Project Overview

-Is an application created with NetBeans IDE which runs on Java on your desktop. It features:

-Users are required to register and log in.

-A method that generates hashes with a secure messaging system

-Messages are stored using the JSON method (Google's Gson library).

-GUI elements are used by interacting with JOptionPane.

Features

-Require basic data to register a new user.

- Log in with the saved details I have.

-Send your messages in short bursts, no more than 250 characters.

- Always enter phone numbers internationally (e.g. +2712345678)

- Message IDs have 10 digits created automatically.

-You will need the message hash for tracking (format ID:Count:FirstWordLastWord)

-Hit the save client button to store your messages in messages.json

-Use the interface to see when each of your messages has been sent.

-Soon you will be able to see your recent send messages.

Technologies Used

- Java
- NetBeans IDE
- Gson (for JSON serialization)
- JOptionPane (for GUI dialogs)

How to Run the Project

-Open the project with NetBeans IDE.

-Include the Gson library as another project library.

-Start by running the class called UserAuthProgPoe.

-Use the steps on the desktop to:

- Add a user to your profile.

-Log in

-Put messages into storage and share them when needed.

Message Hash Purpose

Each message includes a **unique hash** to:

- Ensure message integrity
- Prevent tampering
- Add a basic layer of security for tracking messages

User Experience (UX)

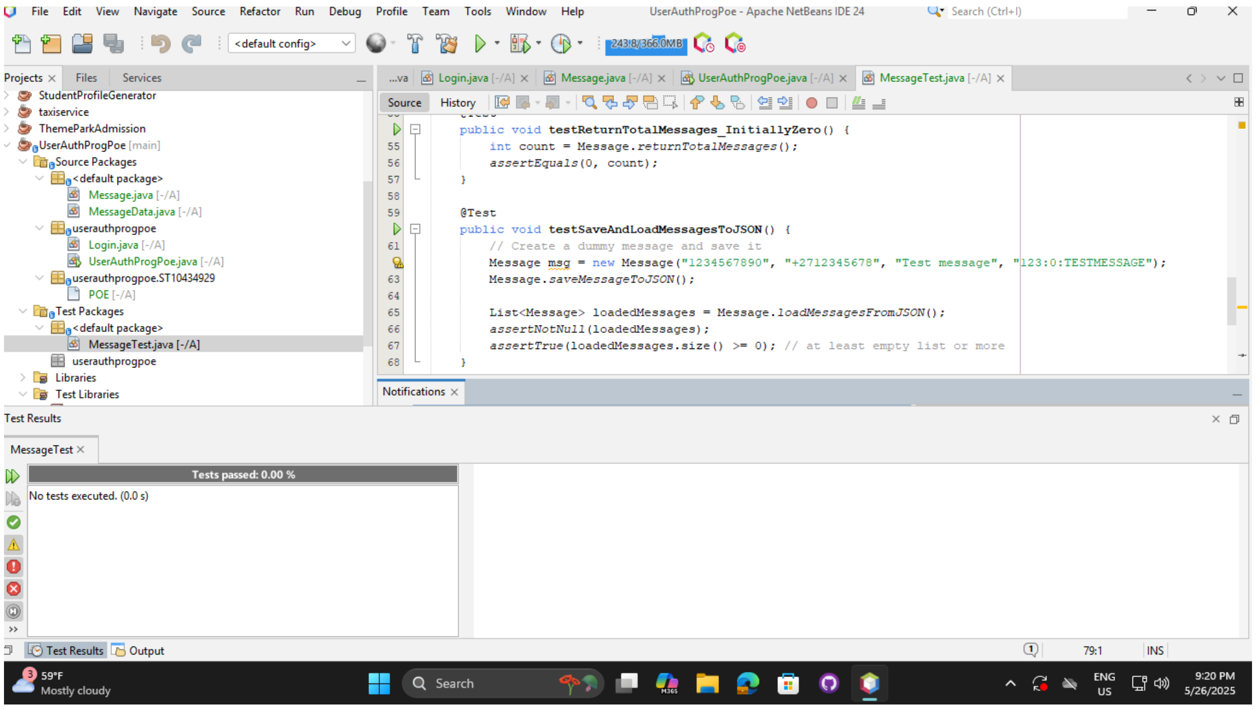
Instead of using a console interface, **JOptionPane** provides:

- Clean popup prompts
- Dialog windows for input and confirmation
- Better usability and error handling for beginners

Credits

Developed by **Dinilla Paulse**
Student Number: **St10434929**
Course: Bachelor in Computer Science
Institution: Varsity College

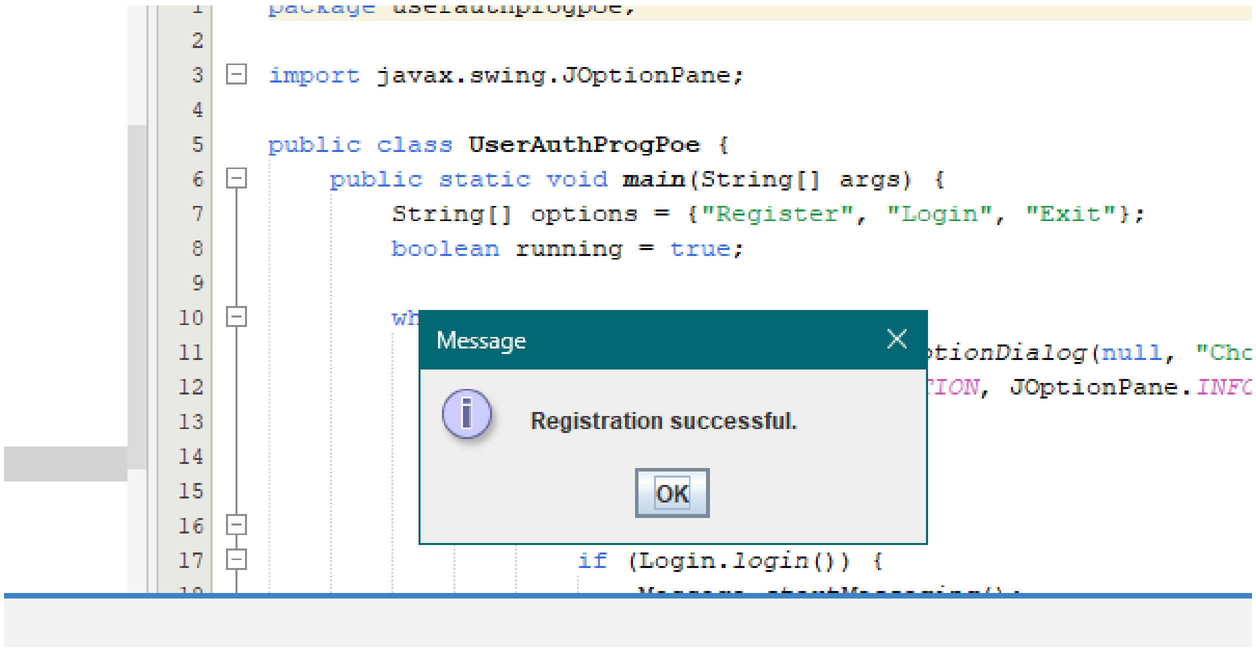
UNIT TESTS:



```java
    public void testReturnTotalMessages_InitiallyZero() {
        int count = Message.returnTotalMessages();
        assertEquals(0, count);
    }

    @Test
    public void testSaveAndLoadMessagesToJSON() {
        // Create a dummy message and save it
        Message msg = new Message("1234567890", "+2712345678", "Test message", "123:0:TESTMESSAGE");
        Message.saveMessageToJSON();

        List<Message> loadedMessages = Message.loadMessagesFromJSON();
        assertNotNull(loadedMessages);
        assertTrue(loadedMessages.size() >= 0); // at least empty list or more
    }
```

```
Tests passed: 0.00 %
No tests executed. (0.0 s)
```

