  );

}

export default SearchResult;

**components/SearchResultVideoCard.jsx**

*/\* eslint-disable react/prop-types \*/*

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

import { BsFillCheckCircleFill } from "react-icons/bs";

import { abbreviateNumber } from "js-abbreviation-number";

import VideoLength from "../shared/VideoLength";

function SearchResultVideoCard({ *video* }) {

  return (

    <Link to={`/video/${video?.videoId}`}>

      <div className="flex flex-col md:flex-row mb-8 md:mb-3 dark:lg:hover:bg-white/[0.1]  lg:hover:bg-black/[0.1] rounded-xl md:p-4">

        <div className="relative flex shrink-0 h-48 md:h-28 lg:h-40 xl:h-48 w-full md:w-48 lg:w-64 xl:w-80 rounded-xl bg-slate-800 overflow-hidden">

          <img

            className="h-full w-full object-cover"

            src={video?.thumbnails[0]?.url}

          />

          {video?.lengthSeconds && <VideoLength time={video?.lengthSeconds} />}

        </div>

        <div className="flex  flex-col ml-4 md:ml-6 mt-4 md:mt-0 overflow-hidden">

          <span className="text-lg md:text-xl font-semibold line-clamp-2 text-black dark:text-white">

            {video?.title}

          </span>

          <span className="empty:hidden text-sm line-clamp-1 md:line-clamp-2 text-black/[0.7] dark:text-white/[0.7] md:pr-24 md:my-4">

            {video?.descriptionSnippet}

          </span>

          <div className="flex items-center">

            <div className="flex items-start mr-3">

              <div className="flex h-9 w-9 rounded-full overflow-hidden">

                <img

                  className="h-full w-full object-cover"

                  src={video?.author?.avatar[0]?.url}

                />

              </div>

            </div>

            <div className="flex flex-col">

              <span className="text-sm font-semibold mt-2 text-black/[0.7] dark:text-white/[0.7] flex items-center">

                {video?.author?.title}

                {video?.author?.badges[0]?.type === "VERIFIED\_CHANNEL" && (

                  <BsFillCheckCircleFill className="text-black/[0.7] dark:text-white/[0.5] text-[12px] lg:text-[10px] xl:text-[12px] ml-1" />

                )}

              </span>

              <div className="flex text-sm font-semibold text-black/[0.7] dark:text-white/[0.7] truncate overflow-hidden">

                <span>{`${abbreviateNumber(

                  video?.stats?.views,

                  2

                )} views`}</span>

                <span className="flex text-[24px] leading-none font-bold text-black/[0.7] dark:text-white/[0.7] relative top-[-10px] mx-1">

                  .

                </span>

                <span className="truncate">{video?.publishedTimeText}</span>

              </div>

            </div>

          </div>

        </div>

      </div>

    </Link>

  );

}

export default SearchResultVideoCard;

**components/SuggestionVideoCard.jsx**

*/\* eslint-disable react/prop-types \*/*

import { abbreviateNumber } from "js-abbreviation-number";

import { BsFillCheckCircleFill } from "react-icons/bs";

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

import VideoLength from "../shared/VideoLength";

const SuggestionVideoCard = ({ *video* }) => {

  return (

    <Link to={`/video/${video?.videoId}`}>

      <div className="mb-3 flex flex-col md:flex-row items-start">

        <div className="relative h-56 md:h-24 lg:h-20 xl:h-24 w-full md:w-40 min-w-[168px] lg:w-32 lg:min-w-[128px] xl:w-40 xl:min-w-[168px] rounded-xl bg-slate-800 overflow-hidden">

          <img

            className="h-full w-full object-cover"

            src={video?.thumbnails[0]?.url}

          />

          {video?.lengthSeconds && <VideoLength time={video?.lengthSeconds} />}

        </div>

        <div className="flex flex-col ml-3 overflow-hidden">

          <span className="text-md lg:text-xs xl:text-sm font-semibold line-clamp-2 text-black dark:text-white">

            {video?.title}

          </span>

          <span className="text-[12px] lg:text-[10px] xl:text-[12px] font-semibold mt-2 text-black/[0.7] dark:text-white/[0.7] flex items-center">

            {video?.author?.title}

            {video?.author?.badges[0]?.type === "VERIFIED\_CHANNEL" && (

              <BsFillCheckCircleFill className="text-white/[0.5] text-[12px] lg:text-[10px] xl:text-[12px] ml-1" />

            )}

          </span>

          <div className="flex text-[12px] lg:text-[10px] xl:text-[12px] font-semibold text-black/[0.7] dark:text-white/[0.7] truncate overflow-hidden">

            <span>{`${abbreviateNumber(video?.stats?.views, 2)} views`}</span>

            <span className="flex text-[24px] leading-none font-bold text-black/[0.7] dark:text-white/[0.7] relative top-[-10px] mx-1">

              .

