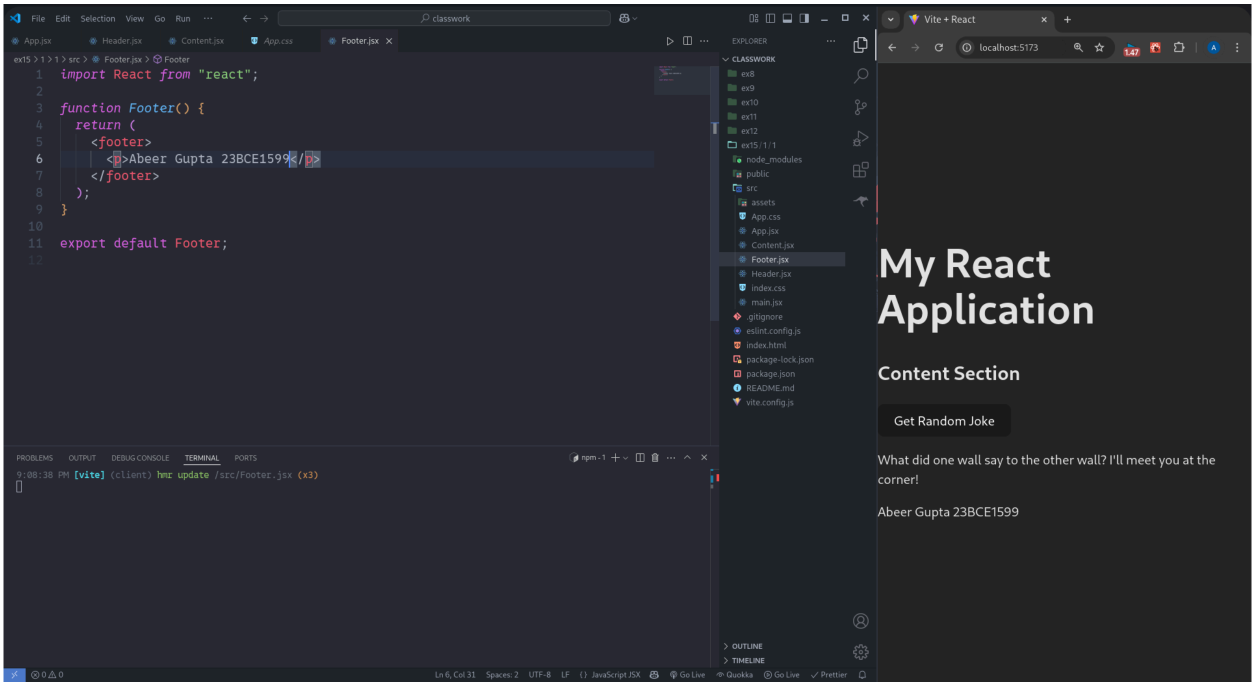
Register Number	23BCE1599	Subject Code	BCSE 308P
Name	Abeer Gupta	Subject Name	Web Programming
Programme	Scope	Slot	TE2
Course	BTech CSE Core	Faculty	Dr. LM Jenila Livingston

BCSE 203E: Web Programming Lab

Exercise 15

1.



```
import React from "react";

function Footer() {
  return (
    <footer>
      <p>Abeer Gupta 23BCE1599</p>
    </footer>
  );
}

export default Footer;
```

```
9:08:38 PM [vite] (client) hmr update /src/Footer.jsx (x3)
```

My React Application

Content Section

Get Random Joke

What did one wall say to the other wall? I'll meet you at the corner!

Abeer Gupta 23BCE1599

Footer.jsx

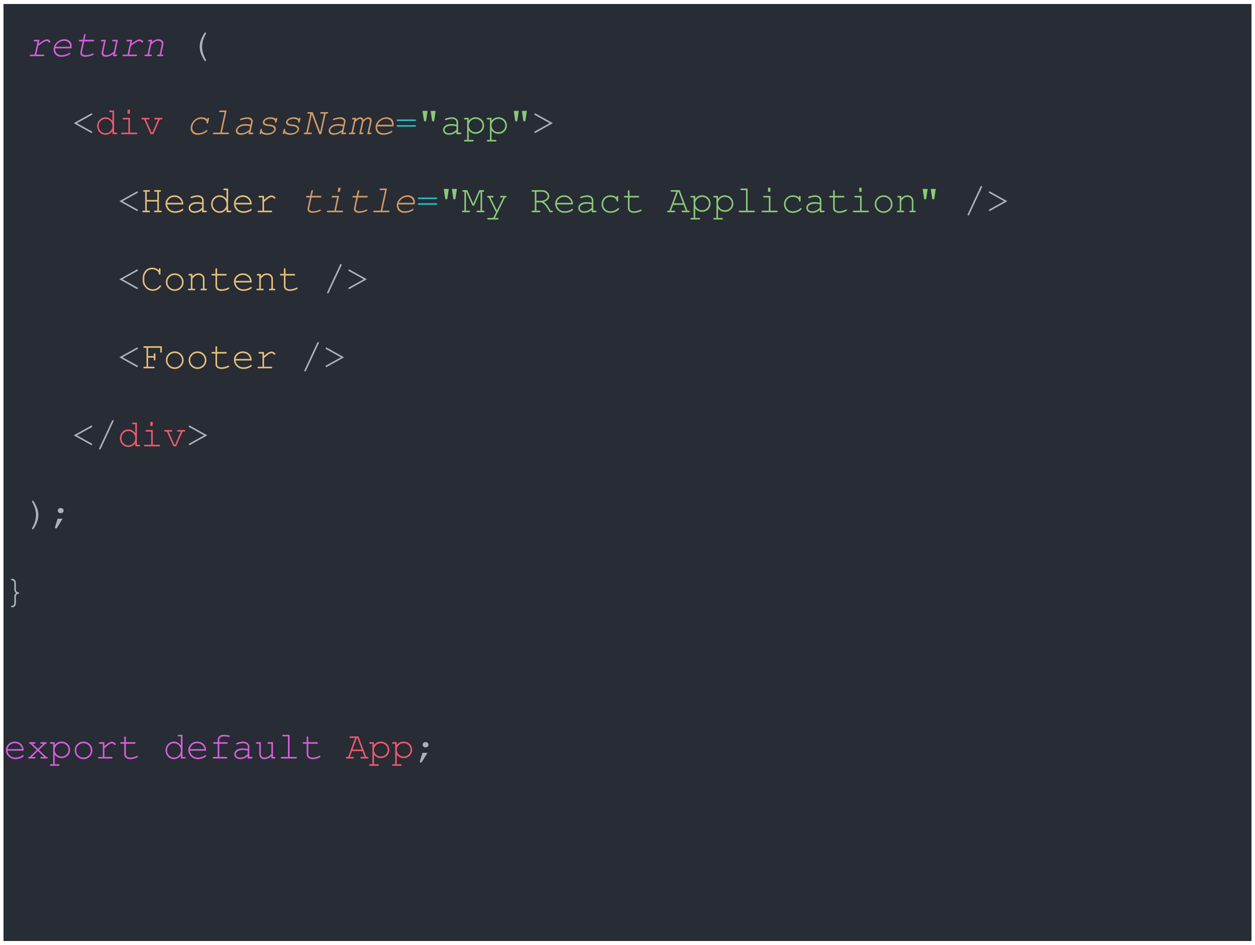
```
import React from "react";


function Footer() {

 return (
   <footer>
     <p>Abeer Gupta 23BCE1599</p>
   </footer>
 );
}

export default Footer;
```

