

# VOC ASSIGNMENT-04(MAJOR PROJECT):-

## Project Title: QUIZ APP

### Introduction:

The Quiz Application is a web-based interactive platform built using HTML, CSS, JavaScript, and React.js. The primary objective of this project is to create a responsive and user-friendly quiz system where users can test their knowledge on various topics such as web development and React.js.

### Implementation:

Tools used: **React, HTML, CSS, Javascript**

=====package.json=====

```
{
  "name": "quizapp",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/dom": "^10.4.1",
    "@testing-library/jest-dom": "^6.7.0",
    "@testing-library/react": "^16.3.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^19.1.1",
    "react-dom": "^19.1.1",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4"
  },
```

```

"scripts": {
  "start": "react-scripts start",
  "build": "react-scripts build",
  "test": "react-scripts test",
  "eject": "react-scripts eject"
},
"eslintConfig": {
  "extends": [
    "react-app",
    "react-app/jest"
  ]
},
"browserslist": {
  "production": [
    ">0.2%",
    "not dead",
    "not op_mini all"
  ],
  "development": [
    "last 1 chrome version",
    "last 1 firefox version",
    "last 1 safari version"
  ]
}
}

```

===== public/index.html =====

```

<!DOCTYPE html>

<html lang="en">

```

```
<head>

  <meta charset="utf-8" />

  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <meta name="theme-color" content="#000000" />

  <meta name="description" content="Web site created using create-react-app" />

  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

  <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

  <title>React App</title>

</head>

<body>

  <noscript>You need to enable JavaScript to run this app.</noscript>

  <div id="root"></div>

</body>

</html>
```

===== [src/index.js](#) =====

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

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

```
reportWebVitals();
```

```
=====src/App.js=====
```

```
import './App.css';
```

```
import Quiz from './Component/Quiz';
```

```
function App() {
```

```
  return (
```

```
    <div className="App">
```

```
      <Quiz />
```

```
    </div>
```

```
  );
```

```
}
```

```
export default App;
```

```
=====src/Quiz.js=====
```

```
import React, { useState } from 'react'
```

```
import QuestionList from './QuestionList'
```

```
import QuizCSS from './Quiz.css'
```

```
export default function Quiz() {
```

```
  const questions = [
```

```
    {
```

```
      question: " A What is React?",
```

```
      options: ['1 A back-end framework', '2 A front-end JavaScript library', '3 A database', '4 A compiler'],
```

```
      answer: "A front-end JavaScript library"
```

```
    },
```

```
    {
```

```

    question: "What is state in React?",
    options: ['A built-in database', 'A way to store component data dynamically', 'A CSS style', 'A function'],
    answer: "A way to store component data dynamically"
  },
  {
    question: "What is the Virtual DOM in React?",
    options: ['A real DOM', 'A copy of the DOM for efficient updates', 'A server-side DOM', 'A JSON file'],
    answer: "A copy of the DOM for efficient updates"
  },
  {
    question: "What is the purpose of key in React lists?",
    options: ['To style the list', 'To uniquely identify elements for rendering efficiency', 'To set state', 'To create routing'],
    answer: "To uniquely identify elements for rendering efficiency",
  },
  {
    question: "What are props in React?",
    options: ['Functions to manage state', 'Arguments passed to components', 'Hooks in React', 'CSS styling'],
    answer: "Arguments passed to components",
  }
]

const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0);
const [currentAnswer, setCurrentAnswer] = useState(null);
const [score, setScore] = useState(0);
const handleClick = (option) => {
  setCurrentAnswer(option)
  if (option === questions[currentQuestionIndex].answer) {

```

```

        setScore(score + 1)
      }
    }
    const handleNextQuestion = () => {
      setCurrentQuestionIndex(currentQuestionIndex + 1);
      setCurrentAnswer(null);
    }
    return (
      <div>
        {currentQuestionIndex < questions.length ? <div>
          <QuestionList question={questions[currentQuestionIndex].question}
            options={questions[currentQuestionIndex].options} handleClick={handleClick}
            currentAnswer={currentAnswer} />
          <button disabled={currentAnswer === null} className={currentAnswer === null ? 'button-
            disable' : 'button'} onClick={handleNextQuestion}>Next Question</button>
          </div> : <div>Your Score is {score}</div>
        }
      </div>
    )
  }
}

```

=====QuestionList.js=====

```

export default function QuestionList({ question, options, handleClick, currentAnswer }) {
  return (
    <div>
      <h2>{question}</h2>
      <ul>
        {options.map((option, index) => (
          <li key={index} onClick={() => handleClick(option)} className={currentAnswer == option ?
            'selected' : ''}>{option}</li>
        ))}
      </ul>
    </div>
  )
}

