

LAB ASSIGNMENT - 8

Name – Sudhanshu Shivam

Branch – CSE

Roll No. – 22CS3059

CODE

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

function CurrencyConverter() {
  const [amount, setAmount] = useState(0);
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] = useState(0);

  // Hard-coded exchange rate
  const exchangeRate = 0.85; // 1 USD = 0.85 EUR

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

```
};
```

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

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

```
const convertCurrency = () => {  
  const result = amount * exchangeRate;  
  setConvertedAmount(result.toFixed(2)); // Round to 2 decimal  
places  
};
```

```
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={handleFromCurrencyChange}>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
      </select>
    </label>
  </div>
  <div>
    <label>
      To Currency:
      <select value={toCurrency}
onChange={handleToCurrencyChange}>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
      </select>
    </label>
  </div>
  <button onClick={convertCurrency}>Convert</button>
</div>
```

```
    {amount} {fromCurrency} is equal to {convertedAmount}  
    {toCurrency}
```

```
</div>
```

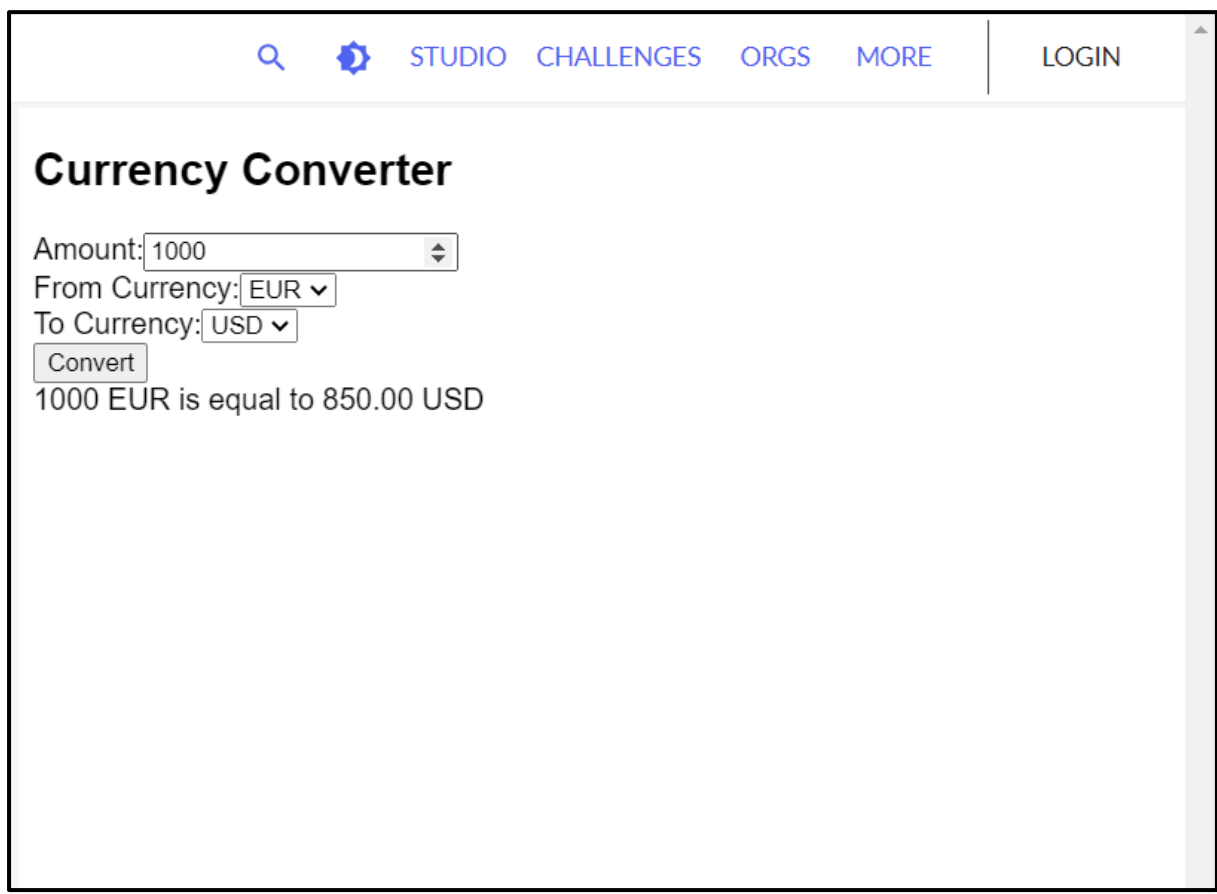
```
</div>
```

```
);
```

```
}
```

```
export default CurrencyConverter;
```