Content.jsx

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

function Content() {
 const [joke, setJoke] = useState("");
  const jokes = [
   "Why don't scientists trust atoms? Because they make up everything!",
   "What did one wall say to the other wall? I'll meet you at the corner!",
   "Why did the scarecrow win an award? Because he was outstanding in his field!",
   "How do you organize a space party? You planet!"
 ];
  const getRandomJoke = () => {
   const randomIndex = Math.floor(Math.random() *
jokes.length);
   setJoke(jokes[randomIndex]);
 };
  return (
   <div className="content">
     <h2>Content Section</h2>
     <button onClick={getRandomJoke}>Get Random
Joke</button>
     {joke && <p>{joke}</p>}
   </div>
 );
}

export default Content;
```

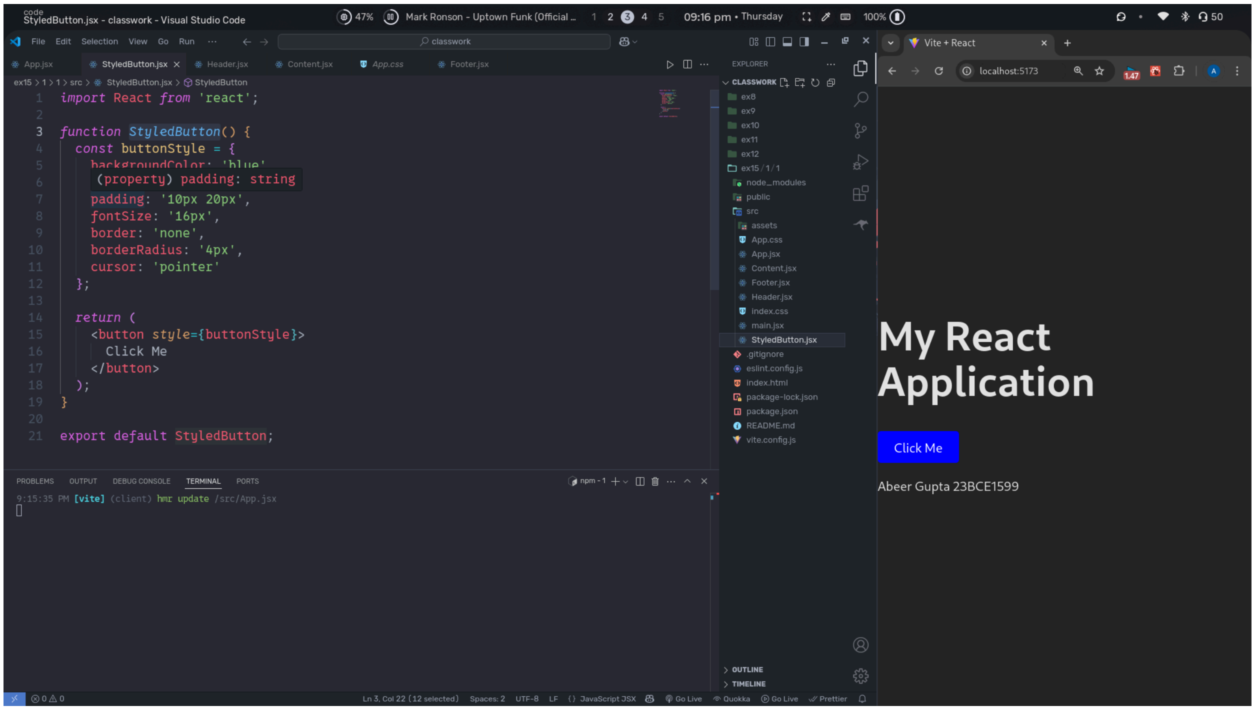
App.jsx

```
import React from "react";
import Header from "./Header";
import Content from "./Content";
import Footer from "./Footer";

function App() {
```



```
  return (
    <div className="app">
      <Header title="My React Application" />
      <Content />
      <Footer />
    </div>
  );
}


export default App;
```

2.



```
import React from 'react';

function StyledButton() {
  const buttonStyle = {
    backgroundColor: 'blue',
    (property) padding: string
    padding: '10px 20px',
    fontSize: '16px',
    border: 'none',
    borderRadius: '4px',
    cursor: 'pointer'
  };

  return (
    <button style={buttonStyle}>
      Click Me
    </button>
  );
}

export default StyledButton;
```

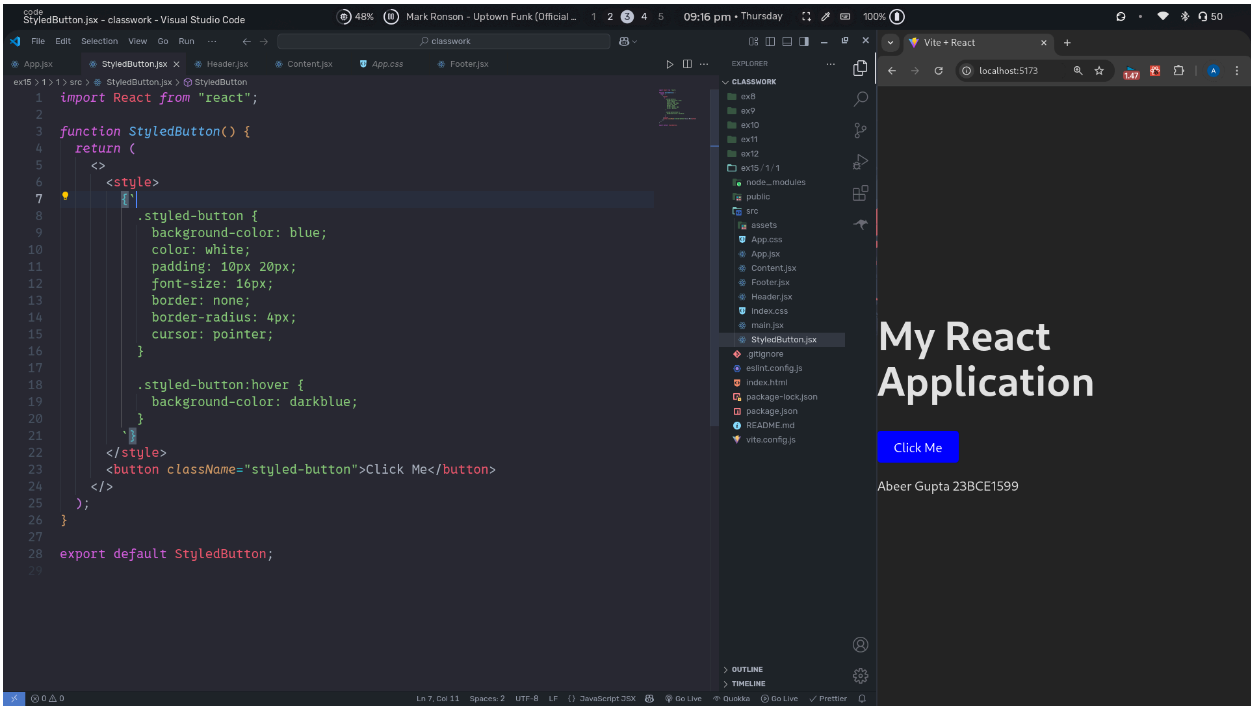
```
9:15:35 PM [vite] (client) hmr update /src/App.jsx
```

My React Application

Click Me

Abeer Gupta 23BCE1599

3.



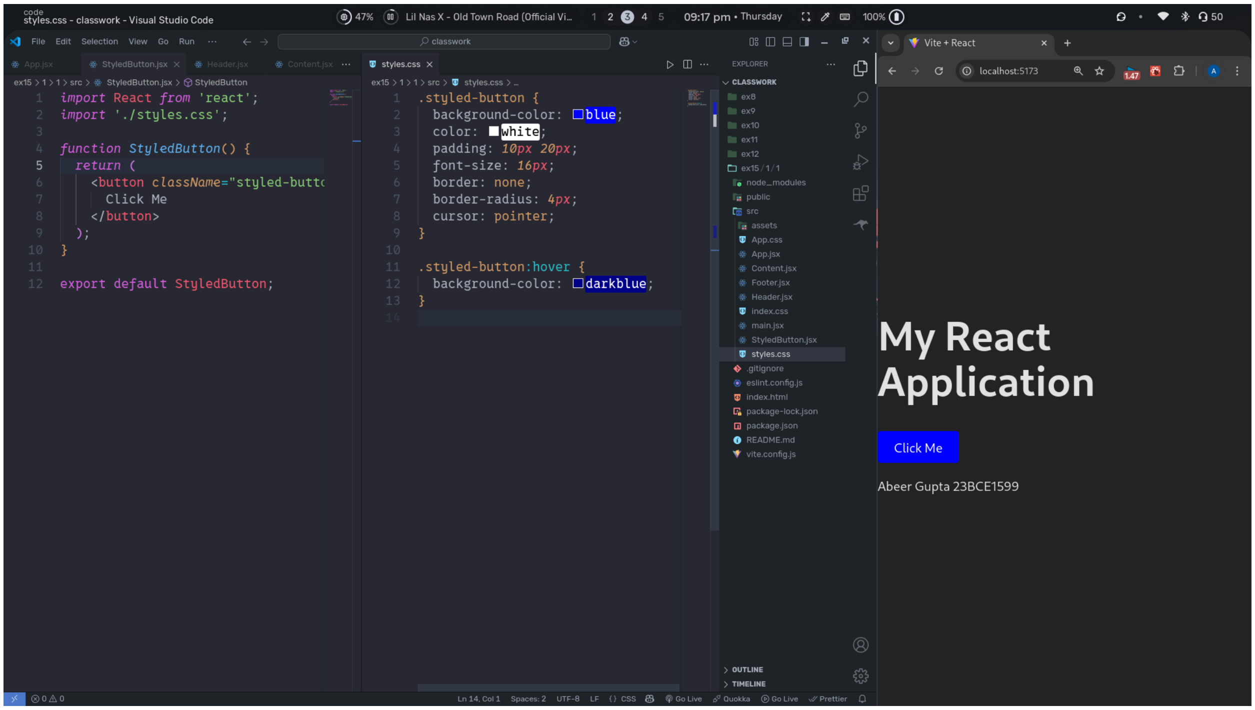
```
import React from "react";

function StyledButton() {
  return (
    <>
      <style>
        {`
          .styled-button {
            background-color: blue;
            color: white;
            padding: 10px 20px;
            font-size: 16px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
          }

          .styled-button:hover {
            background-color: darkblue;
          }
        `}
      </style>
      <button className="styled-button">Click Me</button>
    </>
  );
}

export default StyledButton;
```

My React Application

Click Me

Abeer Gupta 23BCE1599

4.



