**Synopsis**

**Project Title: Guardian Secure - Comprehensive Web Security Application**

***Group Members:***

* Vicky Bhattacharya
* Rupam Chakraborty
* Priyanshu Roy
* Junaid Ali
* Debarati Poddar

***Abstract:***

Guardian Secure is an advanced web security application designed to provide users with robust protection against cyber threats. Leveraging a three-tier architecture, the application employs Flask for backend processing, SQLAlchemy for database management, and Chatterbot for AI-powered chatbot assistance. Guardian Secure offers a user-friendly interface with features such as user authentication, password management, vulnerability scanning, and AI-powered assistance. With a focus on proactive security measures, the application enables users to strengthen their online presence while ensuring data privacy and confidentiality.

***Objectives:***

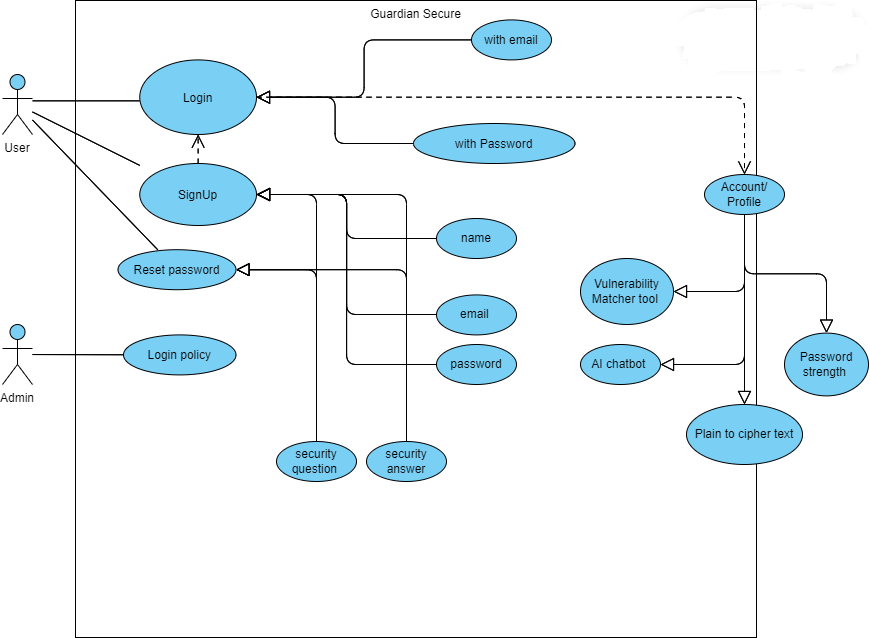
1. User Authentication and Registration:
   * Implement signup and login functionalities to secure access to the application.
   * Validate user credentials using Flask and SQLAlchemy for backend processing.
2. Password Management:
   * Develop a password management system with features for signup, login, and password strength testing.
   * Utilize hashing algorithms for secure password storage and verification.
3. Vulnerability Detection:
   * Create a vulnerability matcher tool to identify potential security risks in web applications.
   * Conduct spidering processes to analyze web app URLs and generate vulnerability reports.
4. AI-Powered Assistance:
   * Integrate an AI chatbot to provide personalized assistance and guidance on security-related queries.
   * Utilize Chatterbot library for natural language understanding and response generation.
5. Text Encryption and Decryption:
   * Enable text encryption and decryption functionality to safeguard sensitive user data.
   * Implement encryption algorithms for secure data transmission and storage.

***Outcomes:***

* Seamless user authentication process with secure signup and login functionalities.
* Enhanced password security through rigorous strength testing and hashing techniques.
* Proactive identification and mitigation of vulnerabilities in web applications.
* Personalized assistance and guidance on cybersecurity best practices through AI-powered chat support.
* Secure encryption and decryption of sensitive text to protect user privacy and confidentiality.

***Methodology: The development methodology involves:***

* Frontend development using HTML, CSS, and JavaScript for an intuitive user interface.
* Backend development with Flask and SQLAlchemy to handle user authentication, password management, and data storage.
* Integration of Chatterbot library for implementing the AI chatbot functionality to provide personalized assistance.
* Implementation of vulnerability scanning mechanisms using spidering processes and active/passive scan options.
* Continuous testing, debugging, and optimization to ensure the reliability, security, and effectiveness of Guardian Secure in real-world scenarios.

 ***Use-Case Diagram:***