

Hooks, Routers and MongoDB CRUD

Aim -

To understand about React Hooks and React Router, perform basic CRUD operations in MongoDB.

Task 1 – React Hooks

Write code to understand different hooks in React JS.

Code –

useStateDemo.js –

```
import React, { useState } from 'react';
function Counter() {
  const [count, setCount] = useState(0);
  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
export default Counter;
```

Output –

You clicked 4 times

Click me

useEffectDemo.js –

```
import React, { useState, useEffect } from 'react';
function Timer() {
  const [seconds, setSeconds] = useState(0);
  useEffect(() => {
    const interval = setInterval(() => {
      setSeconds(s => s + 1);
    }, 1000);
    return () => clearInterval(interval);
  }, []);
  return <p>Timer: {seconds}s</p>;
}
export default Timer;
```

Output -

Timer: 10s

useRefDemo.js –

```
import React, { useRef } from 'react';
function FocusInput() {
  const inputRef = useRef(null);
  const handleClick = () => {
    inputRef.current.focus();
  };
  return (
    <div>
      <input ref={inputRef} type="text" />
      <button onClick={handleClick}>Focus the input</button>
    </div>
  );
}
export default FocusInput;
```

Output –**ThemeContext.js –**

```
import React, { createContext } from 'react';
export const ThemeContext = createContext('grey');
```

useContext.js –

```
import React, { useContext } from 'react';
import { ThemeContext } from './ThemeContext';
function UseContextDemo() {
  const theme = useContext(ThemeContext);
  return (
    <div style={{ backgroundColor: theme, padding: '10px' }}> <h3>useContext Hook</h3>
    <p>Theme: {theme}</p>
  </div>
  );
}
export default UseContextDemo;
```

Output –**useContext Hook**

Theme: grey

Task 2 – React Routing

Write a program to demonstrate routing in React.

Code –**Blogs.js -**

```
const Blogs = () => {  
  return <h1>Blog Articles</h1>;  
};  
export default Blogs;
```

Contact.js –

```
const Contact = () => {  
  return <h1>Contact Me</h1>;  
};  
export default Contact;
```

HomePage.js –

```
const Home = () => {  
  return <h1>Home</h1>;  
};  
export default Home;
```

NoPage.js –

```
const NoPage = () => {  
  return <h1>404</h1>;  
};  
export default NoPage;
```

index.js –

```
import ReactDOM from "react-dom/client";  
import { BrowserRouter, Routes, Route } from "react-router-dom";  
import Layout from "../pages/Layout_link";  
import Home from "../pages/HomePage";  
import Blogs from "../pages/Blogs";
```

```
import Contact from "../pages/Contact";
import NoPage from "../pages/NoPage";

export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route index element={<Home />} />
          <Route path="blogs" element={<Blogs />} />
          <Route path="contact" element={<Contact />} />
          <Route path="*" element={<NoPage />} />
        </Route>
      </Routes>
    </BrowserRouter>
  );
}
```

Output –



Task 3 – MongoDB CRUD

Write a program to demonstrate React Class Components.

Create database and Collection –

```
test> use college
switched to db college

college> db.createCollection("students")
{ ok: 1 }
```

Inserting records (Create) –

```
college> db.students.insertMany([
...   { rno: 101, name: "Aarav Mehta", branch: "CSE", gpa: 8.6 },
...   { rno: 102, name: "Ishita Sharma", branch: "ECE", gpa: 9.1 },
...   { rno: 103, name: "Raj Patel", branch: "Mechanical", gpa: 7.4 },
...   { rno: 104, name: "Sneha Reddy", branch: "CSE", gpa: 8.9 },
...   { rno: 105, name: "Vikram Joshi", branch: "Civil", gpa: 6.8 }
... ]);
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6804ad822241144a0db71236'),
    '1': ObjectId('6804ad822241144a0db71237'),
    '2': ObjectId('6804ad822241144a0db71238'),
    '3': ObjectId('6804ad822241144a0db71239'),
    '4': ObjectId('6804ad822241144a0db7123a')
  }
}
```

Finding records (Read) –

```
college> db.students.find({ gpa: { $gte: 8.5 } });
...
[
  {
    _id: ObjectId('6804ad822241144a0db71236'),
    rno: 101,
    name: 'Aarav Mehta',
    branch: 'CSE',
    gpa: 8.6
  },
  {
    _id: ObjectId('6804ad822241144a0db71237'),
    rno: 102,
    name: 'Ishita Sharma',
    branch: 'ECE',
    gpa: 9.1
  },
  {
    _id: ObjectId('6804ad822241144a0db71239'),
    rno: 104,
    name: 'Sneha Reddy',
    branch: 'CSE',
    gpa: 8.9
  }
]
```

Updating records (Update) –

```
college> db.students.updateOne(  
...   { name: "Vikram Joshi" },  
...   { $set: { branch: "Architecture" } }  
... );  
...  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

Deleting records(Delete) –

```
college> db.students.deleteOne({rno:102});  
{ acknowledged: true, deletedCount: 1 }
```

