**Web Programming Lab**

**Register Number:** 23BCE1777

**Name:** Madamsetty Sai Venkat Akshat

**Subject Code:** BCSE203E

**Subject Name:** Web Programming Lab

**Slot:** TE1/TE2

**Exercise-15**

**Question1:**

**Code:(App.js)**

import React from 'react';

import Header from './components/Header';

import Content from './components/Content';

import Footer from './components/Footer';

const App = () => (

  <div>

    <Header title="Welcome to My React App" />

    <Content />

    <Footer />

  </div>

);

export default App;

**Header.js**

import React from 'react';

const Header = ({ title }) => <h1>{title}</h1>;

export default Header;

**Content.js**

import React, { useState } from 'react';

const jokes = [

  "Why don’t scientists trust atoms? Because they make up everything!",

  "I told my computer I needed a break, and now it won’t stop sending me KitKats.",

  "Why do programmers prefer dark mode? Because light attracts bugs!"

];

const Content = () => {

  const [joke, setJoke] = useState('');

  const getJoke = () => {

    const random = jokes[Math.floor(Math.random() \* jokes.length)];

    setJoke(random);

  };

  return (

    <div>

      <button onClick={getJoke}>Tell me a joke</button>

      <p>{joke}</p>

    </div>

  );

};

export default Content;

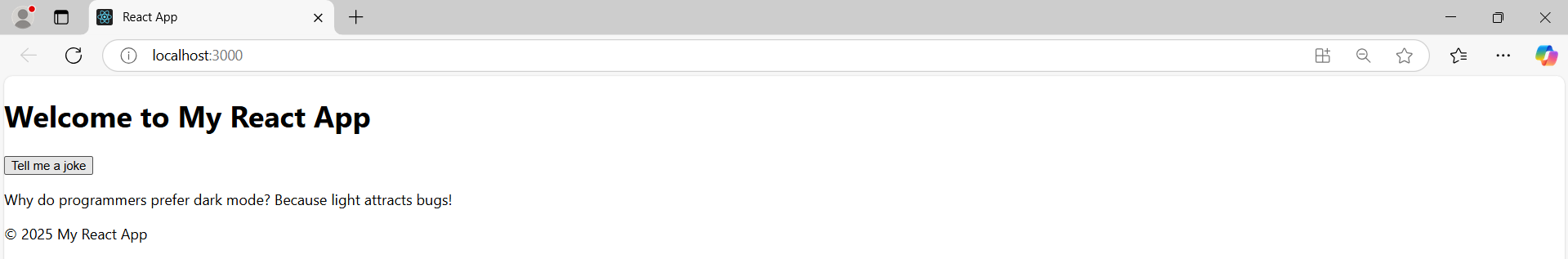
**Footer.js**

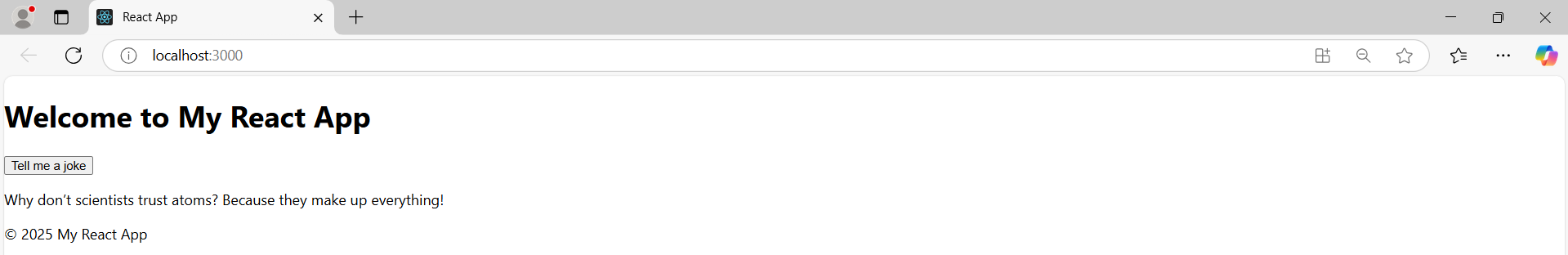
import React from 'react';

const Footer = () => <footer>© 2025 My React App</footer>;

export default Footer;