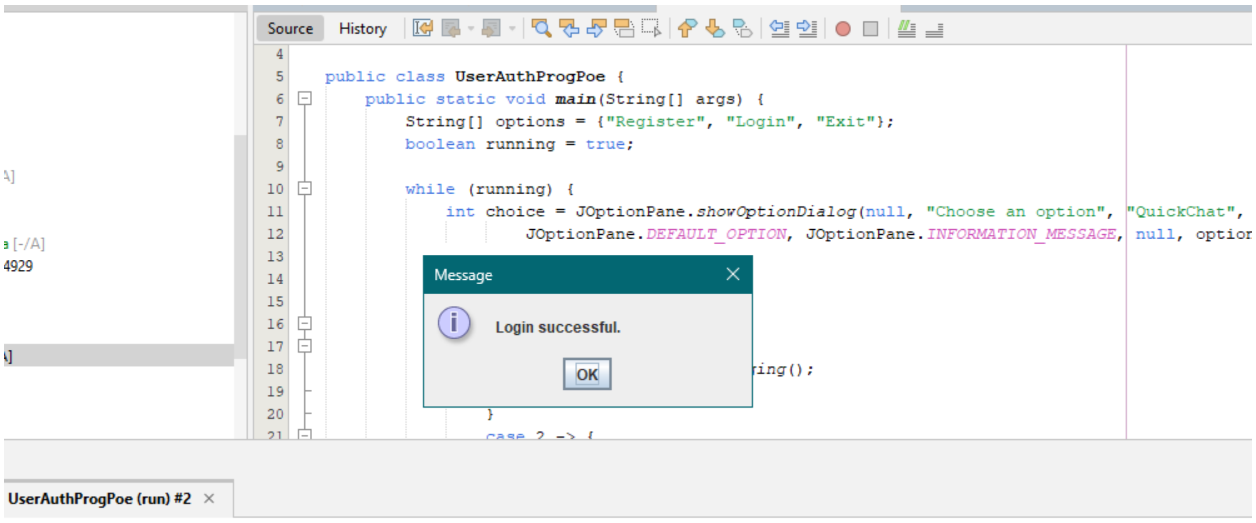
PROOF OF CODE WORKING



```
package userauthprogpoe;

import javax.swing.JOptionPane;

public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        wh                                       tionDialog(null, "Cho
                                               TION, JOptionPane.INFO



                    if (Login.login()) {
```

Message

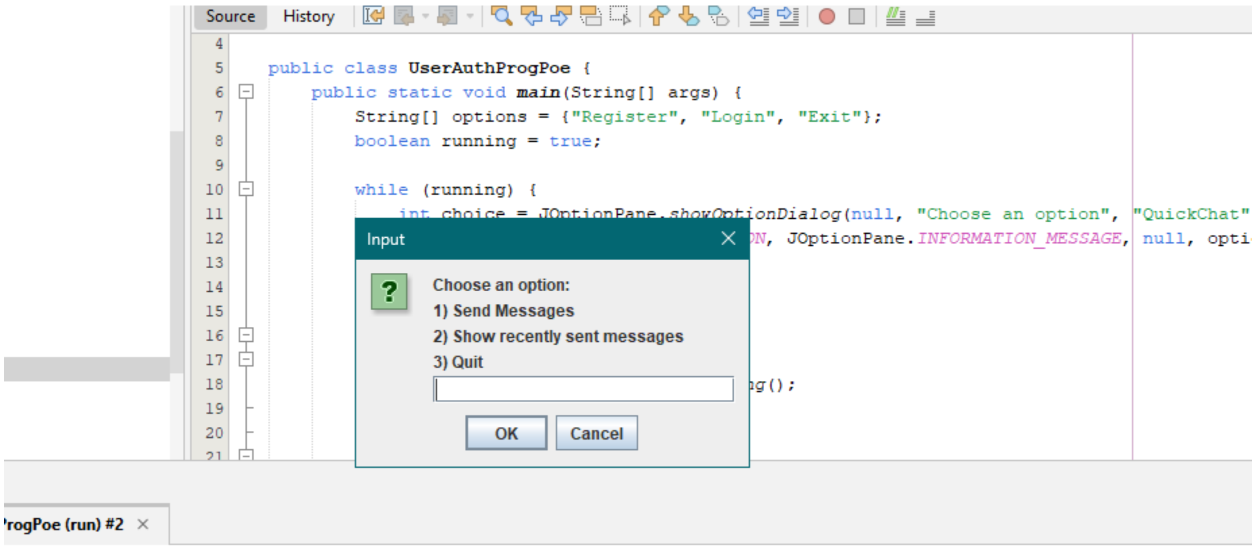
Registration successful.



```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an option", "QuickChat",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, optior
                                                                    ing();

                }
                case 2 -> {
```

