

# 11. eventexamplesapp

## Objectives

- Explain React events
- Explain about event handlers
- Define Synthetic event
- Identify React event naming convention

In this hands-on lab, you will learn how to:

- Implement Event handling concept in React applications
- Use this keyword
- Use synthetic event

## Prerequisites

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

## Notes

Estimated time to complete this lab: 90 minutes.

Create a React Application “eventexamplesapp” to handle various events of the form elements in HTML.

1. Create “Increment” button to increase the value of the counter and “Decrement” button to decrease the value of the counter. The “Increase” button should invoke multiple methods.
  - a. To increment the value
  - b. Say Hello followed by a static message.
2. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.
3. Create a button which invokes synthetic event “OnPress” which display “I was clicked”

Create a “CurrencyConvertor” component which will convert the Indian Rupees to Euro when the Convert button is clicked.

Handle the Click event of the button to invoke the handleSubmit event and handle the conversion of the euro to rupees.

**CODE:**

### **App.js**

```
import React, { Component } from 'react';
import CurrencyConverter from './components/CurrencyConverter';

class App extends Component {
  constructor(props) {
    super(props);
    this.state = {
      count: 5
    };
  }
}
```

```
increment = () => {  
  this.setState(prev => ({ count: prev.count + 1 }));  
};
```

```
sayHello = () => {  
  alert('Hello! Member1');  
};
```

```
incrementAndGreet = () => {  
  this.increment();  
  this.sayHello();  
};
```

```
decrement = () => {  
  this.setState(prev => ({ count: prev.count - 1 }));  
};
```

```
sayWelcome = (msg) => {  
  alert(`Welcome ${msg}`);  
};
```

```
handleClick = (event) => {  
  alert("I was clicked!");  
};
```

```

render() {
  return (
    <div>
      <p>{this.state.count}</p>
      <button onClick={this.incrementAndGreet}>Increment</button>
      <button onClick={this.decrement}>Decrement</button>
      <br /><br />
      <button onClick={() => this.sayWelcome("welcome")}>Say
welcome</button>
      <br /><br />
      <button onClick={this.handleClick}>Click on me</button>
      <br /><br />
      <CurrencyConverter />
    </div>
  );
}
}

```

```
export default App;
```

## CurrencyConverter.js

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

```

class CurrencyConverter extends Component {
  constructor(props) {

```

```
super(props);  
this.state = {  
  amount: '',  
  currency: ''  
};  
}
```

```
handleAmountChange = (e) => {  
  this.setState({ amount: e.target.value });  
};
```

```
handleCurrencyChange = (e) => {  
  this.setState({ currency: e.target.value });  
};
```

```
handleSubmit = (e) => {  
  e.preventDefault();
```

```
  const { amount, currency } = this.state;
```

```
  if (currency.toLowerCase() === 'euro') {  
    const converted = parseFloat(amount) * 0.011; // Example rate  
    alert(`Converted amount in Euro: €${converted.toFixed(2)}`);  
  } else {  
    alert('Only Euro conversion is supported.');
```

