**NAME : SHANGANA YADAV**

**ROLL NO : 22IT3045**

**LAB 8**

T1.

import React, { useState } from 'react';

const ExchangeRate = {

USD: 1,

EUR: 0.85,

INR: 74.56

};

const CurrencyConverter = () => {

const [amount, setAmount] = useState(0);

const [fromCurrency, setFromCurrency] = useState('USD');

const [toCurrency, setToCurrency] = useState('EUR');

const handleAmountChange = (event) => {

setAmount(event.target.value);

};

const handleFromCurrencyChange = (event) => {

setFromCurrency(event.target.value);

};

const handleToCurrencyChange = (event) => {

setToCurrency(event.target.value);

};

const convertCurrency = () => {

const convertedAmount = (amount \* ExchangeRate[toCurrency]) / ExchangeRate[fromCurrency];

return convertedAmount.toFixed(2);

};

return (

<div>

<h2>Currency Converter</h2>

<div>

<label>

Amount:

<input type="number" value={amount} onChange={handleAmountChange} />

</label>

</div>

<div>

<label>

From:

<select value={fromCurrency} onChange={handleFromCurrencyChange}>

{Object.keys(ExchangeRate).map(currency => (

<option key={currency} value={currency}>{currency}</option>

))}

</select>

</label>

</div>

<div>

<label>

To:

<select value={toCurrency} onChange={handleToCurrencyChange}>

{Object.keys(ExchangeRate).map(currency => (

<option key={currency} value={currency}>{currency}</option>

))}

</select>

</label>

</div>

<div>

<p>Converted Amount: {convertCurrency()}</p>

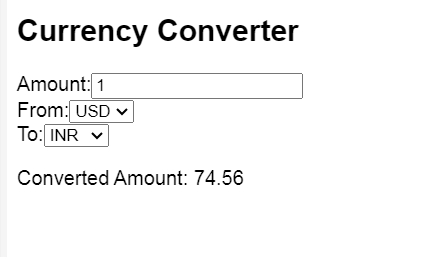
</div>

</div>

);

};

export default CurrencyConverter;



T2.

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

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 handleStart = () => {

setIsRunning(true);

};

const handlePause = () => {

setIsRunning(false);

};

const handleReset = () => {

setTime(0);

setIsRunning(false);

};

const formatTime = (timeInSeconds) => {

const hours = Math.floor(timeInSeconds / 3600);

const minutes = Math.floor((timeInSeconds % 3600) / 60);

const seconds = timeInSeconds % 60;

return `${hours < 10 ? '0' + hours : hours}:${minutes < 10 ? '0' + minutes : minutes}:${seconds < 10 ? '0' + seconds : seconds}`;

};

return (

<div>

<h2>Shangana's Stopwatch</h2>

<p>{formatTime(time)}</p>

<div>

<button onClick={handleStart} disabled={isRunning}>Start</button>

<button onClick={handlePause} disabled={!isRunning}>Pause</button>

<button onClick={handleReset}>Reset</button>

</div>

</div>

);

};

export default StopWatch;



T3.

// App.js

import React, { useState } from 'react';

import ConversationsList from './ConversationsList';

import ChatInterface from './ChatInterface';

const App = () => {

const [conversations, setConversations] = useState([]);

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

const dummyConversations = [

{ id: 1, name: "Shangana" },

{ id: 2, name: "Ravija" },

{ id: 3, name: "Aayona" }

];

const dummyMessages = [

{ id: 1, conversationId: 1, text: "Hello from Shangana" },

{ id: 2, conversationId: 1, text: "Hi there!" },

{ id: 3, conversationId: 2, text: "Hey, how are you?" },

{ id: 4, conversationId: 2, text: "I'm good, thanks!" }

];

const selectConversation = (conversationId) => {

setSelectedConversation(conversationId);

};

const fetchMessagesForConversation = (conversationId) => {

const messages = dummyMessages.filter(msg => msg.conversationId === conversationId);

return messages;

};

return (

<div>

<h1>Messaging Application</h1>

<div className="container">

<ConversationsList

conversations={dummyConversations}

selectConversation={selectConversation}

/>

<ChatInterface

selectedConversation={selectedConversation}

fetchMessagesForConversation={fetchMessagesForConversation}

/>

</div>

</div>

);

};

export default App;

// ConversationsList.js (React)

import React from 'react';

const ConversationsList = () => {

// Sample conversation messages

const conversationMessages = [

"Hi Shangana here,",

"Nice to meet you."

];

return (

<div>

<h2>Conversations</h2>

<ul>

{conversationMessages.map((message, index) => (

<li key={index}>{message}</li>

))}

</ul>

</div>

);

};

export default ConversationsList;

// ChatInterface.js (React)

import React, { useState } from 'react';

const ChatInterface = () => {

const [messages, setMessages] = useState([]);

const [newMessage, setNewMessage] = useState('');

const handleMessageChange = (event) => {

setNewMessage(event.target.value);

};

const handleSubmit = (event) => {

event.preventDefault();

if (newMessage.trim() !== '') {

// Assuming a function sendMessage exists to send the message to the backend

// sendMessage(newMessage);

// For demonstration purposes, let's just update the state directly

setMessages([...messages, { text: newMessage, sender: 'user' }]);

setNewMessage('');

}

};

return (

<div>

<h2>Chat Interface</h2>

<div className="messages">

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

<div key={index} className={`message ${message.sender}`}>

{message.text}

</div>

))}

</div>

<form onSubmit={handleSubmit}>

<input

type="text"

value={newMessage}

onChange={handleMessageChange}

placeholder="Type your message..."

/>

<button type="submit">Send</button>

</form>

</div>

);

};

export default ChatInterface;

