Q1

import React, { useState } from 'react';

const Cur\_c = () => {

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

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

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

const [convertedAmount, setConvertedAmount] = useState(null);

const exchangeRates = {

USD: {

EUR: 0.85, // 1 USD = 0.85 EUR

},

EUR: {

USD: 1.18, // 1 EUR = 1.18 USD

},

};

const handleAmountChange = (e) => {

setAmount(e.target.value);

};

const handleFromCurrencyChange = (e) => {

setFromCurrency(e.target.value);

};

const handleToCurrencyChange = (e) => {

setToCurrency(e.target.value);

};

const handleConvert = () => {

if (!amount || isNaN(amount)) {

alert('Please enter a valid amount.');

return;

}

const exchangeRate = exchangeRates[fromCurrency][toCurrency];

const converted = parseFloat(amount) \* exchangeRate;

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:

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

<option value="USD">USD</option>

<option value="EUR">EUR</option>

</select>

</label>

</div>

<div>

<label>

To:

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

<option value="USD">USD</option>

<option value="EUR">EUR</option>

</select>

</label>

</div>

<button onClick={handleConvert}>Convert</button>

{convertedAmount && (

<div>

<p>

Converted Amount: {convertedAmount} {toCurrency}

</p>

</div>

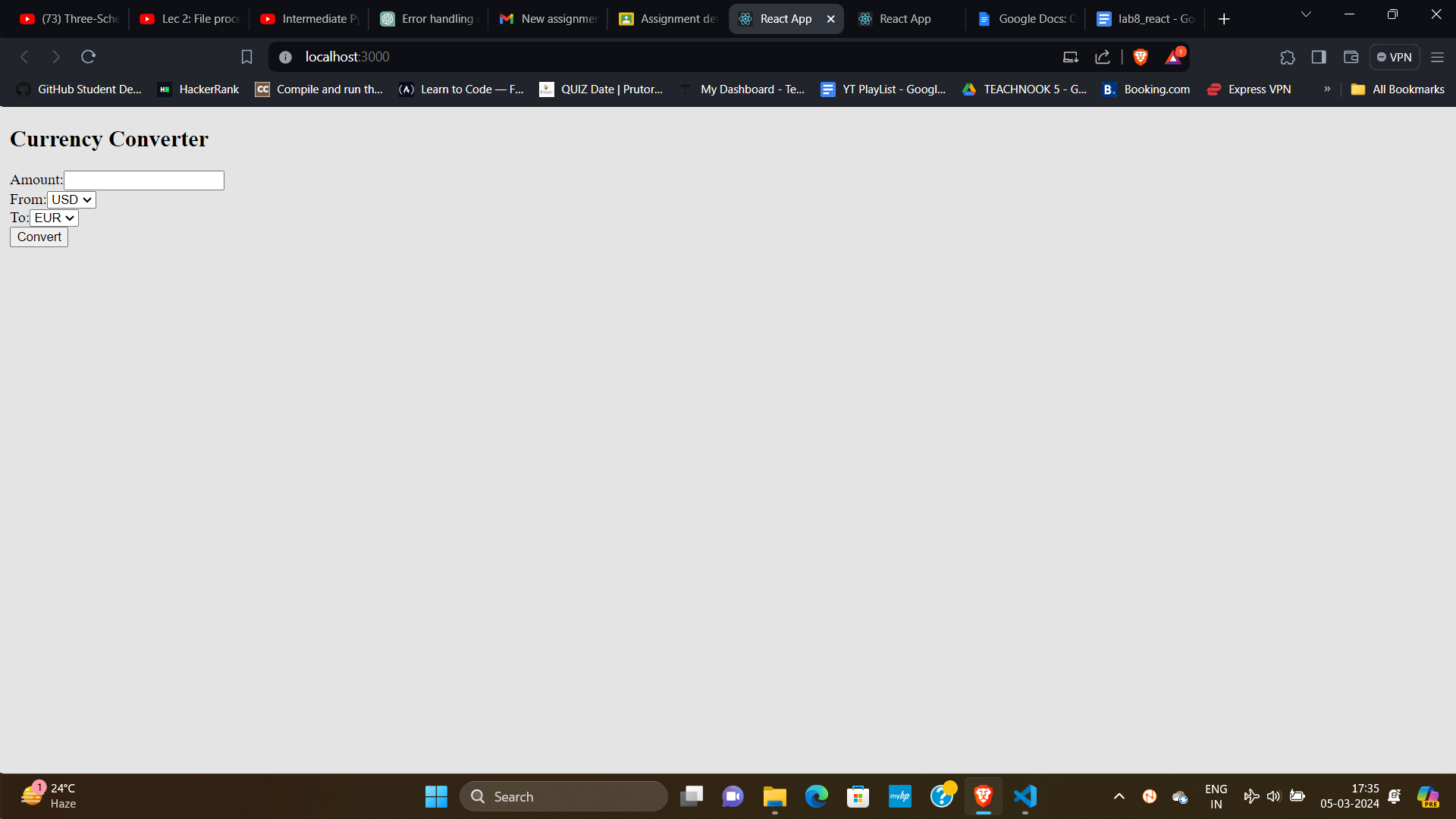
)}

</div>

);

};

export default Cur\_c;