```
import React from 'react';
import './styles.css';

function StyledButton() {
  return (
    <button className="styled-butto
      Click Me
    </button>
  );
}

export default StyledButton;
```

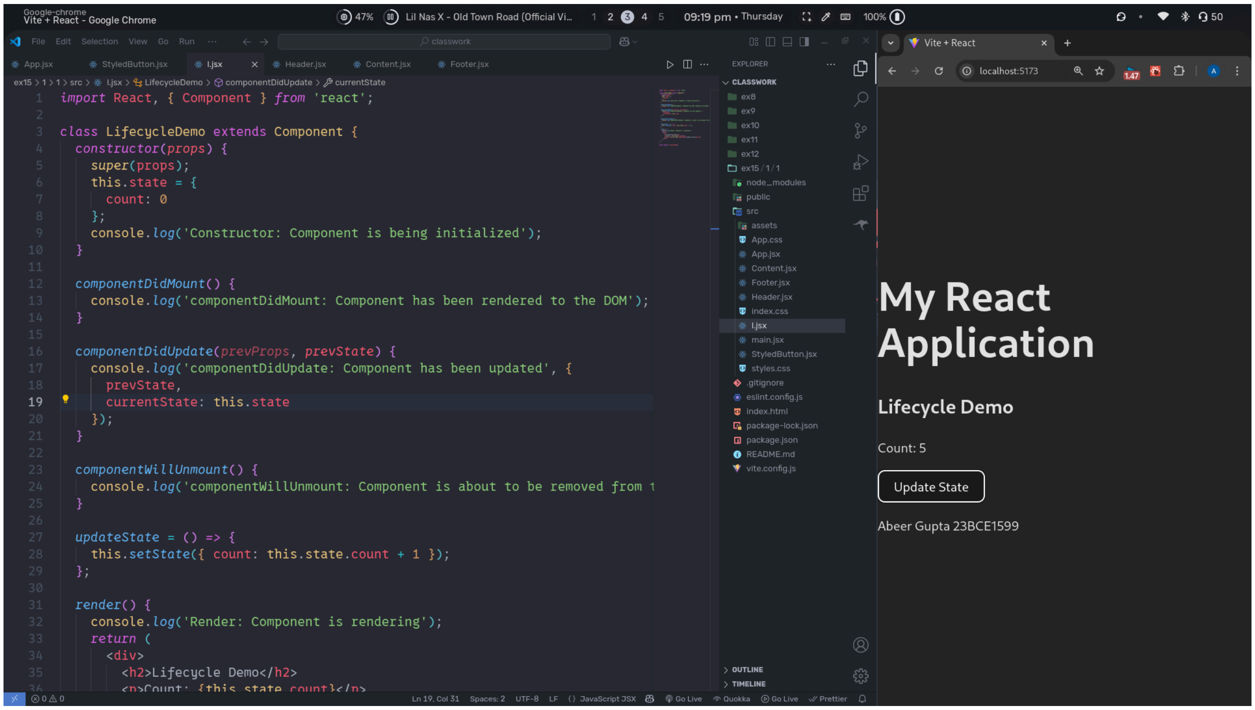
```
.styled-button {
  background-color: blue;
  color: white;
  padding: 10px 20px;
  font-size: 16px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}

.styled-button:hover {
  background-color: darkblue;
}
```

My React Application

Click Me

Abeer Gupta 23BCE1599

5.



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

class LifecycleDemo extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 0
    };
    console.log('Constructor: Component is being initialized');
  }

  componentDidMount() {
    console.log('componentDidMount: Component has been rendered to the DOM');
  }

  componentDidUpdate(prevProps, prevState) {
    console.log('componentDidUpdate: Component has been updated', {
      prevState,
      currentState: this.state
    });
  }

  componentWillUnmount() {
    console.log('componentWillUnmount: Component is about to be removed from t
  }

  updateState = () => {
    this.setState({ count: this.state.count + 1 });
  };

  render() {
    console.log('Render: Component is rendering');
    return (
      <div>
        <h2>Lifecycle Demo</h2>
        <p>Count: {this.state.count}</p>
```

My React Application

Lifecycle Demo

Count: 5

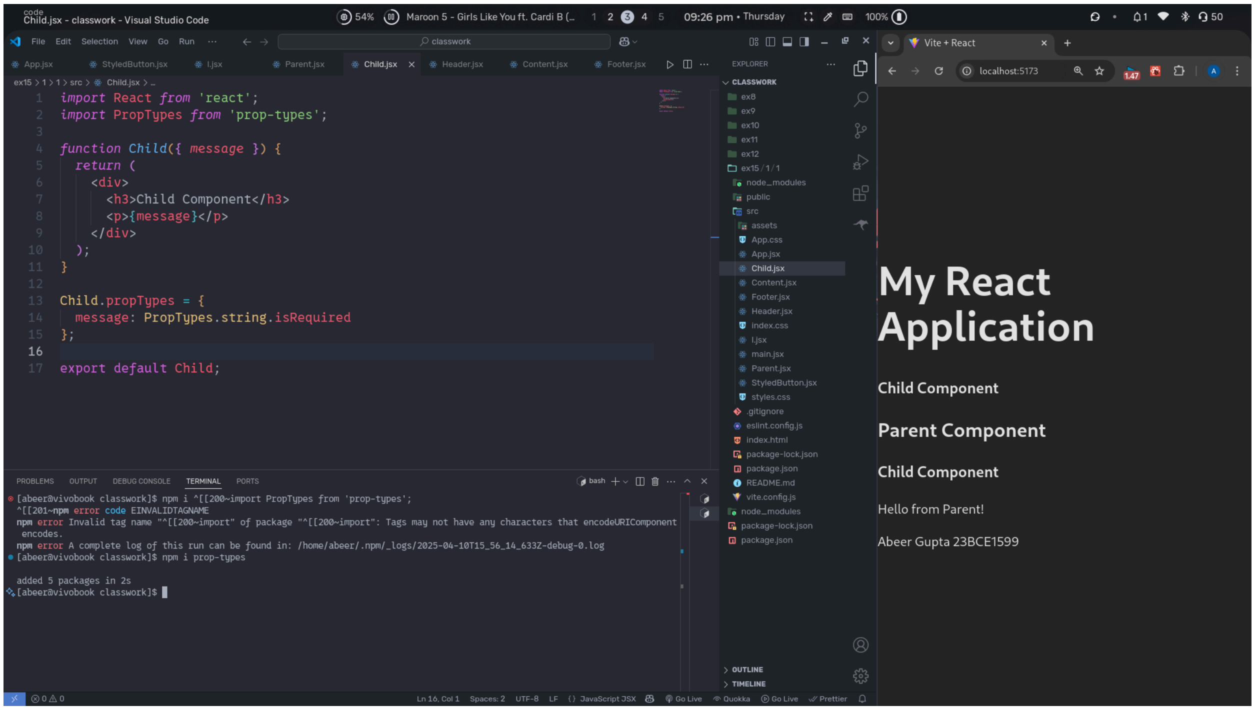
Update State

Abeer Gupta 23BCE1599

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


class LifecycleDemo extends Component {

 constructor(props) {

   super(props);

   this.state = {

     count: 0

   };

   console.log('Constructor: Component is being
initialized');

 }
 componentDidMount() {
   console.log('componentDidMount: Component has been rendered to the DOM');
 }

 componentDidUpdate(prevProps, prevState) {
   console.log('componentDidUpdate: Component has been updated', {
     prevState,
     currentState: this.state
   });
 }

 componentWillUnmount() {
   console.log('componentWillUnmount: Component is about to be removed from the DOM');
 }

 updateState = () => {
   this.setState({ count: this.state.count + 1 });
 };
 render() {
   console.log('Render: Component is rendering');
   return (
     <div>
       <h2>Lifecycle Demo</h2>
       <p>Count: {this.state.count}</p>
       <button onClick={this.updateState}>Update
State</button>
     </div>
   );
 }
}

export default LifecycleDemo;
import React, { useState } from 'react';
import LifecycleDemo from './LifecycleDemo';

function LifecycleDemoContainer() {
 const [isVisible, setIsVisible] = useState(true);

 return (
   <div>
```

```
     <button onClick={() => setIsVisible(!isVisible)}>
       {isVisible ? 'Unmount Component' : 'Mount Component'}
     </button>
     {isVisible && <LifecycleDemo />}
   </div>
 );
}


export default LifecycleDemoContainer;
```

6. And 7.