            </span>

            <span className="truncate">{video?.publishedTimeText}</span>

          </div>

        </div>

      </div>

    </Link>

  );

};

export default SuggestionVideoCard;

**components/VideoCard.jsx**

*/\* eslint-disable react/prop-types \*/*

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

import { BsFillCheckCircleFill } from "react-icons/bs";

import { abbreviateNumber } from "js-abbreviation-number";

import VideoLength from "../shared/VideoLength";

function VideoCard({ *video* }) {

  return (

    <Link to={`/video/${video?.videoId}`}>

      <div className="flex flex-col mb-8">

        <div className="relative h-48 md:h-52 rounded-2xl overflow-hidden">

          <img

            src={video?.thumbnails[0]?.url}

            alt="thumbnails"

            className="h-full w-full object-cover"

          />

          {video.lengthSeconds && <VideoLength time={video?.lengthSeconds} />}

        </div>

        <div className="flex text-white mt-3">

          <div className="flex items-start">

            <div className="flex h-9 w-9 rounded-full overflow-hidden ">

              <img

                src={video?.author?.avatar[0]?.url}

                alt="avatar"

                className="w-full h-full object-cover"

              />

            </div>

            <div className="flex flex-col ml-3 overflow-hidden">

              <span className="text-sm font-semibold line-clamp-2 text-black dark:text-white">

                {video?.title}

              </span>

              <span className="text-[12px] font-semibold text-black/[0.7] dark:text-white/[0.7] flex items-center ">

                {video?.author?.title}

                {video?.author?.badges[0]?.type === "VERIFIED\_CHANNEL" && (

                  <BsFillCheckCircleFill className="text-black/[0.5] dark:text-white/[0.5] text-[12px] ml-1" />

                )}

              </span>

              <div className="flex text-[12px] font-semibold text-black/[0.7] dark:text-white/[0.7] truncate overflow-hidden">

                <span>{`${abbreviateNumber(

                  video?.stats?.views,

                  2

                )} views`}</span>

                <span className="flex  text-[24px] leading-none font-bold text-black/[0.7] dark:text-white/[0.7] relative top-[-10px] mx-1">

                  .

                </span>

                <span className="truncate">{video?.publishedTimeText}</span>

              </div>

            </div>

          </div>

        </div>

      </div>

    </Link>

  );

}

export default VideoCard;

**components/VideoDetail.jsx**

import { useState, useEffect, useContext } from "react";

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

import ReactPlayer from "react-player/youtube";

import { BsFillCheckCircleFill } from "react-icons/bs";

import { AiOutlineLike } from "react-icons/ai";

import { abbreviateNumber } from "js-abbreviation-number";

import { fetchDataFromApi } from "../utils/Api";

import { Context } from "../context/ContextApi";

import SuggestionVideoCard from "./SuggestionVideoCard";

function VideoDetail() {

  const [video, setVideo] = useState();

  const [relatedVideos, setRelatedVideos] = useState();

  const { id } = useParams();

  const { setLoading } = useContext(Context);

  useEffect(() => {

    document.getElementById("root").classList.add("custom-h");

    fetchVideoDetails();

    fetchRelatedVideos();

    window.scrollTo(0, 0);

  }, [id]);

  const fetchVideoDetails = () => {

    setLoading(**true**);

    fetchDataFromApi(`video/details/?id=${id}`).then((*res*) => {

      setVideo(res);

      setLoading(**false**);

    });

  };

  const fetchRelatedVideos = () => {

    setLoading(**true**);

    fetchDataFromApi(`video/related-contents/?id=${id}`).then((*res*) => {

      setRelatedVideos(res);

      setLoading(**false**);

    });

  };

  return (

    <div className="flex justify-center flex-row h-[calc(100%-56px)] bg-white dark:bg-black">

      <div className="w-full max-w-[1280px] flex flex-col lg:flex-row">

        <div className="flex flex-col lg:w-[calc(100%-350px)] xl:w-[calc(100%-400px)] px-4 py-3 lg:py-6 overflow-y-auto">

          <div className="h-[200px] md:h-[400px] lg:h-[400px] xl:h-[550px] ml-[-16px] lg:ml-0 mr-[-16px] lg:mr-0">

            <ReactPlayer

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

              controls

              width="100%"

              height="100%"

              style={{ backgroundColor: "#00000" }}

              playing={**true**}

            />

          </div>

          <div className="text-black dark:text-white font-semibold text-sm md:text-xl mt-4 line-clamp-2">

