**PREREQUISITES:-**

* Node (nodejs.org)
* Knowledge on
  + HTML5
  + CSS3
  + JavaScript (ES6)

**Babel & WebPack :-**

* All the browsers suppoerts ES-5.
* We are using **ES-6** concepts
* Babel is a tranpiler that converts the new JS code to Older ones.
* Minify (CSS, SCSS, SASS, JavaScript)
* WebPack consists of two functionalities
  + Loaders
    - CSS-Loader
    - SASS-Loader
    - Babel-Loader
  + Plugins
* **Composition**
* *The process of combining multiple components together*

**Installation:**

node -v

npm -v

npm install create-react-app -g

create-react-app <project\_name>

npx create-react-app <project\_name>

**Components in React**

* Class components
* For representing a class component we have to use class keyword
* render() for returning a statement
* We have to aquire the properties from base class for implementing.
* Functional components
* We have to use function | ()=> for implementing a functional component
* Can return a statement directly
* High performance than class Component

**Styling components**

  // Inline styles

  <h2 style={{backgroundColor:"green"}}> Sample content </h2>`

  // Stying component using JavaScript object

  var style={

     backgroundColor: "reg",

     color: "#000"

   }

**React fragment**

*We have to use react fragments for avoiding number of divisions.*

   <React.Fragment>

    <Header />

    <h2> Sample heading </h2>

   </React.Fragment>

   or

   <>

      <Header />

      <h2> Hellow world </h2> <h2> Hi everyone</h2>

      <App />

    </>

**State in React**

*The concept of State provides a way to store the data in a component*

  constructor(){

    super();

    this.state={

      "name":"Hanuman",

      "role":"Full stack developer"

    }

  }

**Manipulating data from a state**

changeStateValues=()=>{

    this.setState({

      "name":"Rajesh",

      "role":"MERN developer"

    })

  }

  render(){

    return(

    <>

      <Header />

    <h2 onMouseOver={this.changeStateValues}> {this.state.name} working as a {this.state.role} </h2>

      <App />

    </>

    )}

**Using prevState for manipulating the data of a state:-**

  incrementCounter=()=>{

    this.setState(prevState=>(

      {counter: prevState.counter+1}

    ))

}

**Decrementing the value of the state:-**

 decrementCounter=()=>{

    if(this.state.counter>0){

    this.setState(prevState=>(

      {counter: prevState.counter-1}

    ))

    }

  }

**Using hooks concept for implementing state functionality in functional component**

 import React,{useState} from 'react';

// import Header from './Header'

function App(){

  var initialData=()=>{

    setData({

      name:"Hanuman",

      role:"Full stack developer"

    })

  }

  const [data,setData]=useState({"name":"Hanuman","role":"Full stack developer"})

  return(

    <>

      <h2 onMouseOver={()=>{setData({name:"Rajesh",role:"MERN developer"})}} onMouseOut={initialData}> {data.name} is working as a {data.role} </h2>

    </>

  )

}

export default App;

**HOOKS CONCEPT :-**

import React,{useState} from 'react';

function App() {

  const [data,setData]=useState({"name":"Hanuman","role":"Full stack developer"})

  return (

    <>

    <h2> {data.name} is working as a {data.role} </h2>

    </>

  )

}

export default App;

**Manipulate data :-**

import React,{useState} from 'react';

function App() {

  const [data,setData]=useState({"name":"Hanuman","role":"Full stack developer"})

  return (

    <>

    <h2 onMouseOver={()=>{setData({"name":"Rajesh","role":"MERN Developer"})}}> {data.name} is working as a {data.role} </h2>

    </>

  )

}

export default App;

**APPLYING DYNAMIC STYLES:-**

 import React,{useState} from 'react';

// import Header from './Header'

import './App.css';

function App(){

  var style={

    background:"green"

  }

  var changeBackground=(e)=>{

    e.target.setAttribute('class','changeBackground')

  }

  const [data,setData]=useState({"name":"Hanuman","role":"Full stack developer"})

  return (

    <>

      <h2 onMouseOver={(e)=>{changeBackground(e)}}> {data.name} is working as {data.role}.</h2>

    </>

  )

}

export default App;

**DATA MANIPULATE:-**

import React,{useState} from 'react';

function App() {

  var changeData=()=>{

    setData({

      name:"Rajesh",

      role:"MERN developer"

    })