```
import React from 'react';
import PropTypes from 'prop-types';

function Child({ message }) {
  return (
    <div>
      <h3>Child Component</h3>
      <p>{message}</p>
    </div>
  );
}

Child.propTypes = {
  message: PropTypes.string.isRequired
};

export default Child;
```

```
[abeer@vivobook classwork]$ npm i ^[[200~import PropTypes from 'prop-types';
^[[201~npm error code EINVALIDTAGNAME
npm error Invalid tag name "^[[200~import" of package "^[[200~import": Tags may not have any characters that encodeURIComponent
 encodes.
npm error A complete log of this run can be found in: /home/abeer/.npm/_logs/2025-04-10T15_56_14_633Z-debug-0.log
[abeer@vivobook classwork]$ npm i prop-types

added 5 packages in 2s
[abeer@vivobook classwork]$ 
```

My React Application

Child Component

Parent Component

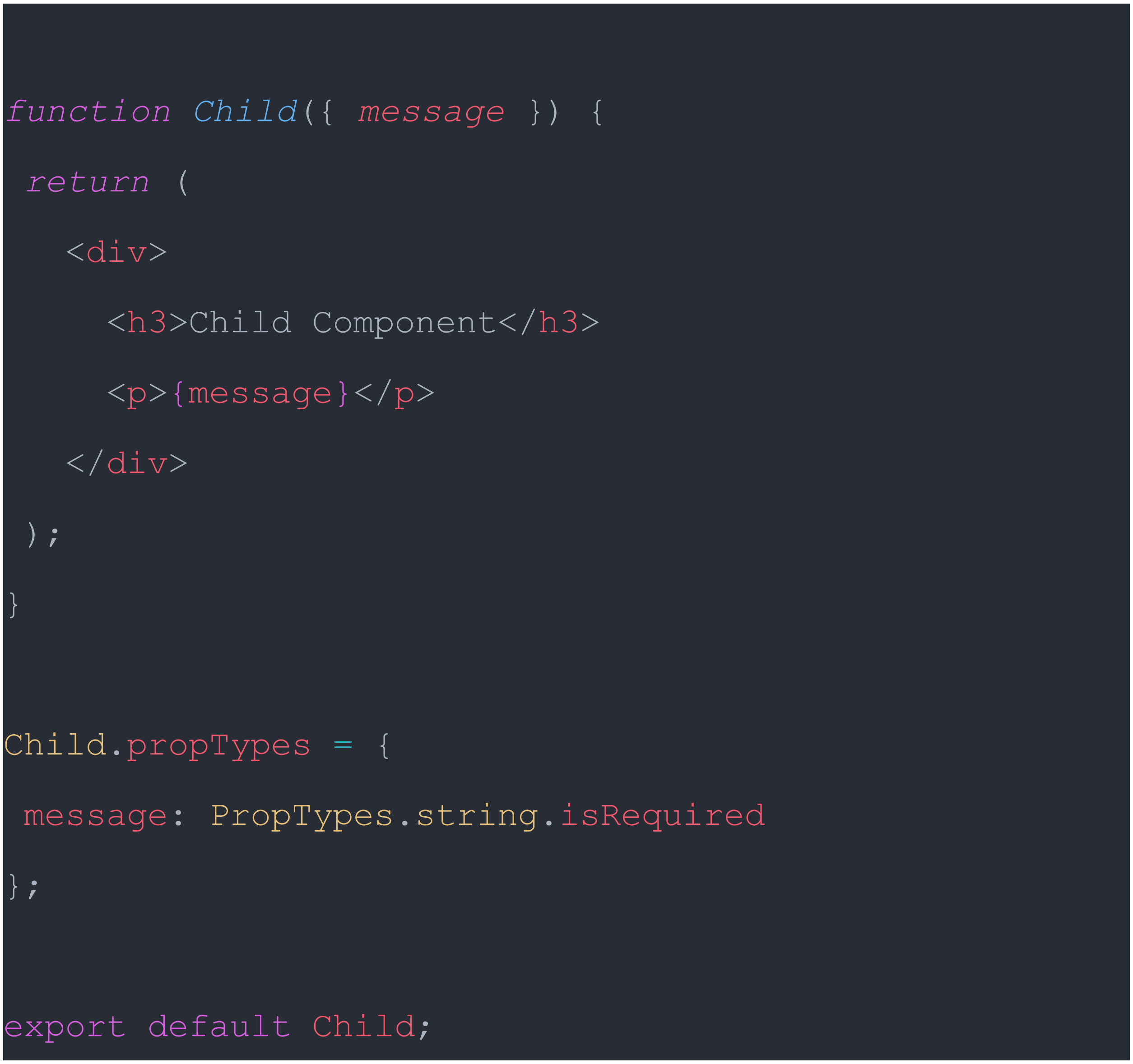
Child Component

Hello from Parent!

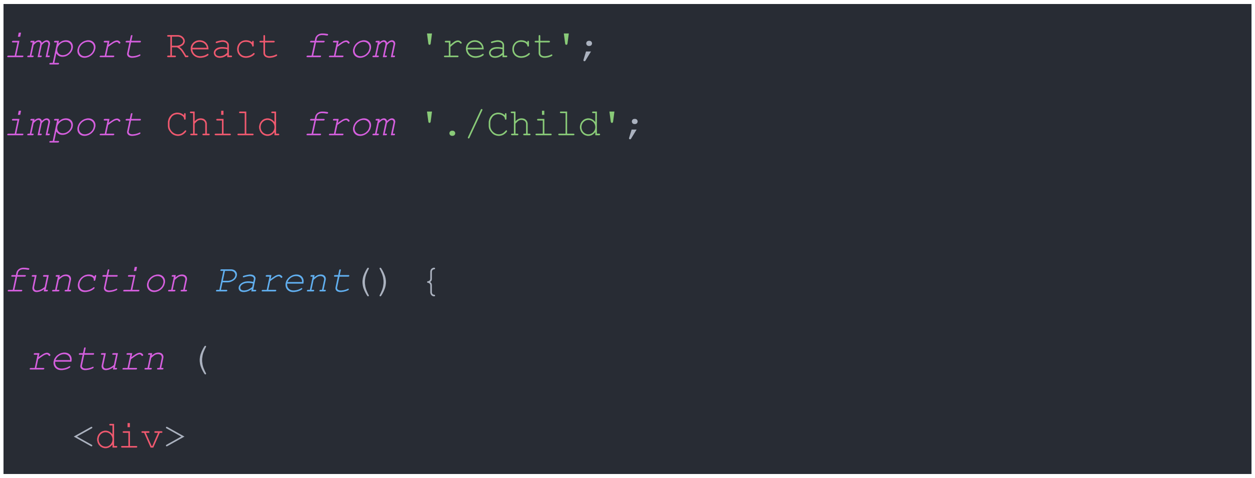
Abeer Gupta 23BCE1599

Child.jsx - classwork - Visual Studio Code
File Edit Selection View Go Run
classwork
App.jsx StyledButton.jsx I.jsx Parent.jsx Child.jsx Header.jsx Content.jsx Footer.jsx
ex15 > 1 > 1 > src > Child.jsx
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 16, Col 1 Spaces: 2 UTF-8 LF JavaScript JSX Go Live Quokka Prettier
Vite + React
localhost:5173

