

# Lab 8

Name: ARYAN KUMAR SINGH

Roll no: 22IT3010

## Question 1

Index.jsx

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));
```

App.jsx

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

const CurrencyConverter = () => {
  const [amount, setAmount] = useState('');
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] = useState(null);

  const exchangeRates = {
    USD: {
      EUR: 0.84,
      INR: 74.19
    },
    EUR: {
      USD: 1.19,
      INR: 83.32
    },
    INR: {

```

```

        USD: 0.013,
        EUR: 0.012
    }
};

const handleAmountChange = (e) => {
    setAmount(e.target.value);
};

const handleFromCurrencyChange = (e) => {
    setFromCurrency(e.target.value);
};

const handleToCurrencyChange = (e) => {
    setToCurrency(e.target.value);
};

const handleConvert = () => {
    const rate = exchangeRates[fromCurrency][toCurrency];
    const converted = parseFloat(amount) * rate;
    setConvertedAmount(converted.toFixed(2));
};

return (
    <div>
        <h2>Currency Converter</h2>
        <div>
            <label htmlFor="amount">Amount:</label>
            <input type="number" id="amount" value={amount}
onChange={handleAmountChange} />
        </div>
        <div>
            <label htmlFor="fromCurrency">From:</label>
            <select id="fromCurrency" value={fromCurrency}
onChange={handleFromCurrencyChange}>
                <option value="USD">USD</option>
                <option value="EUR">EUR</option>
                <option value="INR">INR</option>
            </select>
        </div>
    </div>
);

```

```





        </select>
      </div>
      <div>
        <label htmlFor="toCurrency">To:</label>
        <select id="toCurrency" value={toCurrency}
onChange={handleToCurrencyChange}>
          <option value="USD">USD</option>
          <option value="EUR">EUR</option>
          <option value="INR">INR</option>
        </select>
      </div>
      <button onClick={handleConvert}>Convert</button>
      {convertedAmount && (
        <p>
          Converted Amount: {convertedAmount} {toCurrency}
        </p>
      )}
    </div>
  );
};

```

```
export default CurrencyConverter;
```

## Output

Web View ×

### Currency Converter

Amount:

From:

To:

Converted Amount: 12.00 EUR

?

Web View ×

▶ □ 🌐 ⋮

### Currency Converter

Amount:

From:

To:

Convert

Converted Amount: 7419.00 INR

?

Web View ×

▶ □ 🌐 ⋮

### Currency Converter

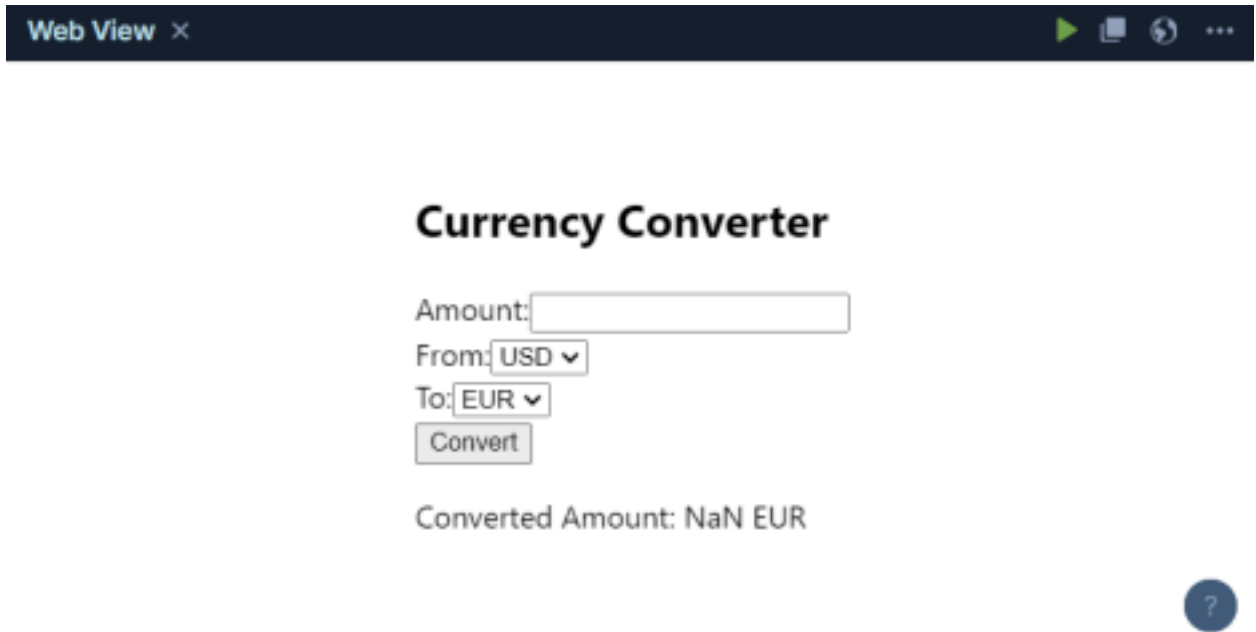
Amount:

From:

To:

Convert

Converted Amount: 7419.00 EUR



## Question 2

index.jsx

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render(<App />, document.getElementById('root'));
```

App.jsx

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

const Stopwatch = () => {
  const [time, setTime] = useState(0);
  const [isRunning, setIsRunning] = useState(false);

  useEffect(() => {
    let intervalId;

    if (isRunning) {
      intervalId = setInterval(() => {
        setTime((prevTime) => prevTime + 1);
      }, 1000);
    }
  }, [isRunning]);

  return (
    <div>
      <span>Time: {time}</span>
      <button onClick={() => setIsRunning(!isRunning)}>
        {isRunning ? 'Stop' : 'Start'}
      </button>
    </div>
  );
}
```

```

    }, 1000);
  } else {
    clearInterval(intervalId);
  }

  return () => clearInterval(intervalId);
}, [isRunning]);
const startTimer = () => {
  setIsRunning(true);
};

const pauseTimer = () => {
  setIsRunning(false);
};

const resetTimer = () => {
  setTime(0);
  setIsRunning(false);
};

const formatTime = (time) => {
  const minutes = Math.floor(time / 60);
  const seconds = time % 60;
  return `${minutes < 10 ? '0' : ''}${minutes}:${seconds < 10 ? '0'
: ''}${seconds}`;
};

return (
  <div className="stopwatch-container"> {/* Add a class for
styling */}
    <h2 className="stopwatch-header">Stopwatch</h2> {/* Add a
class for styling */}
    <div className="stopwatch-content"> {/* Add a class for
styling */}
      <p className="stopwatch-time">{formatTime(time)}</p> {/* Add
a class for styling */}
      <div className="button-group"> {/* Add a class for styling
*/}

```

```

        <button className="start-button" onClick={startTimer}
disabled={isRunning}>Start</button> {/* Add classes for styling
*/}

        <button className="pause-button" onClick={pauseTimer}
disabled={!isRunning}>Pause</button> {/* Add classes for styling */}
        <button className="reset-button"
onClick={resetTimer}>Reset</button> {/* Add classes for styling */}
    </div>
</div>
</div>
);
};

export default Stopwatch;

```

## CSS File

```

.stopwatch-container {
  margin: 20px;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.stopwatch-header {
  font-size: 24px;
  margin-bottom: 10px;
}

.stopwatch-content {
  display: flex;
  align-items: center;
}

.stopwatch-time {
  font-size: 36px;
}

```

```
margin-right: 20px;
}

.button-group button
{ font-size: 18px;
padding: 10px 20px;
margin-right: 10px;
border: none;
border-radius: 5px;
cursor: pointer;
}

.start-button {
background-color: #5cb85c;
color: #fff;
}

.pause-button {
background-color: #d9534f;
color: #fff;
}

.reset-button {
background-color: #f0ad4e;
color: #fff;
}
```

## Output



