VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of the UGC Act, 1956)

Programme	:	**B.Tech.**	Semester	:	**Winter24-25**
Course	:	**BCSE203E: Web Programming Lab**	Slot	:	**TE1/TE2**
Faculty	:	**Dr. LM Jenila Livingston**	Marks	:	**10**

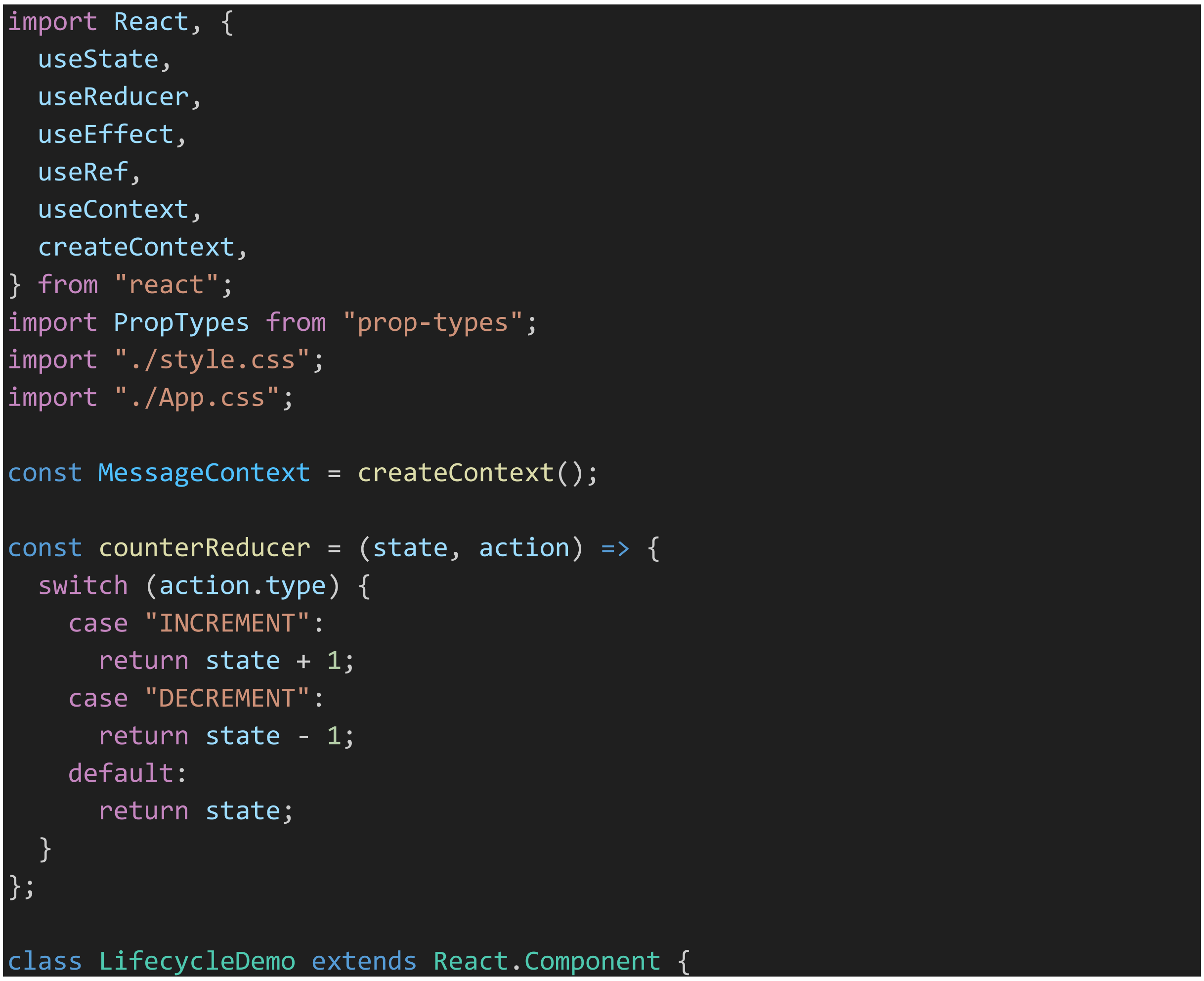
Name: Sara Arora

Register No. 23BCE5036

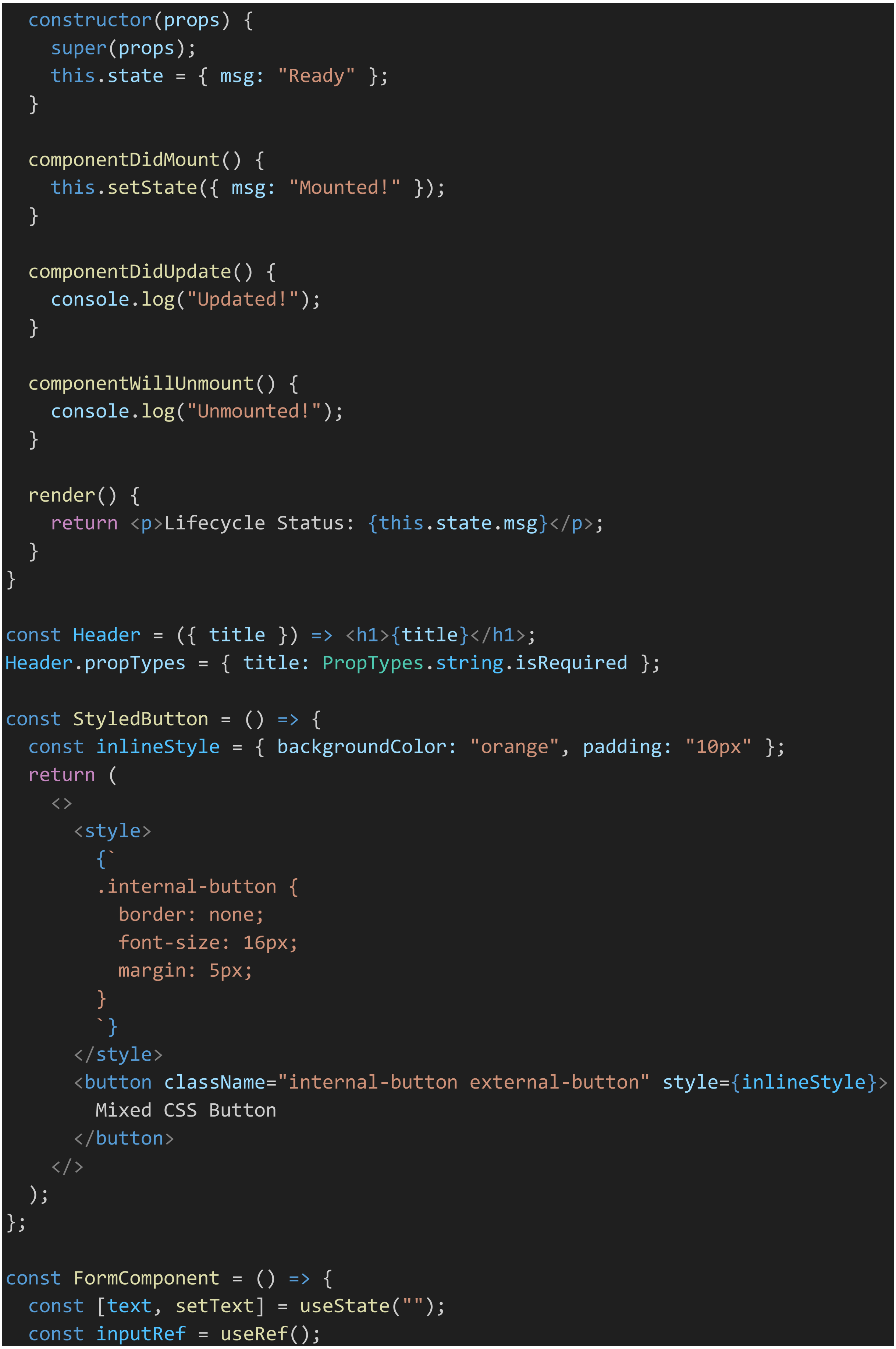
Exercise-15

Code:

App.js



```
import React, {
  useState,
  useReducer,
  useEffect,
  useRef,
  useContext,
  createContext,
} from "react";
import PropTypes from "prop-types";
import "./style.css";
import "./App.css";

const MessageContext = createContext();

const counterReducer = (state, action) => {
  switch (action.type) {
    case "INCREMENT":
      return state + 1;
    case "DECREMENT":
      return state - 1;
    default:
      return state;
  }
};

class LifecycleDemo extends React.Component {
```

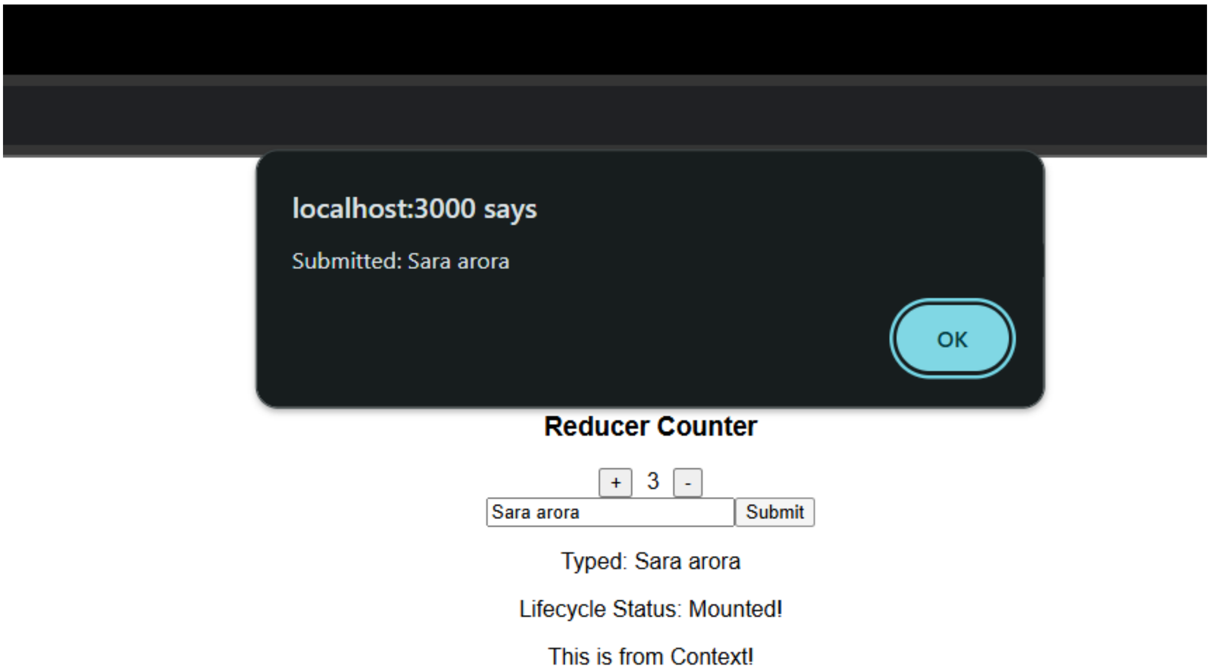


```
  constructor(props) {
    super(props);
    this.state = { msg: "Ready" };
  }

  componentDidMount() {
    this.setState({ msg: "Mounted!" });
  }

  componentDidUpdate() {
    console.log("Updated!");
  }

  componentWillUnmount() {
    console.log("Unmounted!");
  }

  render() {
    return <p>Lifecycle Status: {this.state.msg}</p>;
  }
}

const Header = ({ title }) => <h1>{title}</h1>;
Header.propTypes = { title: PropTypes.string.isRequired };

const StyledButton = () => {
  const inlineStyle = { backgroundColor: "orange", padding: "10px" };
  return (
    <>
      <style>
        {`
        .internal-button {
          border: none;
          font-size: 16px;
          margin: 5px;
        }
        `}
      </style>
      <button className="internal-button external-button" style={inlineStyle}>
        Mixed CSS Button
      </button>
    </>
  );
};

const FormComponent = () => {
  const [text, setText] = useState("");
  const inputRef = useRef();
  const handleSubmit = (e) => {
    e.preventDefault();
    alert(`Submitted: ${inputRef.current.value}`);
  };

  return (
    <form onSubmit={handleSubmit}>
      <input
        ref={inputRef}
        type="text"
        placeholder="Enter something"
        onChange={(e) => setText(e.target.value)}
      />
      <button type="submit">Submit</button>
      <p>Typed: {text}</p>
    </form>
  );
};

const ContextDisplay = () => {
  const msg = useContext(MessageContext);
  return <p>{msg}</p>;
};

const App = () => {
  const [joke, setJoke] = useState("");
  const [count, dispatch] = useReducer(counterReducer, 0);

  useEffect(() => {
    console.log("App Mounted");
  }, []);

  const getJoke = async () => {
    const res = await fetch("https://official-joke-api.appspot.com/random_joke");
    const data = await res.json();
    setJoke(`${data.setup} - ${data.punchline}`);
  };

  return (
    <MessageContext.Provider value="This is from Context!">
      <div className="app">
        <Header title="React All-in-One Demo" />
        <StyledButton />
        <button onClick={getJoke}>Fetch Joke</button>
        <p>{joke}</p>
        <h3>Reducer Counter</h3>
        <button onClick={() => dispatch({ type: "INCREMENT" })}>+</button>
        <span style={{ margin: "0 10px" }}>{count}</span>
        <button onClick={() => dispatch({ type: "DECREMENT" })}>-</button>
        <FormComponent />
        <LifecycleDemo />
        <ContextDisplay />
      </div>
    </MessageContext.Provider>
  );
};

export default App;
```