```
import React from 'react';

import PropTypes from 'prop-types';
```



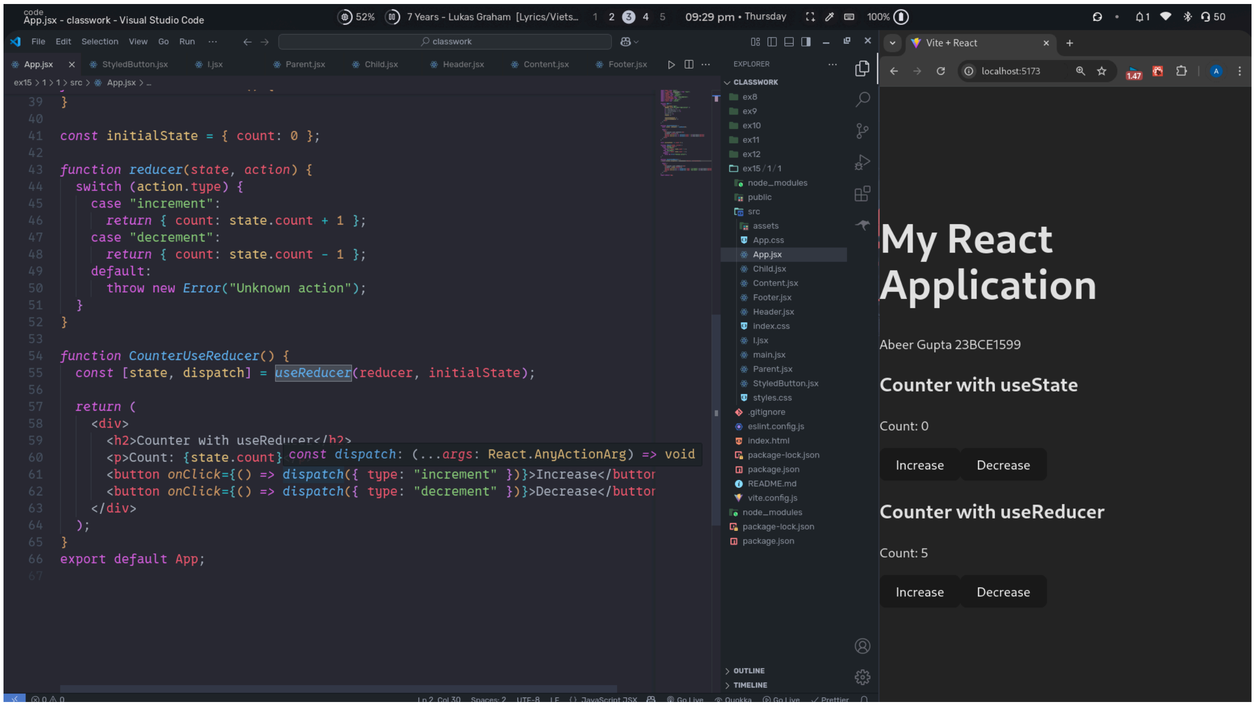
```
function Child({ message }) {
 return (
   <div>
     <h3>Child Component</h3>
     <p>{message}</p>
   </div>
 );
}

Child.propTypes = {
 message: PropTypes.string.isRequired
};

export default Child;
```



```
import React from 'react';
import Child from './Child';

function Parent() {
 return (
   <div>
```

```
      <h2>Parent Component</h2>

      <Child message="Hello from Parent!" />

    </div>

  );

}



export default Parent;
```

8.



```
}

const initialState = { count: 0 };

function reducer(state, action) {
  switch (action.type) {
    case "increment":
      return { count: state.count + 1 };
    case "decrement":
      return { count: state.count - 1 };
    default:
      throw new Error("Unknown action");
  }
}

function CounterUseReducer() {
  const [state, dispatch] = useReducer(reducer, initialState);

  return (
    <div>
      <h2>Counter with useReducer</h2>
      <p>Count: {state.count}
      <button onClick={() => dispatch({ type: "increment" })}>Increase</buttor
      <button onClick={() => dispatch({ type: "decrement" })}>Decrease</buttor
    </div>
  );
}
export default App;
```

```
const dispatch: (...args: React.AnyActionArg) => void
```

My React Application

Abeer Gupta 23BCE1599

Counter with useState

Count: 0

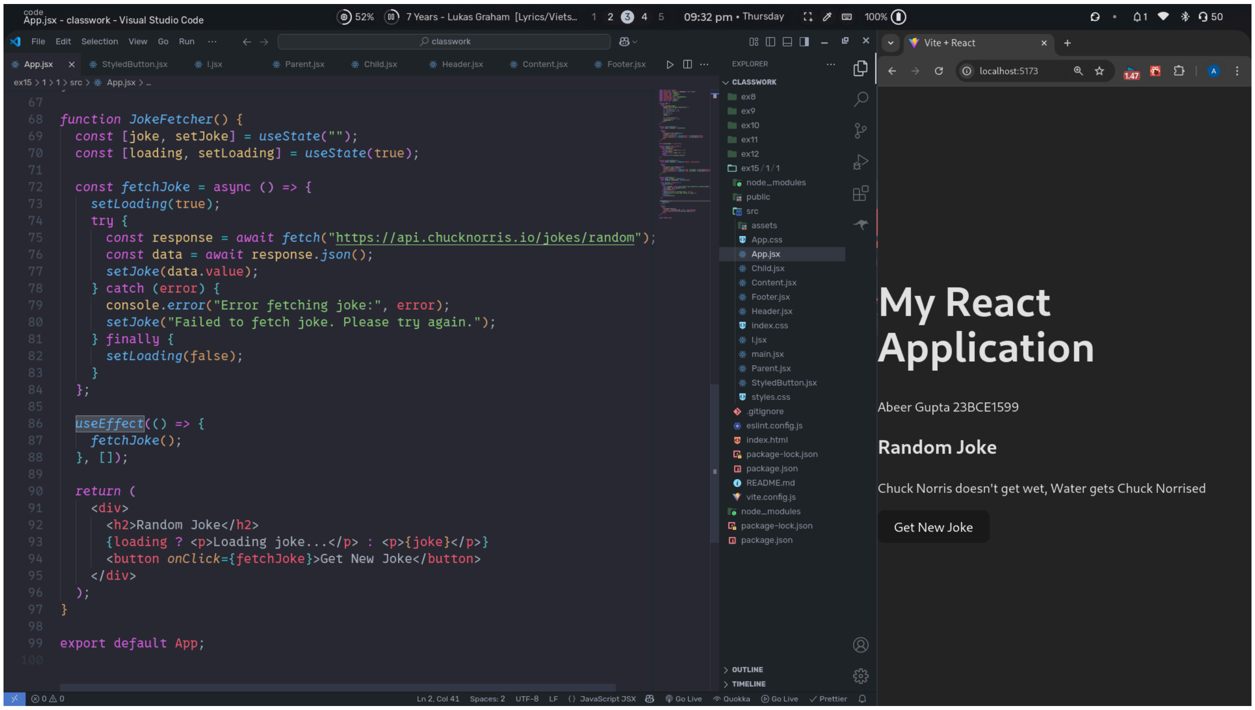
Increase Decrease

Counter with useReducer

Count: 5

Increase Decrease

9.



```jsx

function JokeFetcher() {
  const [joke, setJoke] = useState("");
  const [loading, setLoading] = useState(true);

  const fetchJoke = async () => {
    setLoading(true);
    try {
      const response = await fetch("https://api.chucknorris.io/jokes/random");
      const data = await response.json();
      setJoke(data.value);
    } catch (error) {
      console.error("Error fetching joke:", error);
      setJoke("Failed to fetch joke. Please try again.");
    } finally {
      setLoading(false);
    }
  };

  useEffect(() => {
    fetchJoke();
  }, []);

  return (
    <div>
      <h2>Random Joke</h2>
      {loading ? <p>Loading joke...</p> : <p>{joke}</p>}
      <button onClick={fetchJoke}>Get New Joke</button>
    </div>
  );
}

