Level 3

Task 1

Task: Implement a Chat Application

Develop a chat application using Java and networking libraries like Socket or Java NIO. This task will enhance their knowledge of network programming, client-server architecture, and message exchange.

File Edit Source Navigate Search Project Run Window Help

ChatServer.java   index.html   script.js ×

```javascript
1  document.addEventListener("DOMContentLoaded", function () {
2      const chatOutput = document.getElementById("chat-output");
3      const messageInput = document.getElementById("message-input");
4      const sendButton = document.getElementById("send-button");
5
6      sendButton.addEventListener("click", function () {
7          const message = messageInput.value;
8          if (message.trim() !== "") {
9              chatOutput.innerHTML += `<div class="message user-message">${message}</div>`;
10             messageInput.value = "";
11
12             // Simulate a response from the server (replace this with actual server communication).
13             setTimeout(() => {
14                 chatOutput.innerHTML += `<div class="message server-message">This is a server response.</div>`;
15             }, 1000);
16
17             // Scroll to the bottom of the chat.
18             chatOutput.scrollTop = chatOutput.scrollHeight;
19         }
20     });
21 });
22
```

Outline ×

- document.addEventListener("DOMContentLoaded") c
  - chatOutput
  - messageInput
  - sendButton
- sendButton.addEventListener("click") callback

Console ×

ChatServer [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:07:05 pm) [pid: 7004]

Server is running. Waiting for clients...

Writable   Insert   8 : 37 : 396

31°C Mostly sunny   ENG   05:11 PM
                                  03-10-2023

File   Edit   Source   Source   Navigate   Search   Project   Run   Window   Help

ChatServer.java   index.html ×   script.js

Outline ×

```html
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <link rel="stylesheet" type="text/css" href="style.css">
5  </head>
6  <body>
7      <div class="chat-container">
8          <div class="chat">
9              <div id="chat-output"></div>
10         </div>
11         <input type="text" id="message-input" placeholder="Type your message">
12         <button id="send-button">Send</button>
13     </div>
14     <script src="script.js"></script>
15 </body>
16 </html>
17
```

DOCTYPE:html
> html

Console ×

ChatServer [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:07:05 pm) [pid: 7004]

Server is running. Waiting for clients...

Writable   Smart Insert   1 : 1 : 0

31°C Mostly sunny   ENG   05:11 PM   03-10-2023

File   Edit   Source   Refactor   Navigate   Search   Project   Run   Window   Help

ChatServer.java ×   index.html   script.js

```java
1  import java.io.*;
2  import java.net.*;
3  import java.util.Scanner;
4
5  public class ChatServer {
6      public static void main(String[] args) {
7          try (ServerSocket serverSocket = new ServerSocket(12345)) {
8              System.out.println("Server is running. Waiting for clients...");
9
10             while (true) {
11                 Socket clientSocket = serverSocket.accept();
12                 System.out.println("Client connected: " + clientSocket.getInetAddress().getHostAddress());
13
14                 ClientHandler clientHandler = new ClientHandler(clientSocket);
15                 new Thread(clientHandler).start();
16             }
17         } catch (IOExceptio
18             e.printStackTra
19         }
20     }
```

ⓘ **ClientHandler clientHandler - ChatServer.main(String[])**

Press 'F2' for focus

Outline ×

- ChatServer
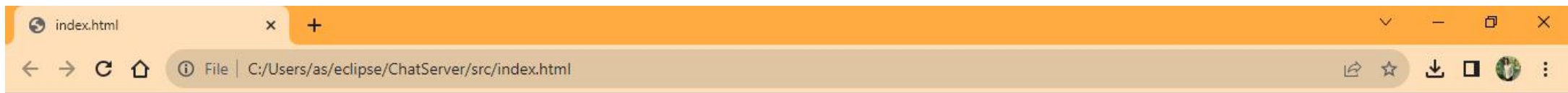 - ᔆ main(String[]) : void
  - ClientHandler

Console ×

ChatServer [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:07:05 pm) [pid: 7004]

Server is running. Waiting for clients...

Writable   Smart Insert   16 : 14 : 635

31°C Mostly sunny   ENG   05:10 PM   03-10-2023

hii
This is a server response.
hello ritesh agrawal
This is a server response.
i am done with all the task of internship
This is a server response.
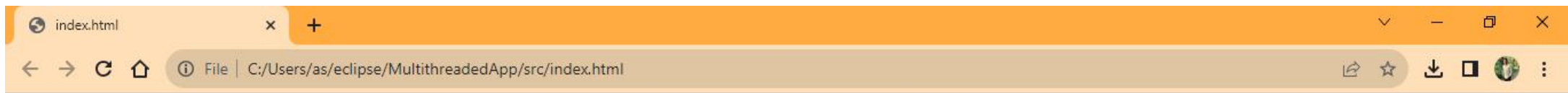
Type your message   Send

Level 3

Task 2

Task: Create a Multi-threaded Application

Build a multi-threaded Java application that demonstrates concurrency concepts like synchronization, locks, and threads. This task will help them understand concurrent programming and handling shared resources.
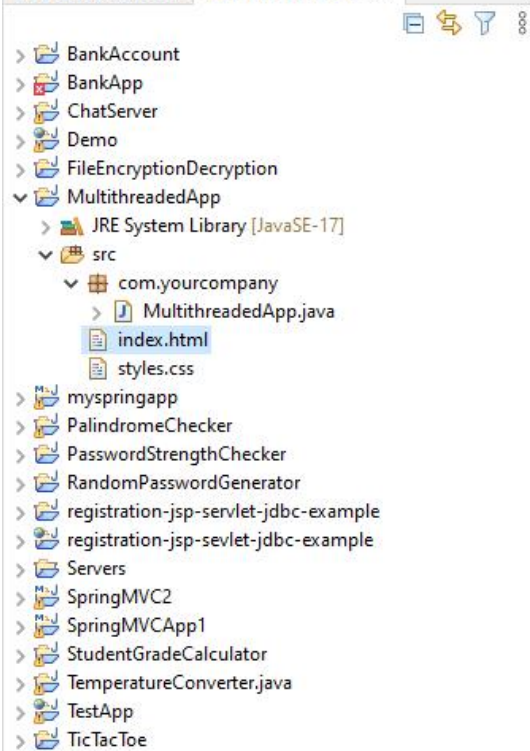
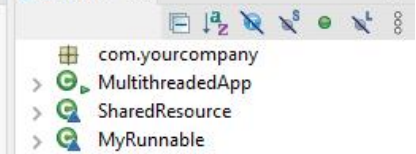**Multithreaded Java App**

Shared Data: 0

Package Explorer   Project Explorer ✕

MultithreadedApp.java ✕   index.html   styles.css

Outline ✕

> BankAccount
> BankApp
> ChatServer
> Demo
> FileEncryptionDecryption
∨ MultithreadedApp
  > JRE System Library [JavaSE-17]
  ∨ src
    ∨ com.yourcompany
      > MultithreadedApp.java
    index.html
    styles.css
> myspringapp
> PalindromeChecker
> PasswordStrengthChecker
> RandomPasswordGenerator
> registration-jsp-servlet-jdbc-example
> registration-jsp-sevlet-jdbc-example
> Servers
> SpringMVC2
> SpringMVCApp1
> StudentGradeCalculator
> TemperatureConverter.java
> TestApp
> TicTacToe

```java
1  // MultithreadedApp.java
2
3  package com.yourcompany;
4
5  public class MultithreadedApp {
6
7      public static void main(String[] args) {
8          SharedResource sharedResource = new SharedResource();
9
10         Thread thread1 = new Thread(new MyRunnable(sharedResource));
11         Thread thread2 = new Thread(new MyRunnable(sharedResource));
12
13         thread1.start();
14         thread2.start();
15     }
16  }
17
18  class SharedResource {
19      private int sharedData = 0;
20
21      public synchronized void increment() {
22          sharedData++;
23      }
24
25      public synchronized int getSharedData() {
26          return sharedData;
27      }
28  }
29
30  class MyRunnable implements Runnable {
31      private SharedResource sharedResource;
```

