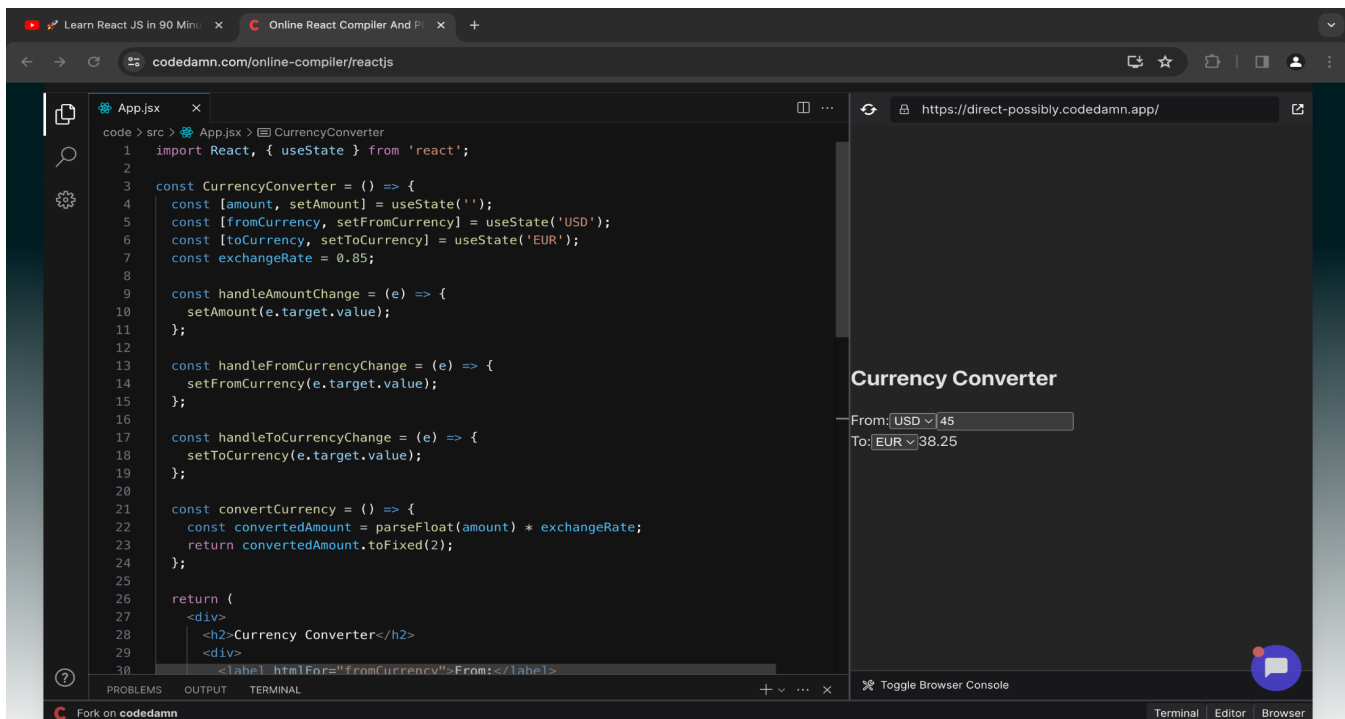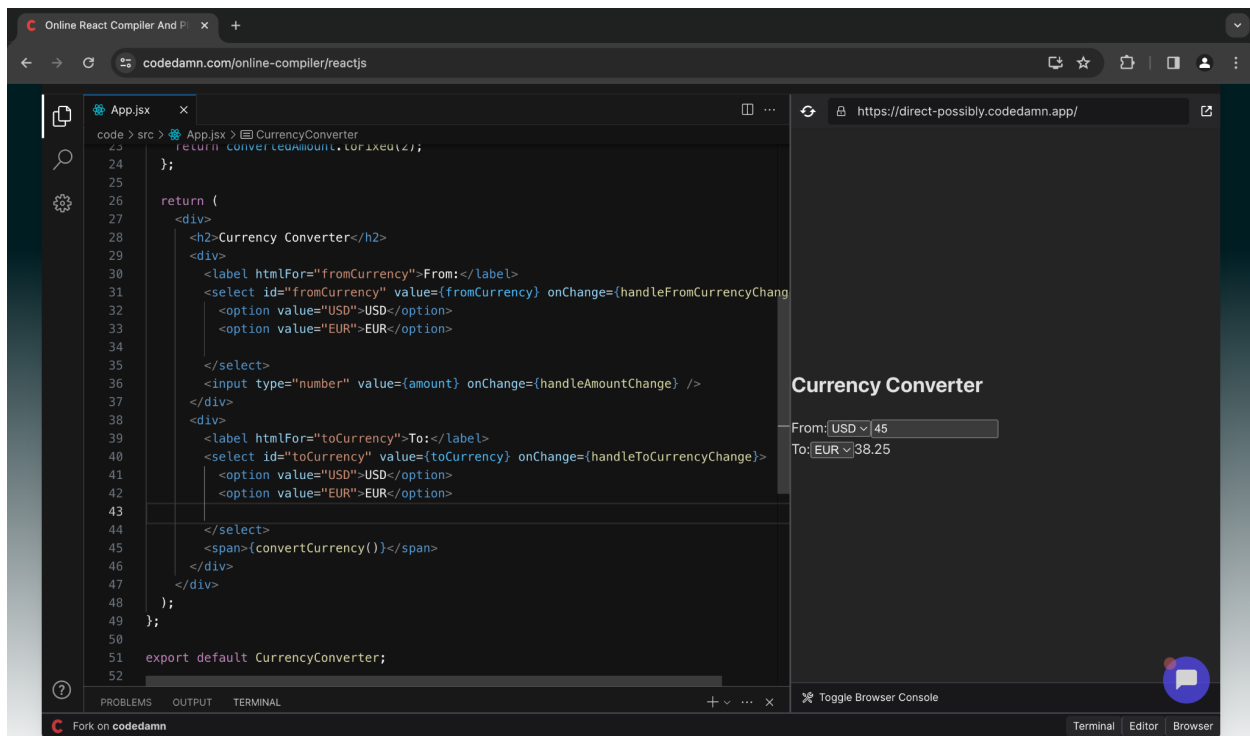# Web-Technology
Lab-8

