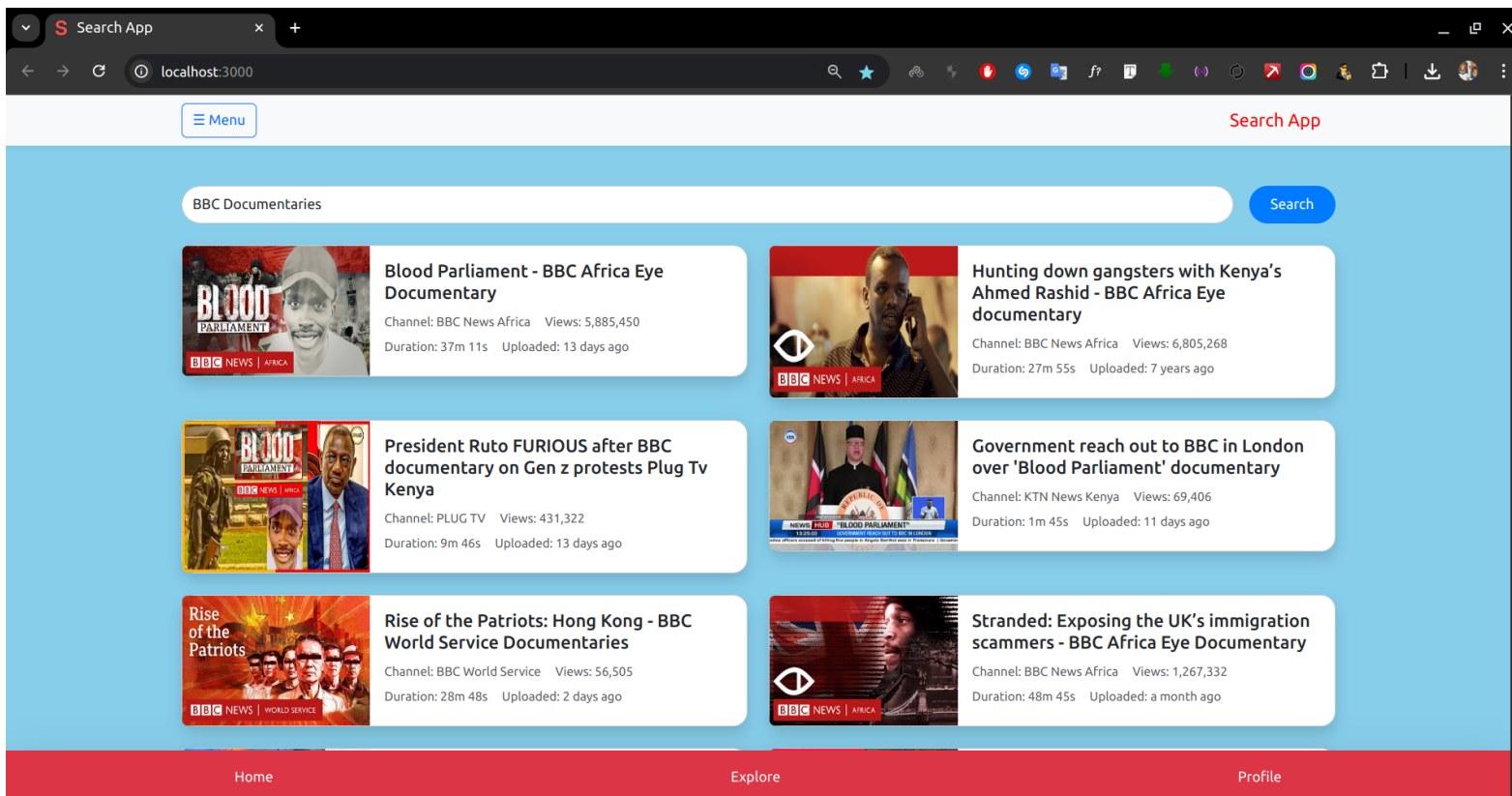


1. Screenshot(s) with search-term(s) along with their respective output.



2.Code from 5 files: index.tsx, APIInfo.js , Lookup.js, Result.js, and useAPIresponse.js

index.tsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import "bootstrap/dist/css/bootstrap.min.css";

const root = ReactDOM.createRoot(document.getElementById("root"));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
reportWebVitals();
```

APIInfo.js

