CYBER-AI ASSISTANT USING JAVA

Mini Project Submitted to

JOY UNIVERSITY

In partial fulfilment of the requirements for the award of the degree of

The Bachelor of Technology in Computer Science

By

Gonupalli Ranjith Kumar

PNR: 2024BTDS087

M Sairam Avinash

PNR: 2024BTDS088

Under the Guidance Of

Prof. Fernando Sahaya Mary Albeena

Assistant Professor

School of Computational Intelligence

BONAFIDE CERTIFICATE

This is to certify that the project titled **"CYBER-AI ASSISTANT USING JAVA"** is a bonafide record of work carried out by **GONUPALLI RANJITH KUMAR**, **Roll No: 2024BTDS087**, submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (AI & DS)** at **JOY UNIVERSITY**, Raja Nagar, Alaganeri, Near Kanyakumari, College Road, Vaddakkankulam, Tamil Nadu – 627116.

This work was carried out during the period **July – November, 2025** under the guidance of:

Submitted for the Viva-voce examination held on _____

ACKNOWLEDGEMENT

I express my sincere thanks to **PROF. FERNANDO SAHAYA MARY ALBEENA,** Assistant Professor, School of Computational Intelligence, for providing me with the necessary support, guidance, and encouragement throughout the completion of this project. Her valuable suggestions and constant motivation have been instrumental in the successful development of this work.

I would also like to extend my heartfelt gratitude to all the faculty members of the **School of Computational Intelligence, Joy University**, for their continuous support and for creating a conducive learning environment that enabled me to carry out this project effectively.

My deep appreciation and thanks go to my **family and friends**, whose constant encouragement, understanding, and support gave me the strength to overcome challenges and complete this project successfully.

I also thank all those who have directly or indirectly contributed to the successful completion of this project work.

Finally, I express my gratitude to the **Almighty** for giving me the strength, wisdom, and perseverance to complete this project successfully.

DECLARATION

I hereby declare that this project work titled **"Cyber-AI Assistant Using Java"** is an original record of work carried out by me under the supervision and guidance of **PROF. FERNANDO SAHAYA MARY ALBEENA,** Assistant Professor, **School of Computational Intelligence**, Joy University. I further declare that this project has not been submitted, in part or full, for the award of any **Degree, Diploma, Associateship, Fellowship, or any other similar title** to any candidate of any other university or institution.

Place:
Date:

GONUPALLI RANJITH KUMAR

Bachelor of Technology
School of Computational Intelligence
Joy University

TABLE OF CONTENT

Project Title: Cyber-AI Assistant Using Java

AIM :

The aim of this project is to design and develop an intelligent **Cyber-AI Assistant** using Java that helps users identify, understand, and prevent cyber security threats through an interactive chatbot interface. This project focuses on building a simple but effective artificial intelligence system capable of recognizing cyber-related keywords such as phishing, malware, hacking, password safety, online fraud, and insecure networks. Using these detected keywords, the chatbot will provide accurate, real-time cyber awareness tips and best practices to ensure digital safety.

The primary objective of the Cyber-AI Assistant is to spread cyber security awareness by responding intelligently to user queries and guiding them on how to stay safe in the digital world. The system aims to simplify the understanding of cyber threats by offering step-by-step assistance, security precautions, and preventive measures that users can follow immediately. It acts as a virtual cyber consultant capable of detecting suspicious situations described by the user and recommending secure actions.

ABSTRACT

The Cyber-AI Assistant is an intelligent, rule-based chatbot system designed to promote cyber awareness and assist users in understanding common cyber threats. Built using Java and implemented through the Swing graphical user interface framework, the system aims to provide real-time, conversational guidance to users in areas such as phishing detection, malware prevention, password security, safe browsing, and account protection. As cyber threats continue to increase in complexity and frequency, individuals often lack the necessary knowledge to identify harmful activities online. This project addresses this gap by offering a simple yet effective AI-powered chatbot capable of recognizing cyber risk indicators based on user input.

The system works on the principle of **Natural Language Processing through keyword-based text analysis**, where user messages are examined for important cyber-related terms such as “hacked,” “virus,” “phishing,” “malware,” “fraud,” “weak password,” and “unsafe wifi.” When such keywords are detected, the chatbot applies its pre-defined knowledge base to provide the most relevant and accurate cybersecurity response. These responses include preventive actions, step-by-step safety instructions, cybersecurity best practices, and awareness tips that empower users to immediately secure their digital activities. Because the system is rule-based and modular, new threat categories can be added easily, making the chatbot extensible and adaptable to emerging cyber trends.