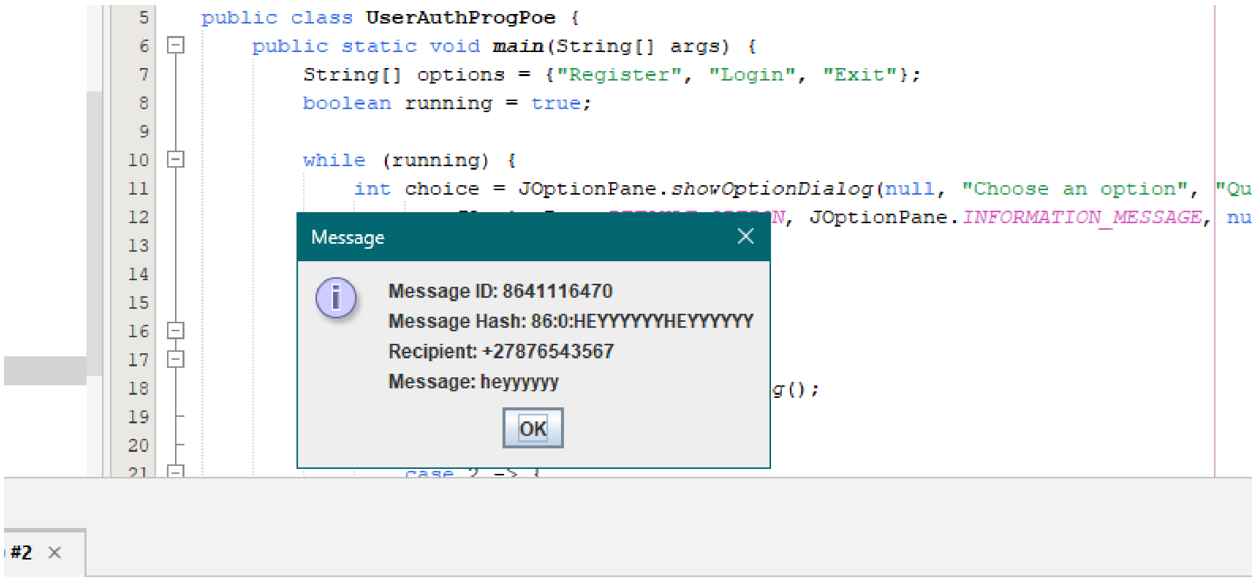
Message

Login successful.



