11. ReactJS-HOL

Code:

App.js:

```
import React from 'react';

import './App.css';

import Counter from './Counter';

import CurrencyConverter from './CurrencyConverter';


function App() {

  return (

    <div className="App">

      <section className="card">

        <Counter />

      </section>

      <section className="card">

        <button onClick={() => sayWelcome('welcome')}>Say Welcome</button>

        <br/><br/>

        <button onClick={() => handlePress()}>Click on me </button>

      </section>

      <section className="card">

        <CurrencyConverter />

      </section>

    </div>

  );

}

function sayWelcome(message) {

  alert(message);

}

function handlePress() {

  alert('I was clicked');

}

export default App;
```

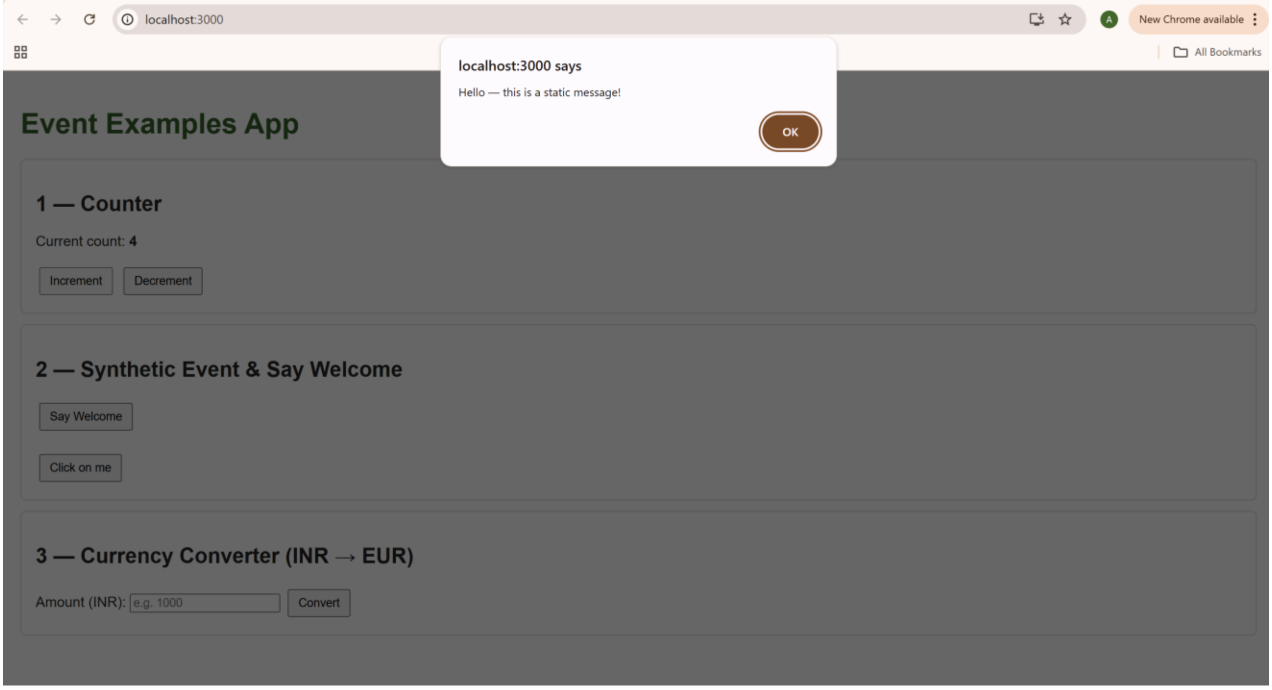
Counter.js:

```
import React, { useState } from 'react';
export default function Counter() {
  const [count, setCount] = useState(0);
  function incrementValue() {
    setCount(prev => prev + 1);
  }
  function sayHello() {
    console.log('Hello — this is a static message!');
    alert('Hello — this is a static message!');
  }
  function handleIncrement() {
    incrementValue();
    sayHello();
  }
  function handleDecrement() {
    setCount(prev => prev - 1);
  }
  return (
    <div>
      <p>Current count: <strong>{count}</strong></p>
      <button onClick={handleIncrement}>Increment</button>
      {' '}
      <button onClick={handleDecrement}>Decrement</button>
    </div>
  );
}
```

CurrencyConverter.java:

```
import React, { useState } from 'react';
export default function CurrencyConverter() {
  const [inr, setInr] = useState('');
  const [result, setResult] = useState(null);
const INR_TO_EUR = 0.011;
function handleSubmit(e) {
  e.preventDefault();
  const numeric = parseFloat(inr);
  if (isNaN(numeric)) {
    alert('Enter a valid number for INR');
    return;
  }
  const euro = numeric * INR_TO_EUR;
  setResult(euro.toFixed(2));
  alert('Converting to Euro Amount is ' + euro.toFixed(2));
}
return (
  <div>
    <form onSubmit={handleSubmit}>
      <label>
        Amount (INR):{' '}
        <input
          type="text"
          value={inr}
          onChange={(e) => setInr(e.target.value)}
          placeholder="e.g. 1000"
        />
      </label>
      {' '}
      <button type="submit">Convert</button>
    </form>
    {result !== null && (
      <div style={{ marginTop: '10px' }}>
        <strong>Result:</strong> € {result}
      </div>
    )}
  </div>
);}
```

Output:



localhost:3000 says

Hello — this is a static message!