The Cyber-AI Assistant emphasizes user accessibility and ease of use. The GUI supports smooth conversation flow, ensuring that even users with minimal technical expertise can understand cybersecurity concepts through simple dialogue. The project's architecture is designed to be lightweight, allowing it to run efficiently on standard computer systems without requiring external resources or internet connectivity. This feature makes the tool useful for educational institutions, cybersecurity awareness programs, and introductory training environments.

EXISTING SYSTEM (CURRENT FUNCTIONALITIES & LIMITATIONS)

Existing Cyber-AI Systems Can:

Below are the common features available in basic cyber-security assistant applications currently in use:

1. Provide basic cyber awareness tips

Most existing AI-based cyber tools are limited to offering simple security advice such as avoiding suspicious links, using strong passwords, and updating software. These are general guidelines that do not change dynamically based on user behavior or threat levels.

2. Warn users about general cyber threats

These systems can respond to common cyber threats such as phishing, malware, and password breaches. They typically use pre-written messages to educate users about risks. However, the warnings are generic and do not adapt to more complex or newer attacks.

3. Work as rule-based security assistants

Existing chatbots operate on keyword detection or predefined patterns. If a user message contains words like "virus" or "hacked," the system shows a matching response. This makes them simple and lightweight, but also limited in intelligence and learning ability.

4. Detect keywords like virus, phishing, hack, password, fraud

Most current systems rely on text analysis using keyword spotting. When the chatbot detects specific cybersecurity-sensitive words, it triggers appropriate predefined replies. However, this model fails when the user describes threats in different wording or complex sentences.

5. Offer simple solutions for cybersecurity problems

These chatbots can give quick, straightforward solutions such as updating antivirus, resetting passwords, or avoiding unknown attachments. These solutions help beginners but are insufficient for real-time cyber threat management.

Limitations of Current Systems:

While basic cyber-AI assistants offer useful information, they have several limitations:

1. Cannot understand complex cybersecurity threats

Most systems depend on simple keyword matching. They cannot interpret long, descriptive messages where users explain real cyber incidents. They also fail to understand indirect or technical descriptions, making them unreliable for serious cases.

2. No real-time threat detection

Modern cyber threats evolve quickly, but basic AI chatbots do not analyze network activity, system processes, or online behavior. They provide information but cannot detect threats happening at the moment, such as ongoing phishing attempts or malware attacks.

3. No machine learning or deep analysis

Current systems lack advanced AI capabilities.
They do not:

- Learn from previous interactions
- Identify patterns over time
- Predict threats
- Adapt responses based on user behavior

Without machine learning, these chatbots stay static and cannot grow in intelligence.

4. Cannot integrate with antivirus or firewall

Most existing cyber chatbots work as standalone educational tools. They cannot interact with operating system security tools such as:

- Antivirus software
- Firewalls
- Real-time scanning engines
- Network monitoring tools

This limits their ability to provide system-level protection.

PROPOSED SYSTEM (ENHANCED FEATURES – DETAILED EXPLANATION)

The proposed **Cyber-AI Assistant** is designed to overcome the shortcomings of existing rule-based cyber systems and to introduce a more intelligent, interactive, and security-focused solution. After implementing this system, users will receive real-time cyber awareness support, instant threat analysis, and personalized security suggestions through a simple Java-based AI chatbot interface.

---

1. Keyword-based Cyber Threat Detection (Enhanced Intelligence)

The proposed system uses a more advanced keyword-detection mechanism that analyzes user input to identify cyber threats.
It recognizes terms such as:

- *phishing*
- *hacked*
- *virus / malware / trojan*
- *scam*

- *password leak*
- *unauthorized access*
- *identity theft*
- *fraud messages*

When such keywords are detected, the system immediately classifies the threat category and responds with relevant solutions.
This makes the chatbot smarter because it understands cyber-related language more deeply than basic systems.

---

2. Instant Cyber Safety Guidance (Smart Response Engine)

For every threat identified, the system provides:

- step-by-step safety instructions,
- recommended security actions,
- and immediate precautions to prevent damage.