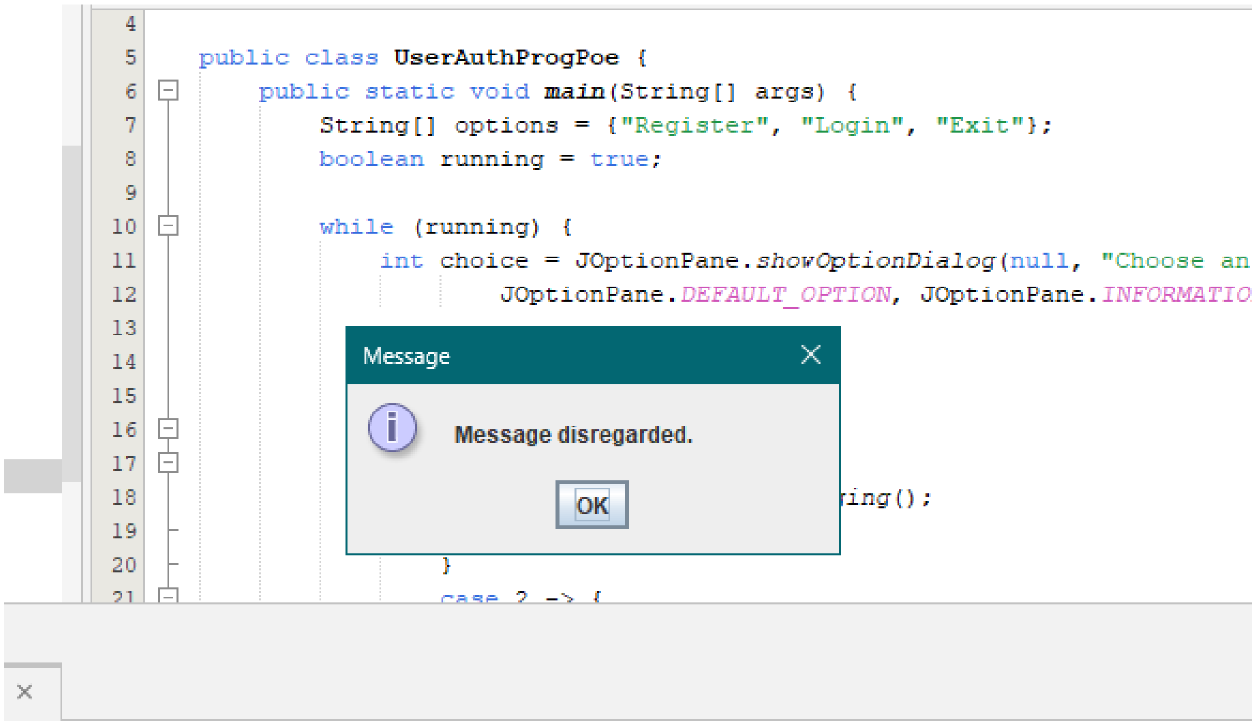
```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an option", "QuickChat"
                                    N, JOptionPane.INFORMATION_MESSAGE, null, opti
```

```
Input

Choose an option:
1) Send Messages
2) Show recently sent messages
3) Quit
```



```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an option", "Qu
                                    N, JOptionPane.INFORMATION_MESSAGE, nu
```

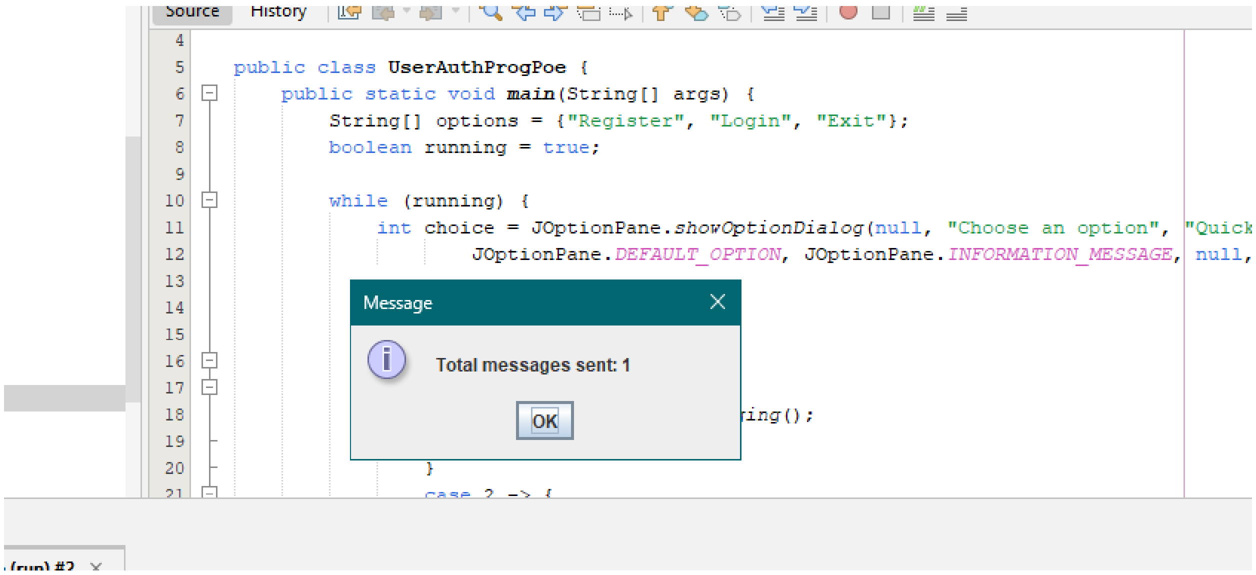
```
Message

Message ID: 8641116470
Message Hash: 86:0:HEYYYYYYHEYYYYYY
Recipient: +27876543567
Message: heyyyyyy
```



```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATIO
```

```
Message
Message disregarded.
OK
```