**Output:**



****

**Question-2,3,4:**

**Code:**

import React from 'react';

import './styles.css'; // External CSS for button 3

const App = () => {

  const inlineStyle = {

    backgroundColor: 'blue',

    padding: '10px 20px',

    fontSize: '16px',

    color: 'white',

    border: 'none',

    borderRadius: '5px',

    margin: '10px'

  };

  return (

    <div>

      {/\* 1. Inline CSS \*/}

      <button style={inlineStyle}>Inline Styled Button</button>

      {/\* 2. Internal CSS \*/}

      <style>

        {`

          .internal-button {

            background-color: green;

            color: white;

            padding: 10px 20px;

            font-size: 16px;

            border: none;

            border-radius: 5px;

            margin: 10px;

          }

        `}

      </style>

      <button className="internal-button">Internal Styled Button</button>

      {/\* 3. External CSS \*/}

      <button className="external-button">External Styled Button</button>

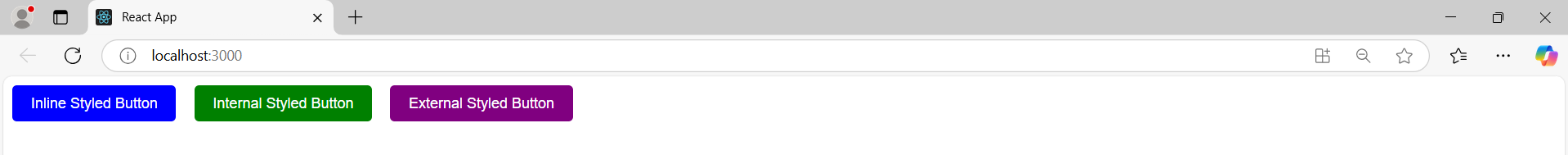
    </div>

  );

};

export default App;

**Output:**



**Question-5:**

import React, { Component, useState } from 'react';

class LifecycleDemo extends Component {

  constructor(props) {

    super(props);

    this.state = { count: 0 };

    console.log('Constructor');

  }

  componentDidMount() {

    console.log('Component Did Mount');

  }

  componentDidUpdate(prevProps, prevState) {

    if (prevState.count !== this.state.count) {

      console.log('Component Did Update');

    }

  }

  componentWillUnmount() {

    console.log('Component Will Unmount');

  }

  render() {

    return (

      <div>

        <h2>Lifecycle Demo</h2>

        <p>Count: {this.state.count}</p>

        <button onClick={() => this.setState({ count: this.state.count + 1 })}>

          Increment

        </button>

      </div>

    );

  }

}

const App = () => {

  const [showComponent, setShowComponent] = useState(true);

