

Name:- Vidya CS

TRACK:JAVA FSE

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.

Step 1: Create the React App(eventexamplesapp)

```
PS C:\Users\DELL\OneDrive\Documents\cogni handson> npx create-react-app eventexamplesapp
>>

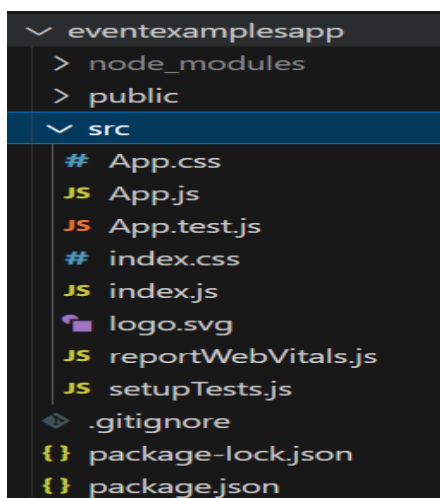
Creating a new React app in C:\Users\DELL\OneDrive\Documents\cogni handson\eventexamplesapp.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

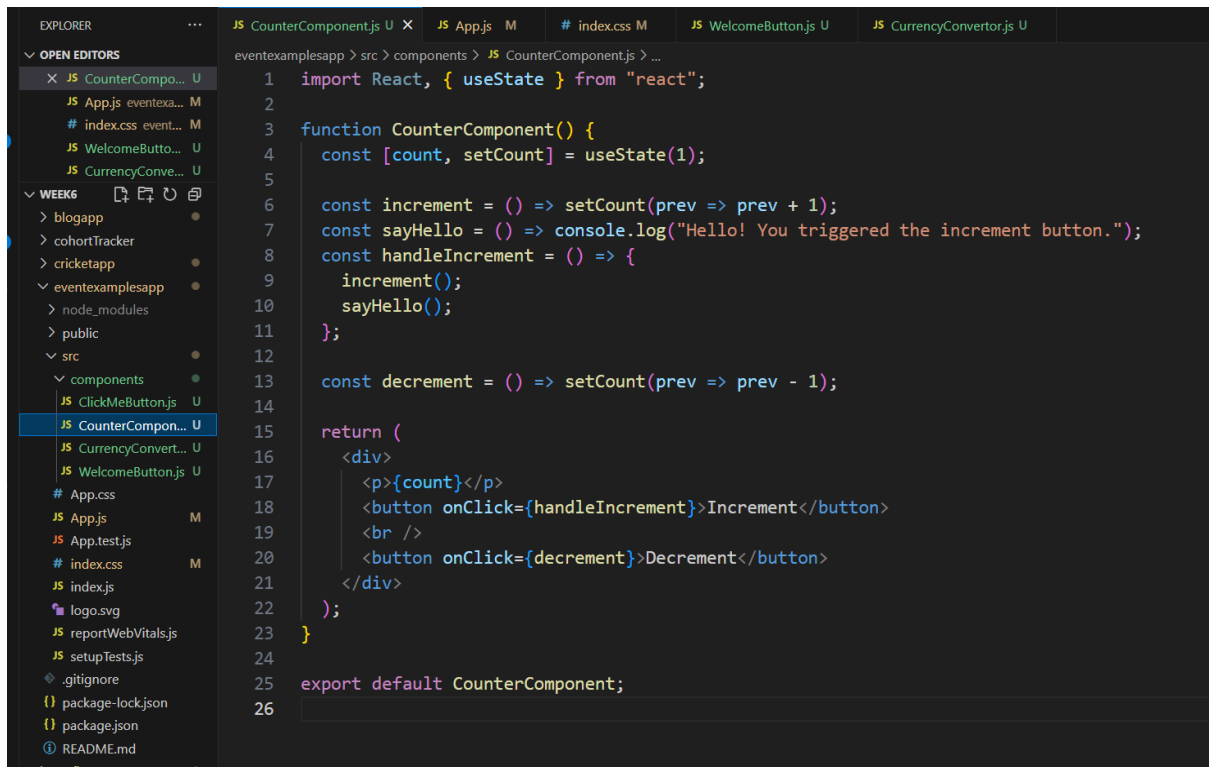
added 1323 packages in 10m

269 packages are looking for funding
  run `npm fund` for details

Initialized a git repository.
```