Event Examples App

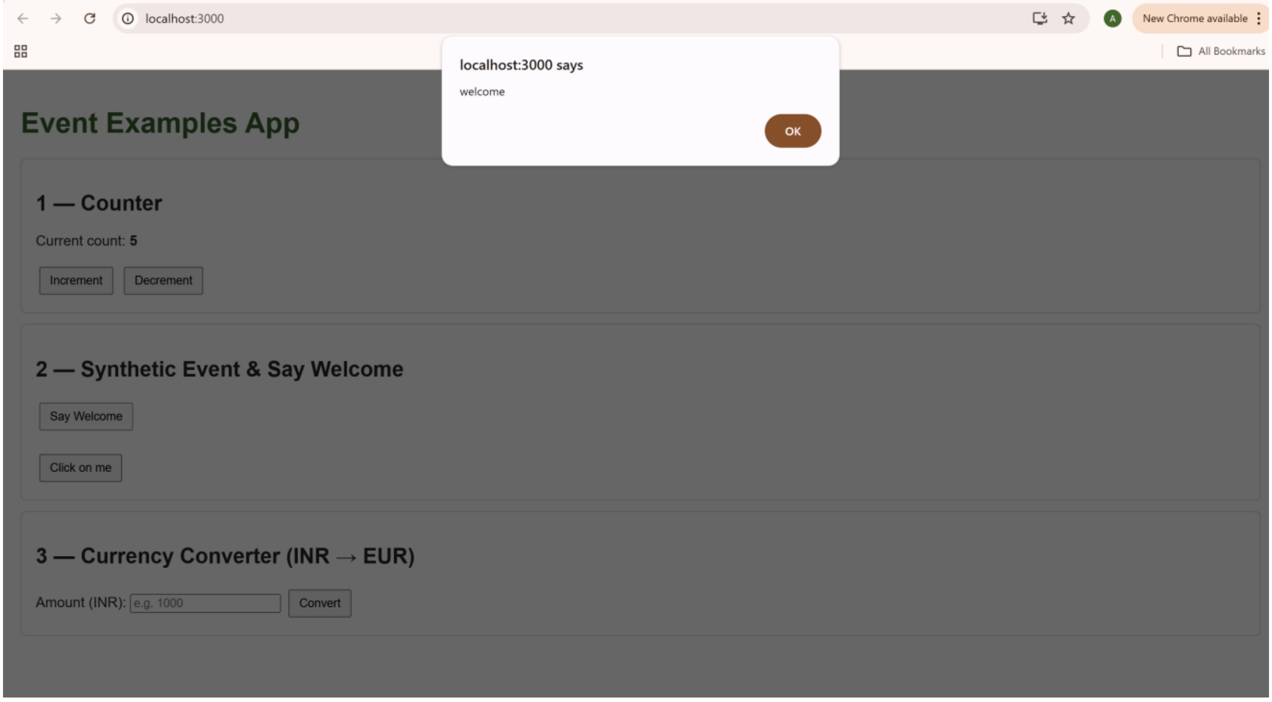
1 — Counter

Current count: **4**

2 — Synthetic Event & Say Welcome

3 — Currency Converter (INR → EUR)