  return (

    <div>

      <h1>Lifecycle Methods Demo</h1>

      <button onClick={() => setShowComponent(!showComponent)}>

        {showComponent ? 'Unmount Component' : 'Mount Component'}

      </button>

      {showComponent && <LifecycleDemo />}

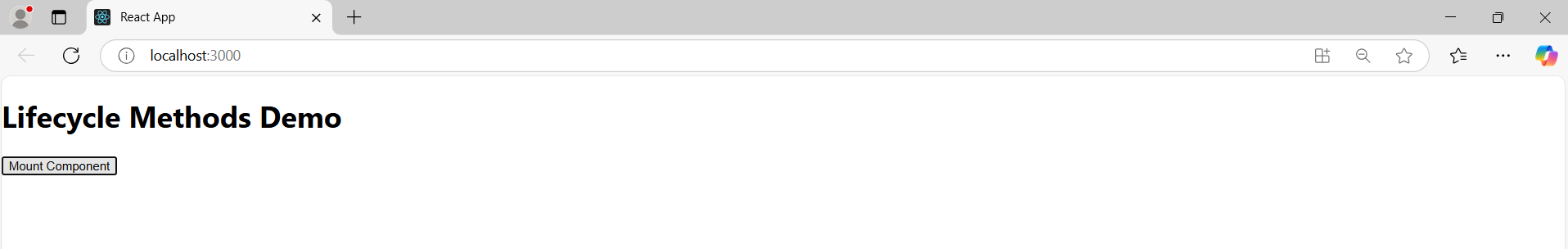
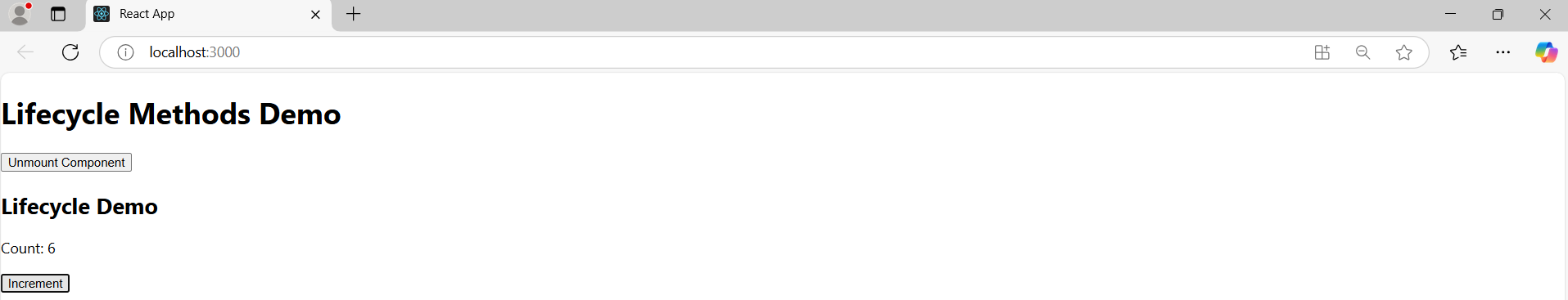
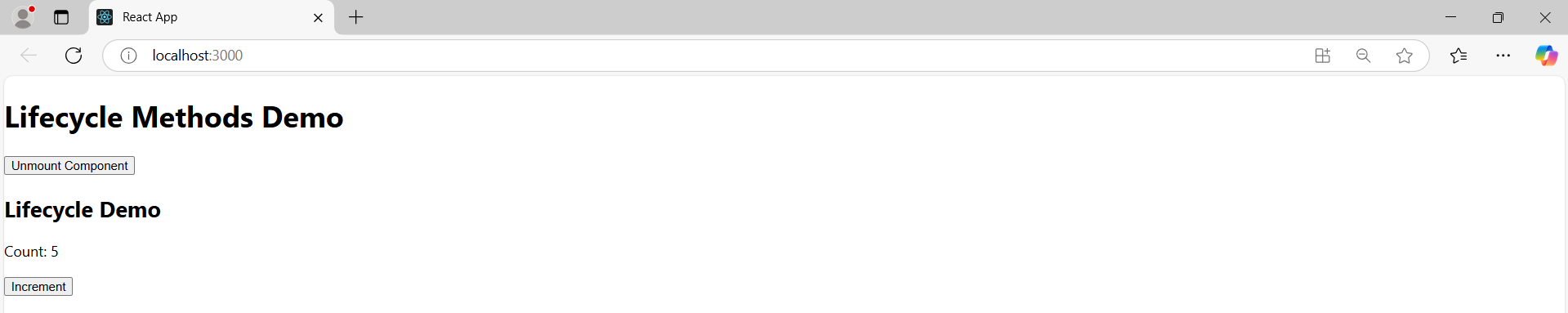
    </div>

  );

};

export default App;

**Output:**



**Question-6:**

import React from 'react';

// Child Component

const Child = ({ message }) => {

  return <h2>Message from Parent: {message}</h2>;

};

// Parent Component

const Parent = () => {

  const msg = "Prop message from parent to child!";

  return (

    <div>

      <h1>Parent Component</h1>

      <Child message={msg} />

    </div>

  );

};

// Main App Component

const App = () => {

  return (

    <div>

      <Parent />

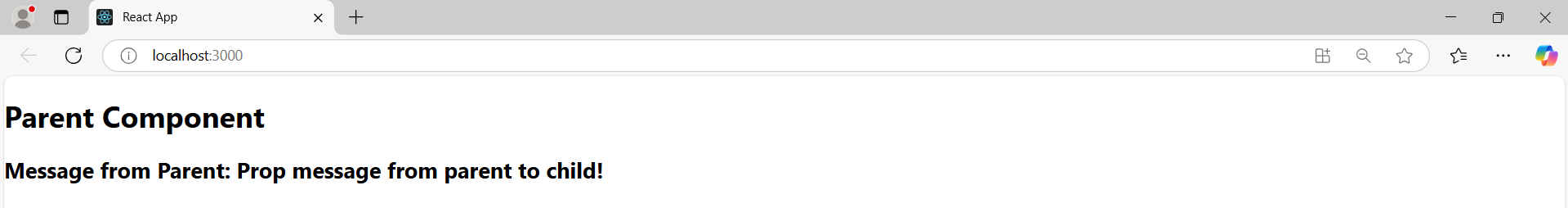
    </div>

  );

