

Assignment

Name→Kalpana Tiwari

Roll No→22CS2025

1Q.)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("");
```

```
const exchangeRate = 0.85;
```

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

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

```
};
```

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

```
const convertCurrency = () => {  
  const converted =  
parseFloat(amount) *  
exchangeRate;
```

```
setConvertedAmount(converted.to  
Fixed(2));  
};
```

```
return (  
  <div>  
    <h2>Currency Converter</h2>  
    <div>  
      <label>  
        Amount:  
        <input type="number"  
value={amount}  
onChange={handleAmountChange  
} />  
      </label>  
    </div>  
    <div>  
      <label>  
        From Currency:
```

```
    <select  
value={fromCurrency}  
onChange={handleFromCurrency  
Change}>
```

```
    <option  
value="USD">USD</option>
```

```
  </select>
```

```
  </label>
```

```
</div>
```

```
<div>
```

```
  <label>
```

```
    To Currency:
```

```
    <select value={toCurrency}  
onChange={handleToCurrencyCha  
nge}>
```

```
<option  
value="EUR">EUR</option>
```

```
</select>
```

```
</label>
```

```
</div>
```

```
<button
```

```
onClick={convertCurrency}>Conve  
rt</button>
```

```
{convertedAmount && (
```

```
<div>
```

```
<h3>Converted
```

```
Amount:</h3>
```

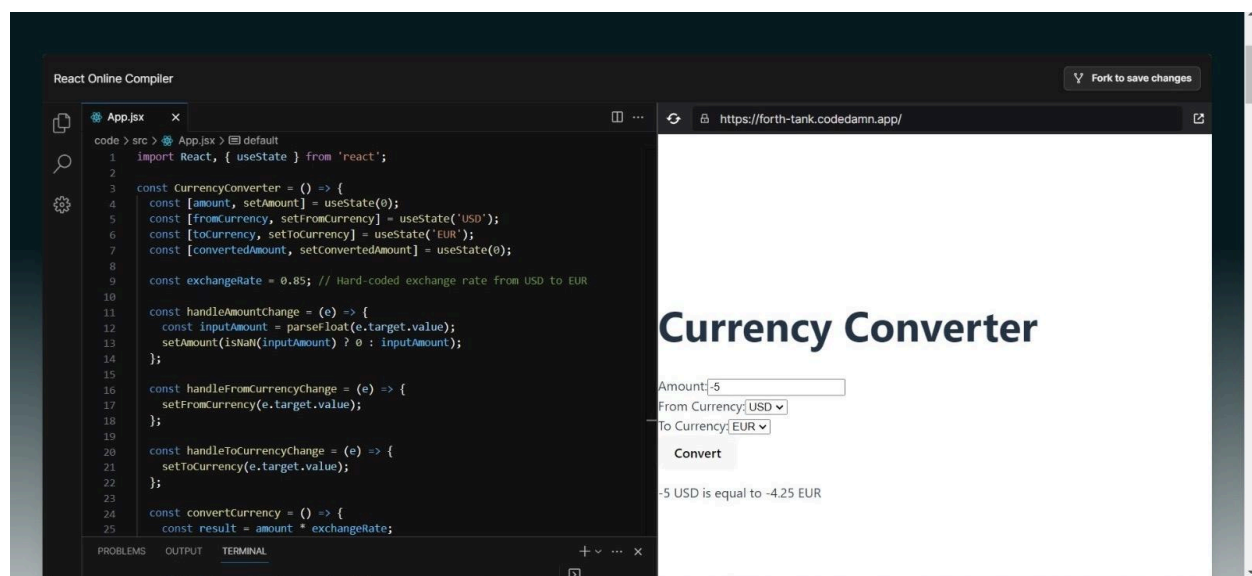
```
<p>{convertedAmount}
```

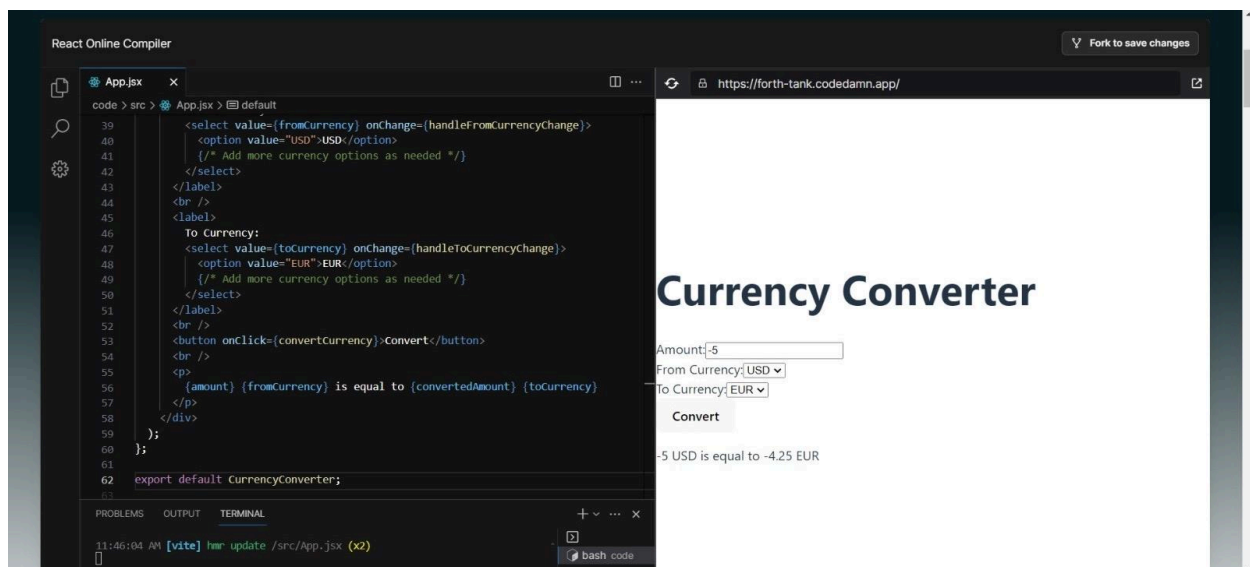
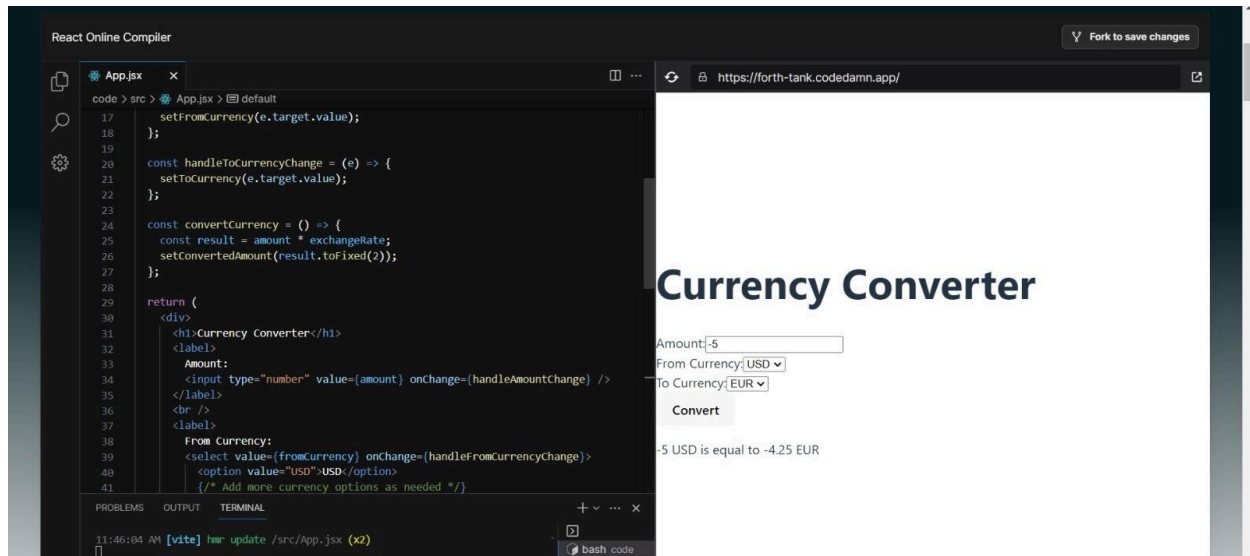
```
{toCurrency}</p>
```