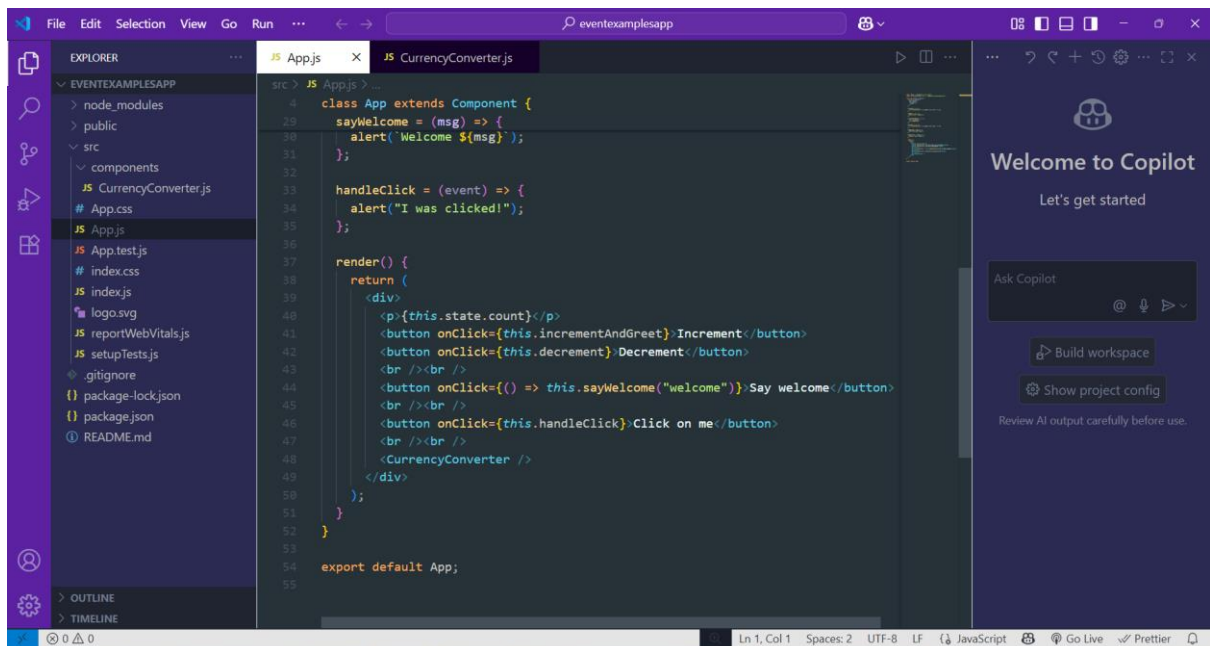
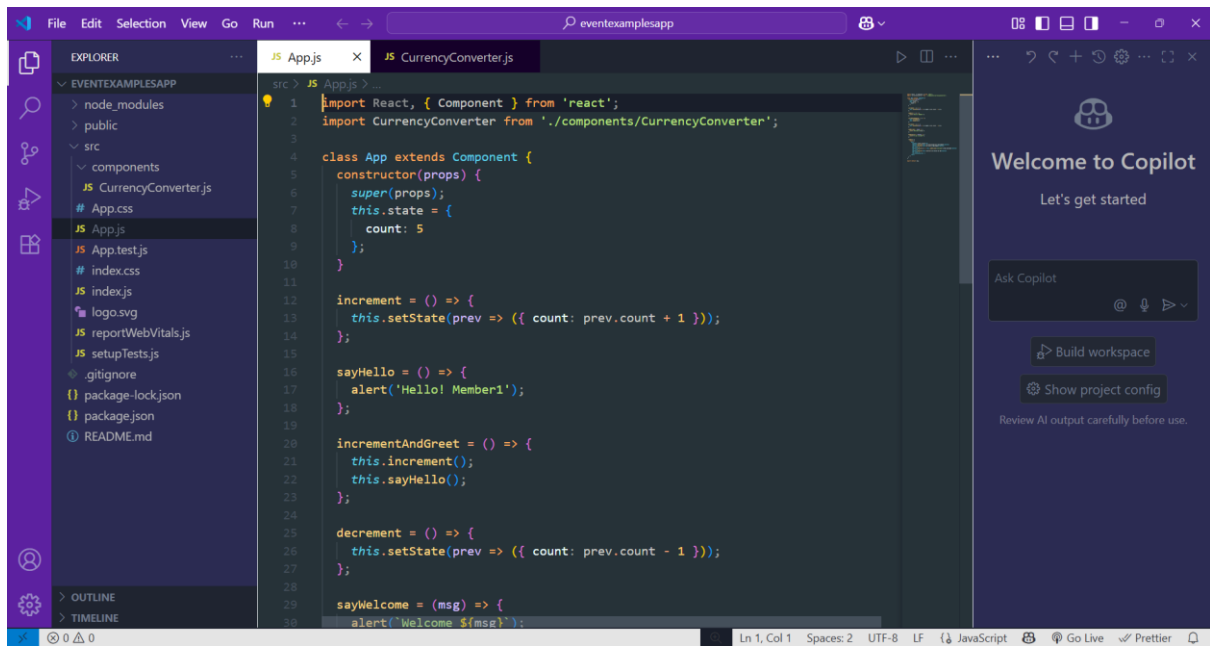
```

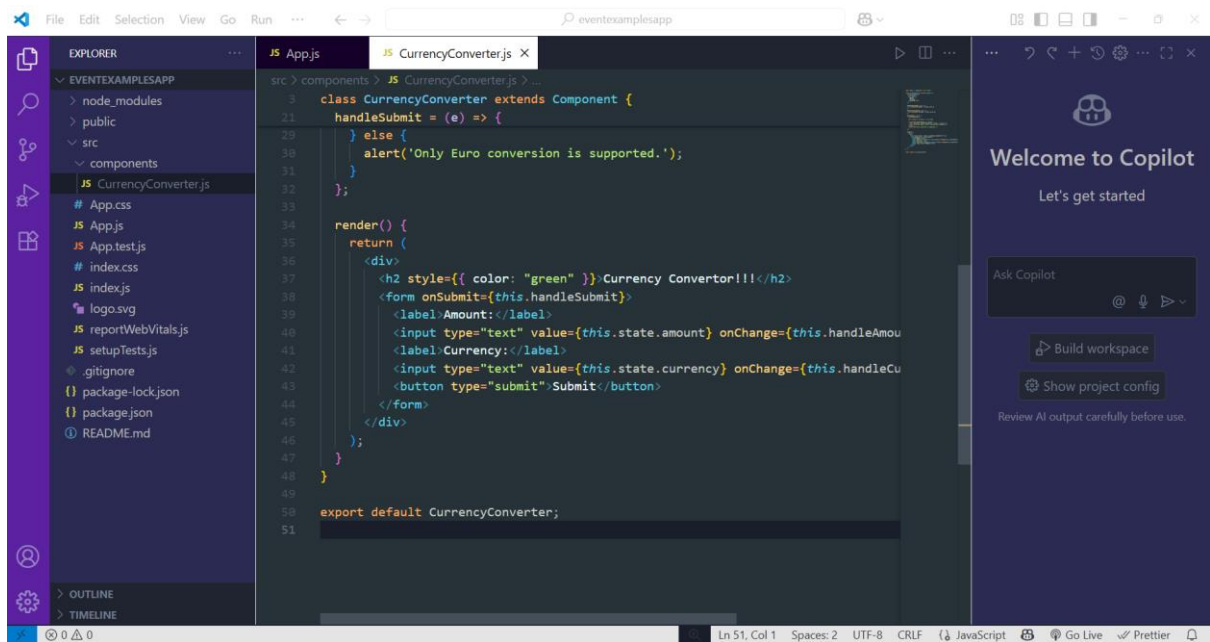
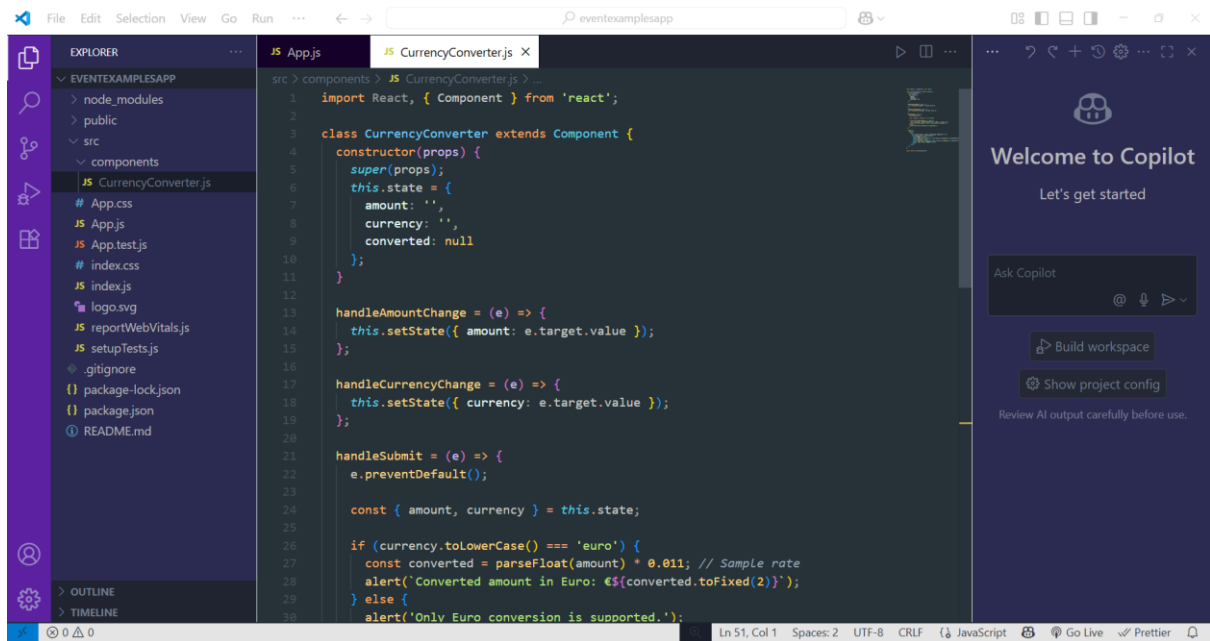
    }
};

render() {
    return (
        <div>
            <h2 style={{ color: "green" }}>Currency Converter!!!</h2>
            <form onSubmit={this.handleSubmit}>
                <label>Amount:</label>
                <input type="text" value={this.state.amount}
onChange={this.handleAmountChange} /><br /><br />
                <label>Currency:</label>
                <input type="text" value={this.state.currency}
onChange={this.handleCurrencyChange} /><br /><br />
                <button type="submit">Submit</button>
            </form>
        </div>
    );
}
}

export default CurrencyConverter;

