============================================================

CyberAiOpenChat - README

============================================================

PROJECT NAME: CyberAiOpenChat

AUTHOR: [Your Name Here]

VERSION: 1.0

DESCRIPTION:

------------

CyberAiOpenChat is a simple AI-driven console chatbot designed

to educate users about cybersecurity concepts. It includes a

friendly interface with audio and ASCII image display, and

provides helpful answers to basic cybersecurity questions.

FEATURES:

---------

- Interactive chatbot experience.

- Welcome audio playback.

- ASCII art logo from an image.

- Custom greeting with validated user name.

- Responds to cybersecurity-related questions.

- Filters out common filler words to identify keywords.

- Exit or return to main menu at any time.

FILES INCLUDED:

---------------

1. Program.cs --> Entry point for the chatbot.

2. ChatBotMenu.cs --> Manages the chatbot menu and flow.

3. AudioAndImage.cs --> Handles image-to-ASCII display and audio playback.

4. QuestionAndIgnore.cs --> Stores replies and processes user questions.

5. Ai.jpg --> Logo image displayed as ASCII art.

6. greeting.wav --> Welcome audio played at startup.

REQUIREMENTS:

-------------

- .NET Framework (or .NET Core)

- Console application environment

- Ensure `Ai.jpg` and `greeting.wav` are located in the project root folder

(NOT inside bin\Debug, but the actual root directory).

HOW TO RUN:

-----------

1. Open the project in Visual Studio or your preferred C# IDE.

2. Build the solution to restore dependencies.

3. Run the application.

4. Follow on-screen instructions:

- Enter your name (letters only).

- Choose to ask questions or exit.

- Ask cybersecurity questions like:

- "What is a firewall?"

- "Tell me about phishing."

- Type 'exit' anytime to return to the menu or quit the program.

EXAMPLES OF SUPPORTED KEYWORDS:

-------------------------------

password, phishing, malware, firewall, vpn, encryption,

2fa, ransomware, antivirus, social engineering,

cybersecurity, hacking, backups, spyware, trojan,

patching, network security, identity theft, botnet,

zero-day

TROUBLESHOOTING:

----------------

- If you don't hear audio: Make sure `greeting.wav` is in the correct path.

- If image doesn't display: Ensure `Ai.jpg` is present and a valid image.

- If unexpected errors appear: Check the console for exception details.

LICENSE:

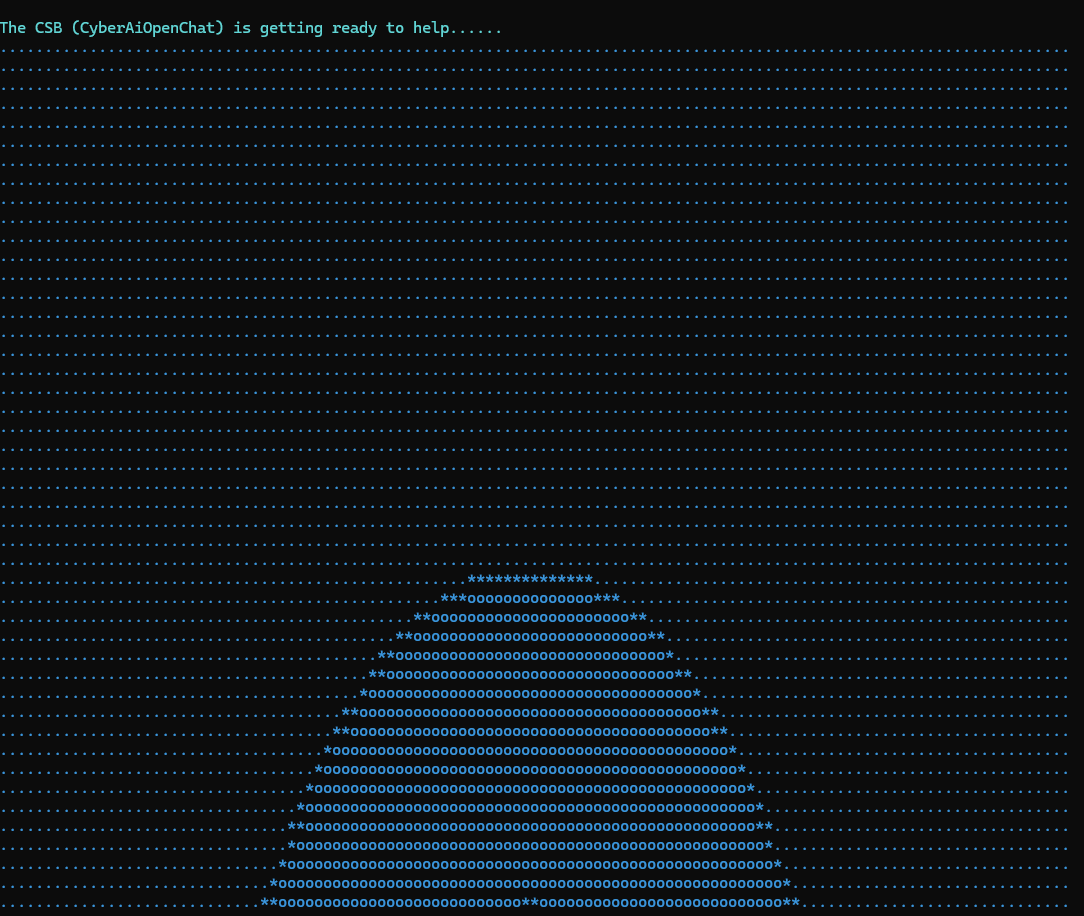
--------

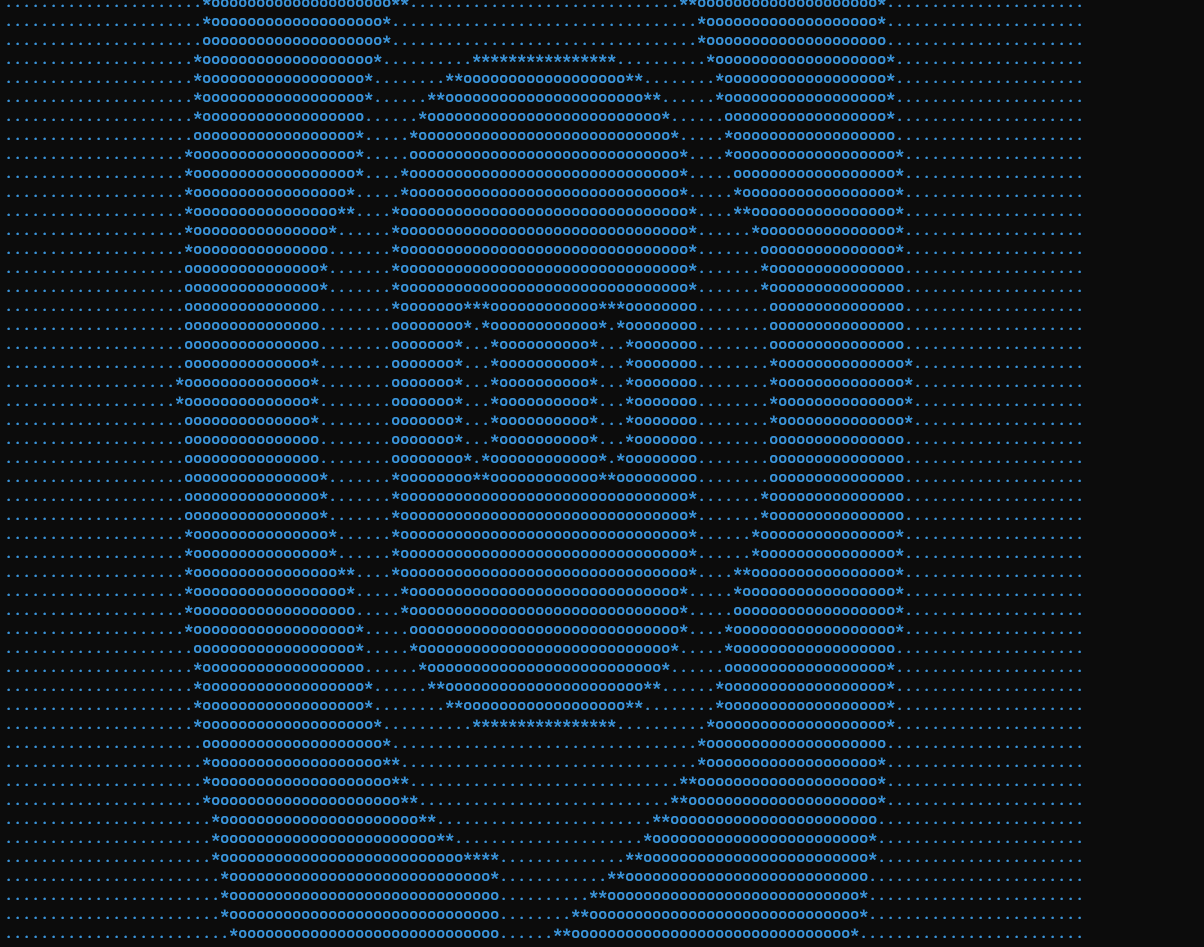
This project is for educational purposes. Feel free to modify and enhance it!

