**MODULE: 9 ReactJs Intro**

**Q1- What is React Js?**

**Ans-** React is a JavaScript library used for building user interfaces, especially for single-page applications where data can change over time without reloading the page. It's maintained by Facebook and a community of developers.

**Q2- What is NPM in React Js?**

**Ans-** NPM (Node Package Manager) is a package manager for Node.js that allows developers to install, manage, and share reusable JavaScript code/packages. It's often used to manage dependencies in React projects.

**Q3- What is Role of Node Js in react Js?**

**Ans-** Node.js is used as a server-side runtime for JavaScript. In React, Node.js can be utilized to set up a development environment, run scripts, manage dependencies (via NPM), and perform server-side rendering.

**Q4- What is CLI command In React Js?**

**Ans-** CLI (Command-Line Interface) commands in React JS allow developers to interact with the React application via the command line. For instance, creating a new React app, running development servers, building production-ready code, etc.

**Q5- What is Components in React Js?**

**Ans-** Components are the building blocks of React applications. They're independent and reusable pieces of code that can contain HTML, JavaScript, CSS, and more. These components can be nested to create complex UIs.

**Q6- What is Header and Content Components in React Js?**

**Ans-** In a React application, Header and Content components could be parts of the UI layout. For instance, the Header component might contain navigation elements, logos, etc., while the Content component could hold the main content of the page.

**Q7- How to install ReactJs on Windows, Linux Operating System? How to Install NPM and How to check version of NPM?**

**Ans-**To install React on Windows or Linux, you'd typically use Node Package Manager (NPM). Here are the basic steps:

1. Install Node.js (which includes NPM) from the official website.
2. Open a terminal (Command Prompt in Windows, Terminal in Linux).
3. Use npx create-react-app my-app to create a new React app named 'my-app'.
4. Change directory to your app: cd my-app.
5. Start the development server: npm start.

**Q8- How to check version of React Js?**

**Ans-** To check the versions:

For NPM: npm -v

For React: In your project folder, check the package.json file or run npm list react.

**Q9- How to change in components of React Js?**

**Ans-** To modify components in React, we'll typically edit the JavaScript files representing those components. These files contain the logic and structure of your UI elements. Any changes made to these files will reflect in your app's UI.

**Q10- How to Create a List View in React Js?**

**Ans-**

**Jsx:**

import React, { useState } from "react";

export default function Listview() {

  const [questions, setQuestions] = useState([

    // create state for list view

    {

      id: 1,

      question: "THE 'REACT WAY' TO RENDER LIST ",

      options: [

        "Use Array.map",

        "Not a for loop",

        "Give each item a unique key",

        "Avoid using array index as the key",

      ],

      // correctAnswer: ["Use Array.map"],

      selectedOptions: [], //For select Some options

    },

  ]);

  // For choose options

  const handleOptionSelect = (questionId, option) => {

    setQuestions((prevQuestions) =>

      prevQuestions.map((q) => {

        if (q.id === questionId) {

          const selectedOptions = q.selectedOptions.includes(option)

            ? q.selectedOptions.filter((selected) => selected !== option)

            : [...q.selectedOptions, option];

          return { ...q, selectedOptions };

        } else {

          return q;

        }

      })

    );

  };

  return (

    <>

      <div className="list-view">

        {questions.map((q) => (

          <div key={q.id}>

            <h1 className="Question">{q.question}</h1>

            <ul>

              {q.options.map((option) => (

                <li key={option}>

                  <label className="custom-checkbox-label">

                    {/\* For checkbox \*/}

                    <input

                      className="custom-checkbox-input"

                      type="checkbox"

                      name={`question\_${q.id}`}

                      value={option}

                      checked={q.selectedOptions.includes(option)}

                      onChange={() => handleOptionSelect(q.id, option)}

                    />

                    {/\* for options \*/}

                    <span className="custom-checkbox-text">{option}</span>

                  </label>

                </li>

              ))}

            </ul>

          </div>

        ))}

      </div>

    </>

  );

}

**App.css:**

/\* list view \*/

body {

  background-color: rgb(126, 233, 97);

}

li {

  list-style: none;

}

.Question{

  text-align: center;

}

.list-view {

  position: relative;

  top: 100px;

}

.custom-checkbox-label {

  display: flex;

  align-items: center;

}

.custom-checkbox-input {

  position: relative;

  left: 30%;

  margin: 5px;

  width: 30px;

  height: 30px;

  border-radius: 50%;

  vertical-align: middle;

  background: gainsboro;

  border: 1px solid gainsboro;

  appearance: none;