Q2

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

const Stopwatch = () => {

const [isRunning, setIsRunning] = useState(false);

const [time, setTime] = useState(0);

const intervalRef = useRef(null);

const startTimer = () => {

setIsRunning(true);

intervalRef.current = setInterval(() => {

setTime((prevTime) => prevTime + 1);

}, 1000);

};

const pauseTimer = () => {

setIsRunning(false);

clearInterval(intervalRef.current);

};

const resetTimer = () => {

setTime(0);

clearInterval(intervalRef.current);

setIsRunning(false);

};

const formatTime = (seconds) => {

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

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

const secs = seconds % 60;

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

.toString()

.padStart(2, '0')}:${secs.toString().padStart(2, '0')}`;

};

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>

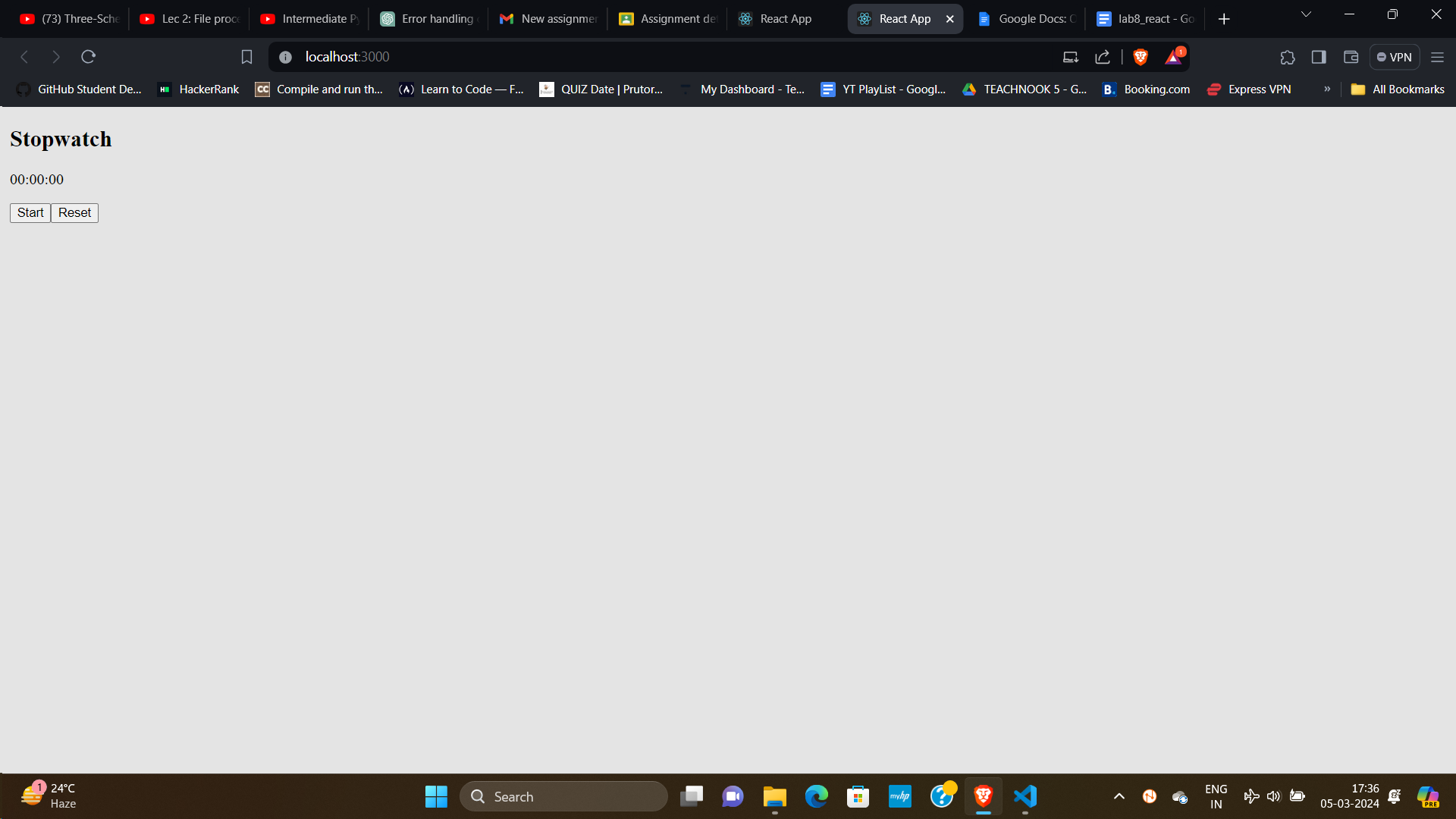
</div>

</div>

);

};

export default Stopwatch;



Q3

import React, { useState } from 'react';

const MessagingApp = () => {

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

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

const sendMessage = () => {

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

return;

}

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

setNewMessage('');

};

return (

<div>

<h2>Messages</h2>

<div style={{ border: '1px solid #ccc', padding: '10px', marginBottom: '10px' }}>

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

<div key={index}>{message.text}</div>

))}

</div>

<input

type="text"

value={newMessage}

onChange={(e) => setNewMessage(e.target.value)}

placeholder="Type your message..."

style={{ marginRight: '10px' }}

/>

<button onClick={sendMessage}>Send</button>

</div>

);

};

export default MessagingApp;