```







```
Windows PowerShell
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mkanm>cd eventexamplesapp
C:\Users\mkanm\eventexamplesapp>npm start

> eventexamplesapp@0.1.0 start
> react-scripts start

(node:25884) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE] DeprecationWarning: 'onAfterSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
(Use 'node --trace-deprecation ...' to show where the warning was created)
(node:25884) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE] DeprecationWarning: 'onBeforeSetupMiddleware' option is deprecated. Please use the 'setupMiddlewares' option.
Starting the development server...
Compiled successfully!

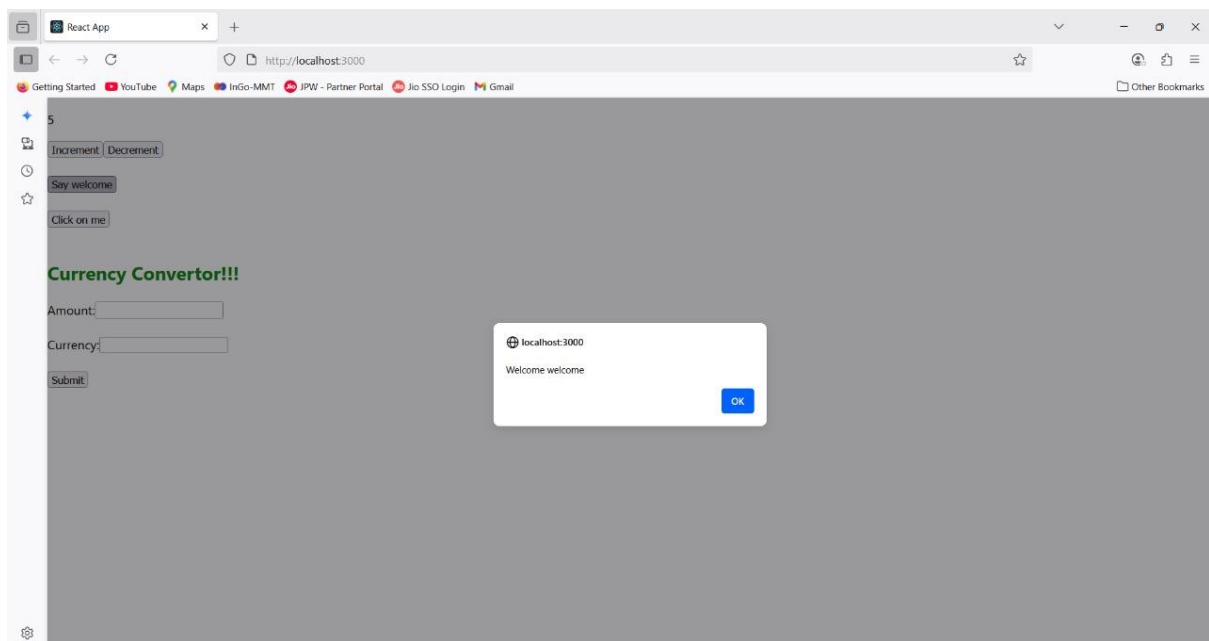
You can now view eventexamplesapp in the browser.