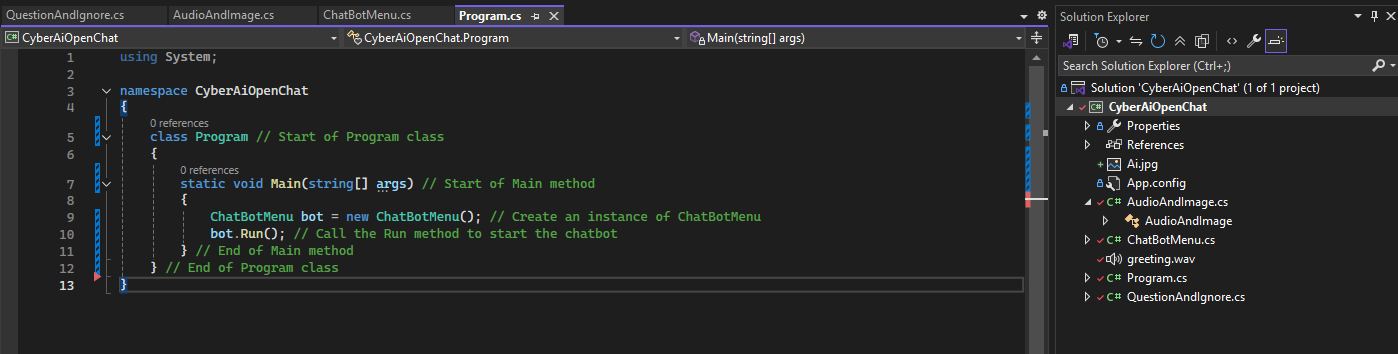
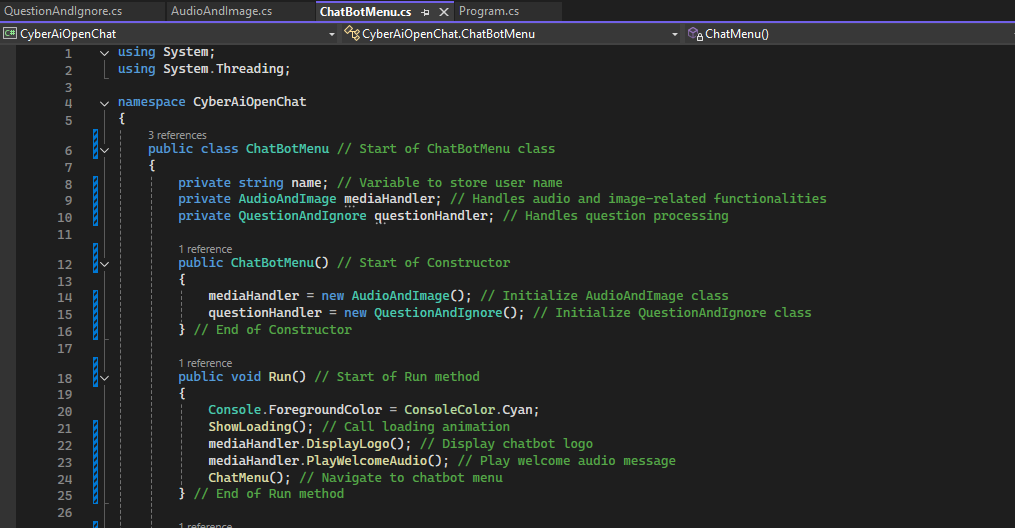
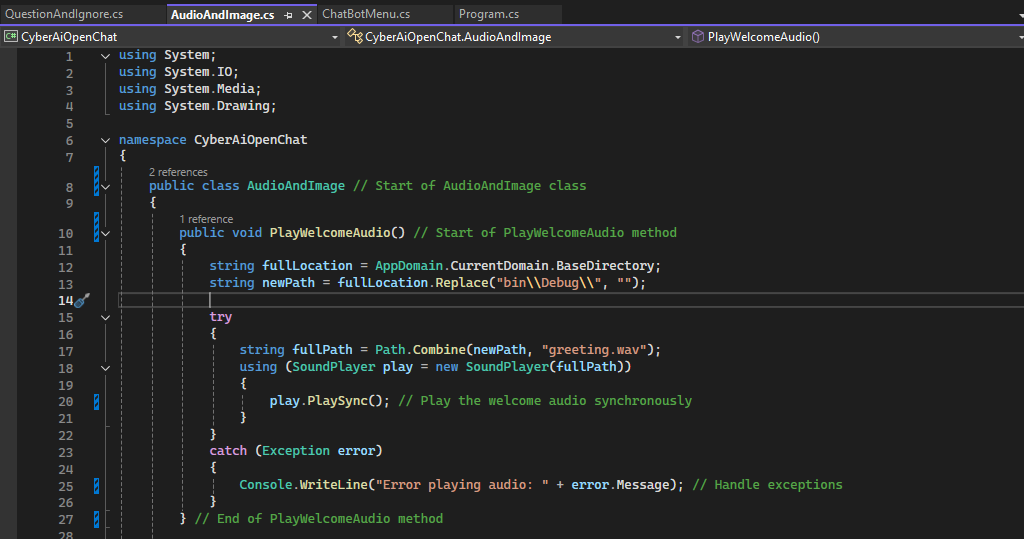