Example:
If the user types *"I got a phishing email from bank"*, the system will respond with:

- Do not click the link
- Do not share OTP or passwords
- Verify the sender's email
- Report the message to authorities

This feature makes the system practical and useful in real-life cyber emergencies.

---

3. Secure Digital Life Recommendations (Daily Cyber Hygiene Tips)

The AI Assistant provides continuous cybersecurity guidance such as:

- how to create a strong password,
- how to identify fake websites,
- safe browsing habits,
- importance of two-factor authentication,
- precautions while using public Wi-Fi,
- safe social media practices,
- how to avoid malware downloads.

This ensures users develop long-term habits for staying safe online.

---

4. User-Friendly GUI (Interactive Chat Environment)

A clean, simple Java Swing interface allows users to chat with the AI assistant in a conversation-like format.
Features include:

- Text input box
- Chat display panel
- Scrollable dialogue history
- Clear, readable message formatting

This makes the system easy for anyone to use, even with minimal technical knowledge.

---

5. Future-ready and Scalable (Designed for Expansion)

The system is built in a modular structure so that new features can be added without rewriting the entire code.
Future enhancements may include:

- Integration of **Machine Learning (ML)** to understand natural language better
- Real-time scanning of suspicious links
- Cyber threat prediction models
- Voice-based cyber assistant
- Integration with antivirus APIs
- Automatic incident reporting
- AI-based phishing email classifier

This ensures that the project can grow into a fully intelligent cybersecurity assistant.

References

1. **Oracle Java Documentation** – https://docs.oracle.com/javase/
   (For Swing components, event handling, and HashMap usage)
2. **GeeksforGeeks Java Tutorials** – https://www.geeksforgeeks.org/java-swing/
   (For understanding GUI design and layout management)
3. **W3Schools Java Guide** – https://www.w3schools.com/java/
   (For Java basics and syntax reference)
4. **TutorialsPoint – Java Programming** – https://www.tutorialspoint.com/java/
   (For event-driven programming and user interface handling)
5. **Cyber Security Awareness Articles** – https://www.cisa.gov/
   (For cyber threat definitions and safety practices used in AI responses)

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import java.text.SimpleDateFormat;

public class CyberAIAssistant extends JFrame implements ActionListener {

   private JTextArea chatArea;
   private JTextField inputField;
   private JButton sendButton;

   private HashMap<String, String> cyberResponses;
   private HashMap<String, String> threatLevels;

   public CyberAIAssistant() {

      setTitle("Cyber Security AI Assistant — Advanced Version");
      setSize(700, 600);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());
```

```
        setLocationRelativeTo(null);

        chatArea = new JTextArea();
        chatArea.setEditable(false);
        chatArea.setFont(new Font("Arial", Font.PLAIN, 16));
        chatArea.setLineWrap(true);
        chatArea.setWrapStyleWord(true);

        JScrollPane scroll = new JScrollPane(chatArea);

scroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLL
BAR_ALWAYS);

        inputField = new JTextField();
        inputField.addActionListener(this);

        sendButton = new JButton("Send");
        sendButton.addActionListener(this);

        JPanel bottom = new JPanel(new BorderLayout());
        bottom.add(inputField, BorderLayout.CENTER);
        bottom.add(sendButton, BorderLayout.EAST);

        add(scroll, BorderLayout.CENTER);
```

```
    add(bottom, BorderLayout.SOUTH);

    initializeResponses();

    printWelcomeMessage();

    setVisible(true);
}

private void printWelcomeMessage() {
    appendBot(
        "👋 Welcome to Cyber Security AI Assistant!\n"
            + "-----------------------------------------------\n"
            + "You can ask me about:\n"
            + "• Virus / Malware\n"
            + "• Hacked account\n"
            + "• Password safety\n"
            + "• Phishing attacks\n"
            + "• Online Scams\n"
            + "• Data Leak / Privacy issues\n"
            + "• Fraud / OTP scam\n"
            + "\nType anything related to cyber security.\n");
}
```