OUTPUT



The screenshot shows a web application interface for a currency converter. At the top, there is a navigation bar with a search icon, a settings icon, and links for 'STUDIO', 'CHALLENGES', 'ORGS', 'MORE', and 'LOGIN'. The main content area is titled 'Currency Converter'. It features a form with the following elements: 'Amount:' followed by a text input containing '1000'; 'From Currency:' followed by a dropdown menu showing 'EUR'; and 'To Currency:' followed by a dropdown menu showing 'USD'. Below these is a 'Convert' button. The result of the conversion is displayed as '1000 EUR is equal to 850.00 USD'.

CODE

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

function Stopwatch() {
  const [timer, setTimer] = useState(0);
  const [isRunning, setIsRunning] = useState(false);

  useEffect(() => {
    let intervalId;

    if (isRunning) {
      intervalId = setInterval(() => {
        setTimer((prevTimer) => prevTimer + 1);
      }, 1000);
    } else {
      clearInterval(intervalId);
    }

    return () => clearInterval(intervalId);
  }, [isRunning]);

  const startTimer = () => {
    setIsRunning(true);
  }
}
```

```
};
```

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

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

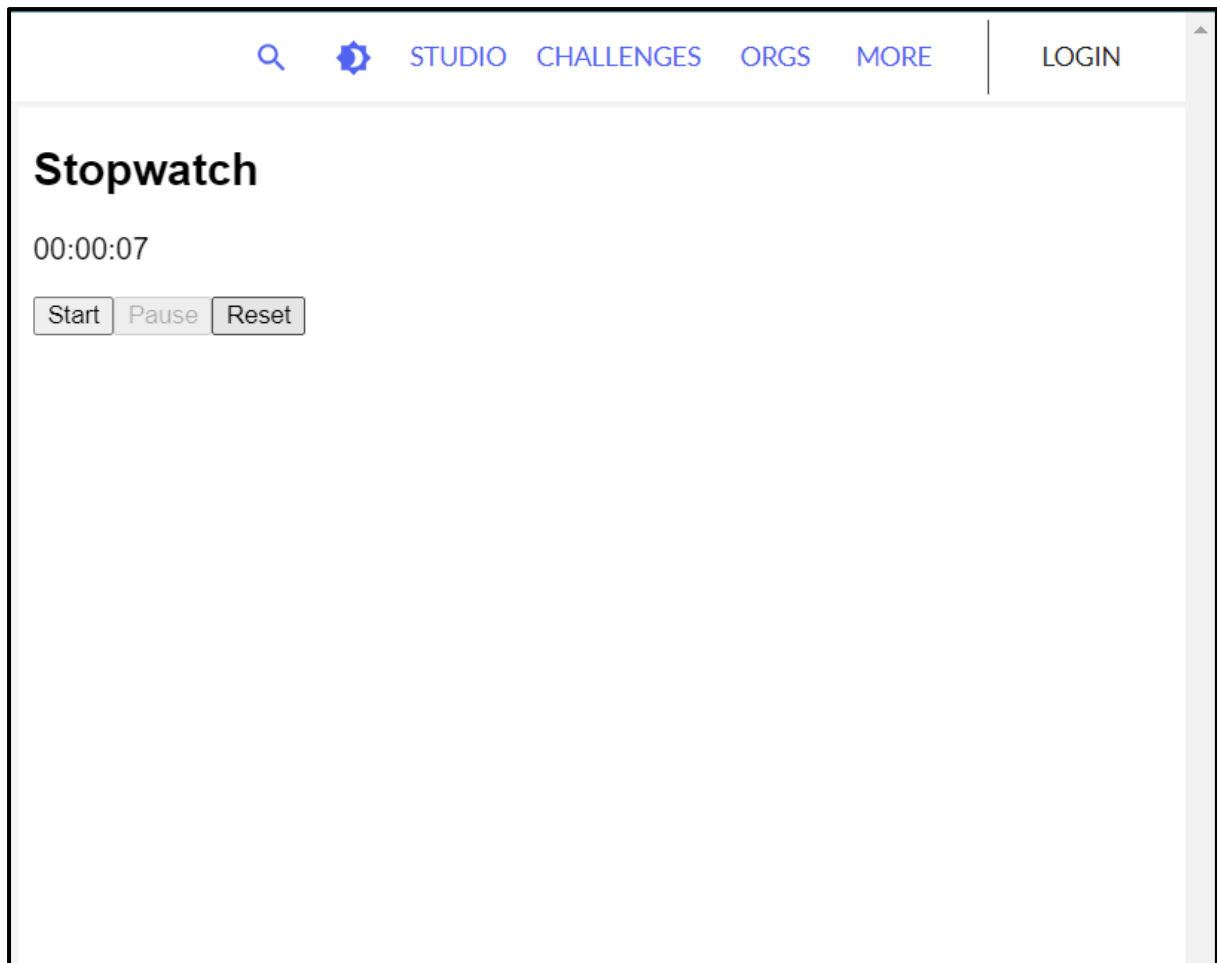
```
const formatTime = (time) => {  
  const hours = Math.floor(time / 3600);  
  const minutes = Math.floor((time % 3600) / 60);  
  const seconds = time % 60;  
  
