GOOGLE SEARCH ENGINE

- BY
- 1.ADITYA SINGH – 1905586
- 2.SAKSHAM GAUR –1905703
- 3.RAJ ARYAN GHOSE –1905866
- 4.KRISHIKA SINGH -1905889

INTRODUCTION

- We have developed a fully functional and responsive Google search engine clone from the scratch using React. This application allows you to do searches like the actual google search engine by the use of Google search API. The main purpose of this project is to learn how we can develop a software that can search for text in publicly accessible documents offered by Web server , it can be images or data contained in the database .
- A Search engine is a software accessed on the internet that searches a database for the information according the queries raised by user . There are many search engines which are currently in used like Bing , yahoo etc. The first search engine developed is Archie , currently Google is the most popular and well-known search engine which is written in Python , C,C ++. Search Engines act as a filter for the wealth of information available on the internet. It helps each one of us to meet daily information needs , solve problems , increase existing knowledge or gain knowledge about specific topic, and it is also used for the purpose of entertainment . In the era of globalization and invention we are moving from “God is answer” to “Google is answer”.

Technologies Used

- 1. **React :**It is a JavaScript library for building user interfaces . It is maintained by Meta and a community of individual developer and community .Basic knowledge of JavaScript,HTML,CSS.
- 2. **Context API:**It is an efficient way to pass data through the component tree. It is a react structure that enables us to exchange unique details and assists in solving prop-drilling(which is a situation when the same data is being sent almost every level due to requirements in the final level) from all levels of your application .
- 3. **Material UI:**It is a simple library that allows us to import and use different components to create UI in our react applications developed by Google .
- 4. **React Router: It** is a standard library for routing in React .It is rich with navigational components that compose declaratively with application which is useful for complex navigational requirements in React Application .
- 5. **Google Search API: It** is a custom JSON API that allows you to get web search results in form of JSON .
- 6. **Firebase: It** is a google platform that helps us create mobile and web application. It is a NoSQL database that lets us store and sync data users in real time.

Requirement Specifications

- The knowledge of React.Js is needed along with its libraries which will be very useful during the implementation of the project such as Material UI and ReactRouter.
- The project has been divided into 4 parts :

❑ Setting up the React App

❑ Building the search component

❑ The Search Result Page

❑ Deploying to Firebase

- Hardware Requirement:

❑ Processor: Intel Core i5

❑ RAM:16 GB

❑ Hard Disk Drive:512 GB

- Software Requirement:

❑ Tool: Visual Studio Code , JDK

❑ Language: React.js , JavaScript

Project Analysis

- Java should be installed in the system since without it node.js won’t be installed and without installing Node.Js npm will not run in the system .
- create-react-app creates and sets up a new React application starter without having to deal with webpack and Babel configurations.
- While installing Material UI one should use the --legacy-peer-deps
- flag at the end of npm install command which ignores all peer dependencies otherwise an error can occur
- In react-router-dom v6(which is the latest as of now), "Switch" is replaced by "Routes" according to latest version.
- rfce is again another way to speed up coding(as a lot of code is reused in React projects). It makes up a code snippet thus avoiding repetition of code.

SYSTEM DESIGN

Setting up the react app

1. Set up the React app
2. Installed and set up the react router
3. Creating Homepage header
4. Added and styled the logo

Component

1. Built UI for search Component
2. Added the search functionality and finished the homepage
3. Made the search component reusable
4. Implemented the global state management with React hooks and the Context API.