```
private void initializeResponses() {

    cyberResponses = new HashMap<>();
    threatLevels = new HashMap<>();

    cyberResponses.put("virus",
        "⚠ **Virus Threat Detected!**\n\n"
            + "Here is your detailed solution:\n"
            + "1️⃣ Disconnect from the internet.\n"
            + "2️⃣ Run a full antivirus scan.\n"
            + "3️⃣ Delete unknown apps & suspicious files.\n"
            + "4️⃣ Do not plug USB devices into infected PC.\n"
            + "5️⃣ Update system and antivirus.\n"
            + "6️⃣ Avoid downloading cracked software.\n");

    threatLevels.put("virus", "THREAT LEVEL: 🔥 HIGH\n");

    cyberResponses.put("hacked",
        "⚠ **Your account may be hacked!**\n\n"
            + "Follow these steps:\n"
            + "1️⃣ Immediately change your password.\n"
            + "2️⃣ Enable 2-Step Verification.\n"
```

```
            + "3️⃣ Log out from all devices.\n"
            + "4️⃣ Check for unauthorized activity.\n"
            + "5️⃣ Contact support team if needed.\n");

threatLevels.put("hacked", "THREAT LEVEL: 🔥 HIGH\n");

cyberResponses.put("password",
    "🔐 **Password Safety Tips**\n"
        + "• Use 12+ characters\n"
        + "• Mix uppercase, lowercase, numbers, symbols\n"
        + "• Avoid names, DOB, phone number\n"
        + "• Set different password for every account\n");

threatLevels.put("password", "THREAT LEVEL: 🟡 MEDIUM\n");

cyberResponses.put("phishing",
    "⚠ **Phishing Alert!**\n\n"
        + "Do the following:\n"
        + "1️⃣ Do NOT click unknown links.\n"
        + "2️⃣ Do NOT share OTP or passwords.\n"
        + "3️⃣ Verify sender email.\n"
        + "4️⃣ Check website address before login.\n");
```

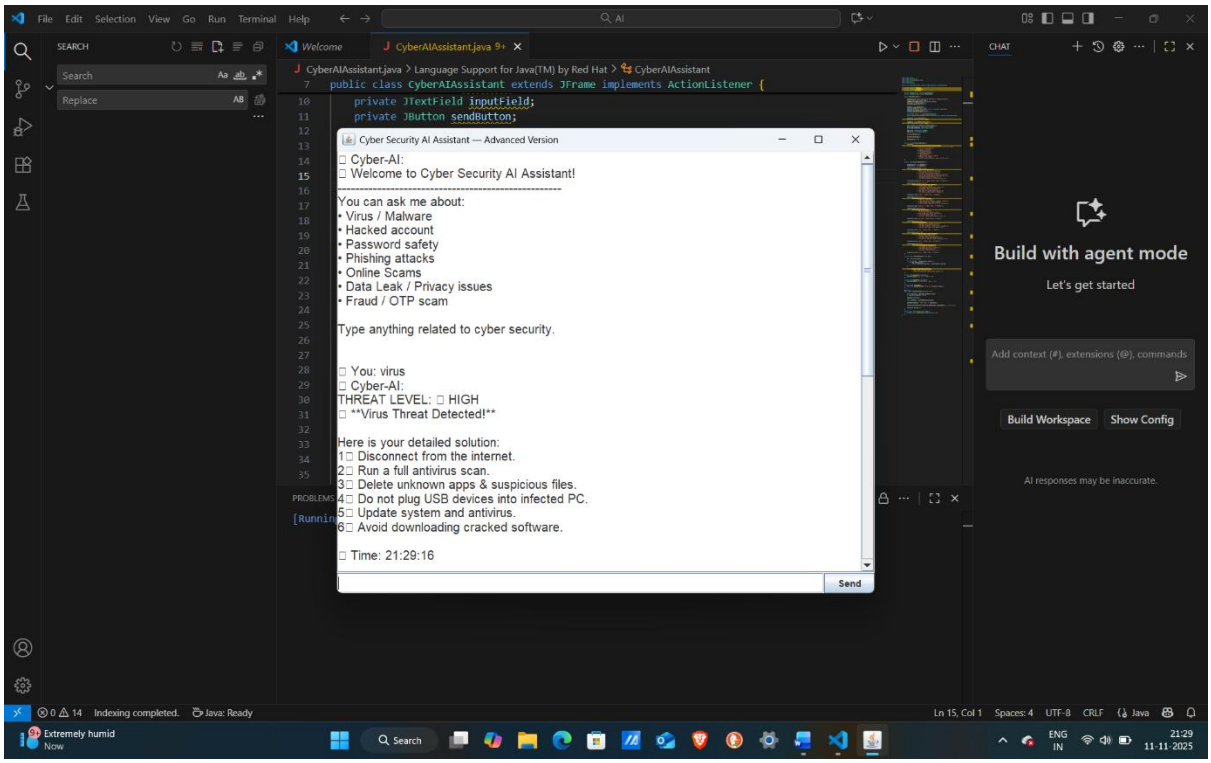
```
threatLevels.put("phishing", "THREAT LEVEL: 🔥 HIGH\n");