  -webkit-appearance: none;

  outline: none;

  cursor: pointer;

}

.custom-checkbox-input:checked {

  appearance: auto;

  clip-path: circle(50% at 50% 50%);

  background-color: blue;

}

.custom-checkbox-text {

  position: relative;

  left: 30%;

  height: 30px;

  width: 400px;

  text-align: left;

  font-size: 18px;

  border: 1px solid #fff;

  padding: 2px 5px;

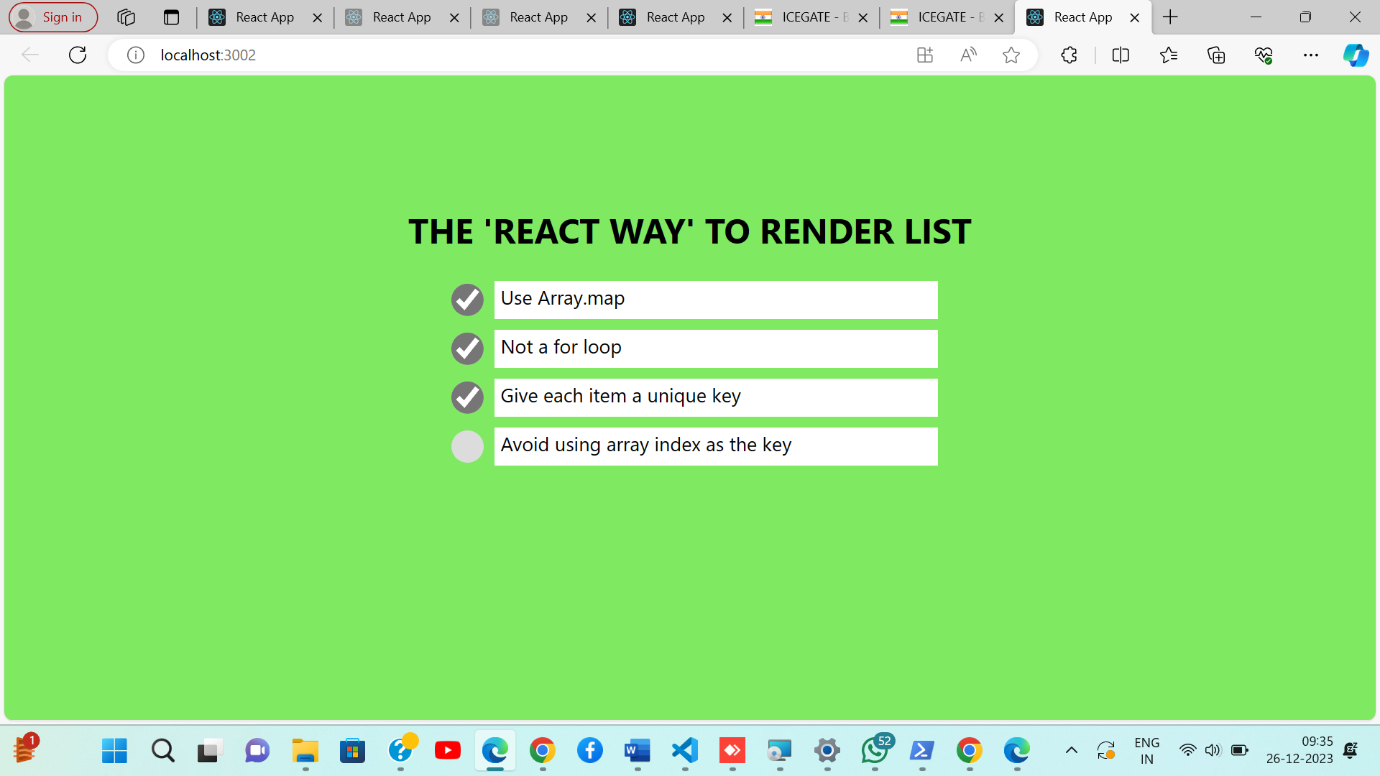
  margin: 5px;

  background: #fff;

}

/\* ------ Listview.css end ------ \*/

**Output:**

****

**Q11- Create Increment decrement state change by button click?**

**Ans-**

**Jsx:**

import React, { useState } from 'react';

function Counter () {

  const [count, setCount] = useState(0);

  return (

    <>

    <div className='main container'>

        <div className="section">

        <h2>React Web</h2>

        <p>{count}</p>

         <button onClick={() => setCount((c) => c+1)}>Increment</button>

      <button className='btn2' onClick={() => setCount((c) => c-1)}>Decrement</button> <br />

