

Web Tech Assignment Lab-8

T1: Code of App.js file in my React App:-

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

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

  const exchangeRate = 0.014;

  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 = amount * exchangeRate;
    setConvertedAmount(converted.toFixed(2));
  };

  return (
    <div>
      <h2>Currency Converter</h2>
      <div>
        <label>Amount:</label>
        <input type="number" value={amount} onChange={handleAmountChange} />
      </div>
      <div>
        <label>From:</label>
        <select value={fromCurrency} onChange={handleFromCurrencyChange}>
          <option value="INR">INR</option>
        </select>
      </div>
      <div>
        <label>To:</label>
        <select value={toCurrency} onChange={handleToCurrencyChange}>
          <option value="USD">USD</option>
        </select>
      </div>
    </div>
```

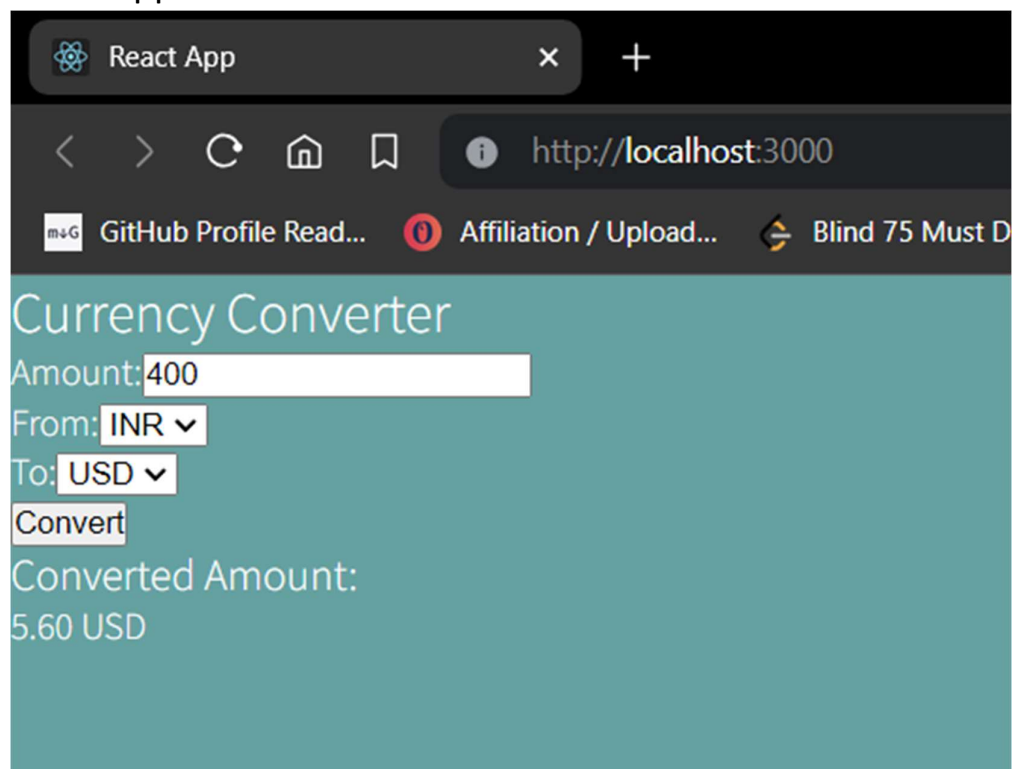
```

    <button onClick={convertCurrency}>Convert</button>
    {convertedAmount > 0 && (
      <div>
        <h3>Converted Amount:</h3>
        <p>{convertedAmount} {toCurrency}</p>
      </div>
    )}
  </div>
);
};

export default CurrencyConverter;

```

React App:



T2: Code of App.js file in my react app:

```

import React, { useState } from 'react';

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

  const startTimer = () => {
    if (!isRunning) {
      const id = setInterval(() => {
        setTime(prevTime => prevTime + 1);
      }, 1000);
    }
  }
}

```

```

    }, 1000);
    setIntervalId(id);
    setIsRunning(true);
  }
};

const pauseTimer = () => {
  clearInterval(intervalId);
  setIsRunning(false);
};

const resetTimer = () => {
  clearInterval(intervalId);
  setTime(0);
  setIsRunning(false);
};