export default App;
```

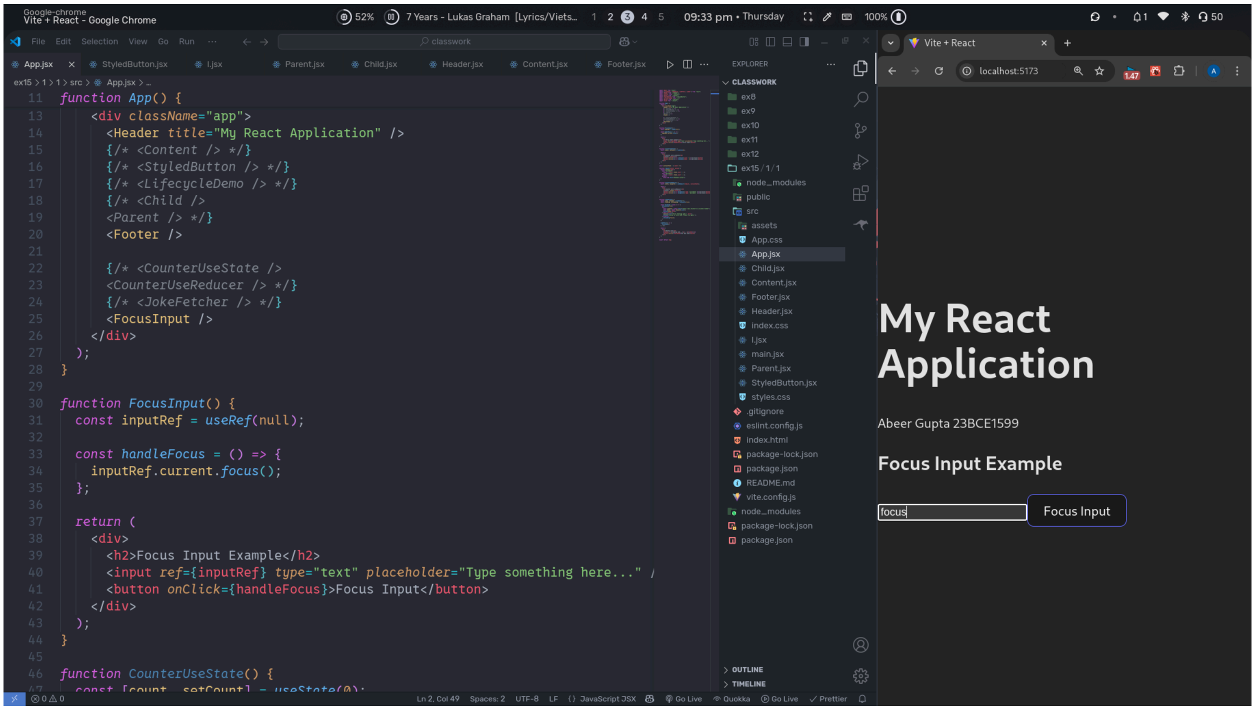
My React Application

Abeer Gupta 23BCE1599

Random Joke

Chuck Norris doesn't get wet, Water gets Chuck Norrised

10.



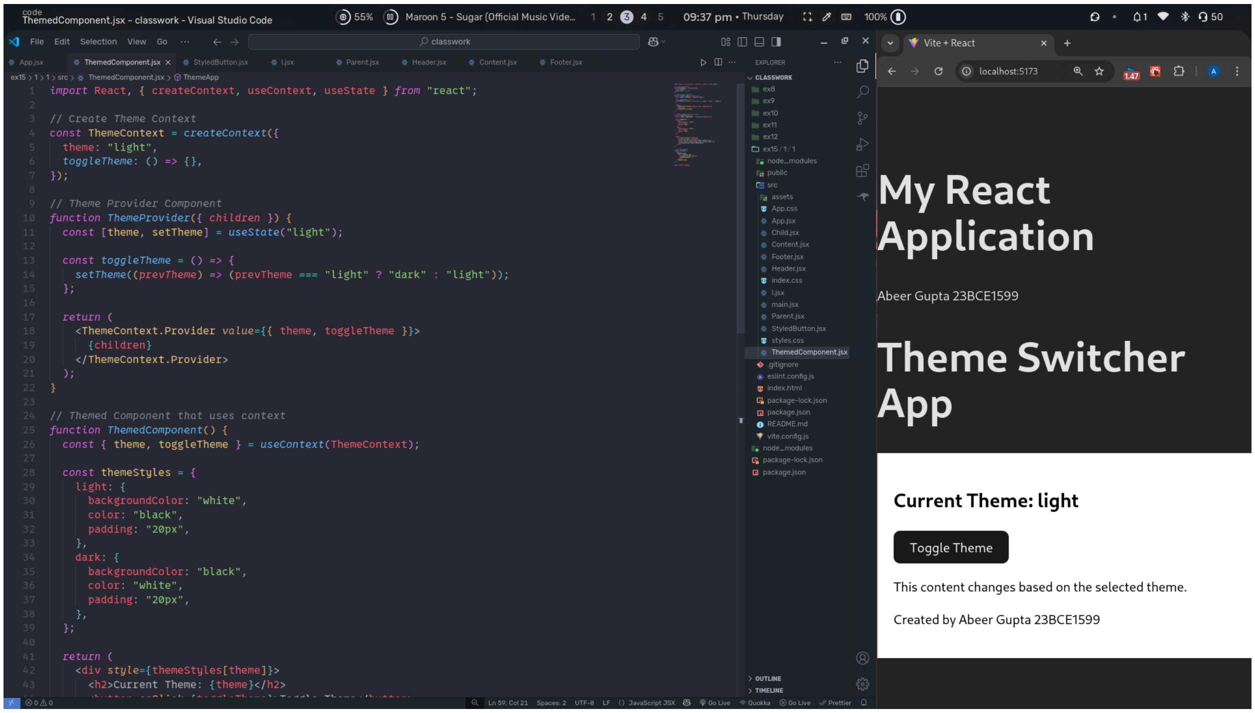
```jsx
function App() {
    <div className="app">
      <Header title="My React Application" />
      {/* <Content /> */}
      {/* <StyledButton /> */}
      {/* <LifecycleDemo /> */}
      {/* <Child />
      <Parent /> */}
      <Footer />

      {/* <CounterUseState />
      <CounterUseReducer /> */}
      {/* <JokeFetcher /> */}
      <FocusInput />
    </div>
  );
}

function FocusInput() {
  const inputRef = useRef(null);

  const handleFocus = () => {
    inputRef.current.focus();
  };

  return (
    <div>
      <h2>Focus Input Example</h2>
      <input ref={inputRef} type="text" placeholder="Type something here..." /
      <button onClick={handleFocus}>Focus Input</button>
    </div>
  );
}

function CounterUseState() {
  const [count, setCount] = useState(0);
```

My React Application

Abeer Gupta 23BCE1599

Focus Input Example

11.



```
import React, { createContext, useContext, useState } from "react";

// Create Theme Context
const ThemeContext = createContext({
  theme: "light",
  toggleTheme: () => {},
});

// Theme Provider Component
function ThemeProvider({ children }) {
  const [theme, setTheme] = useState("light");

  const toggleTheme = () => {
    setTheme((prevTheme) => (prevTheme === "light" ? "dark" : "light"));
  };

  return (
    <ThemeContext.Provider value={{ theme, toggleTheme }}>
      {children}
    </ThemeContext.Provider>
  );
}

// Themed Component that uses context
function ThemedComponent() {
  const { theme, toggleTheme } = useContext(ThemeContext);

  const themeStyles = {
    light: {
      backgroundColor: "white",
      color: "black",
      padding: "20px",
    },
    dark: {
      backgroundColor: "black",
      color: "white",
      padding: "20px",
    },
  };

  return (
    <div style={themeStyles[theme]}>
      <h2>Current Theme: {theme}</h2>
```

My React Application

Abeer Gupta 23BCE1599

Theme Switcher App

Current Theme: light

Toggle Theme

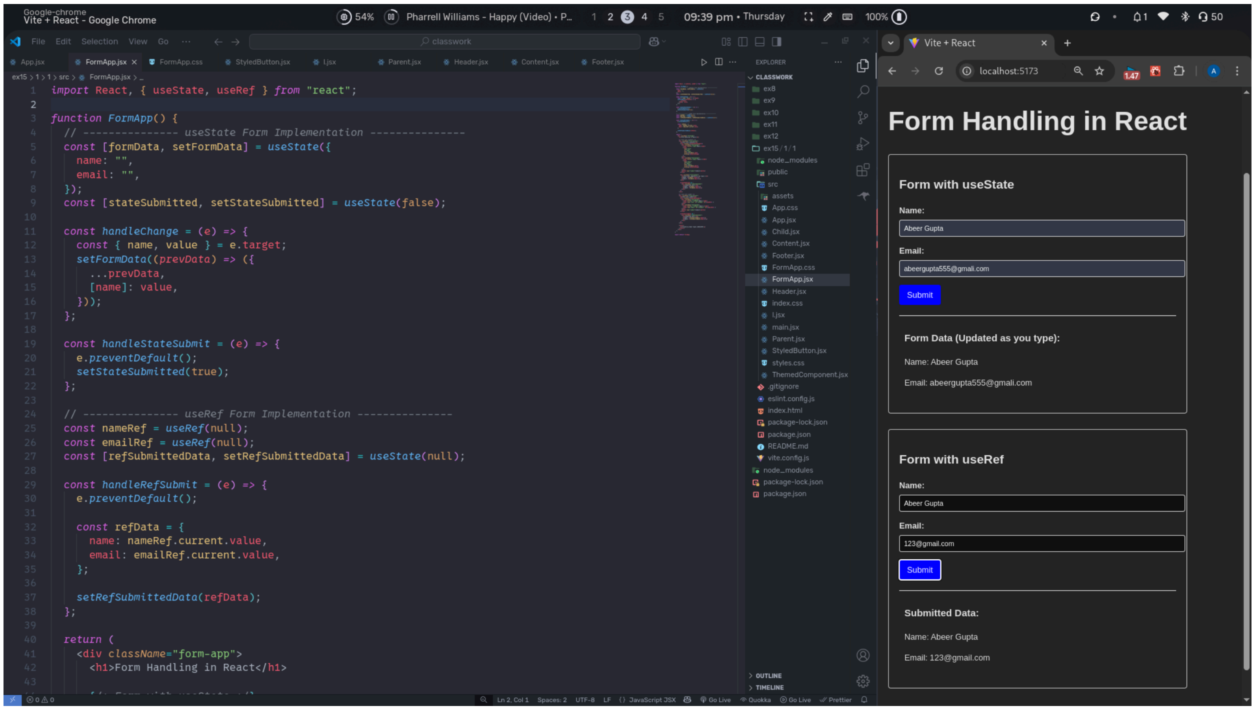
This content changes based on the selected theme.

Created by Abeer Gupta 23BCE1599

```
import React, { createContext, useContext, useState }
from "react";


// Create Theme Context

const ThemeContext = createContext({

 theme: "light",

 toggleTheme: () => {},

});


// Theme Provider Component

function ThemeProvider({ children }) {

 const [theme, setTheme] = useState("light");
 const toggleTheme = () => {
   setTheme((prevTheme) => (prevTheme === "light" ?
"dark" : "light"));
 };

 return (
   <ThemeContext.Provider value={{ theme, toggleTheme
}}>
     {children}
   </ThemeContext.Provider>
 );
}

// Themed Component that uses context
function ThemedComponent() {
 const { theme, toggleTheme } =
useContext(ThemeContext);

 const themeStyles = {
   light: {
     backgroundColor: "white",
     color: "black",
     padding: "20px",
   },
   dark: {
     backgroundColor: "black",
     color: "white",
     padding: "20px",
   },
 };

 return (
   <div style={themeStyles[theme]}>
     <h2>Current Theme: {theme}</h2>
     <button onClick={toggleTheme}>Toggle Theme</button>
     <p>This content changes based on the selected theme.</p>
     <p>Created by Abeer Gupta 23BCE1599</p>
   </div>
 );
}
```

```
// Main App Component

function ThemeApp() {

 return (

   <ThemeProvider>

     <div className="app">

       <h1>Theme Switcher App</h1>

       <ThemedComponent />

     </div>

   </ThemeProvider>

 );

}


export default ThemeApp;
```

12.



```
import React, { useState, useRef } from "react";

function FormApp() {
  // -------------- useState Form Implementation --------------
  const [formData, setFormData] = useState({
    name: "",
    email: "",
  });
  const [stateSubmitted, setStateSubmitted] = useState(false);

  const handleChange = (e) => {
    const { name, value } = e.target;
    setFormData((prevData) => ({
      ...prevData,
      [name]: value,
    }));
  };

  const handleStateSubmit = (e) => {
    e.preventDefault();
    setStateSubmitted(true);
  };

  // -------------- useRef Form Implementation --------------
  const nameRef = useRef(null);
  const emailRef = useRef(null);
  const [refSubmittedData, setRefSubmittedData] = useState(null);

  const handleRefSubmit = (e) => {
    e.preventDefault();

    const refData = {
      name: nameRef.current.value,
      email: emailRef.current.value,
    };

    setRefSubmittedData(refData);
  };

  return (
    <div className="form-app">
      <h1>Form Handling in React</h1>
```

Form Handling in React

Form with useState

Name: Abeer Gupta

Email: abeergupta555@gmali.com

Submit

Form Data (Updated as you type):

Name: Abeer Gupta

Email: abeergupta555@gmali.com

Form with useRef

Name: Abeer Gupta

Email: 123@gmail.com

Submit

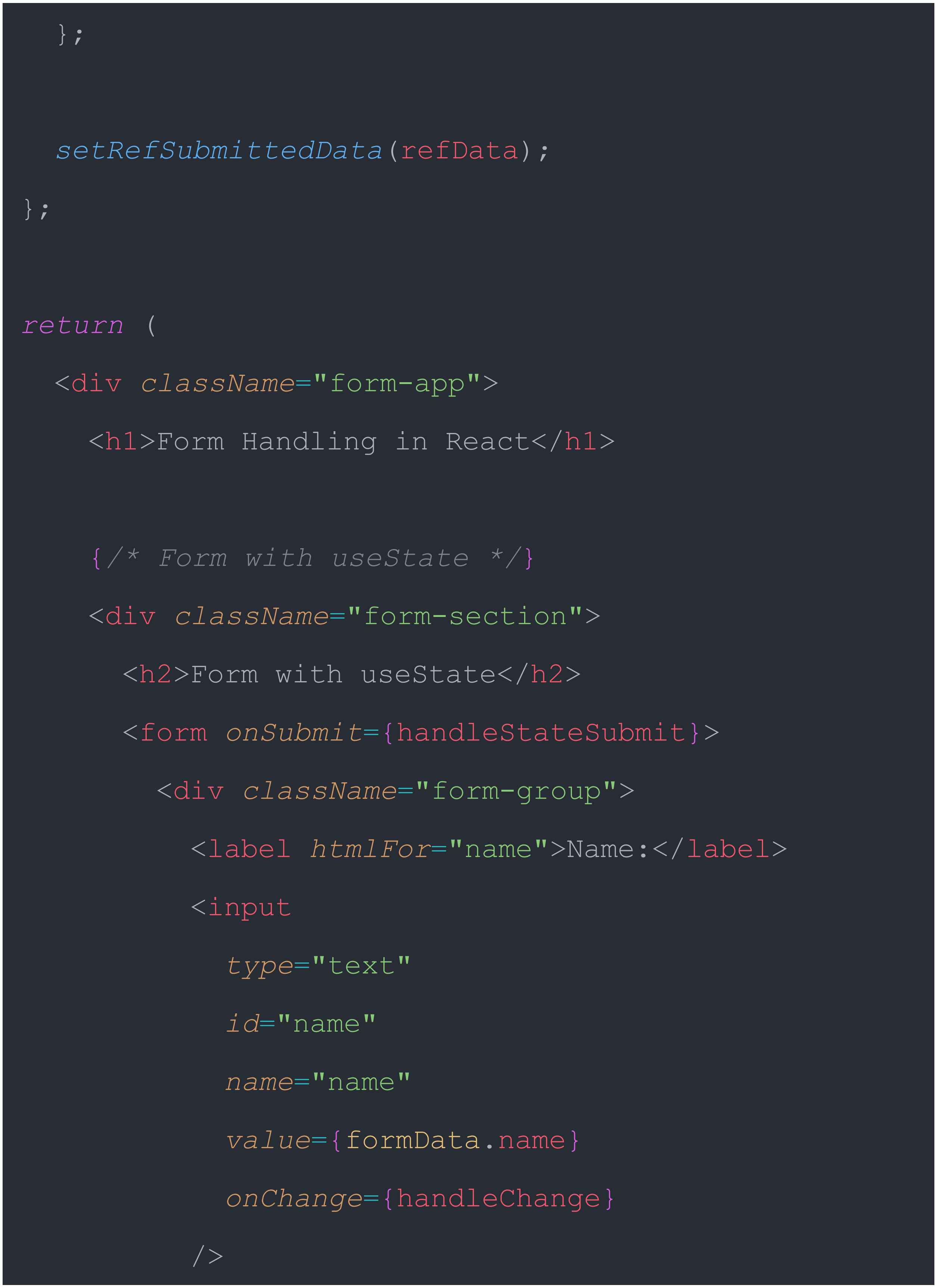
Submitted Data:

Name: Abeer Gupta

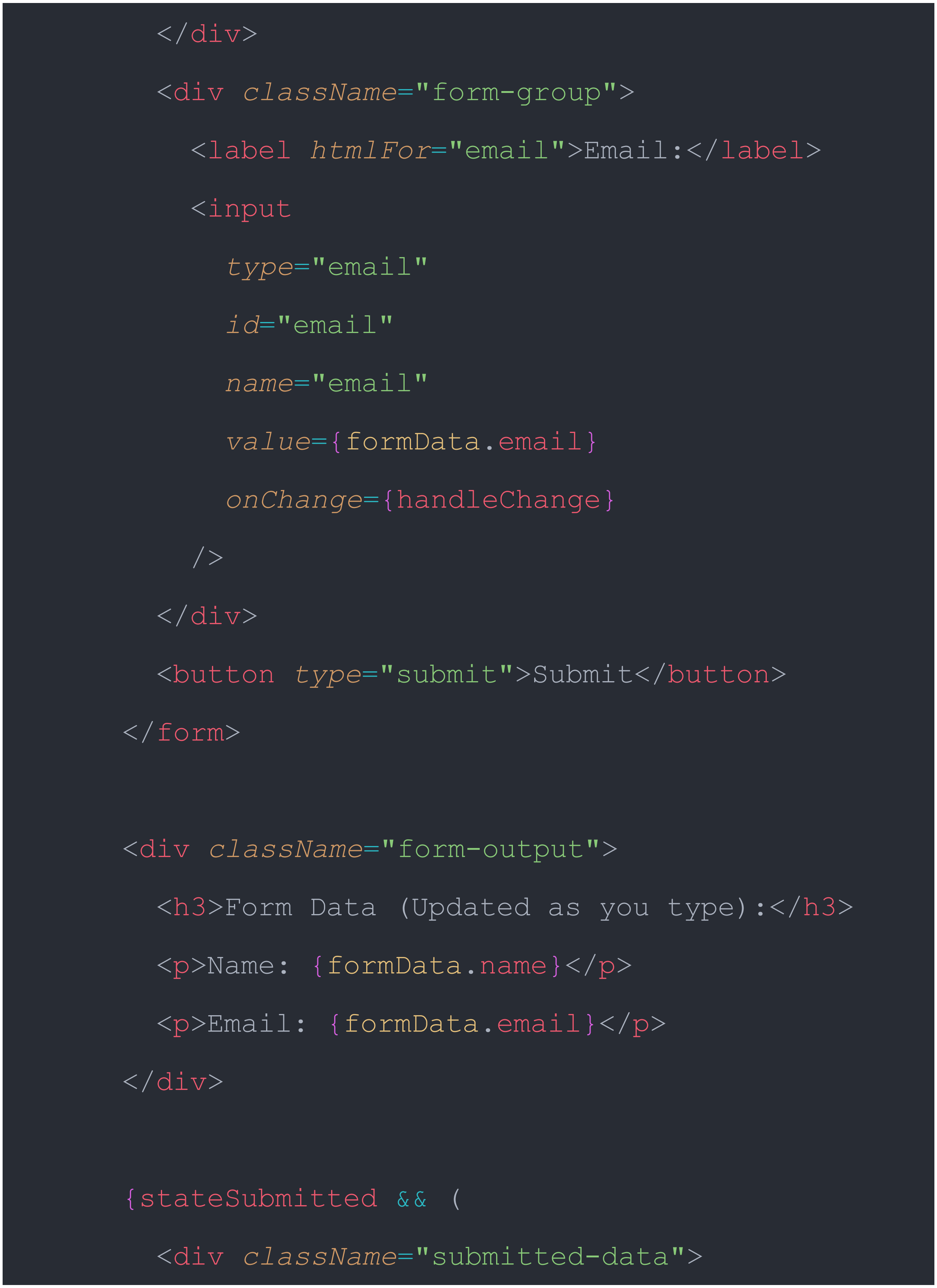
Email: 123@gmail.com