```

```
        </ul>
    </div>
);
}
```

=====Quiz.css=====

```
.App {
text-align: center;
font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
background: linear-gradient(135deg, #e3f2fd, #f9f9f9);
min-height: 100vh;
display: flex;
justify-content: center;
align-items: center;
padding: 20px;
}
```

```
.quiz-container {
background: #ffffff;
padding: 40px;
border-radius: 16px;
box-shadow: 0px 8px 20px rgba(0, 0, 0, 0.1);
max-width: 600px;
width: 100%;
text-align: center;
}
```

```
h2 {
font-size: 24px;
margin-bottom: 20px;
```

```
    color: #222;  
}
```

```
ul {  
    list-style-type: none;  
    padding: 0;  
    margin: 20px 0;  
}
```

```
li {  
    margin: 12px 0;  
    padding: 14px 18px;  
    cursor: pointer;  
    background-color: #f4f4f4;  
    border-radius: 10px;  
    font-size: 16px;  
    transition: all 0.3s ease;  
    border: 1px solid transparent;  
}
```

```
li:hover {  
    background-color: #e1f5fe;  
    transform: scale(1.02);  
    border: 1px solid #29b6f6;  
}
```

```
.selected {  
    background-color: #b3e5fc;  
    border: 1px solid #0288d1;  
    font-weight: bold;  
    color: #01579b;
```



```
}
```

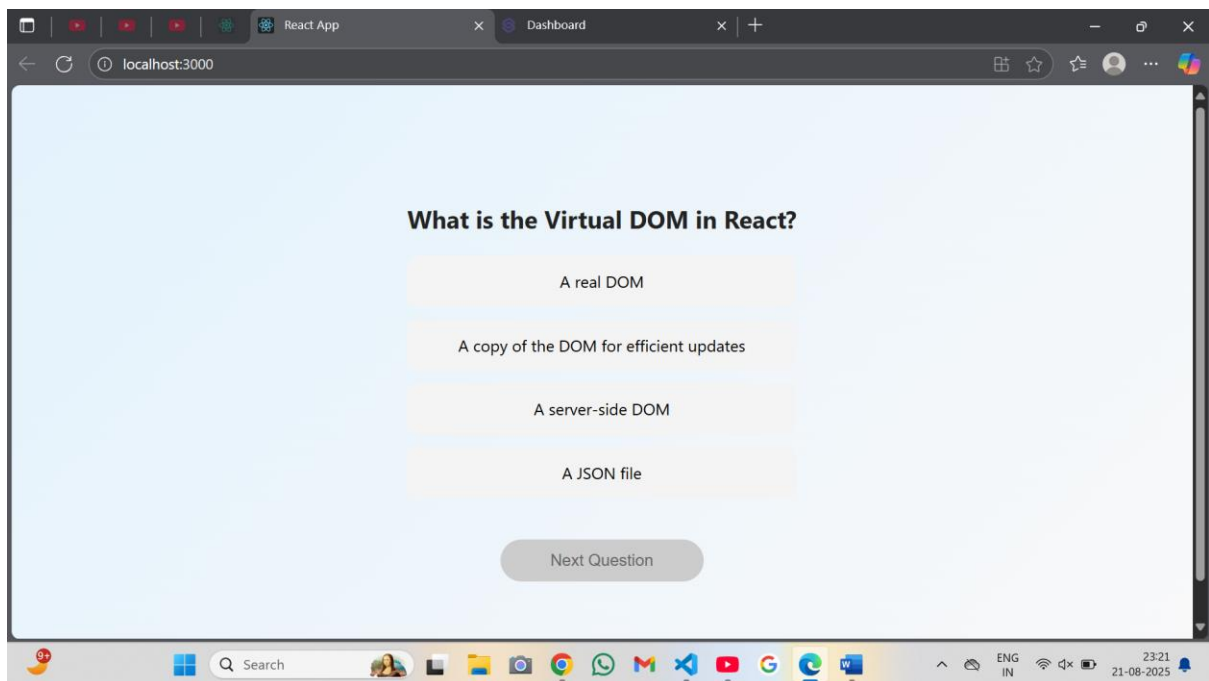
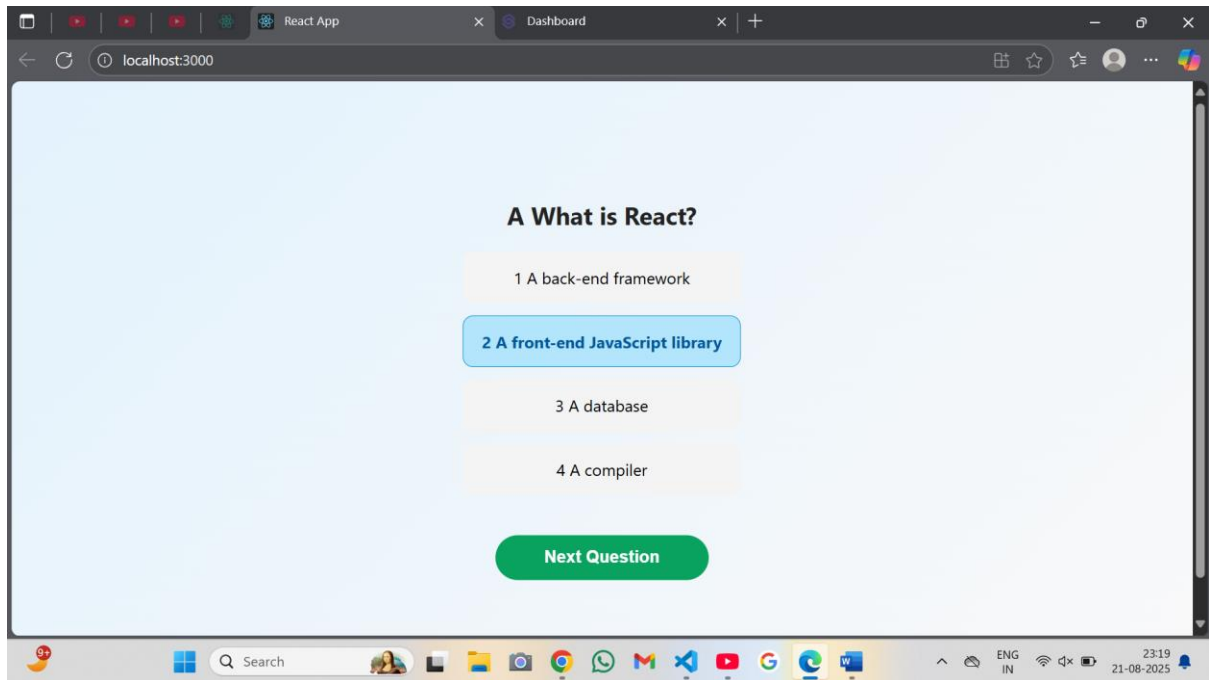
```
.button {  
    background-color: #0dbc72;  
    border-radius: 25px;  
    width: 200px;  
    padding: 12px;  
    font-size: 16px;  
    font-weight: bold;  
    color: white;  
    border: none;  
    cursor: pointer;  
    margin-top: 20px;  
    transition: background 0.3s ease, transform 0.2s ease;  
}
```