Step 2: Create components/CounterComponent.js



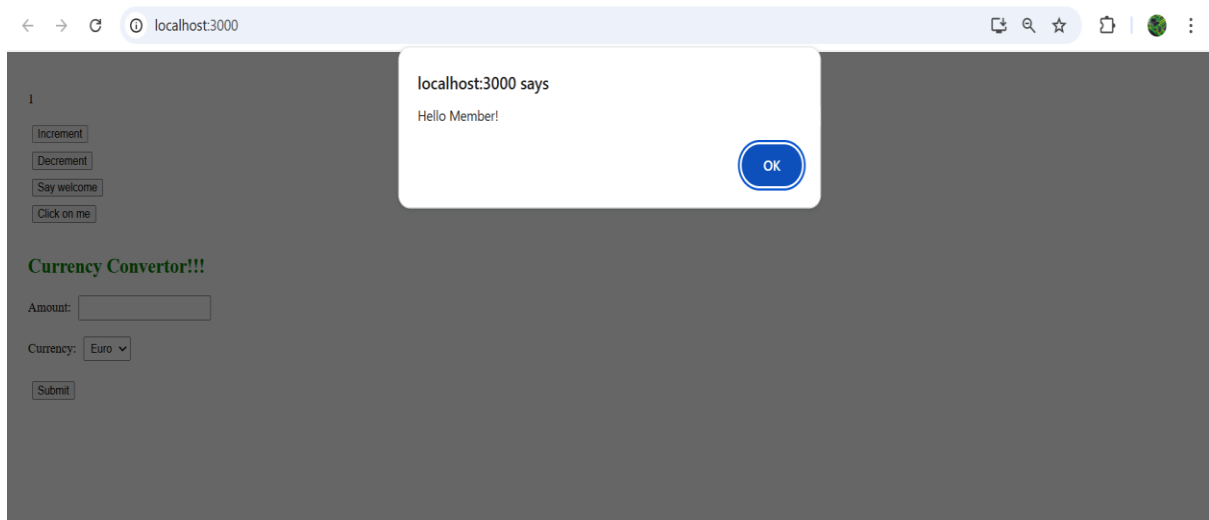
The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following files and folders:

- eventexamplesapp > src > components > JS CounterComponent.js (selected)
- eventexamplesapp > src > components > JS ClickMeButton.js
- eventexamplesapp > src > components > JS CurrencyConvertor.js
- eventexamplesapp > src > components > JS WelcomeButton.js
- eventexamplesapp > src > components > # App.css
- eventexamplesapp > src > components > JS App.js
- eventexamplesapp > src > components > JS App.test.js
- eventexamplesapp > src > components > # index.css
- eventexamplesapp > src > components > JS index.js
- eventexamplesapp > src > components > logo.svg
- eventexamplesapp > src > components > JS reportWebVitals.js
- eventexamplesapp > src > components > JS setupTests.js
- eventexamplesapp > src > components > .gitignore
- eventexamplesapp > src > components > package-lock.json
- eventexamplesapp > src > components > package.json
- eventexamplesapp > src > components > README.md

The code editor shows the following code in CounterComponent.js:

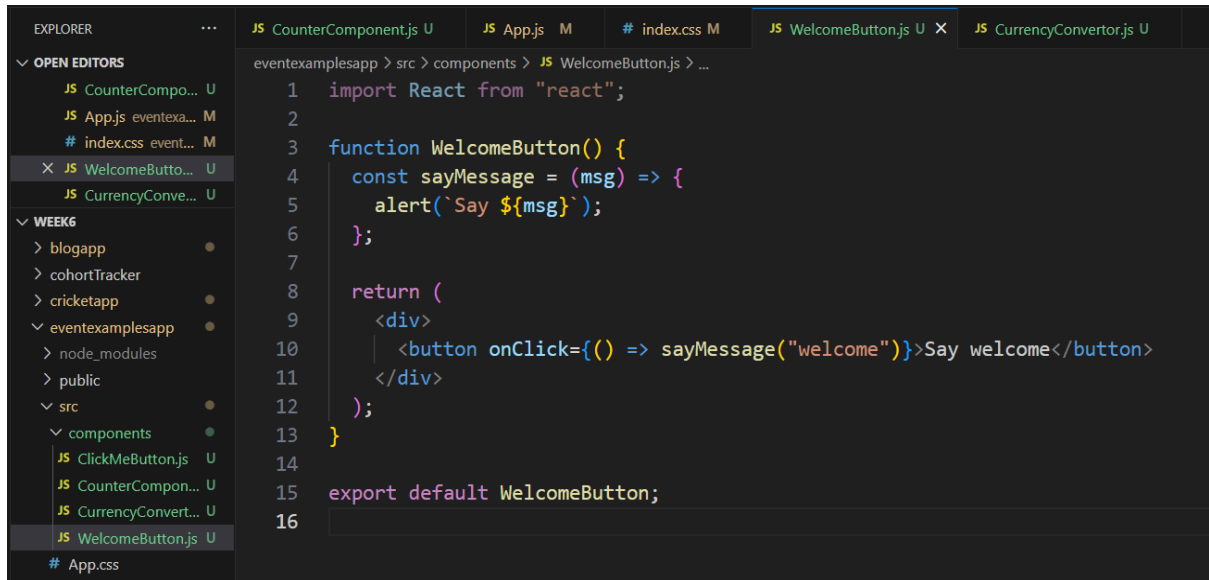
```
1 import React, { useState } from "react";
2
3 function CounterComponent() {
4   const [count, setCount] = useState(1);
5
6   const increment = () => setCount(prev => prev + 1);
7   const sayHello = () => console.log("Hello! You triggered the increment button.");
8   const handleIncrement = () => {
9     increment();
10    sayHello();
11  };
12
13  const decrement = () => setCount(prev => prev - 1);
14
15  return (
16    <div>
17      <p>{count}</p>
18      <button onClick={handleIncrement}>Increment</button>
19      <br />
20      <button onClick={decrement}>Decrement</button>
21    </div>
22  );
23 }
24
25 export default CounterComponent;
26
```

OUTPUT:-



2. Create a button “Say Welcome” which invokes the function which takes “welcome” as an argument.

Step 1: Create components/ WelcomeButton.js



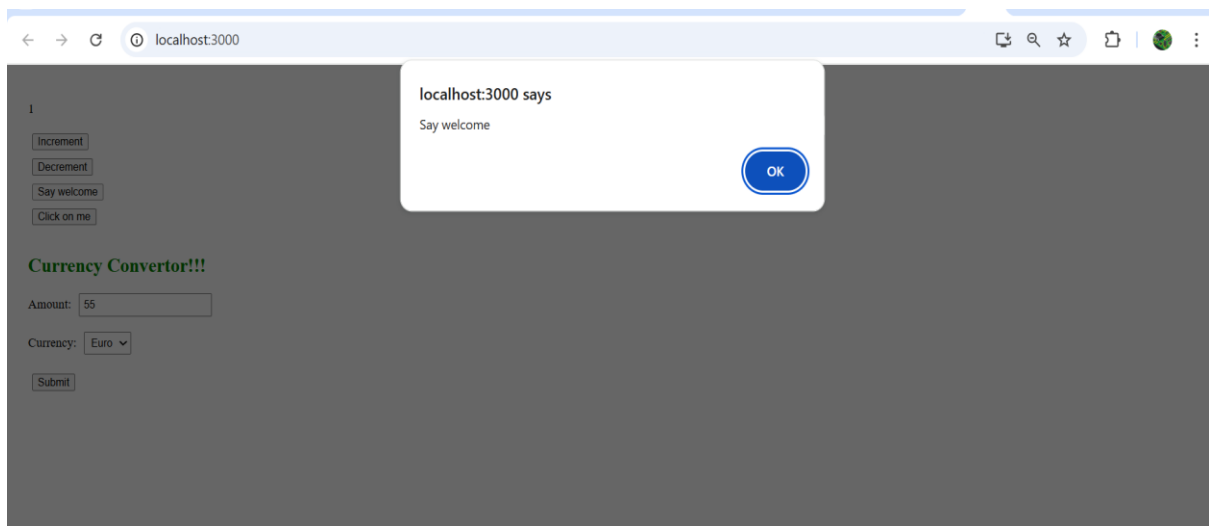
The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the following files and folders:

- OPEN EDITORS
 - JS CounterCompo... U
 - JS App.js eventexa... M
 - # index.css event... M
 - JS WelcomeButto... U (selected)
 - JS CurrencyConve... U
- WEEK6
 - blogapp
 - cohortTracker
 - cricketapp
 - eventexamplesapp
 - node_modules
 - public
 - src
 - components
 - JS ClickMeButton.js U
 - JS CounterCompon... U
 - JS CurrencyConvert... U
 - JS WelcomeButton.js U (selected)

The code editor shows the following code in the WelcomeButton.js file:

```
1 import React from "react";
2
3 function WelcomeButton() {
4   const sayMessage = (msg) => {
5     alert(`Say ${msg}`);
6   };
7
8   return (
9     <div>
10       <button onClick={() => sayMessage("welcome")}>Say welcome</button>
11     </div>
12   );
13 }
14
15 export default WelcomeButton;
```

OUTPUT:-



3 . Create a button which invokes synthetic event “OnPress” which display “I was clicked”

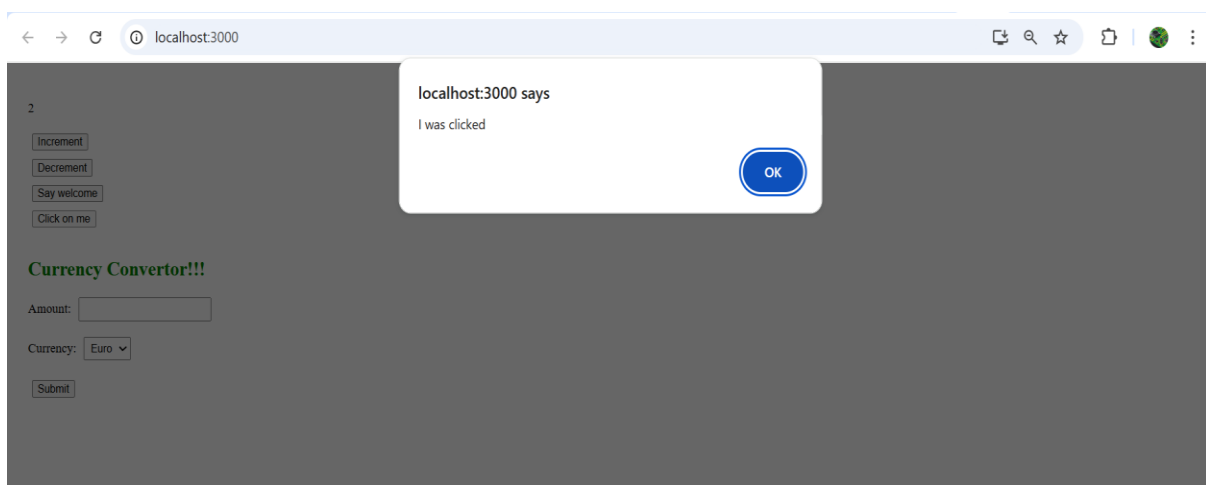
Step 1: Create components/ ClickMeButton.js



The screenshot shows the VS Code editor with the file `ClickMeButton.js` open in the `components` directory of `eventexamplesapp`. The file contains the following code:

```
1 import React from "react";
2
3 function ClickMeButton() {
4   const handleClick = (event) => {
5     alert("I was clicked");
6   };
7
8   return (
9     <div>
10      <button onClick={handleClick}>Click on me</button>
11    </div>
12  );
13 }
14
15 export default ClickMeButton;
16
```