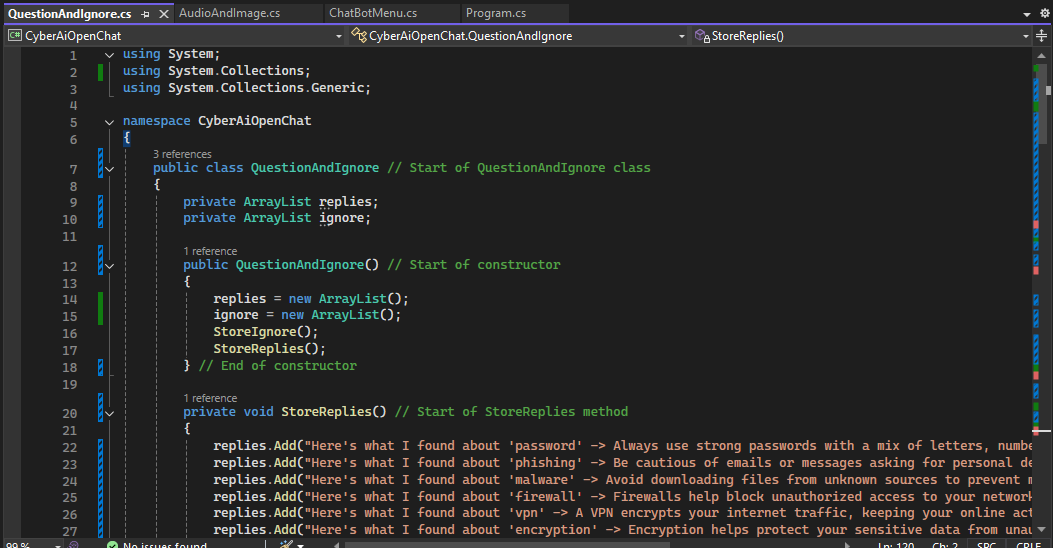
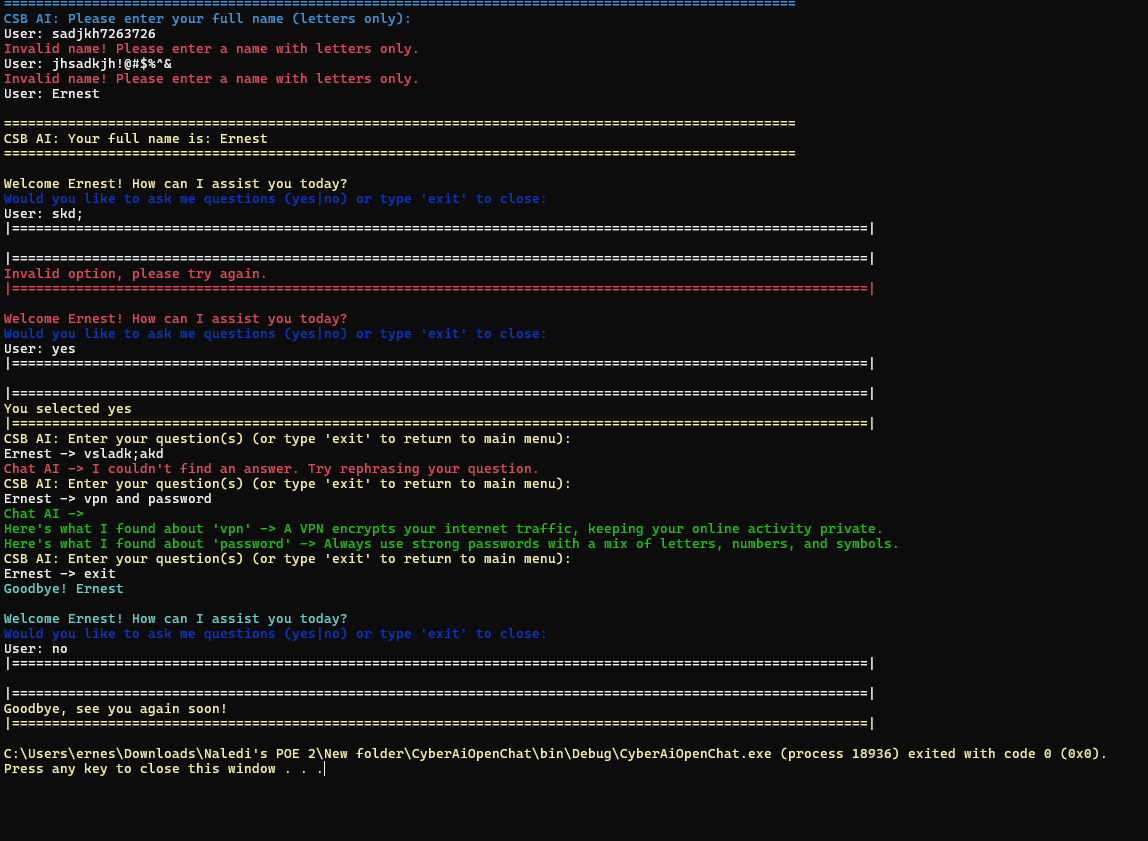
CONTACT:

--------

For questions, reach out to [your\_email@example.com].

============================================================





Part 2

================================================================================

CYBERSECURITY AWARENESS CHATBOT

================================================================================

Project Title: CSB - CyberSecurity Bot (POE Assignment)

Language: C#

Author: [Your Name]

Date: [Date]

--------------------------------------------------------------------------------

DESCRIPTION:

--------------------------------------------------------------------------------

The CyberSecurity Bot (CSB) is a console-based AI chatbot designed to promote

cybersecurity awareness. It interacts with users through typed questions and

provides helpful cybersecurity advice. The chatbot includes features such as:

- Audio greeting

- ASCII logo display

- Keyword-based response system

- Sentiment detection

- Chat history logging and retrieval

- Name validation

- Randomized dynamic responses

- Input filtering and error handling

--------------------------------------------------------------------------------

PROJECT STRUCTURE:

--------------------------------------------------------------------------------

1. Program.cs

- Entry point of the application.

- Instantiates the CyberBot class and runs it.

2. CyberBot.cs

- Main chatbot controller.

- Greets the user, validates the name, and shows a menu.

- Connects all the core components (media handler, memory manager, question handler).

3. AudioImageHandler.cs

- Plays an audio greeting (Greeting\_audio.wav).

- Displays an ASCII-style logo generated from an image (Image to ASCII.jpg).

4. MemoryManager.cs

- Handles saving and retrieving chat history.

- Ensures `chat\_history.txt` file exists and is updated.

5. QuestionHandler.cs

- Manages user input, matches cybersecurity-related keywords.

- Ignores common stopwords (like "what", "is", etc.).

- Detects user sentiment (e.g., nervous, scared) and responds accordingly.

- Provides predefined answers to common cybersecurity topics like phishing, passwords, malware, etc.

--------------------------------------------------------------------------------

HOW TO USE:

--------------------------------------------------------------------------------

1. Run the program.

2. Enter your full name (letters only).

3. Choose from the following options:

- 'y' to ask a cybersecurity-related question.

- 'n' to exit the program.

- 'p' to view the chat history stored in 'chat\_history.txt'.

4. While asking questions:

- Type questions like "What is phishing?" or "How can I protect my password?".

- Type "exit" to return to the main menu.

- Sentiment words like "worried" or "anxious" will trigger supportive responses.

--------------------------------------------------------------------------------

FILES NEEDED:

--------------------------------------------------------------------------------

- Greeting\_audio.wav : Audio file played at startup

- Image to ASCII.jpg : Image used to display an ASCII logo

- chat\_history.txt : Automatically created file that stores past chat logs

--------------------------------------------------------------------------------

FEATURE HIGHLIGHTS:

--------------------------------------------------------------------------------

✔ Keyword Recognition

✔ Randomized and grouped responses

✔ Sentiment Detection

✔ Chat History and Memory Recall

✔ Audio and Image Media Support

✔ Robust Input Validation and Error Handling

✔ Use of Delegates and Collections (List, Dictionary, HashSet)

--------------------------------------------------------------------------------

REQUIREMENTS:

--------------------------------------------------------------------------------

- .NET Framework or .NET Core (for C# Console Apps)

- Image and Audio files placed in the project root directory (not in /bin/Debug)

--------------------------------------------------------------------------------

NOTES:

--------------------------------------------------------------------------------

- Only cybersecurity-related topics are supported.

- The bot ignores generic or unsupported questions.

- The chatbot is designed to educate and support, not provide technical troubleshooting.