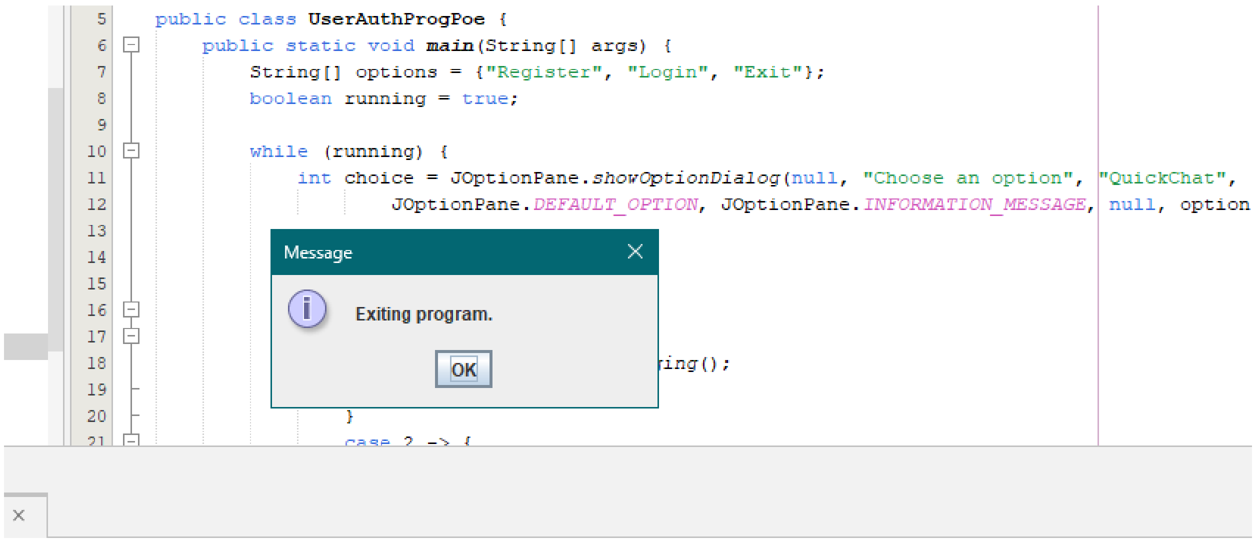
```
ing();
            }
            case 2 -> {
```



```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an option", "Quick
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null,
```

```
Message
Total messages sent: 1
OK
```

```
ing();
            }
            case 2 -> {
```



```
public class UserAuthProgPoe {
    public static void main(String[] args) {
        String[] options = {"Register", "Login", "Exit"};
        boolean running = true;

        while (running) {
            int choice = JOptionPane.showOptionDialog(null, "Choose an option", "QuickChat",
                    JOptionPane.DEFAULT_OPTION, JOptionPane.INFORMATION_MESSAGE, null, option

                                        ing();

                }
                case 2 -> {
```

Message

Exiting program.

Reference:

OpenAI's ChatGPT (2025) the regular expression logic required for South African cell phone number validation in the registration system received assistance from the analytic tool.

OpenAI. (2025). ChatGPT May 22 version) [Large language model]. https://chat.openai.com/