            {video?.title}

          </div>

          <div className="flex justify-between flex-col md:flex-row mt-4">

            <div className="flex">

              <div className="flex items-start ">

                <div className="flex h-11 w-11 rounded-full overflow-hidden">

                  <img

                    src={video?.author?.avatar[0]?.url}

                    alt="avatar"

                    className="h-full w-full object-cover"

                  />

                </div>

              </div>

              <div className="flex flex-col ml-3 ">

                <div className="text-black dark:text-white text-md font-semibold flex items-center">

                  {video?.author?.title}

                  {video?.author?.badges[0]?.type === "VERIFIED\_CHANNEL" && (

                    <BsFillCheckCircleFill className="text-black/[0.5] dark:text-white/[0.5] text-[12px] ml-1" />

                  )}

                </div>

                <div className="text-black/[0.7] dark:text-white/[0.7] text-sm">

                  {video?.author?.stats?.subscribersText}

                </div>

              </div>

            </div>

            <div className="flex text-black dark:text-white mt-4 md:mt-0">

              <div className="flex items-center justify-center h-11 px-6 rounded-3xl dark:bg-white/[0.15] bg-black/[0.2]">

                <AiOutlineLike className="text-xl dark:text-white text-black mr-2" />

                {`${abbreviateNumber(video?.stats?.views, 2)} Likes`}

              </div>

              <div className="flex items-center justify-center h-11 px-6 rounded-3xl dark:bg-white/[0.15] bg-black/[0.2] ml-4">

                {`${abbreviateNumber(video?.stats?.views, 2)} Views`}

              </div>

            </div>

          </div>

        </div>

        <div className="flex flex-col py-6 px-4 overflow-y-auto lg:w-[350px] xl:w-[400px] hide">

          {relatedVideos?.contents?.map((*item*, *index*) => {

            if (item?.type !== "video") return **false**;

            return <SuggestionVideoCard key={index} video={item?.video} />;

          })}

        </div>

      </div>

    </div>

  );

}

export default VideoDetail;

**context/ContextApi.jsx**

*/\* eslint-disable react/prop-types \*/*

import { createContext, useState, useEffect } from "react";

import { fetchDataFromApi } from "../utils/Api";

export const Context = createContext();

export const AppContext = ({ *children* }) => {

  const [loading, setLoading] = useState(**false**);

  const [searchResult, setSearchResult] = useState([]);

  const [selectcategories, setSelectCategories] = useState("New");

  const [mobileMenu, setMobileMenu] = useState(**false**);

  useEffect(() => {

    fetchSelectedCategoryData(selectcategories);

  }, [selectcategories]);

  const fetchSelectedCategoryData = (*query*) => {

    setLoading(**true**);

    fetchDataFromApi(`search/?q=${query}`).then(({ *contents* }) => {

      setSearchResult(contents);

      setLoading(**false**);

    });

  };

  return (

    <Context.Provider

      value={{

        loading,

        setLoading,

        searchResult,

        selectcategories,

        setSelectCategories,

        mobileMenu,

        setMobileMenu,

      }}

    >

      {children}

    </Context.Provider>

  );

};

**shared/loader.jsx**

function Loader() {

  return (

    <div className="load-bar">

      <div className="bar"></div>

      <div className="bar"></div>

      <div className="bar"></div>

    </div>

  );

}

export default Loader;

**shared/VideoLength.jsx**

import moment from "moment/moment";

function VideoLength({ *time* }) {

  const videoLengthInSeconds = moment()

    .startOf("day")

    .seconds(time)

    .format("H:mm:ss");

  return (

    <div className="absolute bottom-2 right-2 bg-black py-[2px] px-2 text-[11px] rounded text-white">

      {videoLengthInSeconds}

    </div>

  );

}

export default VideoLength;

**utils/Api.jsx**

import axios from "axios";

const BASE\_URL = "https://youtube138.p.rapidapi.com";

const options = {

  params: {

    hl: "en",

    gl: "in",

  },

  headers: {

    'x-rapidapi-Key': '12a0d48e45msh6517c15b49f95e8p186612jsn8584f8fcb890',

    'x-rapidapi-host': 'youtube138.p.rapidapi.com'

  }

};

export const fetchDataFromApi = async (*url*) => {

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

  return data;

};