cyberResponses.put("scam",
    "⚠ **Scam Warning!**\n"
        + "1️⃣ Don’t trust unknown callers.\n"
        + "2️⃣ Don’t share bank details.\n"
        + "3️⃣ Don’t respond to prize/lottery messages.\n"
        + "4️⃣ Block and report scam numbers.\n");

threatLevels.put("scam", "THREAT LEVEL: 🔥 HIGH\n");

cyberResponses.put("fraud",
    "⚠ **Banking Fraud Attempt Detected!**\n"
        + "Follow these steps now:\n"
        + "1️⃣ Never share OTP or PIN.\n"
        + "2️⃣ Use official banking app only.\n"
        + "3️⃣ Report fraudulent number immediately.\n");

threatLevels.put("fraud", "THREAT LEVEL: 🔥 HIGH\n");

cyberResponses.put("leak",
    "⚠ **Possible Data Leak Detected!**\n"
```

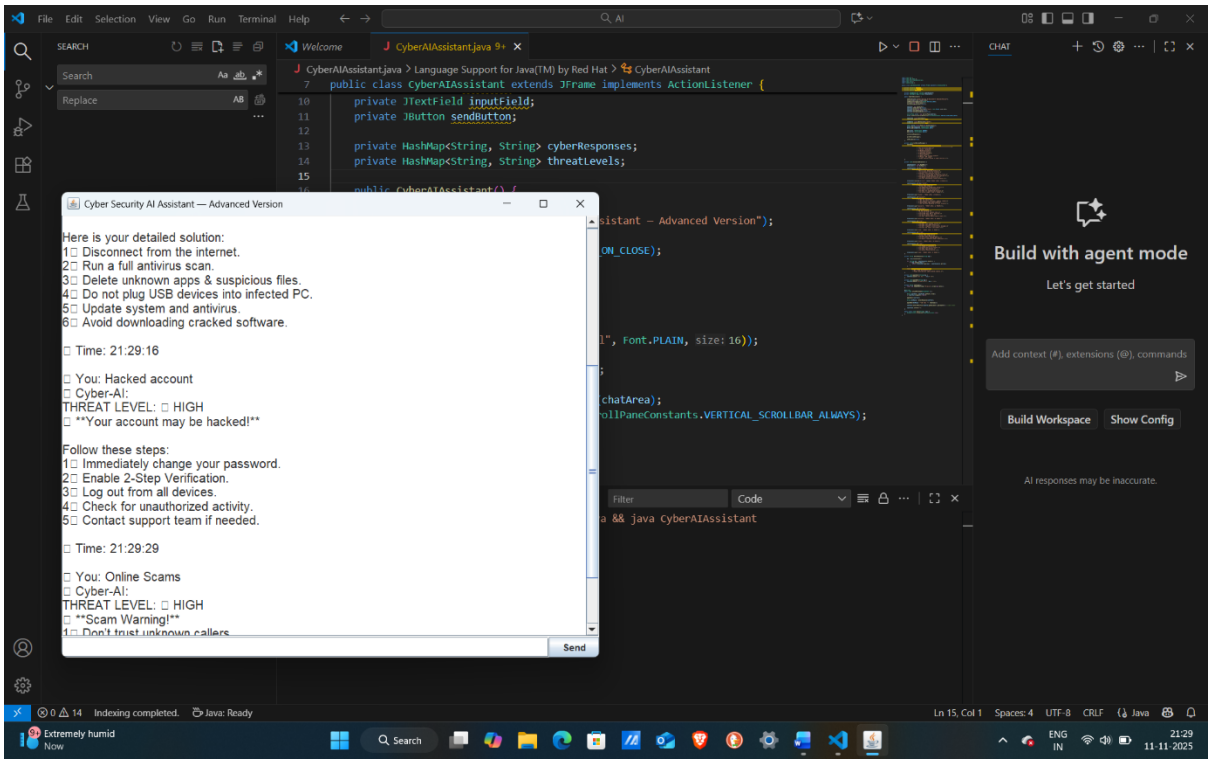
```
                + "1️⃣ Change all passwords.\n"
                + "2️⃣ Enable 2FA on all accounts.\n"
                + "3️⃣ Check login history.\n"
                + "4️⃣ Remove unknown devices.\n");