};

export default App;

**Output:**

****

**Question-7:**

import React from 'react';

import PropTypes from 'prop-types';

const Child = ({ msg }) => {

  return <h2>Validated Message: {msg}</h2>;

};

Child.propTypes = {

  msg: PropTypes.string.isRequired,

};

const Parent = () => {

  return (

    <div>

      <h1>Question 7</h1>

      <Child msg="Message from parent" />

    </div>

  );

};

const App = () => {

  return (

    <div>

      <Parent />

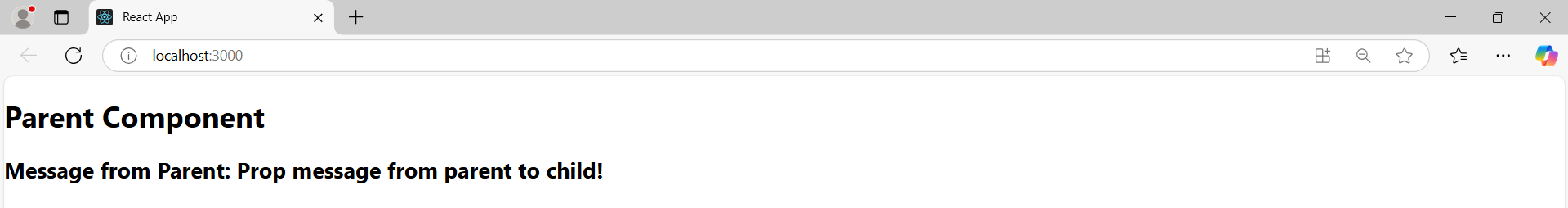
    </div>

  );

};

export default App;

**Output:**



**Question-8:**

import React, { useState, useReducer } from 'react';

const CounterUseState = () => {

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

  return (

    <div>

      <h2>Counter (useState)</h2>

      <p>Count: {count}</p>

      <button onClick={() => setCount(count + 1)}>Increase</button>

      <button onClick={() => setCount(count - 1)}>Decrease</button>

    </div>

  );

};

const reducer = (state, action) => {

  switch (action.type) {

    case 'increment':

      return { count: state.count + 1 };

    case 'decrement':

      return { count: state.count - 1 };

    default:

      return state;

  }

};

const CounterUseReducer = () => {

  const [state, dispatch] = useReducer(reducer, { count: 0 });

  return (

    <div>

      <h2>Counter (useReducer)</h2>

      <p>Count: {state.count}</p>

      <button onClick={() => dispatch({ type: 'increment' })}>Increase</button>

      <button onClick={() => dispatch({ type: 'decrement' })}>Decrease</button>

    </div>

  );

};