**utils/Constants.jsx**

import { AiFillHome, AiOutlineFlag } from "react-icons/ai";

import { MdLocalFireDepartment, MdLiveTv } from "react-icons/md";

import { FiFilm } from "react-icons/fi";

import { IoGameControllerSharp } from "react-icons/io5";

import { ImNewspaper } from "react-icons/im";

import { RiLightbulbLine, RiFeedbackLine } from "react-icons/ri";

import { FiSettings, FiHelpCircle } from "react-icons/fi";

import { BiLike } from "react-icons/bi";

import { MdOutlineSubscriptions } from "react-icons/md";

export const categories = [

  { name: "New", icon: <AiFillHome />, type: "home" },

  { name: "Shorts", icon: <MdLocalFireDepartment />, type: "category" },

  {

    name: "Subscription",

    icon: <MdOutlineSubscriptions />,

    type: "category",

    divider: **true**,

  },

  { name: "Your Channel", icon: <FiFilm />, type: "category" },

  { name: "History", icon: <MdLiveTv />, type: "category" },

  { name: "Your Videos", icon: <IoGameControllerSharp />, type: "category" },

  { name: "Watch later", icon: <ImNewspaper />, type: "category" },

  { name: "Liked Videos", icon: <BiLike />, type: "category", divider: **true** },

  { name: "Learning", icon: <RiLightbulbLine />, type: "category" },

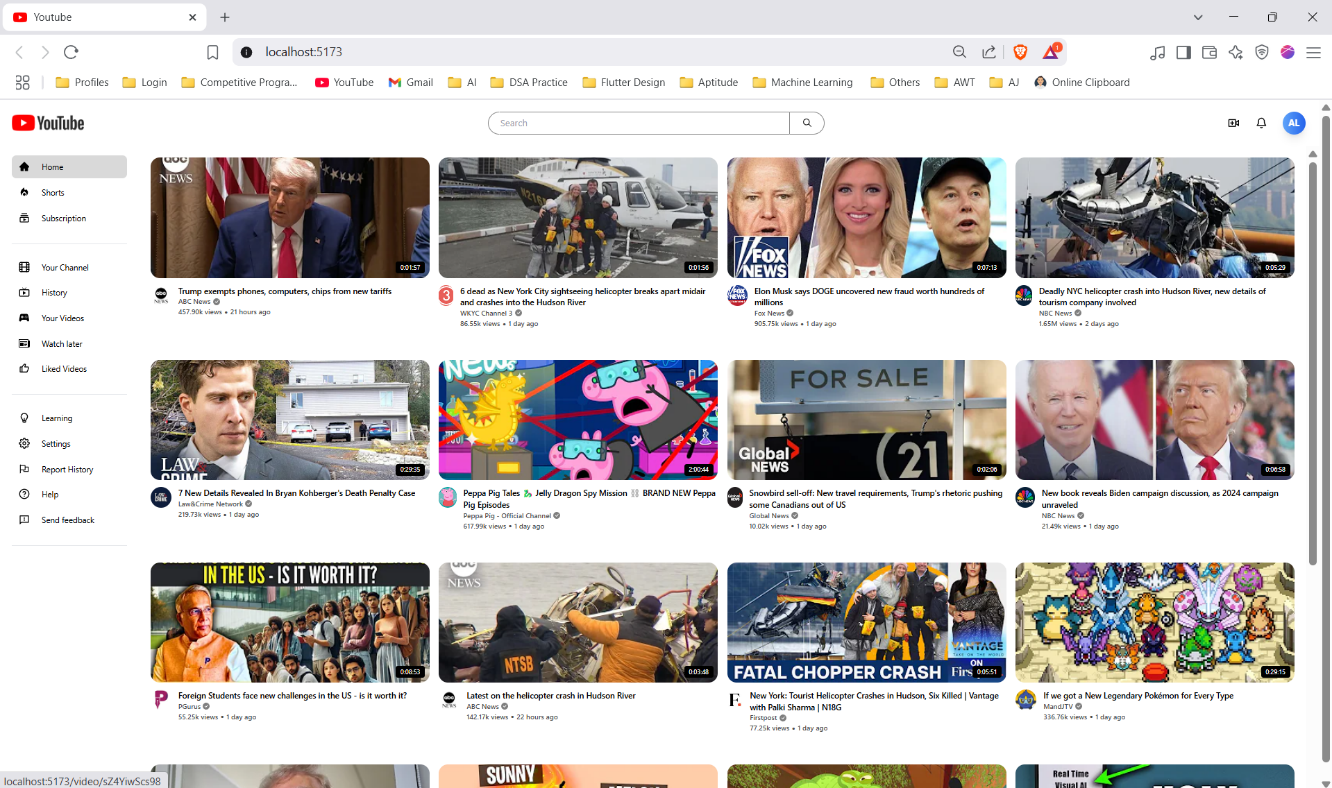
  { name: "Settings", icon: <FiSettings />, type: "menu" },

  { name: "Report History", icon: <AiOutlineFlag />, type: "menu" },

  { name: "Help", icon: <FiHelpCircle />, type: "menu" },

  { name: "Send feedback", icon: <RiFeedbackLine />, type: "menu" },

];

**Output :-**