  return `${hours.toString().padStart(2, '0')}:${minutes  
    .toString()  
    .padStart(2, '0')}:${seconds.toString().padStart(2, '0')}`;  
};
```

```
return (  
  <div>  
    <h2>Stopwatch</h2>
```

```
<div>
  <p>{formatTime(timer)}</p>
</div>
<div>
  <button onClick={startTimer} disabled={isRunning}>
    Start
  </button>
  <button onClick={pauseTimer} disabled={!isRunning}>
    Pause
  </button>
  <button onClick={resetTimer}>Reset</button>
</div>
</div>
);
}
```

```
export default Stopwatch;
```

OUTPUT



CODE

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

const MessagingApp = () => {
  const [conversations, setConversations] = useState([
    { id: 1, name: 'Family', messages: [] },
    { id: 2, name: 'Friends', messages: [] },
  ]);

  const [selectedConversation, setSelectedConversation] =
    useState(null);

  const [newMessage, setNewMessage] = useState("");
```



```

const handleConversationClick = (conversationId) => {
  setSelectedConversation(conversationId);
};

const handleSendMessage = () => {
  if (newMessage.trim() !== "" && selectedConversation) {
    const updatedConversations = conversations.map((conversation)
=>
      conversation.id === selectedConversation
        ? {
            ...conversation,
            messages: [
              { text: newMessage, timestamp: new
Date().toLocaleTimeString() },
              ...conversation.messages,
            ],
          }
        : conversation
    );
    setConversations(updatedConversations);
    setNewMessage("");
  }
};

```

```

return (
  <div>
    <h1>Messaging App</h1>

    <div style={{ display: 'flex' }}>
      {/* List of Conversations */}
      <div style={{ flex: '1', borderRight: '1px solid #ccc', padding:
'10px' }}>
        <h2>Conversations</h2>
        <ul>
          {conversations.map((conversation) => (
            <li key={conversation.id} onClick={() =>
handleConversationClick(conversation.id)}>
              {conversation.name}
            </li>
          ))}
        </ul>
      </div>

      {/* Chat Interface */}
      <div style={{ flex: '3', padding: '10px' }}>
        {selectedConversation && (
          <div>

```

```
<h2>Chat with {conversations.find((conv) => conv.id ===
selectedConversation).name}</h2>
```

```
<div style={{ maxHeight: '300px', overflowY: 'auto', border:
'1px solid #ccc', padding: '10px' }}>
```

```
  {conversations
```

```
    .find((conv) => conv.id === selectedConversation)
```

```
    .messages.map((message, index) => (
```

```
      <div key={index} style={{ borderBottom: '1px solid #eee',
paddingBottom: '5px' }}>
```

```
        <strong>{message.timestamp}</strong>: {message.text}
```

```
      </div>
```

```
    )})
```

```
</div>
```

```
<textarea
```

```
  rows="3"
```

```
  value={newMessage}
```

```
  onChange={(e) => setNewMessage(e.target.value)}
```

```
<button onClick={handleSendMessage}>Send</button>
```

```
</div>
```

```
  )}
```

```
</div>
```

```
</div>
```

```
</div>
```

```
);
```

```
};
```

```
export default MessagingApp;
```

OUTPUT