const App = () => {

  return (

    <div>

      <h1>Question 8</h1>

      <CounterUseState />

      <CounterUseReducer />

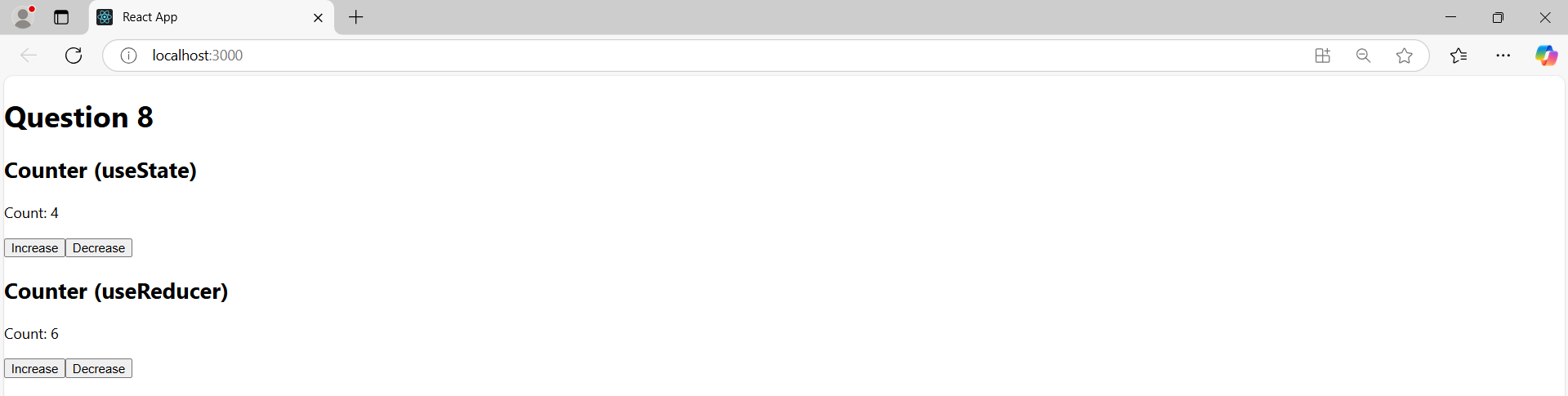
    </div>

  );

};

export default App;

**Output:**



**Question-9:**

import React, { useEffect, useState } from 'react';

const JokeComp = () => {

  const [joke, setJoke] = useState('');

  const fetchJoke = () => {

    fetch('https://official-joke-api.appspot.com/random\_joke')

      .then(res => res.json())

      .then(data => {

        setJoke(data.setup + ' - ' + data.punchline);

      })

      .catch(() => {

        setJoke('Error fetching joke.');

      });

  };

  useEffect(() => {

    fetchJoke();

  }, []);

  return (

    <div>

      <h2>Random Joke</h2>

      <p>{joke}</p>

      <button onClick={fetchJoke}>Get New Joke</button>

    </div>

  );

};

const App = () => {

  return (

    <div>

      <h1>Question 9</h1>

      <JokeComp />

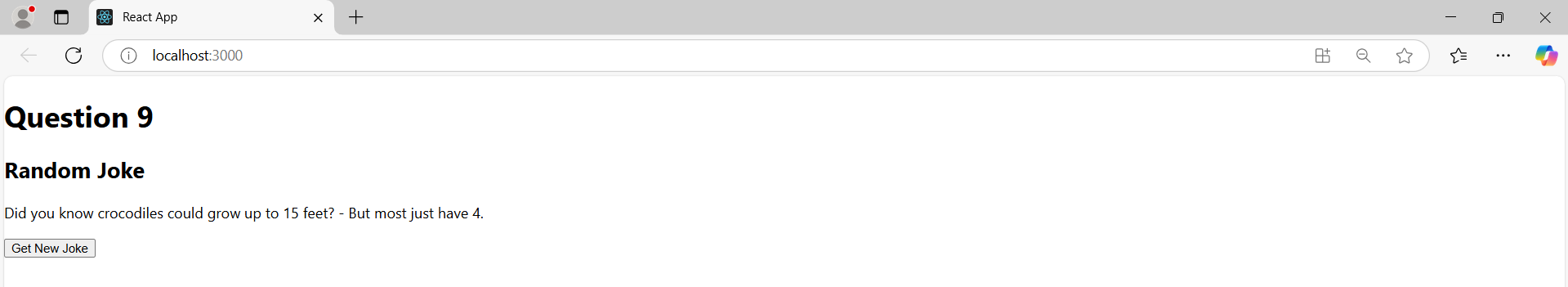
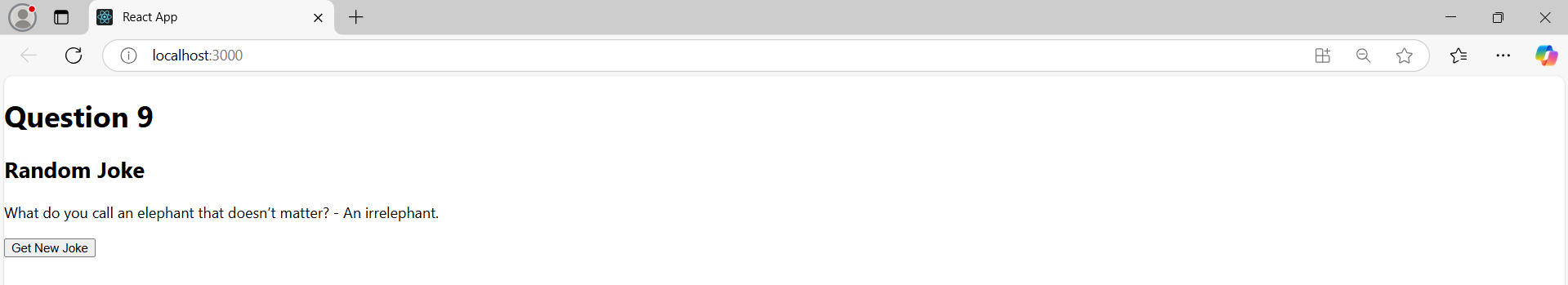
    </div>

