

Q1.

```
1 import { useEffect, useState } from 'react';
2 import axios from 'axios';
3 import Dropdown from 'react-dropdown';
4 import { HiSwitchHorizontal } from 'react-icons/hi';
5 import 'react-dropdown/style.css';
6 import './App.css';
7
8 function App() {
9   const [info, setInfo] = useState([]);
10   const [input, setInput] = useState(0);
11   const [from, setFrom] = useState("usd");
12   const [to, setTo] = useState("inr");
13   const [options, setOptions] = useState([]);
14   const [output, setOutput] = useState(0);
15   useEffect(() => {
16     axios.get(
17       `https://cdn.jsdelivr.net/gh/fawazahmed0/currency-api@1/latest/currencies/${from}.json`
18     ).then((res) => {
19       setInfo(res.data[from]);
20     });
21   }, [from]);
22   useEffect(() => {
23     setOptions(Object.keys(info));
24     convert();
25   }, [info])
26   function convert() {
27     var rate = info[to];
28     setOutput(input * rate);
29   }
30   function flip() {
31     var temp = from;
32     setFrom(to);
33     setTo(temp);
34   }
35   return (
36     <div className="App">
37       <div className="heading">
38         <h1>Currency converter</h1>
39       </div>
40       <div className="container">
41         <div className="left">
42           <h3>Amount</h3>
43           <input type="text"
44             placeholder="Enter the amount"
45             onChange={(e) => setInput(e.target.value)} />
46         </div>
47         <div className="middle">
```

```
47         <div className="middle">
48           <h3>From</h3>
49           <Dropdown options={options}
50             onChange={(e) => { setFrom(e.value) }}
51             value={from} placeholder="From" />
52         </div>
53         <div className="switch">
54           <HiSwitchHorizontal size="30px"
55             onClick={() => { flip() }} />
56         </div>
57         <div className="right">
58           <h3>To</h3>
59           <Dropdown options={options}
60             onChange={(e) => { setTo(e.value) }}
61             value={to} placeholder="To" />
62         </div>
63       </div>
64       <div className="result">
65         <button onClick={() => { convert() }}>Convert</button>
66         <h2>Converted Amount:</h2>
67         <p>[input + " " + from + " = " + output.toFixed(2) + " " + to]</p>
68       </div>
69     </div>
70   );
71 }
72
73 export default App;
74
```

```

2
3 @import url('https://fonts.googleapis.com/css2?family=Pacifico&display=swap');
4
5 .App {
6   height: 100vh;
7   width: 100%;
8   display: flex;
9   align-items: center;
10  flex-direction: column;
11  background-image: linear-gradient(120deg, ■ #fdfbfb 0%, ■ #ebedee 100%);
12 }
13
14 .heading {
15   font-family: 'Pacifico', cursive;
16   font-size: 35px;
17 }
18
19 .container {
20   height: auto;
21   width: 800px;
22   display: flex;
23   justify-content: space-around;
24   align-items: center;
25 }
26
27 input {
28   padding-left: 5px;
29   font-size: 20px;
30   height: 36px;
31 }
32
33 .middle,
34 .right {
35   width: 120px;
36 }
37
38 .switch {
39   padding: 5px;
40   background-color: ■ rgb(226, 252, 184);
41   border-radius: 50%;
42   cursor: pointer;
43 }
44
45 .result {
46   box-sizing: border-box;
47   width: 800px;

```

```

47   width: 800px;
48   padding-left: 30px;
49 }
50
51 button {
52   width: 100px;
53   height: 30px;
54   font-weight: bold;
55   font-size: 20px;
56   margin-top: 15px;
57   border: 2px solid ■ forestgreen;
58   background-color: ■ rgb(226, 252, 184);
59   cursor: pointer;
60 }
61
62 p,
63 h3,
64 button,
65 .switch {
66   color: ■ forestgreen;
67 }
68
69 p {
70   font-size: 30px;
71 }
72

```

Currency converter

Amount

Enter the amount I

From

usd



To

inr

Convert

Converted Amount:

0 usd = 0.00 inr

Currency converter

Amount

500

From

usd



To

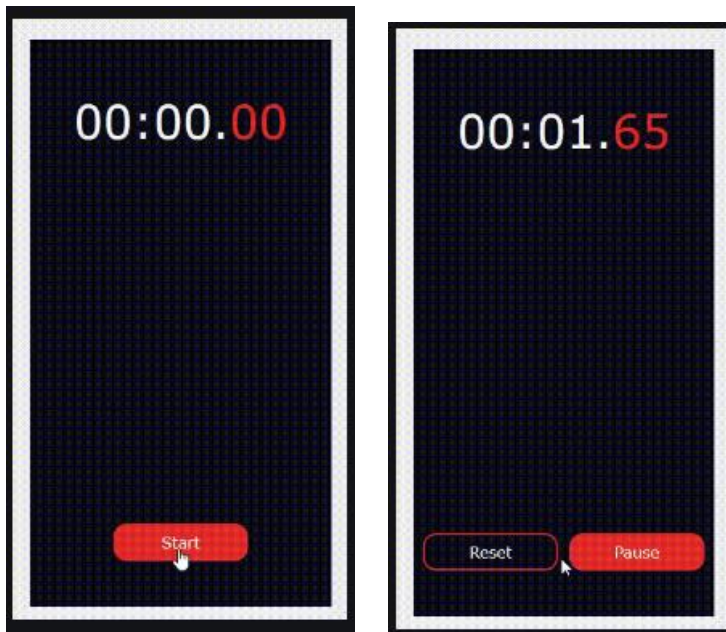
eur

Convert

Converted Amount:

500 usd = 426.03 eur

Q2.



```
JS app.js > ...
1  import './App.css';
2  import Stopwatch from './Components/StopWatch/StopWatch.js';
3
4  function App() {
5    return (
6      <div className="App">
7        <StopWatch />
8      </div>
9    );
10 }
11
12 export default App;
13
```

```
JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App.css';
5
6  ReactDOM.render([
7    <React.StrictMode>
8      <App />
9    </React.StrictMode>,
10    document.getElementById('root')
11  ]);
12
```

```

JS timer.js > ...
1  import React from "react";
2  import "../Timer.css";
3
4  export default function Timer(props) {
5      return (
6          <div className="timer">
7              <span className="digits">
8                  {"0" + Math.floor((props.time / 60000) % 60).slice(-2)}:
9              </span>
10             <span className="digits">
11                 {"0" + Math.floor((props.time / 1000) % 60).slice(-2)}.
12             </span>
13             <span className="digits mili-sec">
14                 {"0" + ((props.time / 10) % 100).slice(-2)}
15             </span>
16         </div>
17     );
18 }
19

```

```

Timer.css > ...
1  .timer {
2      margin: 3rem 0;
3      width: 100%;
4      display: flex;
5      height: 12%;
6      justify-content: center;
7      align-items: center;
8  }
9
10 .digits {
11     font-family: Verdana, Geneva, Tahoma, sans-serif;
12     font-size: 3rem;
13     color: #f5f5f5;
14 }
15
16 .mili-sec {
17     color: #e42a2a;
18 }
19

```

```

JS ControlButtons.js > ...
1  import React from "react";
2  import "../ControlButtons.css";
3
4  export default function ControlButtons(props) {
5      const StartButton = (
6          <div className="btn btn-one btn-start"
7              onClick={props.handleStart}>
8              Start
9          </div>
10     );
11     const ActiveButtons = (
12         <div className="btn-grp">
13             <div className="btn btn-two"
14                 onClick={props.handleReset}>
15                 Reset
16             </div>
17             <div className="btn btn-one"
18                 onClick={props.handlePauseResume}>
19                 {props.isPaused ? "Resume" : "Pause"}
20             </div>
21         </div>
22     );
23
24     return (
25         <div className="Control-Buttons">
26             <div>{props.active ? ActiveButtons : StartButton}</div>
27         </div>
28     );
29 }
30

```

```

ControlButtons.css > pre>.Control-Buttons
1  pre>.Control-Buttons {
2      margin: 3rem 0;
3      width: 100%;
4      box-sizing: border-box;
5      display: flex;
6      align-items: center;
7      justify-content: center;
8  }
9
10 .btn-grp {
11     display: flex;
12     align-items: center;
13     justify-content: space-around;
14 }
15
16 .btn {
17     font-family: Verdana, Geneva, Tahoma, sans-serif;
18     width: 10vw;
19     height: 5vh;
20     border-radius: 14px;
21     margin: 0px 6px;
22     display: flex;
23     border: 2px solid #e42a2a;
24     justify-content: center;
25     align-items: center;
26     cursor: pointer;
27     color: #f5f5f5;
28 }
29
30 .btn-one {
31     background-color: #e42a2a;
32 }
33

```

```
App.css > ...
1 .App {
2   background-color: #191919;
3   width: 100vw;
4   height: 100vh;
5   display: flex;
6   flex-direction: column;
7   align-items: center;
8   justify-content: center;
9 }
10
```

```
JS stopwatch.jsx > ...
1 import React, { useState } from "react";
2 import "./StopWatch.css";
3 import Timer from "../Timer/Timer";
4 import ControlButtons from "../ControlButtons/ControlButtons";
5
6 function Stopwatch() {
7   const [isActive, setIsActive] = useState(false);
8   const [isPaused, setIsPaused] = useState(true);
9   const [time, setTime] = useState(0);
10
11   React.useEffect(() => {
12     let interval = null;
13
14     if (isActive && isPaused === false) {
15       interval = setInterval(() => {
16         setTime((time) => time + 10);
17       }, 10);
18     } else {
19       clearInterval(interval);
20     }
21     return () => {
22       clearInterval(interval);
23     };
24   }, [isActive, isPaused]);
25
26   const handleStart = () => {
27     setIsActive(true);
28     setIsPaused(false);
29   };
30
31   const handlePauseResume = () => {
32     setIsPaused(!isPaused);
33   };
34
35   const handleReset = () => {
36     setIsActive(false);
37     setTime(0);
38   };
39
40   return (
41     <div className="stop-watch">
42       <Timer time={time} />
43       <ControlButtons
44         active={isActive}
45         isPaused={isPaused}
46         handleStart={handleStart}
47         handlePauseResume={handlePauseResume}
48         handleReset={handleReset}
49       />
50     </div>
51   );
52 }
53
54 export default Stopwatch;
55
```

```
stopwatch.css > ...
1 .stop-watch {
2   height: 85vh;
3   width: 23vw;
4   background-color: #0d0c1b;
5   display: flex;
6   flex-direction: column;
7   align-items: center;
8   justify-content: space-between;
9 }
10
```

