NAME: RISHIT TIWARI
ROLL NO.: 22CS3050

WEB TECH LAB – 08

Q1.

```jsx
import React, { useState } from "react";

const CurrencyConverter = () => {
  const [amount, setAmount] = useState("");
  const [fromCurrency, setFromCurrency] = useState("INR");
  const [toCurrency, setToCurrency] = useState("USD");
  const exchangeRateINRtoUSD = 0.012;
  const exchangeRateUSDtoINR = 82.91;

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

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

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

  const convertCurrency = () => {
    if (fromCurrency === "INR" && toCurrency === "USD") {
      return (amount * exchangeRateINRtoUSD).toFixed(3);
    } else if (fromCurrency === "USD" && toCurrency === "INR") {
      return (amount * exchangeRateUSDtoINR).toFixed(3);
    } else {
      return "Invalid conversion";
    }
  };

  return (
    <div>
      <h1>Currency Converter</h1>
      <div>
        <label>
          Amount:
          <input type="number" value={amount} onChange={handleAmountChange} />
        </label>
      </div>
      <div>
        <label>
          From Currency:
          <select value={fromCurrency} onChange={handleFromCurrencyChange}>
            <option value="INR">INR</option>
            <option value="USD">USD</option>
          </select>
        </label>
      </div>
      <div>
        <label>
          To Currency:
          <select value={toCurrency} onChange={handleToCurrencyChange}>
            <option value="USD">USD</option>
            <option value="INR">INR</option>
          </select>
        </label>
      </div>

      <div>
        Converted Amount: {convertCurrency()} {toCurrency}
      </div>
    </div>
  );
};

export default CurrencyConverter;
```

Currency Converter

Amount:

From Currency: INR

To Currency: USD

Converted Amount: 0.000 USD

Currency Converter

Amount: 1

From Currency: INR

To Currency: USD

Converted Amount: 0.012 USD

Currency Converter

Amount: 1

From Currency: USD

To Currency: INR

Converted Amount: 82.910 INR

Currency Converter

Amount: 10

From Currency: USD

To Currency: INR

Converted Amount: 829.100 INR

Q2.

```tsx
import React, { useState, useEffect } from "react";
import "./App.css";

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

  useEffect(() => {
    let intervalId;
    if (isRunning) {
      intervalId = setInterval(() => {
        setTime((prevTime) => prevTime + 1);
      }, 1000);
    } else {
      clearInterval(intervalId);
    }

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

  const startStopwatch = () => {
    setIsRunning(true);
  };

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

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

  const formatTime = (seconds) => {
    const hours = Math.floor(seconds / 3600);
    const minutes = Math.floor((seconds % 3600) / 60);
    const remainingSeconds = seconds % 60;

    return `${hours.toString().padStart(2, "0")}:${minutes
      .toString()
      .padStart(2, "0")}:${remainingSeconds.toString().padStart(2, "0")}`;
  };

  return (
    <div className="flex">
      <div className="head">
        <h1>Stopwatch</h1>
      </div>
      <div className="watch">
        <p>{formatTime(time)}</p>
      </div>
      <div className="time">
        {!isRunning ? (
          <button onClick={startStopwatch}>Start</button>
        ) : (
          <button onClick={pauseStopwatch}>Pause</button>
        )}
        <button onClick={resetStopwatch}>Reset</button>
      </div>
    </div>
  );
};

export default Stopwatch;
```

Stopwatch

00:00:00

Start | Reset

Stopwatch

00:00:05

Pause | Reset

Stopwatch

00:00:00

Start | Reset

Q3.

```jsx
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) {
      setConversations((prevConversations) =>
        prevConversations.map((conversation) =>
          conversation.id === selectedConversation
            ? {
                ...conversation,
                messages: [
                  {
                    text: newMessage,
                    timestamp: new Date().toLocaleTimeString(),
                  },
                  ...conversation.messages,
                ],
              }
            : conversation
        )
      );

      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;
```

Messaging App

Conversations

- Family
- Friends

Messaging App

Conversations

- Family
- Friends

Chat with Family

Hello guys

Send

Messaging App

Conversations

- Family
- Friends

Chat with Family

5:47:01 PM: Hello guys

Send

Messaging App

Conversations

- Family
- Friends

Chat with Family

5:47:15 PM: How are you?
5:47:01 PM: Hello guys

Send

Messaging App

Conversations

- Family
- Friends

Chat with Friends

5:47:28 PM: wassupppp

Send