  );

};

export default App;

**Output:**



**Question-10:**

import React, { useRef } from 'react';

const FocusInput = () => {

  const inputRef = useRef(null);

  const handleClick = () => {

    inputRef.current.focus();

  };

  return (

    <div>

      <h2>useRef Example</h2>

      <input type="text" ref={inputRef} />

      <button onClick={handleClick}>Focus Input</button>

    </div>

  );

};

const App = () => {

  return (

    <div>

      <h1>Question10</h1>

      <FocusInput />

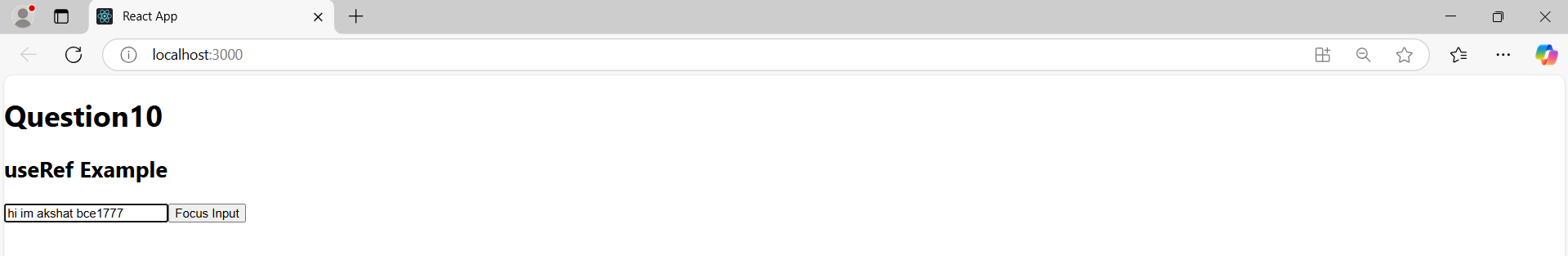
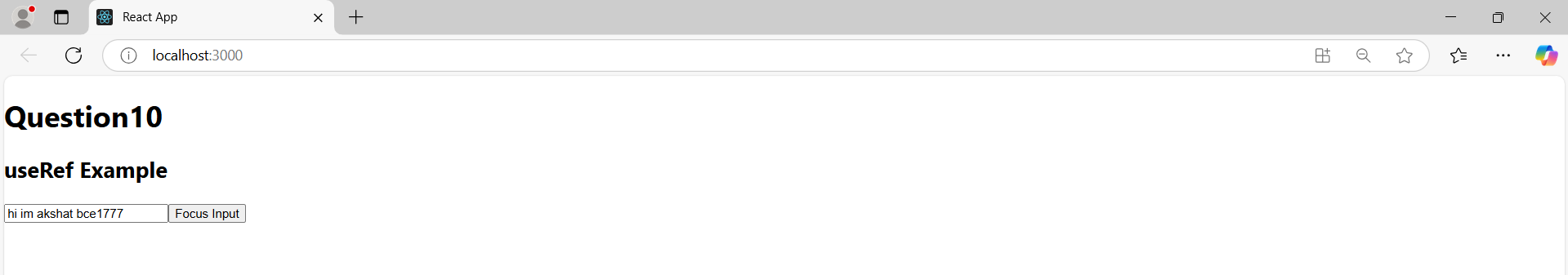
    </div>

  );

};

export default App;

**Output:**



**Question-11:**

import React, { useState, createContext, useContext } from 'react';

// Create Context

const ThemeCtx = createContext();

const Box = () => {

  const t = useContext(ThemeCtx);

  const style = {

    backgroundColor: t === 'dark' ? '#222' : '#eee',

    color: t === 'dark' ? '#fff' : '#000',

    padding: '20px',

    marginTop: '20px'

  };

  return (

    <div style={style}>

      <p>This is {t} mode</p>

    </div>

  );

};

const App = () => {

  const [t, setT] = useState('light');

  const toggle = () => {

    setT(t === 'light' ? 'dark' : 'light');