com.yourcompany
> MultithreadedApp
> SharedResource
> MyRunnable

Console ✕

<terminated> MultithreadedApp [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:34:41 pm – 5:34:42 pm) [pid: 142

```
Thread 15 finished. Shared data: 2000
Thread 14 finished. Shared data: 2000
```

index.html - MultithreadedApp/src

31°C Mostly sunny   ENG   05:35 PM
                                             03-10-2023

File   Edit   Source   Source   Navigate   Search   Project   Run   Window   Help

MultithreadedApp.java   index.html ×   styles.css

```html
1  <!-- index.html -->
2
3  <!DOCTYPE html>
4  <html>
5  <head>
6      <link rel="stylesheet" type="text/css" href="styles.css">
7  </head>
8  <body>
9      <div class="container">
10         <h1>Multithreaded Java App</h1>
11         <p>Shared Data: <span id="sharedData">0</span></p>
12     </div>
13 </body>
14 <script src="script.js"></script>
15 </html>
16
```

Outline ×

- index.html
- DOCTYPE:html
- > html

Console ×

<terminated> MultithreadedApp [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:34:41 pm – 5:34:42 pm) [pid: 14252]

```
Thread 15 finished. Shared data: 2000
Thread 14 finished. Shared data: 2000
```

Writable   Smart Insert   16 : 1 : 331

31°C Mostly sunny   ENG   05:35 PM
03-10-2023

File Edit Source Source Navigate Search Project Run Window Help

MultithreadedApp.java   index.html   styles.css ×

```css
1  /* styles.css */
2
3  body {
4      font-family: Arial, sans-serif;
5  }
6
7  .container {
8      text-align: center;
9      margin: 50px;
10  }
11
12  h1 {
13      font-size: 24px;
14  }
15
16  p {
17      font-size: 18px;
18  }
19
```

Outline ×

> body
> .container
> h1
> p

Console ×

<terminated> MultithreadedApp [Java Application] C:\Users\as\AppData\Local\Programs\Eclipse Adoptium\jdk-17.0.3.7-hotspot\bin\javaw.exe (03-Oct-2023, 5:34:41 pm – 5:34:42 pm) [pid: 14252]

```
Thread 15 finished. Shared data: 2000
Thread 14 finished. Shared data: 2000
```

Writable   Smart Insert   19 : 1 : 196

31°C Mostly sunny   ENG   05:35 PM 03-10-2023

Level 3

Task 3

Task: Build a Desktop Application

Develop a desktop application using JavaFX or Swing. The application could be a calculator, file manager, or any other tool that showcases your user interface design skills and event handling.