Creating Search Result Page

1. Set the search results Page
2. Set the google search API
3. Completed the search header.

Deploying to Firebase

Step 1: Setting up the react App

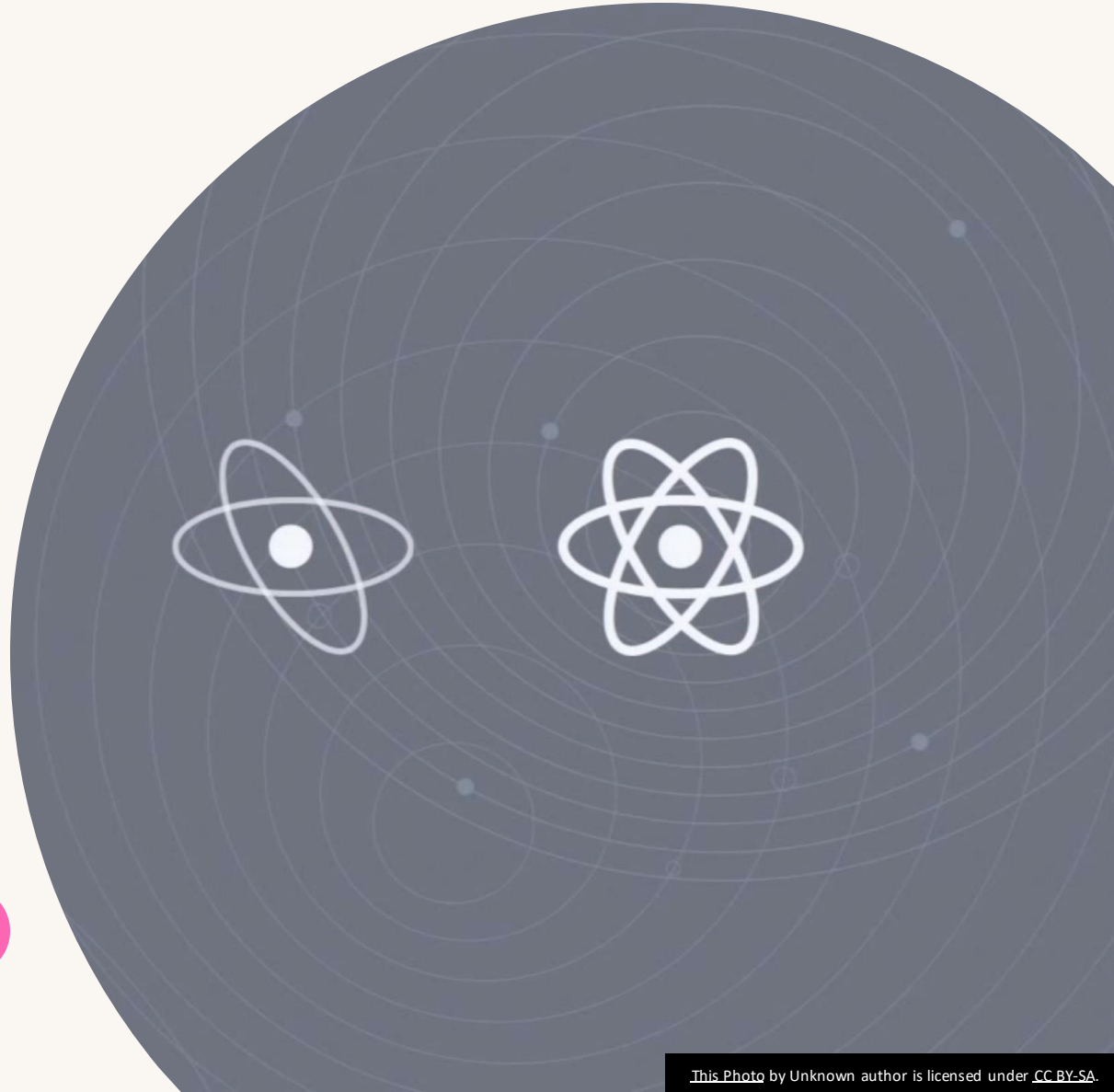
- We have divided it into 4 parts:

✓ Set up the react app- Here important thing to note is that Create react app : creates and sets up a new React application starter without having to deal with webpack and Babel configurations and npm start starts the application

✓ Setting up the Homepage –Here we created Home.js and Home.css files .The homepage will contain two main important components i.e Header and body

✓ Building Homepage Header-Here we wrapped up the entire application in the<router> component so that every component has access to the router and rendered the home component inside the <route path ="/"> component

✓ Add the homepage logo



Step 2:The Search Component

- Here we have divided this step into 4 parts :

❖ Building the search component :The reason for creating the search bar is to reuse this later in the SearchPage. Here we imported two icons from material UI that is one of the library of react.js

❖ Adding the Search Functionality :Here we have added some of the functionality of the search component and to track the input text whenever the user types something in the search bar , we have to make use of React App .

❖ Making Search Component reusable :To use the search component in the search page we have to add some code to hide the two buttons when the component is rendered in the search page.

❖ Context API +React Hooks :We implemented the global state management for our app with React Hooks (which is a great alternative for Redux) and Comtext API which is an efficient way to pass data through the component tree .

Step 3:The Search result page

- We have divided this step into 3 main substep :

✓ Set the search result page –Here we have created the js file of SearchPage and before setting up the search result page we did set up the Google Search API .

✓ Set the Google Search API-Her we created a js file (keys.js) where we stored the API key.

✓ Complete the Search Result Header

Step 4: Deploying to firebase

- We have divided this project into 2 parts :
- ✓ Complete the search page result :In this stage of development, we completed the application
- ✓ Firebase Setup : We finally deployed the application to firebase by first installing the firebase tools .

References

Search engine - Wikipedia

IEEE - The world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

React – A JavaScript library for building user interfaces (reactjs.org)

Firebase (google.com)

Context – React (reactjs.org)

Acknowledgement

- We are profoundly grateful to **PRABHU PRASAD DEV** of *School of Computer Engineering, KIIT, BBSR* for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

-----END-----