  };

  return (

    <ThemeCtx.Provider value={t}>

      <div style={{ padding: '20px' }}>

        <h1>Question 11</h1>

        <button onClick={toggle}>Toggle Theme</button>

        <Box />

      </div>

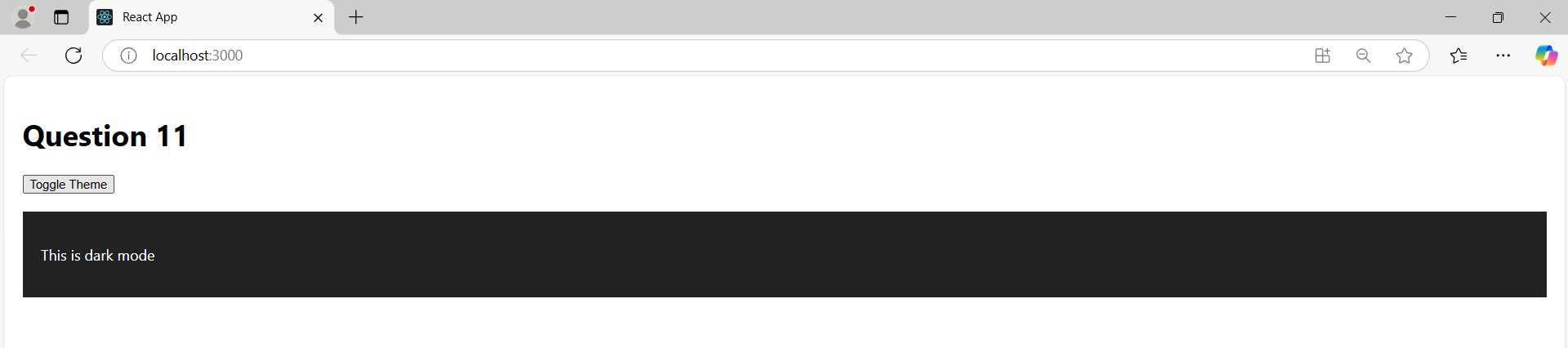
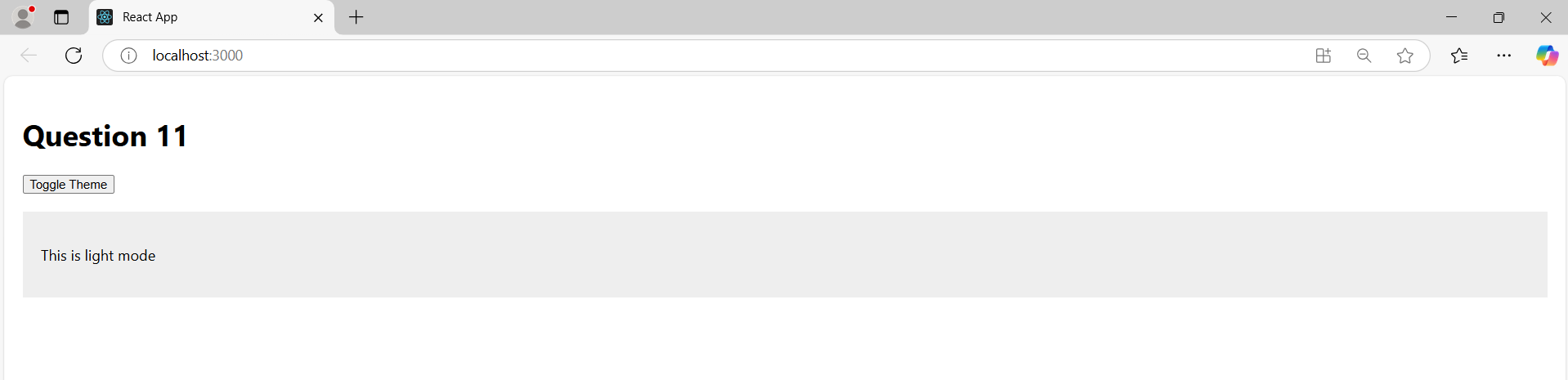
    </ThemeCtx.Provider>

  );

};

export default App;

**Output:**



**Question-12:**

import React, { useState, useRef } from 'react';

// Part (i) – useState

const FormState = () => {

  const [name, setN] = useState('');

  const [email, setE] = useState('');

  const handleSubmit = (e) => {

    e.preventDefault();