```
import axios from "axios";

const API_KEY = "AIzaSyCBQB7fm-F6nGcoS1WIFgbiVc6Y6HqXW5I";
const BASE_URL = "https://www.googleapis.com/youtube/v3/";

const api = axios.create({
  baseURL: BASE_URL,
  params: {
    key: API_KEY,
    language: "en-US",
    include_adult: false,
  },
});

export default api;
```

Lookup.js

```
import React, { useState } from "react";

function Lookup({ onSearch }) {
  const [term, setTerm] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    onSearch(term);
  };

  return (
    <form
      onSubmit={handleSubmit}
      className="d-flex justify-content-center mb-4"
    >
      <input
        type="text"
        className="form-control me-2 rounded-pill w-50"
        placeholder="Search..."
        value={term}
        onChange={(e) => setTerm(e.target.value)}
      />
      <button className="btn btn-danger rounded-pill px-4" type="submit">
        Search
      </button>
    </form>
  );
}

export default Lookup;
```

Result.js

```
import React from "react";
import moment from "moment";
import { Card, Row, Col } from "react-bootstrap";

function Result({ results }) {
  if (!results || !Array.isArray(results) || results.length === 0) {
    return <p className="text-white">No results found.</p>;
  }

  const formatDuration = (isoDuration) => {
    const duration = moment.duration(isoDuration);
    let formatted = "";
    if (duration.hours()) formatted += duration.hours() + "h ";
    if (duration.minutes() || duration.hours()) formatted += duration.minutes() + "m ";
    formatted += duration.seconds() + "s";
    return formatted.trim();
  };

  const formatViewCount = (viewCount) => parseInt(viewCount).toLocaleString();
```

```

const timeSinceUpload = (publishedAt) => moment(publishedAt).fromNow();

const getVideoUrl = (videoId) => `https://www.youtube.com/watch?v=${videoId}`;

return (
  <Row>
    {results.map((item, idx) =>
      item?.snippet ? (
        <Col key={idx} md={6} lg={4} className="mb-4">
          <Card className="shadow rounded" style={{ background: "#ffffffdd" }}>
            {item.snippet.thumbnails?.medium?.url && (
              <Card.Img
                variant="top"
                src={item.snippet.thumbnails.medium.url}
                alt={item.snippet.title}
              />
            )}
            <Card.Body>
              <Card.Title>
                <a
                  href={getVideoUrl(item.id)}
                  target="_blank"
                  rel="noopener noreferrer"
                  style={{ textDecoration: "none", color: "#d63384" }}
                >
                  {item.snippet.title}
                </a>
              </Card.Title>
              <Card.Text>
                <strong>Channel:</strong> {item.snippet.channelTitle}
                <br />
                <strong>Views:</strong> {formatViewCount(item.statistics.viewCount)}
                <br />
                <strong>Duration:</strong> {formatDuration(item.contentDetails.duration)}
                <br />
                <strong>Uploaded:</strong> {timeSinceUpload(item.snippet.publishedAt)}
              </Card.Text>
            </Card.Body>
          </Card>
        </Col>
      ) : (
        <p key={idx}>Invalid item</p>
      )
    )}
  </Row>
);
}

export default Result;

```

useAPIresponse.js

```

import { useEffect, useState } from "react";
import api from "../APIInfo";
import moment from "moment";

const useAPIresponse = () => {
  const [results, setResults] = useState([]);

```

```

const [loading, setLoading] = useState(false);
const [error, setError] = useState(null);

const performSearch = async (term = "skateboarding dog") => {
  setLoading(true);
  setError(null);

  try {
    const searchResponse = await api.get("search", {
      params: {
        part: "snippet",
        q: term,
        type: "video",
        maxResults: 10,
      },
    });

    const videoItems = searchResponse.data.items || [];

    if (!videoItems.length) {
      setResults([]);
      setLoading(false);
      return;
    }

    const videoIds = videoItems.map((item) => item.id.videoId).join(",");

    const videosResponse = await api.get("videos", {
      params: {
        part: "snippet,statistics,contentDetails",
        id: videoIds,
      },
    });

    setResults(videosResponse.data.items || []);
  } catch (err) {
    console.error("YouTube API error:", err);
    setError(err);
    setResults([]);
  } finally {
    setLoading(false);
  }
};

useEffect(() => {
  performSearch();
}, []);

return { results, loading, error, performSearch };
};

export default useAPIresponse;

```