      <button onClick={() => setCount(0)}>Reset</button>

    </div>

    </div>

    </>

  );

};

export default Counter;

**App.css**

/\* Counter css \*/

\*{

  background-color: rgb(81, 81, 235);

}

h2{

  color: white;

}

button{

  background-color: rgb(60, 60, 60);

  color: white;

  padding: 10px;

  margin-bottom: 20px;

}

.btn2{

  margin-left: 10px;

}

.main{

  display: flex;

  position: relative;

  align-items: center;

  justify-content: center;

}

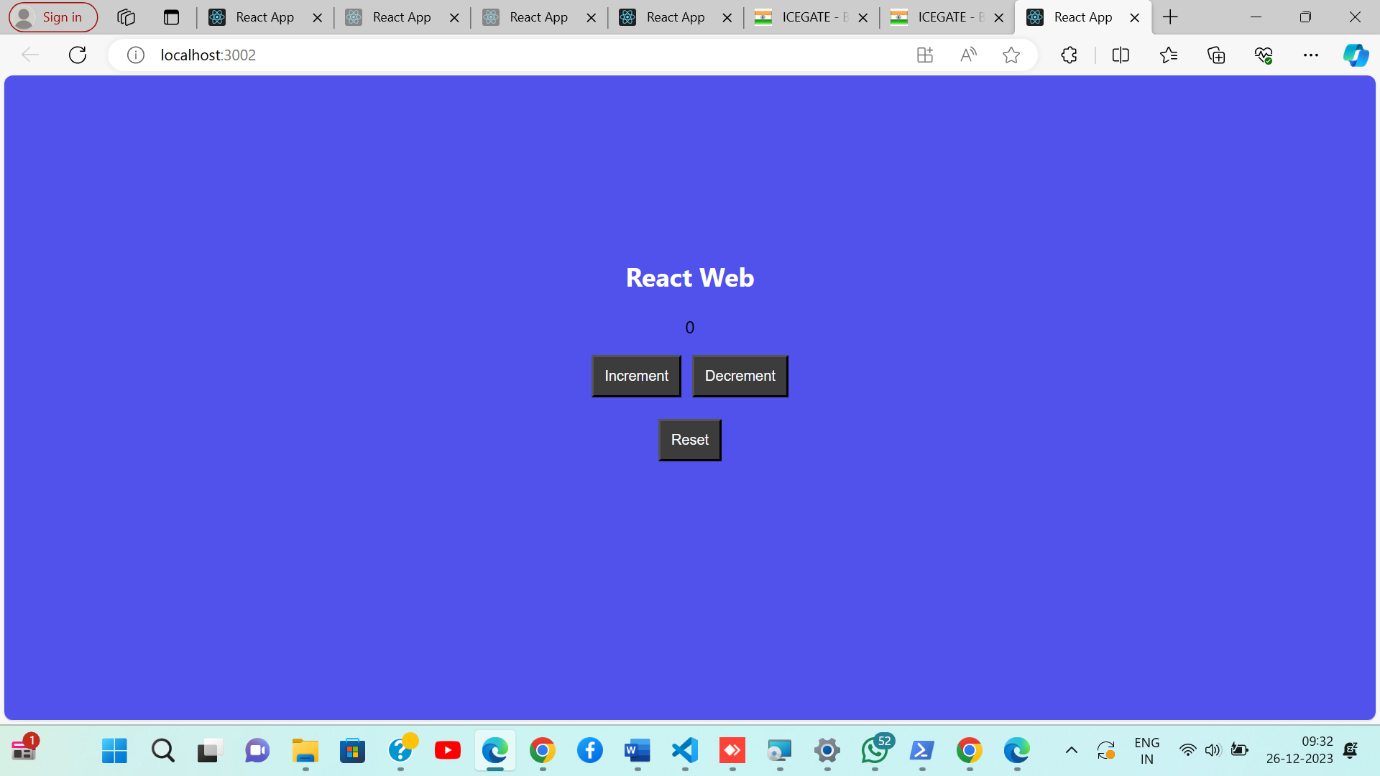
.section{

 align-content: center;

 padding-top: 150px;

 text-align: center;

}

**Output:**