  };

  return (

    <div>

      <h2>Form with useState</h2>

      <form onSubmit={handleSubmit}>

        <input

          type="text"

          placeholder="Name"

          value={name}

          onChange={(e) => setN(e.target.value)}

        />

        <br />

        <input

          type="email"

          placeholder="Email"

          value={email}

          onChange={(e) => setE(e.target.value)}

        />

        <br />

        <button type="submit">Submit</button>

      </form>

      <p>Name: {name}</p>

      <p>Email: {email}</p>

    </div>

  );

};

// Part (ii) – useRef

const FormRef = () => {

  const nRef = useRef();

  const eRef = useRef();

  const [out, setOut] = useState({ n: '', e: '' });

  const handleSubmit = (e) => {

    e.preventDefault();

    setOut({

      n: nRef.current.value,

      e: eRef.current.value

    });

  };

  return (

    <div>

      <h2>Form with useRef</h2>

      <form onSubmit={handleSubmit}>

        <input type="text" placeholder="Name" ref={nRef} />

        <br />

        <input type="email" placeholder="Email" ref={eRef} />

        <br />

        <button type="submit">Submit</button>

      </form>

      <p>Name: {out.n}</p>

      <p>Email: {out.e}</p>

    </div>

  );

};

const App = () => {

  return (

    <div style={{ padding: '20px' }}>

      <h1>Question 12</h1>

      <FormState />

      <hr />

      <FormRef />

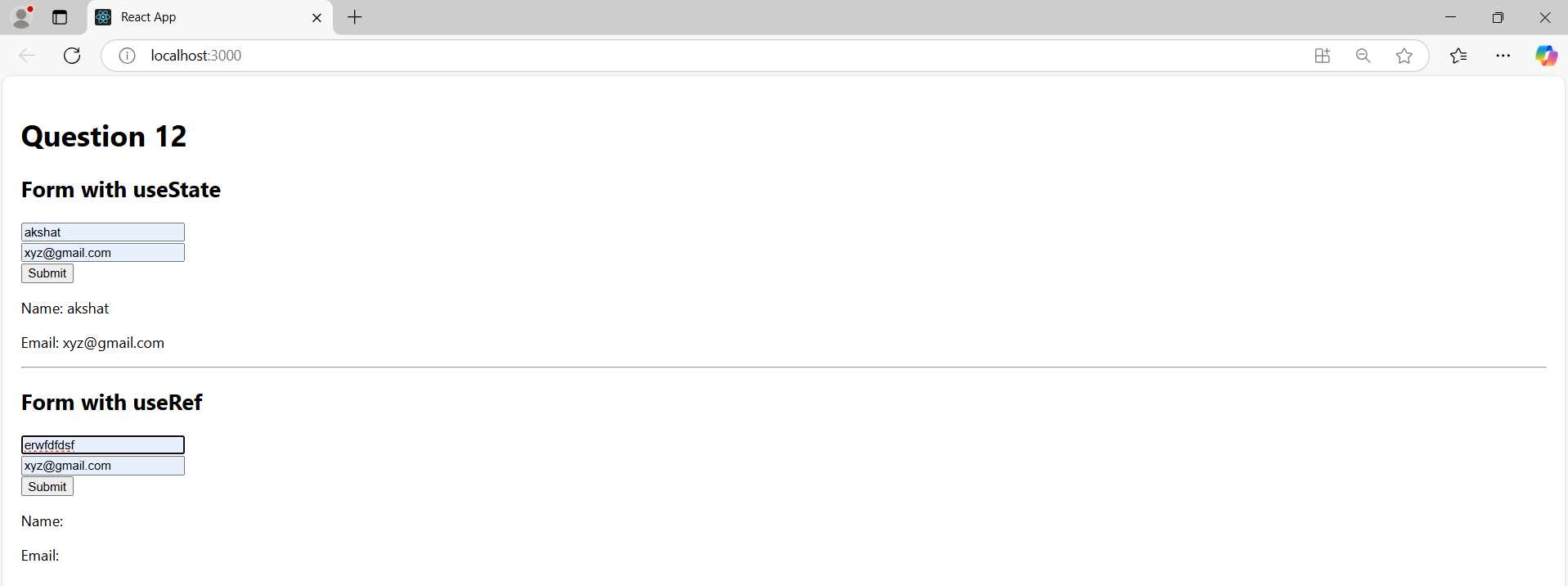
    </div>

  );

};

export default App;

**Output:**



Useref displays when submit is clicked

