Enhanced Cybersecurity Chatbot & Task Manager

- **Subtitle:** A WPF Desktop Application for Interactive Cybersecurity Assistance
  - **ST10437798 Lisakhanya Notununu**
    - **27 June 2025**

Bridging Cybersecurity Knowledge with Personal Productivity

- **Key Points:**
  - **Interactive Chatbot:** Provides immediate answers and guidance on cybersecurity topics.
  - **Intelligent Task Management:** Helps users organize security-related tasks with reminders.
  - **Cybersecurity Quiz:** Engages users to test and reinforce their cybersecurity knowledge.
  - **Activity Logging:** Maintains a transparent record of user interactions and system actions.
  - **Personal Touch:** Customizable startup ASCII art display.
- **Visual Idea:** A screenshot of your running application, highlighting the chat, task, and input areas.

Powering the Application: Key Technologies

- **Key Technologies:**
  - **C#:** The primary programming language for all application logic.
  - **WPF (Windows Presentation Foundation):** Microsoft's UI framework for building rich, interactive desktop applications. This defines how the windows, buttons, textboxes, etc., look and behave.
  - **LINQ (Language Integrated Query):** Used extensively for querying and manipulating data collections (e.g., filtering tasks, ordering quiz questions).
  - **Regular Expressions (`System.Text.RegularExpressions`):** Essential for "Natural Language Processing (NLP) simulation," allowing the chatbot to understand and act on chat-based task commands (e.g., "add task to...").
  - **`System.IO`:** Handles file system operations, specifically used for reading the ASCII art file.
  - **`System.Windows.Threading.DispatcherTimer`:** Manages time-based events, crucial for the automated task reminder system, ensuring it runs smoothly without freezing the user interface.
- **Visual Idea:** Icons for C#, WPF, LINQ, Regex, Folder, Clock/Timer.

- **Concept:** The application follows an object-oriented approach, separating concerns into distinct classes, each responsible for a specific part of the functionality.
- **Diagram Idea:** A simple block diagram showing:
- `MainWindow` (top layer, interacts with all others)
- Arrows pointing from `MainWindow` to:
  - `EnhancedResponseSystem`
  - `CybersecurityTaskManager`
  - `CybersecurityQuizGame`
- `ActivityLogEntry` (used by `MainWindow` and `CybersecurityTaskManager`)
- `TaskItem`, `QuizQuestion` (data structures, used by managers/games)

The Central User Interface and Coordinator

- **Purpose:** This `partial class` manages the WPF window, displays all UI elements defined in `MainWindow.xaml`, and acts as the central point for user interaction and coordination between different backend systems.

- **Key Responsibilities:**
  - **Initialization:** Sets up `EnhancedResponseSystem`, `CybersecurityTaskManager`, and `CybersecurityQuizGame` instances.
  - **Display Management:** Methods like `DisplayChatbotMessage()` and `DisplayUserMessage()` update the chat area.
  - **Input Processing:** `ProcessUserInput()` is the core method that takes user input and intelligently routes it to the appropriate logic (chatbot, quiz, or task manager).
  - **UI Updates:** `RefreshTaskListDisplay()` dynamically updates the list of tasks shown to the user.
  - **Event Handling:** Responds to button clicks (`SendButton_Click`, `AddTaskButton_Click`, etc.) and textbox events.
  - **Reminder Timer:** Initializes and manages the `DispatcherTimer` for task reminders.
  - **Activity Logging:** Calls the `LogActivity()` method to record interactions.
  - **ASCII Art Display:** Calls `DisplayAsciiArt()` to show a custom image on startup.

- **Key Classes (Brief Role):**
- `MainWindow`: The main user interface, orchestrates interactions between UI elements and backend logic.
- `EnhancedResponseSystem`: Handles chat responses, including keyword matching, sentiment analysis (simulated), and delegation of task commands.
- `CybersecurityTaskManager`: Manages the lifecycle of user-defined tasks.
- `CybersecurityQuizGame`: Manages the quiz logic, questions, and scoring.
- `ActivityLogEntry`, `TaskItem`, `QuizQuestion`: Data structures to represent log entries, tasks, and quiz questions respectively.