  }

 const [data,setData]=useState({"name":"Hanuman","role":"Full stack developer"})

  return (

    <>

    <h2 onMouseOver={changeData}>{data.name} => {data.role}</h2>

    </>

  )

}

export default App;

**App.js Header.js**

|  |  |
| --- | --- |
| import logo from './logo.svg';  import './App.css';  import Header from './Header'  function App(){      return(      <>      <h2> MY NAME RAJA </h2>      <Header name="raja" role="FRESHER" />      </>    )  }  export default App; | let Header=(props)=>{      return(          <header>          <h2> {props.name} working as a {props.role}</h2>          </header>      )  }  export default Header; |

**REGULAR EXPRESSION FOR EMAIL VALIDATIONS:-**

 import React,{useState} from 'react';

export default function FormInput(){

    function testMobile(e){

        var reg=/^\S+@\S+[.]\S{1,}$/

        if(reg.test(e.target.value)){

            setContent({

                background:"green",

                text:"Valid Email id"

            })

        } else {

            setContent({

                background:"red",

                text:"Invalid Email id"

            })

        }

    }

    var [content, setContent]=useState({background:"#fff",text:"hi"})

    return(

        <div>

            <article style={{background:content.background, textAlign:"center"}}> {content.text} </article> <br />

            <form>

                <input type="text" placeholder="Email id" onKeyUp={(event)=>{testMobile(event)}}/>

            </form>

        </div>

    )

}

# ROUTER:-

## npm install react-router-dom –save

**APP.JS**

|  |
| --- |
| import React,{useState} from 'react';  import './App.css';  import {BrowserRouter,Route,Link,Switch} from 'react-router-dom';  import Sample from './Sample';  import {profiles} from './data.json';  function App(){   return(      <>         <BrowserRouter>           <Route exact path='/' component={Home}/>           <Route exact path='/sample' component={Sample}/>          </BrowserRouter>     </>    )  }  let Home=()=>{    return(      <article className="parent">      {profiles.map((item,index)=>(        <div className="child" key={index}>        <h2> {item.name} </h2>        <h4>{item.role} </h4>        <a href={"mailto"+item.email}> {item.email} </a>        <link className="button" to= {{pathname:'/sample'}}> view profile </link>        </div>      ))}        </article>    )  }  export default App; |
| **App.js**  .parent{    display: flex;    flex-flow: row wrap;    justify-content: center;  }  .child{    width: 40%;    border: 1px solid red;    padding: 1%;    margin: 1%;    box-shadow: 0px 8px 8px -8px;    text-align: center;  }  .child h2{    color: #18379d;  }    .button{    display: block;    background: blue;    margin: 1%;    color: #fff;    padding: 1%;    text-decoration: none;} |

**Data.json**

|  |
| --- |
| {      "profiles":[          {               "name": "raja",               "role" : "b.tech",               "email": "drajasekhar21@gmail.com",               "mobile": "9160232535"          },{                "name": "nagendra",               "role" : "diploma",               "email": "venkatanagendra70@gmail.com",               "mobile": "9705960022"          }      ]  } |

**Sample.js**

|  |
| --- |
| export default function Sample() {      return (  <>      <h2> this is sample component </h2>  </>      )    } |

# To-do-List :-

* We need to create an input field along with the button
* We have to create an empty array and empty string in the state.

**state={**

**formInfo:” ”.**

**data: [ ]**

**}**

* We have to store the information temporarily which we are getting from the input field. We have to do this by an onkeyup event.
* When we are gonna click on the button, we’ve to store the information from the input field to the array of state
* We’ve to render the information in the view part.

### Live search from JSON:-

import React from 'react';

import './App.css'

import {profiles} from './data.json'

export default class Search extends React.Component{

    constructor(){

        super();

        this.state={

            search:""

        }

    }

    searchData=(e)=>{

        this.setState({search: e.target.value})

    }

    render(){

        const {search} =this.state;

        console.log(search)

        const filteredNames=profiles.filter(profileInfo=>{

            return profileInfo.name.toLowerCase().indexOf(search.toLowerCase())!==-1

        })

        return(

            <div className="container">

                <input type="text" placeholder="Search..." onKeyUp={(event)=>{this.searchData(event)}}/>

                <div className="parent">

                    {

                        filteredNames.map((item,index)=>(

                            <div className="child" key={index}>

                                {item.name}

                             </div>

                        ))

                    }

                </div>

            </div>

        )

    }

}