Q3.

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <title>GeeksforGeeks ChatRoom</title>
6 <link rel="stylesheet" href="style.css">
7 </head>
8
9 <body>
10 <div class="container">
11 <div class="logo">ChatRoom</div>
12 <form class="form" id="form">
13 <input class="input" type="text" placeholder="Name" id="myname">
14 <input class="input" type="text" placeholder="Message" id="message">
15 <button class="button">Send Message</button>
16 </form>
17 <div class="messageArea" id="messageArea"></div>
18 </div>
19
20 <script src="/socket.io/socket.io.js"></script>
21 <script>
22 let socket = io();
23
24 let form = document.getElementById('form');
25 let myname = document.getElementById('myname');
26 let message = document.getElementById('message');
27 let messageArea = document.getElementById('messageArea');
28
29 form.addEventListener('submit', (e) => {
30 e.preventDefault();
31
32 if (message.value) {
33 socket.emit('send name', myname.value);
34 socket.emit('send message', message.value);
35 message.value = '';
36 }
37 });
38
39 socket.on('send name', (username) => {
40 let name = document.createElement('p');
41 name.classList.add('message');
42 name.innerHTML = '<span class="username">${username}</span>';
43 messageArea.appendChild(name);
44 });
45
46 socket.on('send message', (chat) => {
47 let chatContent = document.createElement('p');
48 chatContent.classList.add('message');
49 chatContent.textContent = chat;
50 messageArea.appendChild(chatContent);
51 });
52 </script>
53 </body>
54 </html>
```

```
2
3 body {
4 background-color: #90EE90;
5 margin: 0;
6 padding: 0;
7 font-family: Arial, sans-serif;
8 }
9
10 .container {
11 max-width: 400px;
12 margin: 50px auto;
13 padding: 20px;
14 background-color: white;
15 border-radius: 10px;
16 box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
17 }
18
19 .logo {
20 text-align: center;
21 font-size: 24px;
22 font-weight: bold;
23 margin-bottom: 20px;
24 color: #008000;
25 }
26
27 .form {
28 display: flex;
29 flex-direction: column;
30 align-items: center;
31 }
32
33 .input {
34 border: 1px solid #ccc;
35 border-radius: 5px;
36 padding: 10px;
37 margin-bottom: 10px;
38 width: 100%;
39 }
```



```

40
41 .button {
42     background-color: #008000;
43     color: white;
44     border: none;
45     border-radius: 5px;
46     padding: 10px;
47     width: 100%;
48     cursor: pointer;
49 }
50
51 .messageArea {
52     margin-top: 20px;
53     border: 1px solid #ccc;
54     border-radius: 5px;
55     padding: 10px;
56     height: 200px;
57     overflow-y: scroll;
58 }
59
60 .message {
61     margin-bottom: 5px;
62 }
63
64 .username {
65     font-weight: bold;
66     color: #008000;
67 }
68

```

```

JS socket.io.js ?...
1  const http = require('http');
2  const fs = require('fs');
3  const server = http.createServer((req, res) => {
4      if (req.url === '/') {
5          fs.readFile(__dirname + '/index.html', (err, data) => {
6              if (err) {
7                  res.writeHead(500);
8                  res.end('Error loading index.html');
9              } else {
10                 res.writeHead(200, { 'Content-Type': 'text/html' });
11                 res.end(data);
12             }
13         });
14     }
15 });
16 const io = require('socket.io')(server);
17 const port = 5000;
18 io.on('connection', (socket) => {
19     socket.on('send name', (user) => {
20         io.emit('send name', user);
21     });
22
23     socket.on('send message', (chat) => {
24         io.emit('send message', chat);
25     });
26 });
27 server.listen(port, () => {
28     console.log(`Server is listening at the port: ${port}`);
29 });
30

```

ChatRoom

User 1

I am doing good, |

Send Message

User 1:

Hi

User 2:

Hello

ChatRoom

User 2

Message

Send Message

User 1:

Hi

User 2:

Hello