**Project Title: Phishing Attack Simulation Framework**

**Project Overview:**

This project aims to simulate phishing attacks in a controlled environment to raise awareness and provide educational insights into how phishing works. The simulation will mimic real-world phishing attacks, allowing users (or employees in a corporate setting) to experience the tactics used by attackers. The framework will feature various types of phishing attempts (e.g., email phishing, spear-phishing, clone phishing) and will include features to track responses and provide training on identifying and preventing phishing attempts.

**Key Features:**

1. **Email Phishing Simulation**:
   * The system will send simulated phishing emails to users with links or attachments that resemble real-world phishing emails.
   * Emails will contain common techniques, such as malicious links, fake login pages, and urgent call-to-action messages.
2. **Data Collection and Reporting**:
   * Track who clicked on phishing links or opened phishing attachments.
   * Collect data on the success rate of the simulated attacks and report on the types of phishing emails that were most effective.
   * Create a comprehensive report that includes graphs, percentages of participants who fell for the phishing attempt, and recommendations for improvements.
3. **User Interaction Tracking**:
   * The system will detect if users provide credentials on fake phishing pages, click on malicious links, or download harmful attachments.
   * Provide real-time alerts or notifications when users interact with these phishing elements.
4. **Phishing Awareness Training**:
   * After each simulation, users will be given a short training session or detailed explanation of how the phishing attack was conducted.
   * Include tips on how to identify phishing emails (e.g., examining URLs, checking email sender information, spotting suspicious language).
5. **Spear-Phishing Simulation**:
   * More targeted phishing emails, based on publicly available information about users, to simulate advanced phishing techniques.
   * Allows the framework to customize messages to make them seem highly relevant to the user (e.g., referencing recent activities, known associates, or topics of interest).
6. **Feedback System**:
   * Allow users to report suspected phishing emails as part of the simulation.
   * Test whether users can recognize phishing attempts and provide feedback on their response time.
7. **Clone Phishing**:
   * Simulate an attack where legitimate email content is copied and a malicious version of the email is sent to users, aiming to deceive those who are expecting the email.
8. **Visualization and Reporting Dashboard**:
   * Real-time dashboard that visualizes the number of users who fall for phishing attempts, click rates, login attempts, and reporting rates.
   * Use visualizations (e.g., bar charts, line graphs) to display attack results, phishing attempt success rates, and participant performance over time.
9. **Integration with Email Systems**:
   * Simulate phishing emails through commonly used email systems (e.g., Gmail, Outlook).
   * Option to generate phishing emails with various difficulty levels based on user awareness levels.

**Technical Stack:**

1. **Backend**:
   * **Python** for handling the simulation logic, data tracking, and report generation.
   * **Flask** or **Django** for managing web server operations and API endpoints.
   * **SMTP/IMAP libraries** for sending and receiving phishing emails, such as smtplib and imaplib in Python.
2. **Frontend**:
   * **HTML/CSS/JavaScript** for phishing email design and visualization of login pages.
   * **Bootstrap** for styling and responsiveness of fake phishing pages and the reporting dashboard.
3. **Visualization**:
   * **Plotly** or **Matplotlib** for generating real-time visualizations of simulation results.
4. **Phishing Email Templates**:
   * A collection of phishing email templates that can be easily customized for different scenarios.
   * Include a library of real-world phishing examples to demonstrate varying levels of sophistication.
5. **Deployment**:
   * **Docker** for easy deployment of the framework in different environments.
   * **AWS SES** (Simple Email Service) or **SendGrid** for sending mass phishing emails if needed in larger corporate environments.