<u>Style.css</u>

```
.external-button {
    color: white;
    background-color: green;
    padding: 10px;
    border-radius: 5px;
  }

  .app {
    font-family: sans-serif;
    text-align: center;
  }
```

Output:



localhost:3000 says

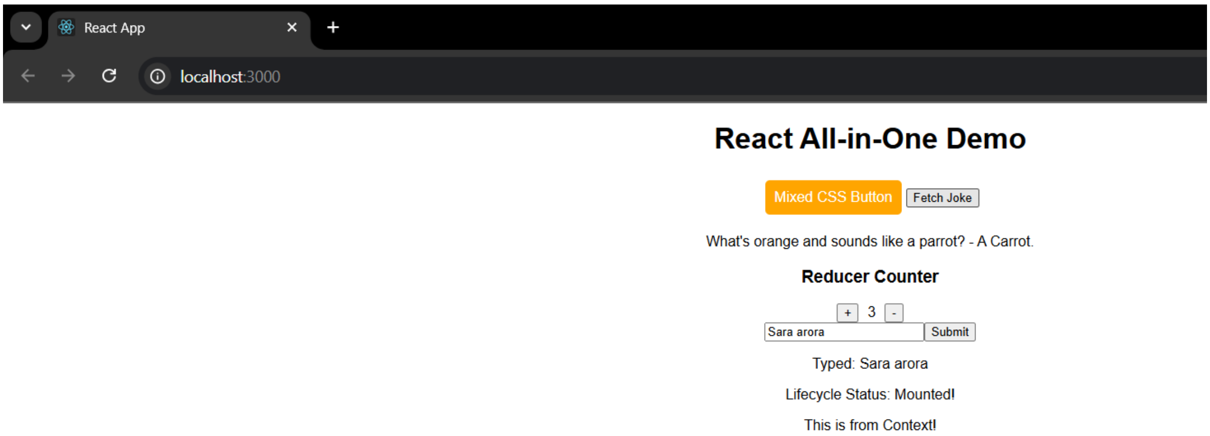
Submitted: Sara arora

Reducer Counter

Typed: Sara arora

Lifecycle Status: Mounted!

This is from Context!



React All-in-One Demo

What's orange and sounds like a parrot? - A Carrot.

Reducer Counter

Typed: Sara arora

Lifecycle Status: Mounted!

This is from Context!