    threatLevels.put("leak", "THREAT LEVEL: 🔥 HIGH\n");
}

private String detectResponse(String msg) {

    msg = msg.toLowerCase();

    for (String key : cyberResponses.keySet()) {
        if (msg.contains(key)) {
            return threatLevels.get(key) + cyberResponses.get(key);
        }
    }

    return "✔ I understand your message.\n"
            + "But I need more details.\n"
            + "Please describe your cyber problem clearly.\n";
}
```

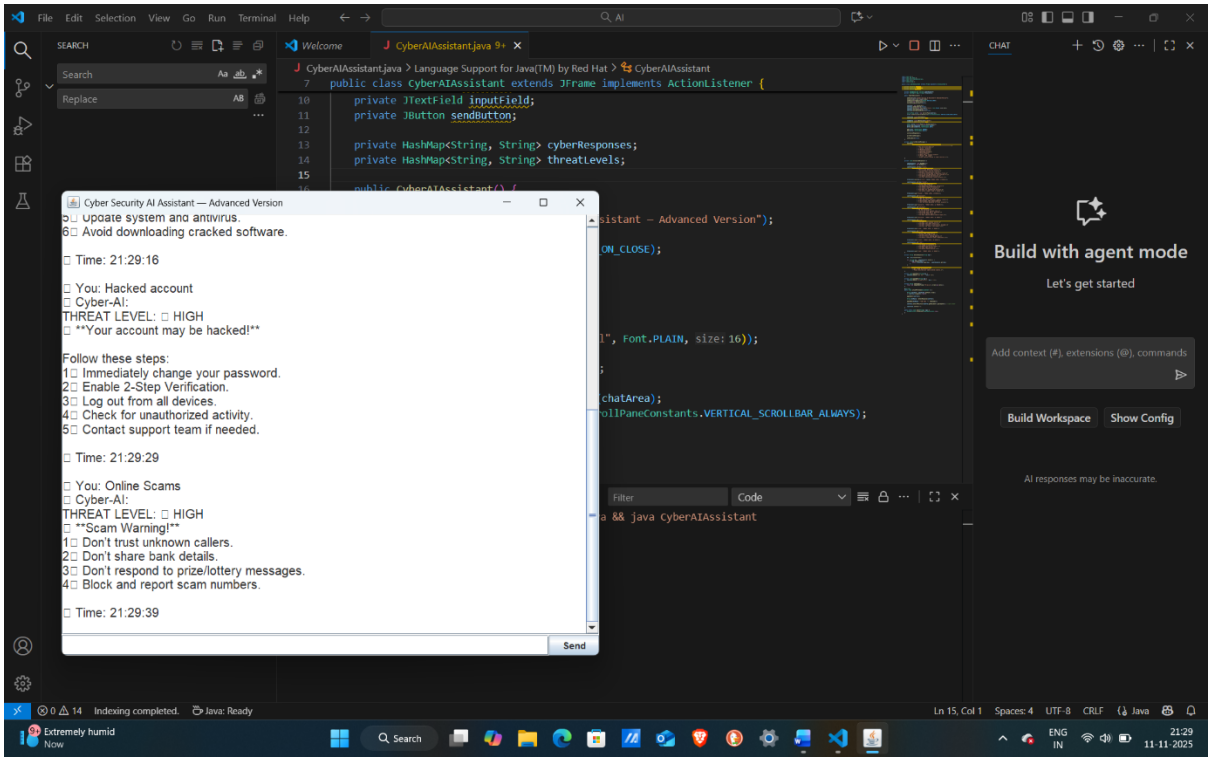
```
private void appendUser(String msg) {
   chatArea.append("\n 👱 You: " + msg + "\n");
}

private void appendBot(String msg) {
   chatArea.append(" 🤖 Cyber-AI:\n" + msg + "\n");
}

private String timestamp() {
   return new SimpleDateFormat("HH:mm:ss").format(new Date());
}

@Override
public void actionPerformed(ActionEvent e) {

   String userText = inputField.getText().trim();
   if (userText.isEmpty()) return;

   appendUser(userText);

   String botReply = detectResponse(userText);

   appendBot(botReply + "\n⏱️ Time: " + timestamp());
```

```
        chatArea.setCaretPosition(chatArea.getDocument().getLength()); //
auto scroll

        inputField.setText("");
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(CyberAIAssistant::new);
    }
}
```

OUTPUT


Cyber Security AI Assistant — Advanced Version
Cyber-AI:
Welcome to Cyber Security AI Assistant!
You can ask me about:
• Virus / Malware
• Hacked account
• Password safety
• Phishing attacks
• Online Scams
• Data Leak / Privacy issues
• Fraud / OTP scam
Type anything related to cyber security.
You: virus
Cyber-AI:
THREAT LEVEL: HIGH
Virus Threat Detected!
Here is your detailed solution:
1 Disconnect from the internet.
2 Run a full antivirus scan.
3 Delete unknown apps & suspicious files.
4 Do not plug USB devices into infected PC.
5 Update system and antivirus.
6 Avoid downloading cracked software.
Time: 21:29:16
Send
Build with agent mode
Let's get started
Build Workspace
Show Config


Cyber Security AI Assistant — Advanced Version
Here is your detailed solution:
1□ Disconnect from the internet.
2□ Run a full antivirus scan.
3□ Delete unknown apps & suspicious files.
4□ Do not plug USB devices into infected PC.
5□ Update system and antivirus.
6□ Avoid downloading cracked software.
□ Time: 21:29:16
□ You: Hacked account
□ Cyber-AI:
THREAT LEVEL: □ HIGH
□ **Your account may be hacked!**
Follow these steps:
1□ Immediately change your password.
2□ Enable 2-Step Verification.
3□ Log out from all devices.
4□ Check for unauthorized activity.
5□ Contact support team if needed.
□ Time: 21:29:29
□ You: Online Scams
□ Cyber-AI:
THREAT LEVEL: □ HIGH
□ **Scam Warning!**
Send
Build with agent mode
Let's get started
Build Workspace
Show Config


Cyber Security AI Assistant — Advanced Version
6□ Avoid downloading cracked software.
□ Time: 21:29:16
□ You: Hacked account
□ Cyber-AI:
THREAT LEVEL: □ HIGH
□ **Your account may be hacked!**
Follow these steps:
1□ Immediately change your password.
2□ Enable 2-Step Verification.
3□ Log out from all devices.
4□ Check for unauthorized activity.
5□ Contact support team if needed.
□ Time: 21:29:29
□ You: Online Scams
□ Cyber-AI:
THREAT LEVEL: □ HIGH
□ **Scam Warning!**
1□ Don't trust unknown callers.
2□ Don't share bank details.
3□ Don't respond to prize/lottery messages.
4□ Block and report scam numbers.
□ Time: 21:29:39
Send
Build with agent mode
Let's get started
Build Workspace
Show Config

Conclusion

The Cyber AI Assistant developed using Java Swing successfully demonstrates how rule-based artificial intelligence can be applied to detect and guide users through common cyber security threats. By integrating keyword-based analysis, threat-level identification, and step-by-step solutions, the system acts as a simple but effective digital safety companion.

The project showcases the use of graphical user interfaces, event handling, collections, and real-time text processing in Java. It helps users recognize dangers such as viruses, phishing, scams, password issues, fraud, hacked accounts, and data leaks, while providing immediate actionable recommendations to stay protected online.

Overall, this mini-project highlights the importance of cyber awareness and proves that even a lightweight AI model can make digital interactions safer, more informative, and user-friendly. Future enhancements such as machine-learning-based responses, chat logging, dark themes, and multi-language support can further enhance the system's usability and intelligence.

RESULT

The Cyber AI Assistant application was successfully developed and executed using Java Swing. The system accurately detected user-entered cyber-related keywords such as *virus, hacking, phishing, scam, password issues,* and *data leak*, and provided appropriate multi-step solutions along with threat-level indicators.

The graphical interface worked effectively, allowing smooth interaction between the user and the AI assistant. For every input, the program displayed a clear response with solutions, timestamps, and formatted guidance.
Thus, the intended objectives of creating a user-friendly, intelligent cyber-awareness tool were achieved successfully.