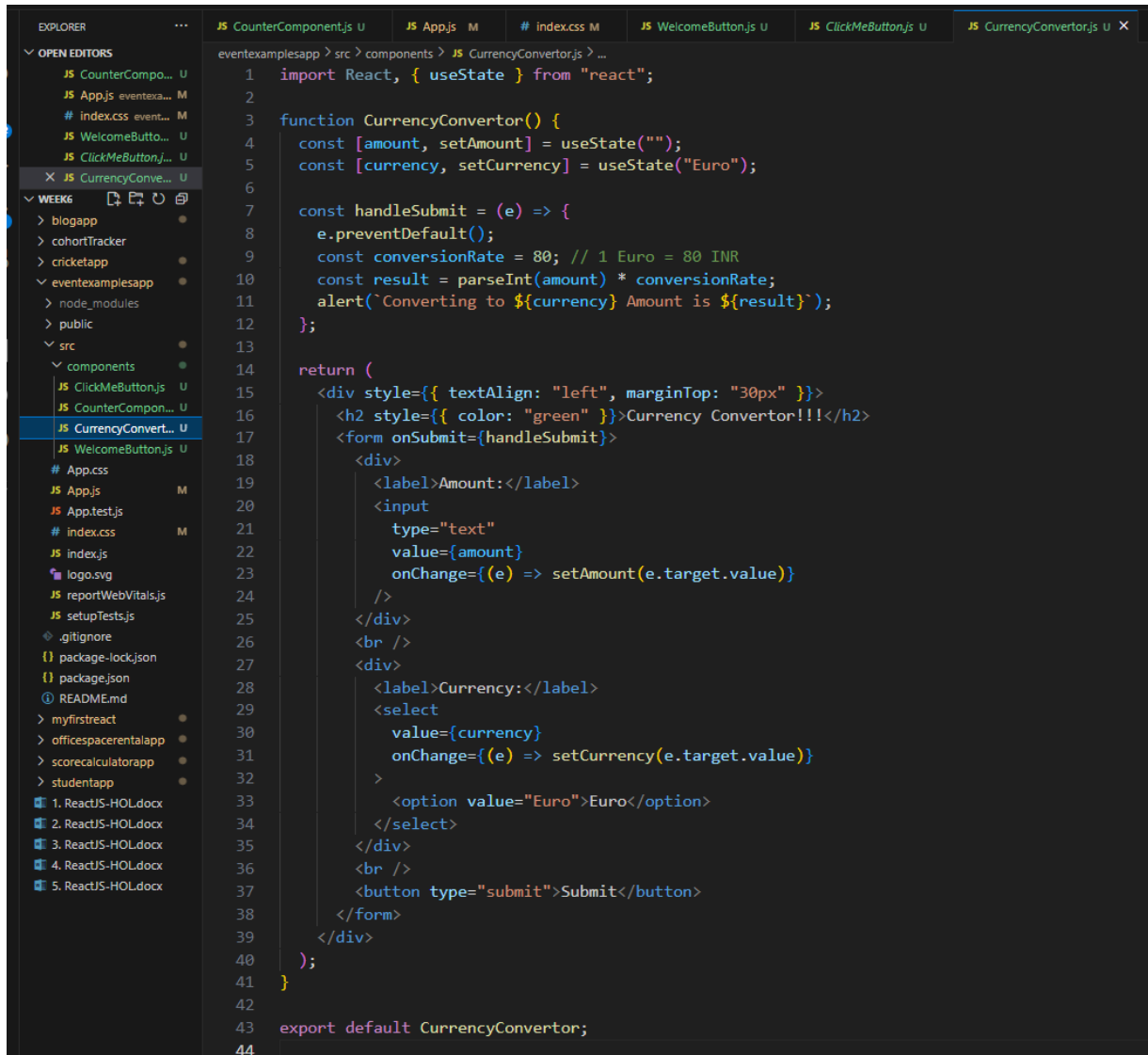
OUTPUT:-



4 .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.

Step 1: Create components/ CurrencyConvertor.js



```
1 import React, { useState } from "react";
2
3 function CurrencyConvertor() {
4   const [amount, setAmount] = useState("");
5   const [currency, setCurrency] = useState("Euro");
6
7   const handleSubmit = (e) => {
8     e.preventDefault();
9     const conversionRate = 80; // 1 Euro = 80 INR
10    const result = parseInt(amount) * conversionRate;
11    alert(`Converting to ${currency} Amount is ${result}`);
12  };
13
14  return (
15    <div style={{ textAlign: "left", marginTop: "30px" }}>
16      <h2 style={{ color: "green" }}>Currency Convertor!!!</h2>
17      <form onSubmit={handleSubmit}>
18        <div>
19          <label>Amount:</label>
20          <input
21            type="text"
22            value={amount}
23            onChange={(e) => setAmount(e.target.value)}
24          />
25        </div>
26        <br />
27        <div>
28          <label>Currency:</label>
29          <select
30            value={currency}
31            onChange={(e) => setCurrency(e.target.value)}
32          >
33            <option value="Euro">Euro</option>
34          </select>
35        </div>
36        <br />
37        <button type="submit">Submit</button>
38      </form>
39    </div>
40  );
41 }
42
43 export default CurrencyConvertor;
44
```

OUTPUT:-