```
</div>
```

```
    )}  
  </div>  
  
);  
};
```

```
export default CurrencyConverter;
```





2Q.)import React, { useState, useRef } from 'react';

const Stopwatch = () => {


```
const [isRunning, setIsRunning] =
useState(false);
const [elapsedTime,
setElapsedTime] = useState(0);
const intervalRef = useRef(null);

const startTimer = () => {
  setIsRunning(true);
  intervalRef.current =
setInterval(() => {

setElapsedTime((prevElapsedTime
) => prevElapsedTime + 10);
  }, 10);
};
```

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

```
clearInterval(intervalRef.current);  
};
```

```
const resetTimer = () => {  
  setIsRunning(false);
```

```
clearInterval(intervalRef.current);  
  setElapsedTime(0);  
};
```

```
const formatTime = (time) => {  
  const minutes = Math.floor(time  
/ 60000);
```

```
    const seconds =  
Math.floor((time % 60000) / 1000);  
    const milliseconds =  
Math.floor((time % 1000) / 10);
```

```
    return  
`${minutes.toString().padStart(2,  
'0')}:${seconds  
    .toString()  
    .padStart(2,  
'0')}.${milliseconds.toString().padSt  
art(2, '0')}`;  
};
```

```
return (  
  <div>
```

```
<h2>Stopwatch</h2>
```

```
<p>{formatTime(elapsedTime)}</p>
```

```
<div>
```

```
  {!isRunning ? (
```

```
    <button
```

```
onClick={startTimer}>Start</button
```

```
>
```

```
  ) : (
```

```
    <button
```

```
onClick={pauseTimer}>Pause</but
```

```
ton>
```

```
  )}
```

```

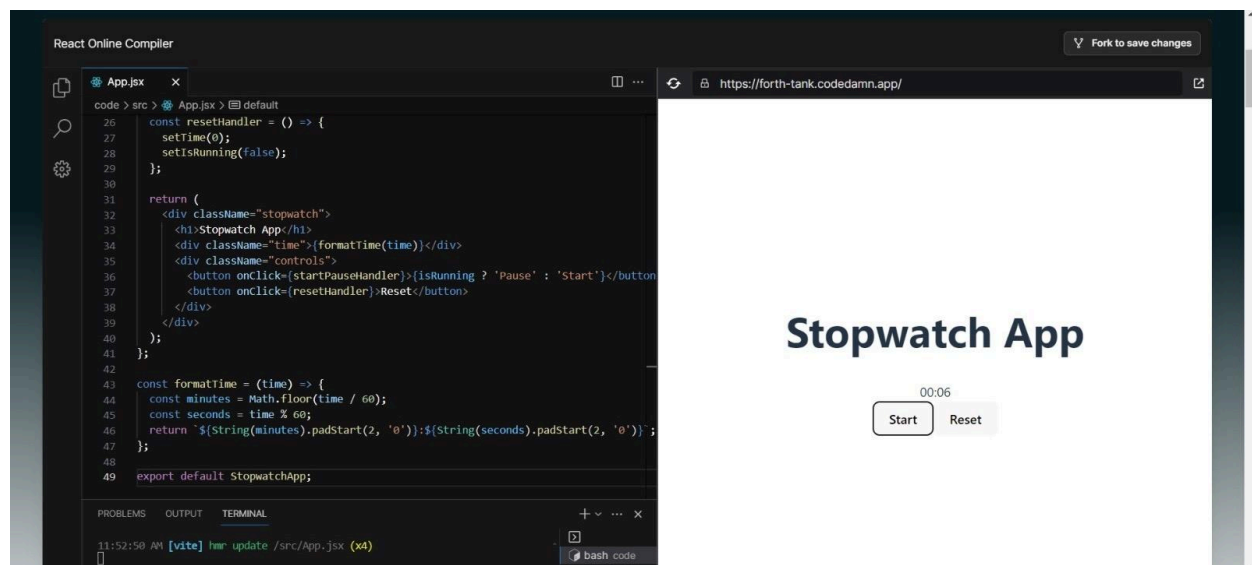
    <button
onClick={resetTimer}>Reset</button
n>