Name – Nitesh saini
Roll n. – 22cs2015

Q1- Develop a currency converter application that allows users to input an amount in one currency and convert it to another. For the sake of this challenge, you can use a hard-coded exchange rate. Take advantage of React state and event handlers to manage the input and conversion calculations.

```
App.jsx ×

code > src > App.jsx > CurrencyConverter

23        return convertedAmount.toFixed(2);
24    };
25
26    return (
27      <div>
28        <h2>Currency Converter</h2>
29        <div>
30          <label htmlFor="fromCurrency">From:</label>
31          <select id="fromCurrency" value={fromCurrency} onChange={handleFromCurrencyChang
32            <option value="USD">USD</option>
33            <option value="EUR">EUR</option>
34
35          </select>
36          <input type="number" value={amount} onChange={handleAmountChange} />
37        </div>
38        <div>
39          <label htmlFor="toCurrency">To:</label>
40          <select id="toCurrency" value={toCurrency} onChange={handleToCurrencyChange}>
41            <option value="USD">USD</option>
42            <option value="EUR">EUR</option>
43
44          </select>
45          <span>{convertCurrency()}</span>
46        </div>
47      </div>
48    );
49  };
50
51  export default CurrencyConverter;
52
```

https://direct-possibly.codedamn.app/

Currency Converter

From: USD ⌄ 45
To: EUR ⌄ 38.25

PROBLEMS   OUTPUT   TERMINAL

Toggle Browser Console

Fork on codedamn

Terminal   Editor   Browser

Q2 - Create a stopwatch application through which users can start, pause and reset the timer.

Use React state, event handlers and the setTimeout or setInterval functions to manage the

timer's state and actions.

App.jsx ×

code > src > App.jsx > ...

```jsx
1   import React, { useState, useEffect } from 'react';
2
3   const Stopwatch = () => {
4     const [time, setTime] = useState(0);
5     const [isRunning, setIsRunning] = useState(false);
6
7     useEffect(() => {
8       let timer;
9       if (isRunning) {
10        timer = setInterval(() => {
11          setTime((prevTime) => prevTime + 1);
12        }, 1000);
13      } else {
14        clearInterval(timer);
15      }
16      return () => clearInterval(timer);
17    }, [isRunning]);
18
19    const startTimer = () => {
20      setIsRunning(true);
21    };
22
23    const pauseTimer = () => {
24      setIsRunning(false);
25    };
26
27    const resetTimer = () => {
28      setTime(0);
29      setIsRunning(false);
30    };
```

PROBLEMS   OUTPUT   TERMINAL

Fork on codedamn

Terminal   Editor   Browser

https://direct-possibly.codedamn.app/

Stopwatch

00:03

[Start] [Reset]

Toggle Browser Console

App.jsx ×

code > src > App.jsx > ...

```jsx
29      setIsRunning(false);
30    };
31
32    const formatTime = (timeInSeconds) => {
33      const minutes = Math.floor(timeInSeconds / 60);
34      const seconds = timeInSeconds % 60;
35      return `${minutes < 10 ? '0' : ''}${minutes}:${seconds < 10 ? '0' : ''}${seconds}`;
36    };
37
38    return (
39      <div>
40        <h2>Stopwatch</h2>
41        <div>
42          <span>{formatTime(time)}</span>
43        </div>
44        <div>
45          {!isRunning ? (
46            <button onClick={startTimer}>Start</button>
47          ) : (
48            <button onClick={pauseTimer}>Pause</button>
49          )}
50          <button onClick={resetTimer}>Reset</button>
51        </div>
52      </div>
53    );
54  };
55
56  export default Stopwatch;
57
```

PROBLEMS   OUTPUT   TERMINAL

Fork on codedamn

Terminal   Editor   Browser

https://direct-possibly.codedamn.app/

Stopwatch

00:03

[Start] [Reset]

Toggle Browser Console