Level 3

Task 4

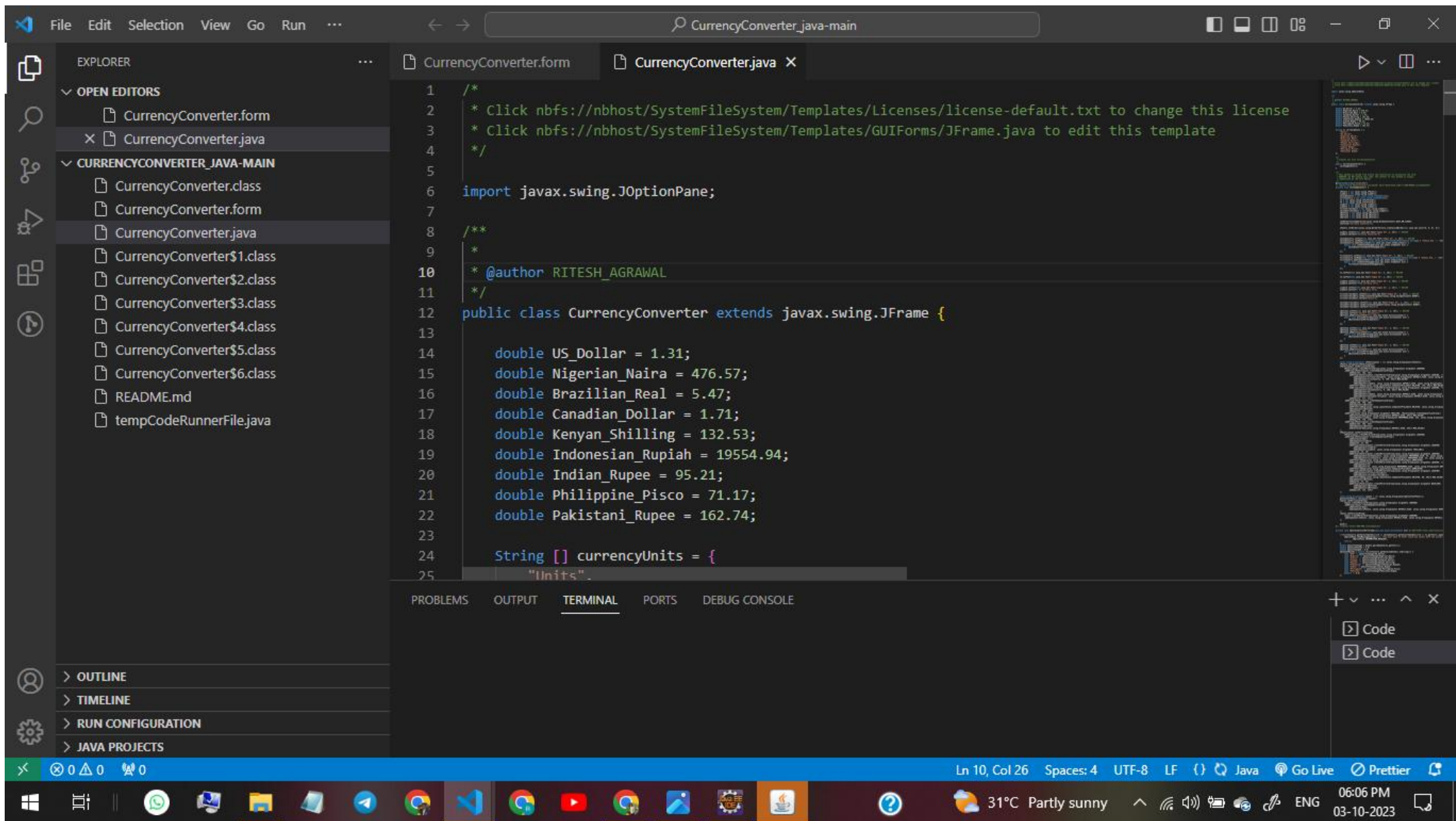**Task: Currency Converter**

Description: Build a program that converts currency between different exchange rates. Prompt the user to input an amount in a base currency and select the target currency. Retrieve the current exchange rate from an API and perform the conversion. Display the converted amount to the user.

Skills: API integration, JSON parsing, basic input/output operations.

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

import javax.swing.JOptionPane;

/**
 *
 * @author RITESH_AGRAWAL
 */
public class CurrencyConverter extends javax.swing.JFrame {

    double US_Dollar = 1.31;
    double Nigerian_Naira = 476.57;
    double Brazilian_Real = 5.47;
    double Canadian_Dollar = 1.71;
    double Kenyan_Shilling = 132.53;
    double Indonesian_Rupiah = 19554.94;
    double Indian_Rupee = 95.21;
    double Philippine_Pisco = 71.17;
    double Pakistani_Rupee = 162.74;

    String [] currencyUnits = {
        "Units",
```

Currency Converter

From Currency of

| Choose One.. ▾ |

| |

Units

To Currency of

| Choose One.. ▾ |

| |

Units

Convert Currency

Reset

Exit

Currency Converter

From Currency of

USA ▾

100

US Dollar

To Currency of

India ▾

7267.94

Indian Rupee

Convert Currency

Reset

Exit