Take Home Challenge

Take-Home Assessment: PHI Redaction Application

Objective

Create a simple C# application that processes simulated lab order documents, identifies and redacts Protected Health Information (PHI), and outputs the sanitized data. The application should include minimal front-end code (e.g., a button to initiate processing) but focus primarily on backend logic.

Assignment Overview

You are provided with a set of sample lab order text files that contain PHI. Your task is to build a C# application that:

- **Reads** the input lab order files.
- **Identifies** PHI within the documents.
- **Redacts** or masks the PHI.
- **Exports** the output file to the user's file system.

Requirements

1. **Application Functionality**
   - **Input**:
     - The application should allow the user to select one or more lab order text files for processing. This can be through a file selection dialog opened by a button click.
   - **Processing**:
     - Identify and redact PHI elements in the lab orders. PHI elements to consider include:

      - Patient Name
      - Date of Birth
      - Social Security Number
      - Addresses
      - Phone Numbers
      - Email Addresses
      - Medical Record Numbers
      - Any other identifiers you deem necessary
    - Replace the PHI with placeholder text (e.g., `[REDACTED]`).
  - **Output**:
    - Save the sanitized lab orders to an output directory specified in the application.
    - Maintain the original file names with a suffix indicating they have been processed (e.g., `_sanitized.txt`).

2. **Front-End Interface**
   - A simple graphical user interface (GUI) with:
     - A button to select input files.
     - A button to initiate the processing of selected files.
     - Optional: Display messages or logs indicating the status of the processing.

3. **Backend Logic**
   - Implement robust parsing of the lab order documents.
   - Use appropriate methods (e.g., regular expressions) to identify PHI.
   - Ensure the application handles errors gracefully (e.g., invalid files, read/write permissions).

4. **Code Quality**
   - Write clean, maintainable, and well-documented code.
   - Include comments explaining your logic and any important decisions.
   - Follow best practices for C# development and application architecture.

5. **Documentation**
   - Provide a `README.md` file that includes:

  - Instructions on how to build and run your application.
  - An explanation of your approach to identifying and redacting PHI.
  - Any assumptions or decisions made during development.

Constraints

- **Time Limit**: Please spend **no more than 4 hours** on this assignment. We value your time and are interested in your approach rather than a perfect or complete solution.
- **Technology Stack**:
  - Use C# with .NET Core or .NET 5/6.
  - Use any javascript front end framework of your choice, React or Angular are preferred.
- **External Libraries**:
  - You may use standard libraries and NuGet packages.
  - Avoid using heavy dependencies that are unnecessary for the task.

Sample Data

You can create your own sample lab order text files for testing purposes. These files should simulate real lab orders and include PHI as specified in the requirements.

Example Content of a Lab Order File:

```
Patient Name: John Doe
Date of Birth: 01/23/1980
Social Security Number: 123-45-6789
Address: 123 Main St, Anytown, USA
Phone Number: (555) 123-4567
Email: john.doe@example.com
Medical Record Number: MRN-0012345

Order Details:
- Complete Blood Count (CBC)
- Comprehensive Metabolic Panel (CMP)
```

Submission Instructions

1. **Code Repository**
   - Create a public Git repository (e.g., on GitHub or GitLab) for your project.
   - Commit all source code and project files necessary to build and run your application.
   - Do not include any compiled binaries or unnecessary files.
2. **README File**
   - Include clear instructions on how to build and run your application.
   - Explain your approach to identifying and redacting PHI.
   - Note any assumptions, limitations, or areas for future improvement.
3. **Email Notification**
   - Once you have completed the assignment, email us the link to your repository.
   - Ensure the repository is accessible for cloning and review.

Evaluation Criteria

We will evaluate your submission based on the following factors:

1. **Functionality**
   - Does the application meet the specified requirements?
   - Does it correctly identify and redact PHI in the sample documents?
2. **Code Quality**
   - Is the code clean, readable, and well-organized?
   - Are best practices in C# development followed?
   - Are there appropriate comments and documentation?
3. **Approach and Problem-Solving**
   - Is your method for identifying PHI effective and efficient?
   - Do you demonstrate an understanding of handling sensitive data?
4. **User Experience**

- Is the GUI intuitive and user-friendly, even though minimal?
- Does the application handle errors gracefully?

5. **Documentation**
   - Is the `README.md` comprehensive and helpful?
   - Are build and run instructions clear and accurate?
6. **Time Management**
   - We understand time is limited. Partial solutions that demonstrate thoughtful approaches are acceptable.

Additional Notes

- **Assumptions**: You may make reasonable assumptions about the format and content of the lab order files. Please document these in your `README.md`.
- **PHI Identification**: Focus on the identifiers listed but feel free to include additional ones if appropriate.
- **Testing**: While thorough testing is valuable, we understand the time constraint. Include basic tests or describe how you would test the application given more time.

Questions and Support

If you have any questions or need clarifications about the assignment, please feel free to reach out via email. We're here to help.
