Chaitanya sai

22cs 3026

Q1

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

const CurrencyConverter = () => {
  // State variables
  const [amount, setAmount] = useState('');
  const [fromCurrency, setFromCurrency] = useState('USD');
  const [toCurrency, setToCurrency] = useState('EUR');
  const [convertedAmount, setConvertedAmount] = useState('');

  // Hard-coded exchange rates
  const exchangeRates = {
    USD: {
      EUR: 0.85,
      GBP: 0.72,
      CAD: 1.27
      // Add more currencies as needed
    },
    EUR: {
      USD: 1.18,
      GBP: 0.85,
      CAD: 1.48
      // Add more currencies as needed
    },
    GBP: {
      USD: 1.39,
      EUR: 1.18,
    CAD: 1.74
    // Add more currencies as needed
  },
  CAD: {
    USD: 0.79,
    EUR: 0.68,
    GBP: 0.57
    // Add more currencies as needed
  }
};

// Function to handle amount change
const handleAmountChange = (event) => {
  const value = event.target.value;
  setAmount(value);
};

// Function to handle from currency change
const handleFromCurrencyChange = (event) => {
  const value = event.target.value;
  setFromCurrency(value);
};

// Function to handle to currency change
const handleToCurrencyChange = (event) => {
  const value = event.target.value;
  setToCurrency(value);
};

// Function to handle conversion
const handleConvert = () => {
  const exchangeRate = exchangeRates[fromCurrency][toCurrency];
  const result = parseFloat(amount) * exchangeRate;
  setConvertedAmount(result.toFixed(2));
};

return (
  <div>
    <h1>Currency Converter</h1>
    <div>
      <label>Amount:</label>
      <input type="number" value={amount} onChange={handleAmountChange} />
    </div>
    <div>
      <label>From Currency:</label>
      <select value={fromCurrency} onChange={handleFromCurrencyChange}>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
        <option value="GBP">GBP</option>
        <option value="CAD">CAD</option>
      </select>
    </div>
    <div>
      <label>To Currency:</label>
      <select value={toCurrency} onChange={handleToCurrencyChange}>
        <option value="USD">USD</option>
        <option value="EUR">EUR</option>
        <option value="GBP">GBP</option>
        <option value="CAD">CAD</option>
      </select>
    </div>
    <button onClick={handleConvert}>Convert</button>
```

```
      <div>
        {convertedAmount && (
          <p>
            Converted Amount: {convertedAmount} {toCurrency}
          </p>
        )}
      </div>
    </div>
  );
};

export default CurrencyConverter;
```

Q2

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

const Stopwatch = () => {
  // State variables
  const [isRunning, setIsRunning] = useState(false);
  const [elapsedTime, setElapsedTime] = useState(0);

  // Function to start the timer
  const startTimer = () => {
    setIsRunning(true);
  };

  // Function to pause the timer
  const pauseTimer = () => {
    setIsRunning(false);
  };

  // Function to reset the timer
```

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

useEffect(() => {
  let intervalId;

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

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

// Function to format time in HH:MM:SS format
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>
      <h1>Stopwatch</h1>
      <div>
        <p>{formatTime(elapsedTime)}</p>
      </div>
      <div>
        {!isRunning ? (
          <button onClick={startTimer}>Start</button>
        ) : (
          <button onClick={pauseTimer}>Pause</button>
        )}
        <button onClick={resetTimer}>Reset</button>
      </div>
    </div>
  );
};

export default Stopwatch;
```

Q3

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

const MessagingApp = () => {
  // State variables
  const [conversations, setConversations] = useState([]);
  const [selectedConversation, setSelectedConversation] = useState(null);
  const [newMessage, setNewMessage] = useState('');
```

```
// Simulated messages
const simulatedMessages = {
  conversation1: [
    { id: 1, text: 'Hello!', sender: 'user1' },
    { id: 2, text: 'Hi there!', sender: 'user2' },
  ],
  conversation2: [
    { id: 1, text: 'How are you?', sender: 'user1' },
    { id: 2, text: 'I\'m fine, thanks!', sender: 'user2' },
  ],
};

useEffect(() => {
  // Simulated conversations
  const conversationsData = [
    { id: 'conversation1', name: 'Conversation 1' },
    { id: 'conversation2', name: 'Conversation 2' },
  ];
  setConversations(conversationsData);
}, []);

// Function to handle conversation selection
const handleConversationSelect = (conversationId) => {
  setSelectedConversation(conversationId);
};

// Function to handle message sending
const handleMessageSend = () => {
  // Add new message to selected conversation
  const updatedMessages = [...simulatedMessages[selectedConversation], { id: Date.now(), text: newMessage, sender: 'user1' }];
```

```
    simulatedMessages[selectedConversation] = updatedMessages;
    setNewMessage('');
  };

  return (
    <div>
      <h1>Messaging App</h1>
      <div className="conversations">
        <h2>Conversations</h2>
        <ul>
          {conversations.map((conversation) => (
            <li key={conversation.id} onClick={() => handleConversationSelect(conversation.id)}>
              {conversation.name}
            </li>
          ))}
        </ul>
      </div>
      <div className="messages">
        <h2>Messages</h2>
        {selectedConversation && (
          <div>
            {simulatedMessages[selectedConversation].map((message) => (
              <div key={message.id} className={message.sender === 'user1' ? 'message sent' : 'message
received'}>
                {message.text}
              </div>
            ))}
          </div>
        )}
        <div className="message-input">
          <input type="text" value={newMessage} onChange={(e) => setNewMessage(e.target.value)} />
          <button onClick={handleMessageSend}>Send</button>
        </div>
      </div>
    </div>
  );
};

export default MessagingApp;
```