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

  const formattedTime = [
    hours.toString().padStart(2, '0'),
    minutes.toString().padStart(2, '0'),
    seconds.toString().padStart(2, '0')
  ].join(':');

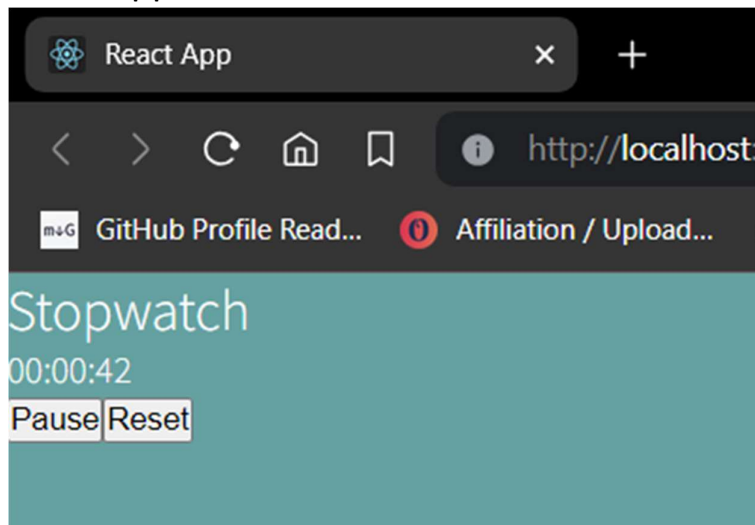
  return formattedTime;
};

return (
  <div>
    <h2>Stopwatch</h2>
    <div>
      <p>{formatTime(time)}</p>
    </div>
    <div>
      {!isRunning ? (
        <button onClick={startTimer}>Start</button>
      ) : (
        <button onClick={pauseTimer}>Pause</button>
      )}
      <button onClick={resetTimer}>Reset</button>
    </div>
  </div>
);
};

export default Stopwatch;

```

React App :-



T3: Code from App.js of react app:

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

const App = () => {
  const [userAMessages, setUserAMessages] = useState([]);
  const [userBMessages, setUserBMessages] = useState([]);
  const [newMessageA, setNewMessageA] = useState("");
  const [newMessageB, setNewMessageB] = useState("");

  const handleMessageSendA = () => {
    if (newMessageA.trim() !== "") {
      const newMessageObj = {
        text: newMessageA,
        sender: 'A',
        timestamp: new Date().toISOString()
      };
      setUserAMessages([...userAMessages, newMessageObj]);
      setNewMessageA("");
    }
  };

  const handleMessageSendB = () => {
    if (newMessageB.trim() !== "") {
      const newMessageObj = {
        text: newMessageB,
        sender: 'B',
        timestamp: new Date().toISOString()
      };
      setUserBMessages([...userBMessages, newMessageObj]);
      setNewMessageB("");
    }
  };
};
```

```

return (
  <div className="app">
    <div className="user-section">
      <h2>User A</h2>
      <div className="message-section">
        <h3>Messages from User B</h3>
        <div className="message-container">
          {userBMessages.map((message, index) => (
            <div key={index} className="message from-b">
              <p>{message.text}</p>
              <p className="timestamp">Sent by B at: {new
Date(message.timestamp).toLocaleString()}</p>
            </div>
          ))}
        </div>
        <input
          type="text"
          value={newMessageA}
          onChange={(e) => setNewMessageA(e.target.value)}
          placeholder="Type your message..."
        />
        <button onClick={handleMessageSendA}>Send</button>
      </div>
    </div>
    <div className="user-section">
      <h2>User B</h2>
      <div className="message-section">
        <h3>Messages from User A</h3>
        <div className="message-container">
          {userAMessages.map((message, index) => (
            <div key={index} className="message from-a">
              <p>{message.text}</p>
              <p className="timestamp">Sent by A at: {new
Date(message.timestamp).toLocaleString()}</p>
            </div>
          ))}
        </div>
        <input
          type="text"
          value={newMessageB}
          onChange={(e) => setNewMessageB(e.target.value)}
          placeholder="Type your message..."
        />
        <button onClick={handleMessageSendB}>Send</button>
      </div>
    </div>
  </div>
);
};

export default App;

```

React App:

