WEB DEVELOPMENT

LAB ASSIGNMENT 8

Name:- Tanmay Amrutkar

Roll no:- 22IT3006

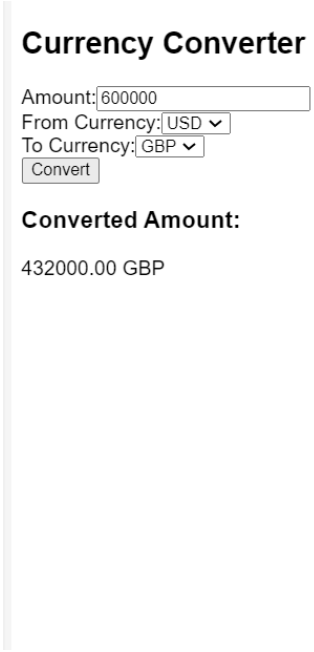
Q1.

Code:-

```
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 exchangeRates = {
    USD: {
      EUR: 0.84,
      GBP: 0.72,
      JPY: 109.25,
    },
    EUR: {
      USD: 1.19,
      GBP: 0.86,
      JPY: 130.90,
    },
    GBP: {
      USD: 1.39,
      EUR: 1.17,
      JPY: 150.31,
    },
    JPY: {
      USD: 0.0091,
      EUR: 0.0076,
      GBP: 0.0066,
    },
  };

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

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

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

```
  const convertCurrency = () => {
    const rate = exchangeRates[fromCurrency][toCurrency];
    const converted = amount * rate;
    setConvertedAmount(converted.toFixed(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={handleFromCurrencyChange}>
            <option value="USD">USD</option>
            <option value="EUR">EUR</option>
            <option value="GBP">GBP</option>
            <option value="JPY">JPY</option>
          </select>
        </label>
      </div>
      <div>
        <label>
          To Currency:
          <select value={toCurrency} onChange={handleToCurrencyChange}>
            <option value="USD">USD</option>
            <option value="EUR">EUR</option>
            <option value="GBP">GBP</option>
            <option value="JPY">JPY</option>
          </select>
        </label>
      </div>
      <button onClick={convertCurrency}>Convert</button>
      {convertedAmount && (
        <div>
          <h3>Converted Amount:</h3>
          <p>{convertedAmount} {toCurrency}</p>
        </div>
      )}
    </div>
  );
};

export default CurrencyConverter;
```

Output:-



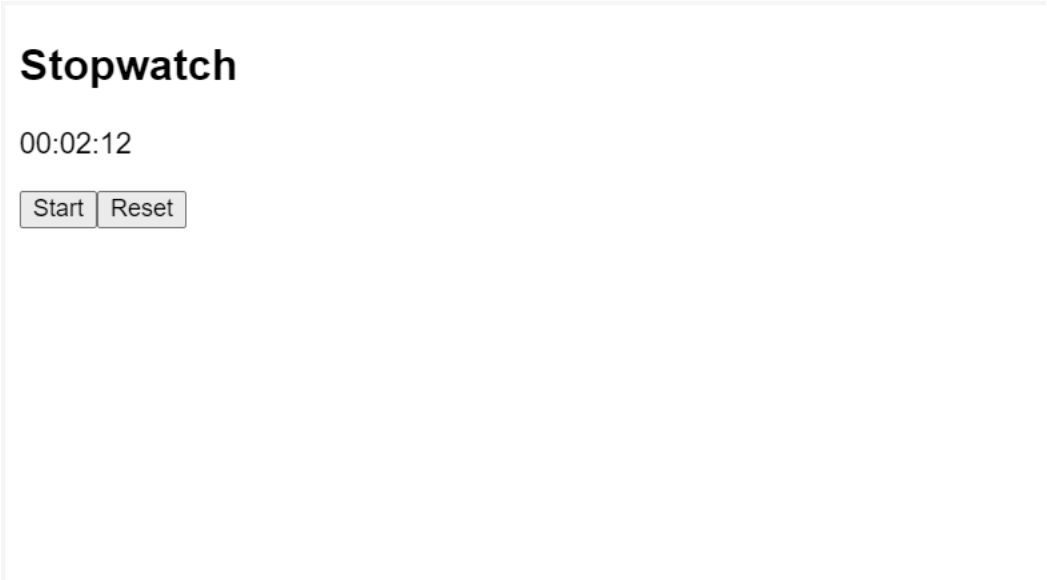
Q2.

Code:-

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

const Stopwatch = () => {
  // State for whether the stopwatch is running
  const [isRunning, setIsRunning] = useState(false);
  // State for elapsed time
  const [elapsedTime, setElapsedTime] = useState(0);

  // useEffect to handle starting and stopping the stopwatch
  useEffect(() => {
    let intervalId;

    if (isRunning) {
      // Start the stopwatch
      intervalId = setInterval(() => {
        // Increment elapsed time every 10 milliseconds
        setElapsedTime(prevElapsedTime => prevElapsedTime + 10);
      }, 10);
    } else {
      // Stop the stopwatch
      clearInterval(intervalId);
    }
```

```
    // Cleanup function to clear interval when component unmounts or is
re-rendered
    return () => clearInterval(intervalId);
  }, [isRunning]);

  // Function to toggle running state of the stopwatch
  const toggleRunning = () => {
    setIsRunning(!isRunning);
  };

  // Function to reset stopwatch to initial state
  const reset = () => {
    setIsRunning(false);
    setElapsedTime(0);
  };

  // JSX to render stopwatch UI
  return (
    <div>
      <h2>Stopwatch</h2>
      {/* Display elapsed time in minutes:seconds:milliseconds format */}
      <p>{`${String(Math.floor((elapsedTime / 60000) % 60)).padStart(2,
'0')}:${String(Math.floor((elapsedTime / 1000) % 60)).padStart(2,
'0')}:${String(Math.floor((elapsedTime / 10) % 100)).padStart(2, '0')}`}</p>
      {/* Button to start/pause stopwatch */}
      <button onClick={toggleRunning}>{isRunning ? 'Pause' : 'Start'}</button>
      {/* Button to reset stopwatch */}
      <button onClick={reset}>Reset</button>
    </div>
  );
};

export default Stopwatch;
```

Output:-



Q3.

Code:-

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

const MessagingApp = () => {
  const [conversations, setConversations] = useState([
    { id: 1, title: 'Conversation 1', messages: [] },
    { id: 2, title: 'Conversation 2', messages: [] },
    { id: 3, title: 'Conversation 3', messages: [] }
  ]);
  const [selectedConversation, setSelectedConversation] = useState(null);
  const [newMessage, setNewMessage] = useState('');

  const selectConversation = (conversation) => {
    setSelectedConversation(conversation);
  };

  const sendMessage = () => {
    if (!newMessage.trim() || !selectedConversation) return;

    const updatedConversations = conversations.map(conv => {
      if (conv.id === selectedConversation.id) {
        return {
          ...conv,
          messages: [...conv.messages, { text: newMessage, timestamp: new
Date().toISOString() }]
        };
      }
      return conv;
    });

    setConversations(updatedConversations);
    setNewMessage('');
```

```
  };

  return (
    <div>
      <div>
        <h2>Conversations</h2>
        <ul>
          {conversations.map(conversation => (
            <li key={conversation.id} onClick={() =>
selectConversation(conversation)}>
              {conversation.title}
            </li>
          ))}
        </ul>
      </div>
      <div>
        <h2>{selectedConversation ? selectedConversation.title : 'Select a
conversation'}</h2>
        <div style={{ maxHeight: '300px', overflowY: 'auto' }}>
          {selectedConversation && selectedConversation.messages.map((message,
index) => (
            <div key={index}>{message.text} - {message.timestamp}</div>
          ))}
        </div>
        <input type="text" value={newMessage} onChange={e =>
setNewMessage(e.target.value)} />
        <button onClick={sendMessage}>Send</button>
      </div>
    </div>
  );
};

export default MessagingApp;
```

Output:-

Conversations

- Conversation 1
- Conversation 2
- Conversation 3

Conversation 1

hello!!tanmay - 2024-04-05T09:25:28.038Z

Send

Conversations

- Conversation 1
- Conversation 2
- Conversation 3

Conversation 2

how are you? - 2024-04-05T09:25:39.368Z

Send

Conversations

- Conversation 1
- Conversation 2
- Conversation 3

Conversation 3

i am fine - 2024-04-05T09:25:51.256Z

Send