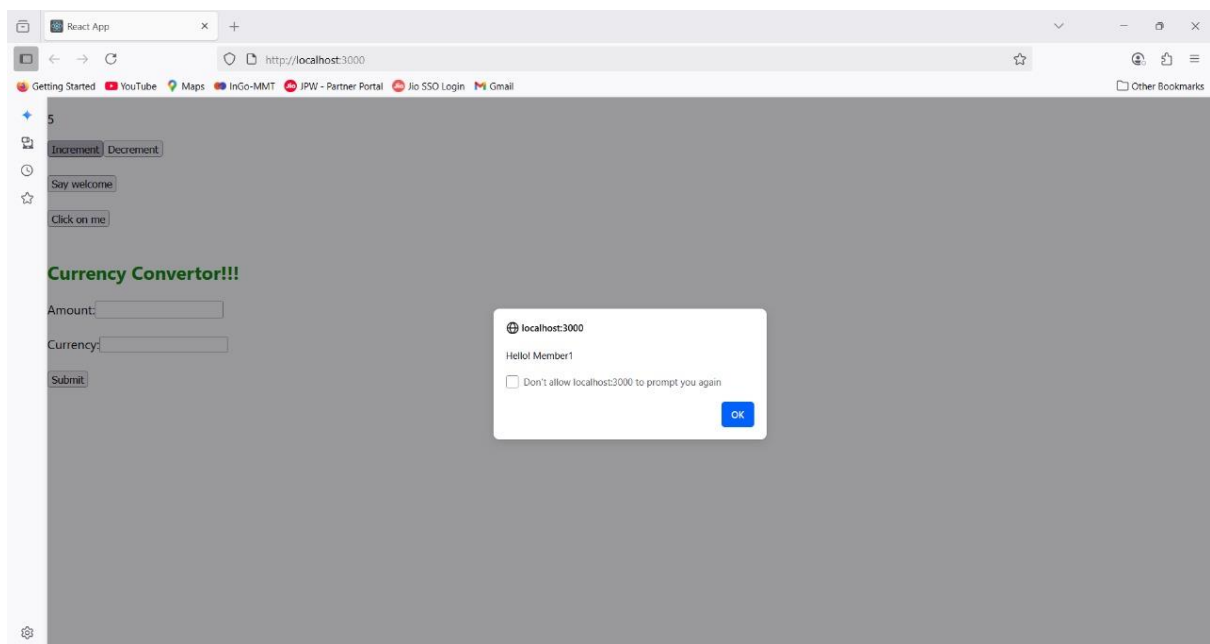
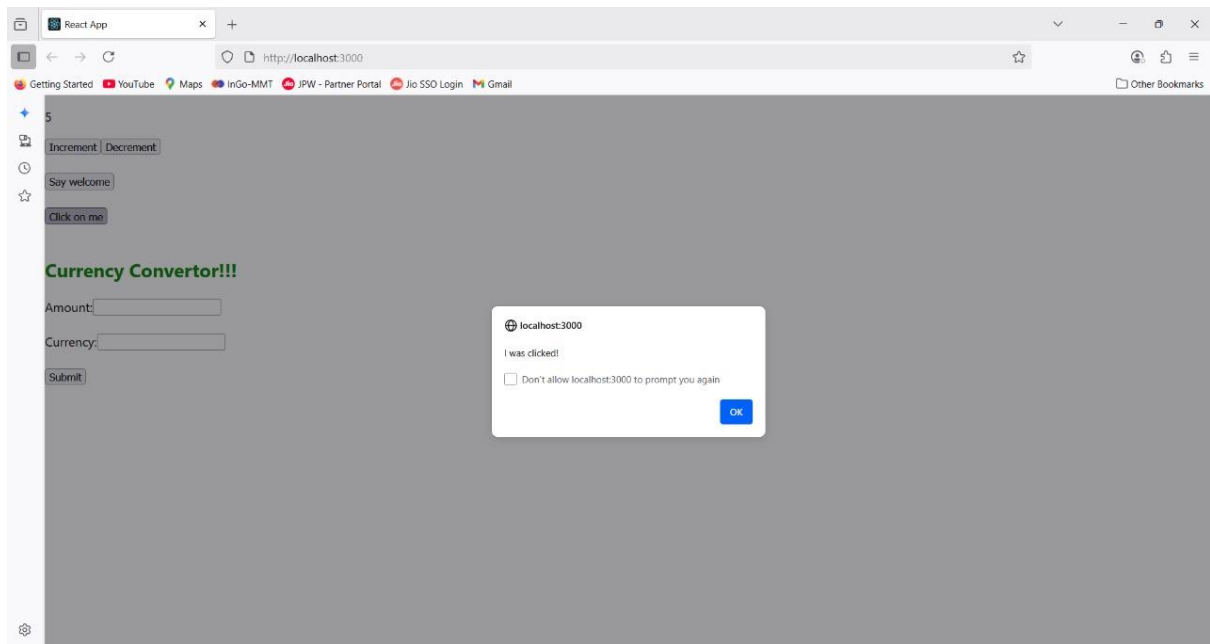
Local:      http://localhost:3000
On Your Network:  http://192.168.0.135:3000

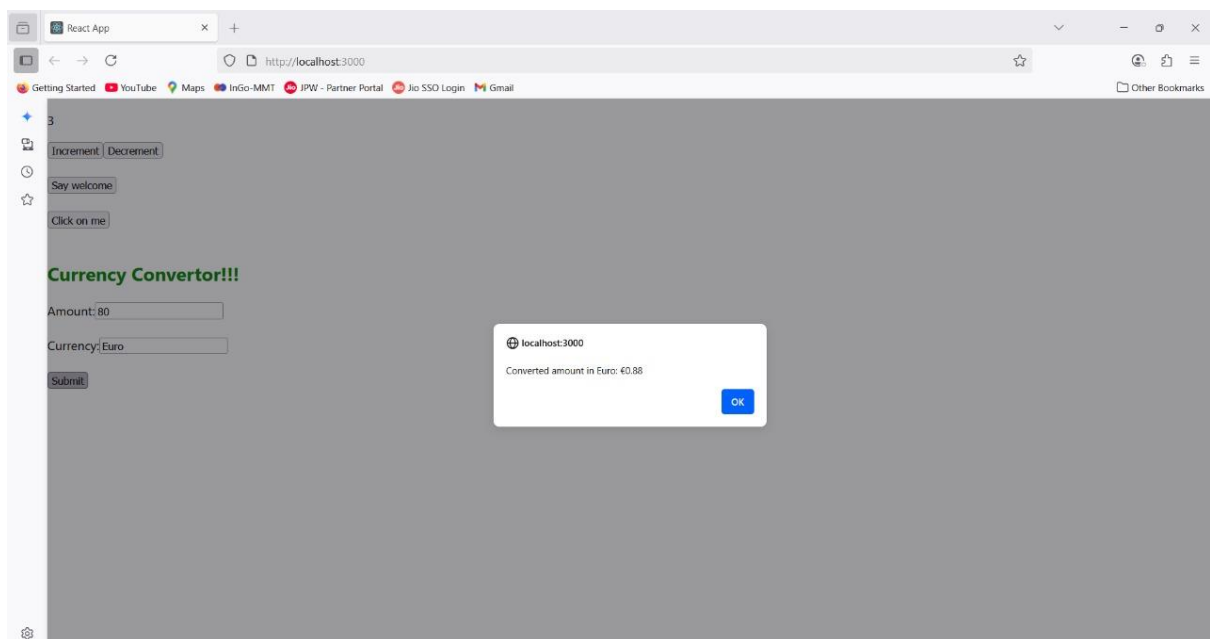
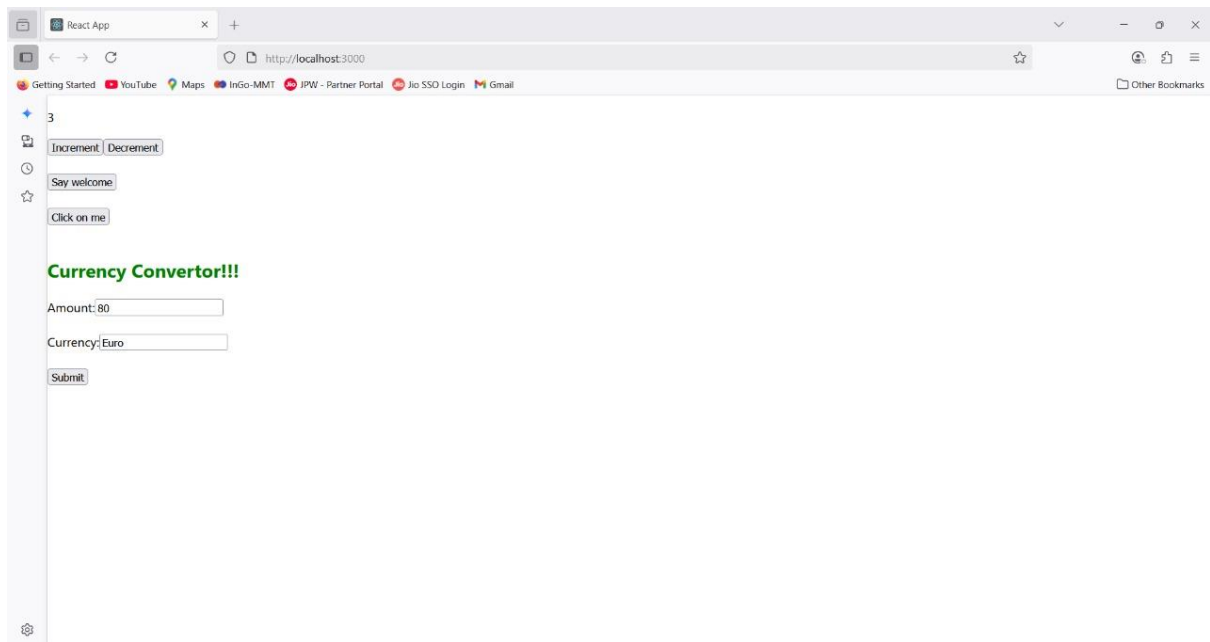
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

## OUTPUT:







KANMANI MURUGHAIYAN  
SUPERSET ID: 6407636