```
import React, { useState, useRef } from "react";


function FormApp() {

 const [formData, setFormData] = useState({

   name: "",

   email: "",

 });

 const [stateSubmitted, setStateSubmitted] =
useState(false);


 const handleChange = (e) => {

   const { name, value } = e.target;
```

```
    setFormData((prevData) => ({
      ...prevData,
      [name]: value,
    }));
  };

  const handleStateSubmit = (e) => {
    e.preventDefault();
    setStateSubmitted(true);
  };

  const nameRef = useRef(null);
  const emailRef = useRef(null);
  const [refSubmittedData, setRefSubmittedData] = useState(null);

  const handleRefSubmit = (e) => {
    e.preventDefault();

    const refData = {
      name: nameRef.current.value,
      email: emailRef.current.value,
```

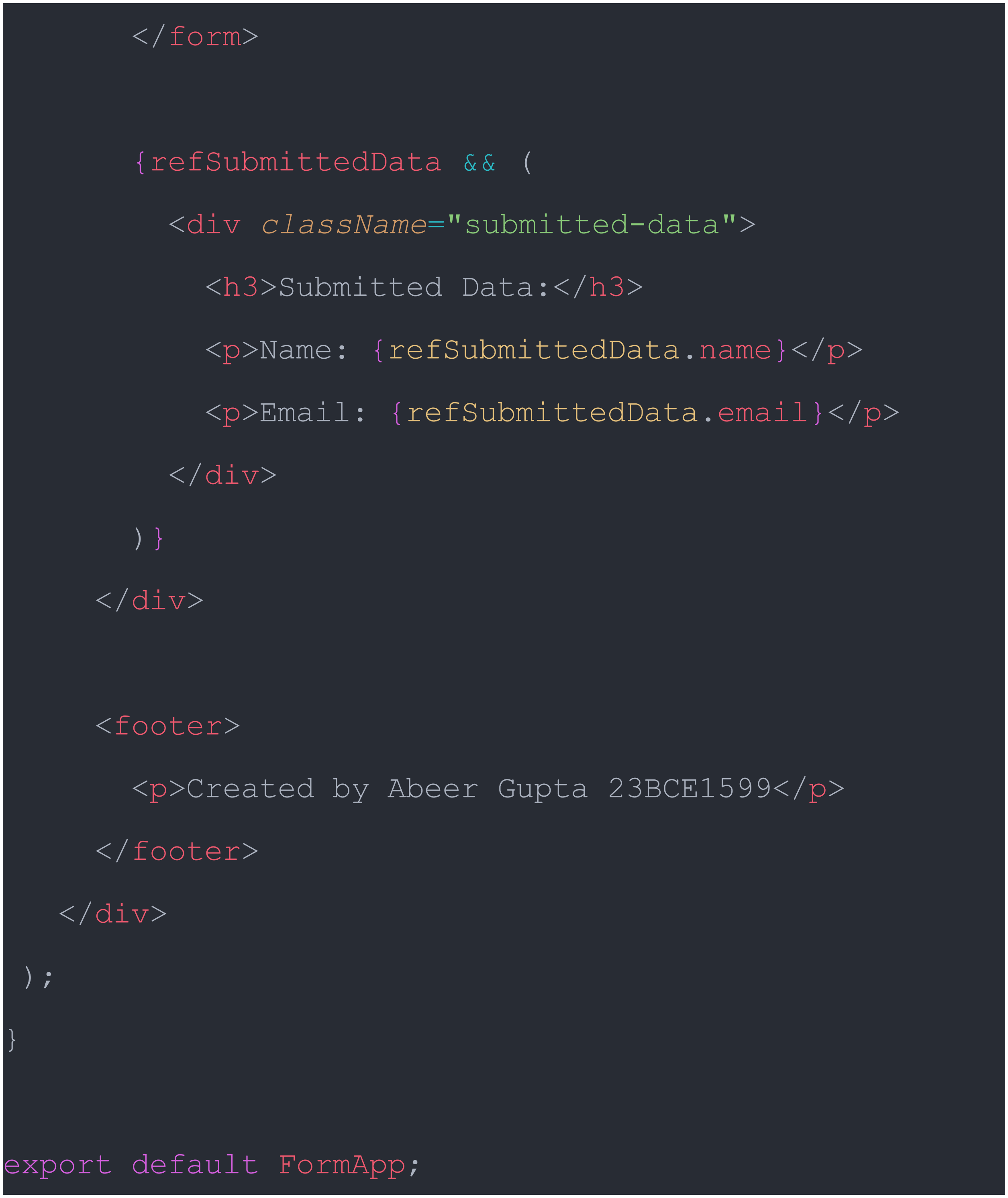


```
  };

  setRefSubmittedData(refData);
};

return (
  <div className="form-app">
    <h1>Form Handling in React</h1>

    {/* Form with useState */}
    <div className="form-section">
      <h2>Form with useState</h2>
      <form onSubmit={handleStateSubmit}>
        <div className="form-group">
          <label htmlFor="name">Name:</label>
          <input
            type="text"
            id="name"
            name="name"
            value={formData.name}
            onChange={handleChange}
          />
```



```
        </div>
        <div className="form-group">
          <label htmlFor="email">Email:</label>
          <input
            type="email"
            id="email"
            name="email"
            value={formData.email}
            onChange={handleChange}
          />
        </div>
        <button type="submit">Submit</button>
      </form>

      <div className="form-output">
        <h3>Form Data (Updated as you type):</h3>
        <p>Name: {formData.name}</p>
        <p>Email: {formData.email}</p>
      </div>

      {stateSubmitted && (
        <div className="submitted-data">
```

```
           <h3>Submitted Data:</h3>
           <p>Name: {formData.name}</p>
           <p>Email: {formData.email}</p>
         </div>
       )}
     </div>

     {/* Form with useRef */}
     <div className="form-section">
       <h2>Form with useRef</h2>
       <form onSubmit={handleRefSubmit}>
         <div className="form-group">
           <label htmlFor="refName">Name:</label>
           <input type="text" id="refName"
ref={nameRef} />
         </div>
         <div className="form-group">
           <label htmlFor="refEmail">Email:</label>
           <input type="email" id="refEmail"
ref={emailRef} />
         </div>
         <button type="submit">Submit</button>
```



```
        </form>

        {refSubmittedData && (
          <div className="submitted-data">
            <h3>Submitted Data:</h3>
            <p>Name: {refSubmittedData.name}</p>
            <p>Email: {refSubmittedData.email}</p>
          </div>
        )}
      </div>

      <footer>
        <p>Created by Abeer Gupta 23BCE1599</p>
      </footer>
    </div>
  );
}

export default FormApp;
```