Amount (INR): e.g. 1000



localhost:3000 says

welcome

Event Examples App

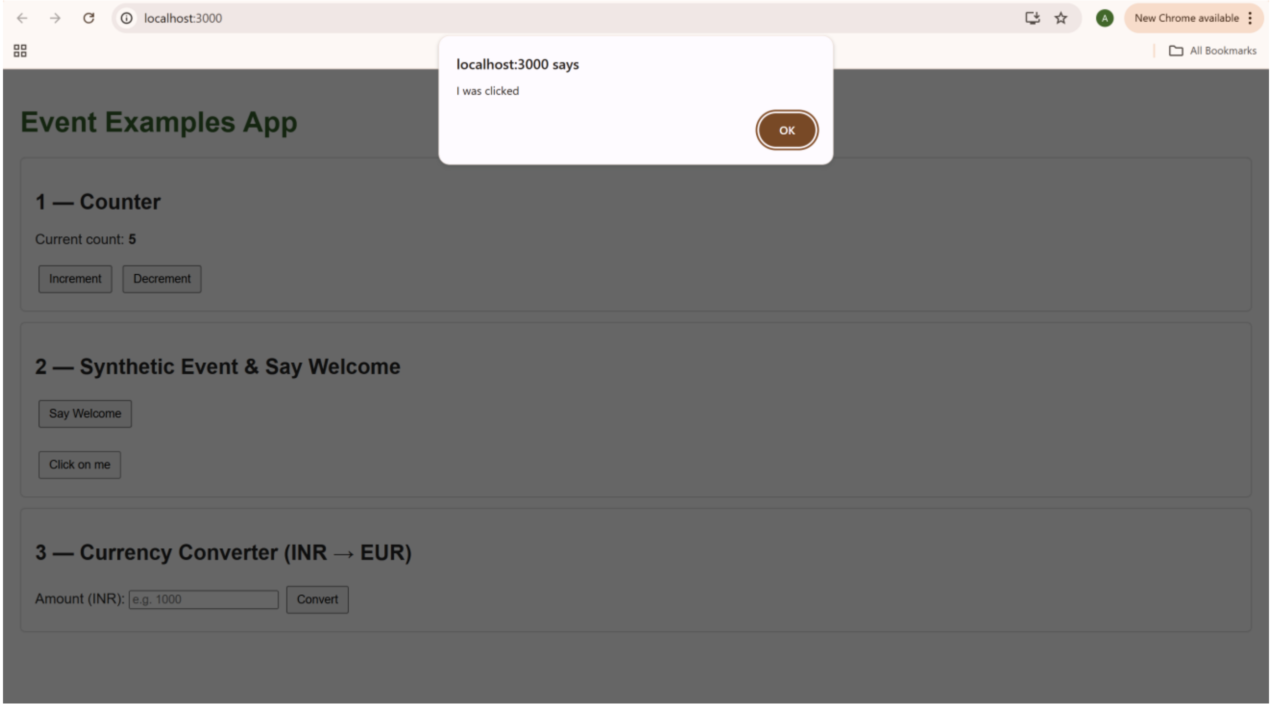
1 — Counter

Current count: **5**

2 — Synthetic Event & Say Welcome

3 — Currency Converter (INR → EUR)

Amount (INR): e.g. 1000



localhost:3000

localhost:3000 says

I was clicked

Event Examples App

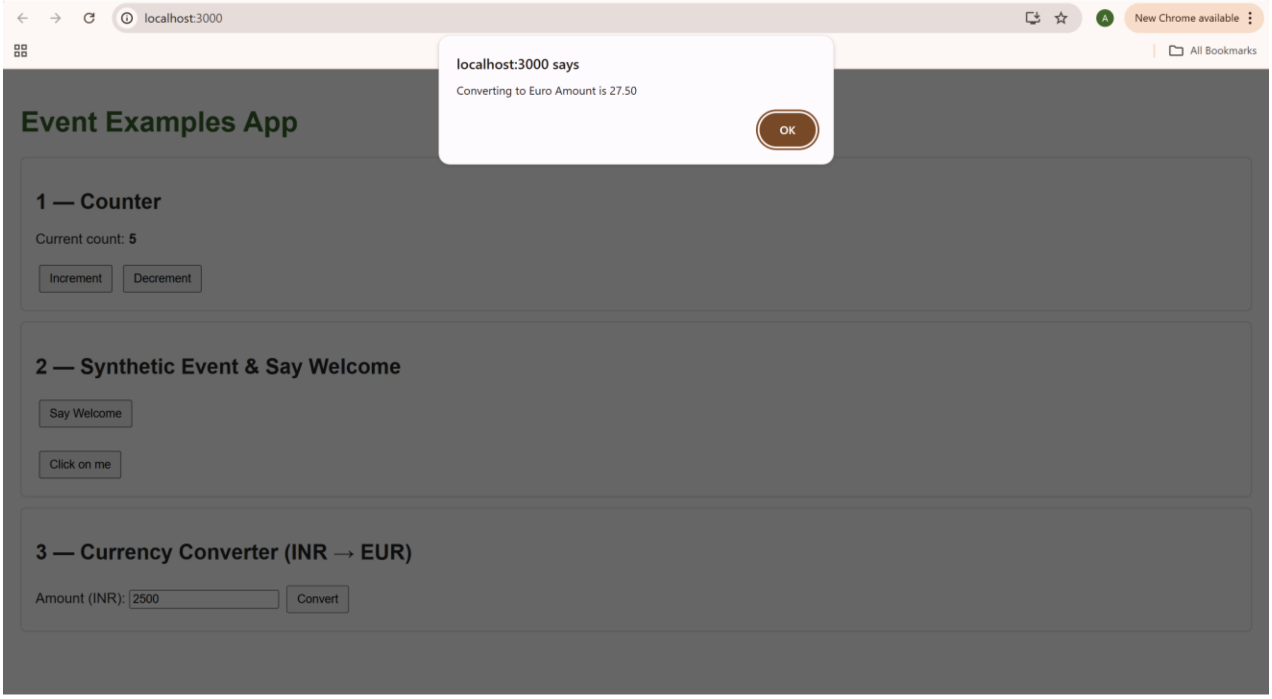
1 — Counter

Current count: **5**

2 — Synthetic Event & Say Welcome

3 — Currency Converter (INR → EUR)

Amount (INR): e.g. 1000



localhost:3000

localhost:3000 says

Converting to Euro Amount is 27.50

Event Examples App

1 — Counter

Current count: **5**

2 — Synthetic Event & Say Welcome

3 — Currency Converter (INR → EUR)

Amount (INR): 2500