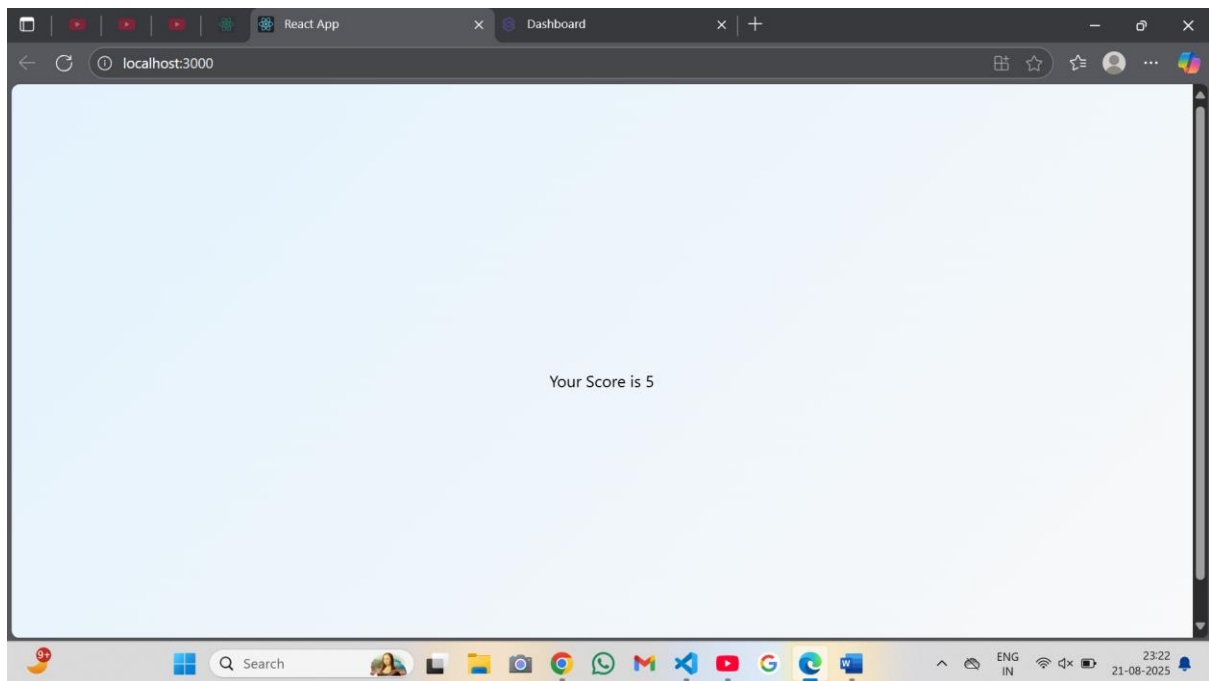
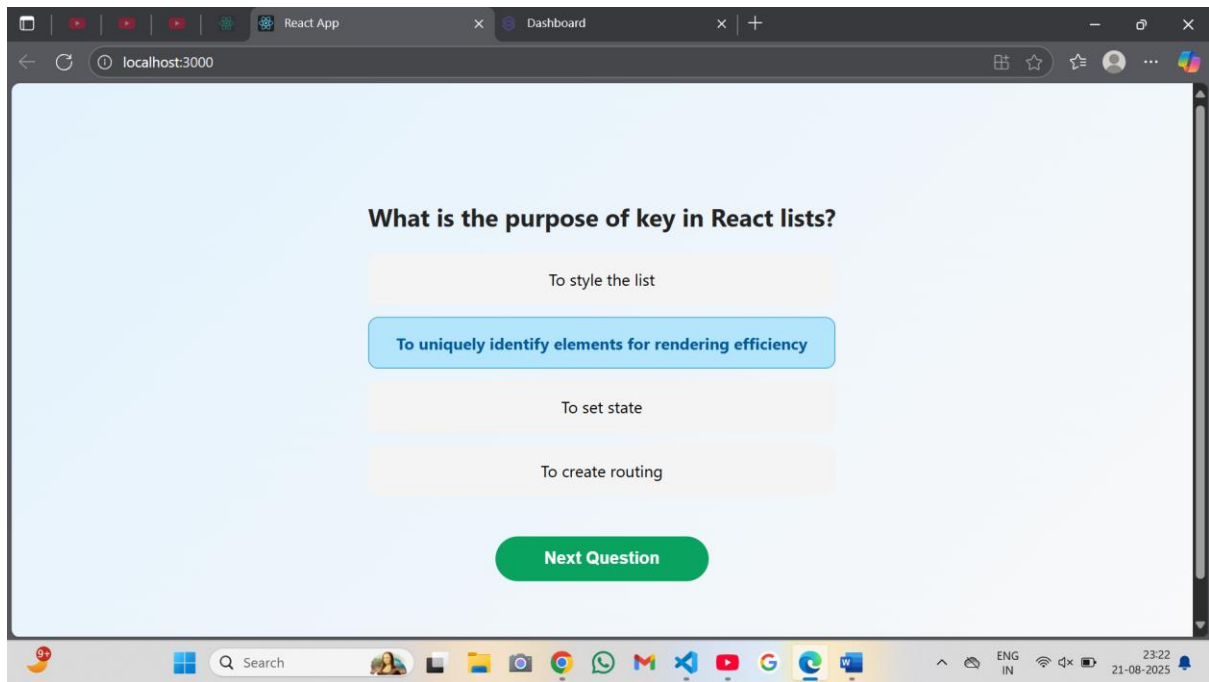
```
.button:hover {  
    background-color: #0aa25f;  
    transform: scale(1.05);  
}
```

```
.button-disable {  
    background-color: #ccc;  
    border-radius: 25px;  
    width: 200px;  
    padding: 12px;  
    font-size: 16px;  
    color: #666;  
    border: none;  
    cursor: not-allowed;  
    margin-top: 20px;
```

}

=====output screenshot:=====





**Demo Video Link: <C:\Users\shris\Videos\Captures\React App and 6 more pages - Personal - Microsoft Edge 2025-08-21 23-29-47.mp4>**