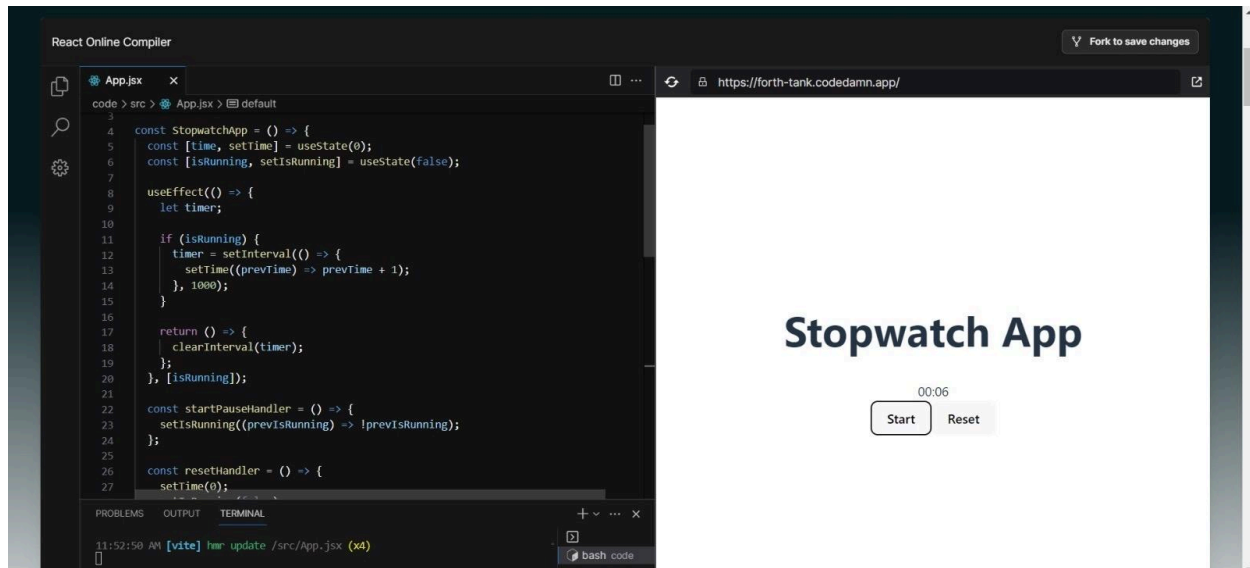
    </div>
</div>

);
};

```

export default Stopwatch;





3Q.)// App.js

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

import firebase from 'firebase/app';

import 'firebase/database';

import ConversationList from
'./ConversationList';

```
import ChatInterface from
'./ChatInterface';
import MessageInput from
'./MessageInput';

// Initialize Firebase
const firebaseConfig = {
  // Your Firebase config details
};

if (!firebase.apps.length) {

firebase.initializeApp(firebaseConfi
g);
}
```

```
const App = () => {  
  const [selectedConversation,  
  setSelectedConversation] =  
  useState(null);  
  const [messages, setMessages]  
  = useState([]);  
  
  useEffect(() => {  
    const conversationsRef =  
    firebase.database().ref('conversations');  
    conversationsRef.on('value',  
    (snapshot) => {  
      // Update conversation list  
    });  
  });  
}
```



```
// Clean up listener  
return () =>  
conversationsRef.off('value');  
}, []);
```

```
const handleConversationSelect  
= (conversationId) => {  
  // Set selected conversation  
};
```

```
return (  
  <div>  
    <ConversationList  
      conversations={/*  
Conversation data */}  
    >
```

```
onSelect={handleConversationSelect}
    />
    <ChatInterface
messages={messages} />
    <MessageInput
conversationId={selectedConversation} />
    </div>
);
};

export default App;
```

