**Capstone Project: Building a Cybersecurity Training Web Application**

**CSEC 601: Web Application Security**

**Objective**

This project aims to design, develop, and secure a web-based game application that teaches cybersecurity principles to individuals and corporate employees through interactive training scenarios. The project will follow a **DevSecOps** methodology, integrating security practices throughout the development lifecycle. The project will span three months, with students divided into specialized teams.

**Project Goals**

1. Develop a secure and engaging web application using modern development frameworks.
2. Integrate core cybersecurity training concepts into game design to simulate real-world threats.
3. Implement DevSecOps practices, including CI/CD pipelines, automated testing, and security scans.
4. Deliver a final application that is functional, secure, and deployable in corporate environments.

**Team Structure**

The class will be divided into **three specialized groups**, each responsible for a critical aspect of the project:

1. **Front-End Development Group**
   * **Role:** Design and implement the user interface and game mechanics.
   * **Responsibilities:**
     + Create an intuitive, responsive UI using frameworks like React or Vue.js.
     + Develop interactive game components aligned with training goals.
     + Ensure accessibility and usability for diverse users.
2. **Back-End Development Group**
   * **Role:** Develop the application logic, API, and database management.
   * **Responsibilities:**
     + Use frameworks like Node.js, Django, or Flask to handle back-end functionality.
     + Implement secure user authentication and data handling.
     + Set up a scalable database (e.g., PostgreSQL or MongoDB).
3. **Security and DevSecOps Group**
   * **Role:** Embed security into the development process and ensure the application meets security standards.
   * **Responsibilities:**
     + Implement CI/CD pipelines with tools like GitHub Actions or Jenkins.
     + Conduct security testing using tools like OWASP ZAP, Snyk, and Burp Suite.
     + Automate dependency checks and vulnerability scans.

**Project Timeline**

1. **Month 1 (January 20 – February 20):**
   * Research and planning: Identify tools, frameworks, and DevSecOps strategies.
   * Front-end and back-end groups start basic scaffolding of the application.
   * Security group sets up the CI/CD pipeline and conducts initial threat modeling.
2. **Month 2 (February 20 – March 20):**
   * Develop core game features and back-end functionality.
   * Integrate automated security testing into the pipeline.
3. **Month 3 (March 20 – April 20):**
   * Finalize development and address bugs from testing.
   * Conduct comprehensive penetration tests and usability reviews.
   * Prepare and deliver the final presentation.

**Expected Outcomes**

By the end of the project, the class will deliver:

1. A fully functional and secure web application that gamifies cybersecurity training.
2. A robust CI/CD pipeline integrated with security testing tools.
3. A live demonstration showcasing the application's capabilities and value.

**Significance**

This project provides hands-on experience in web application development, DevSecOps practices, and cybersecurity integration. It emphasizes teamwork, technical skills, and secure coding principles, preparing students for real-world challenges in secure software development.

**Leadership Structure**

1. **Overall Project Manager**
   * Oversees the project’s overall progress and ensures all groups work in sync.
   * **Responsibilities:**
     + Organize weekly meetings and consolidate progress reports.
     + Resolve conflicts and dependencies across groups.
     + Provide regular updates to the instructor.
2. **Technical Manager**
   * Sets up and maintains the technical infrastructure, including GitHub repositories and CI/CD pipelines.
   * **Responsibilities:**
     + Create the repository structure with branches for development, testing, and production.
     + Standardize the tools and frameworks used across groups.
     + Provide technical guidance and support.
3. **Group Leaders**
   * Each group will have a leader responsible for managing specific tasks and deliverables.
   * **Responsibilities:**
     + Coordinate group activities and ensure timely completion of milestones.
     + Communicate progress and challenges to the project manager.

**Group Roles**

Each group will include these core roles to ensure all tasks are covered effectively:

1. **Group Manager:** Oversees and coordinates the group’s activities and progress.
2. **Build Master:** Manages the CI/CD pipelines and ensures seamless integration.
3. **Risk Manager (Part-Time):** Identifies, assesses, and mitigates risks.
4. **Development Lead:** Plans and oversees coding tasks and ensures modular, secure development.
5. **Test Lead:** Manages testing activities and ensures comprehensive QA processes.

**Workflow and Collaboration**

1. **Weekly Check-Ins:**
   * The project manager meets with group leaders to review progress.
   * Group leaders hold meetings with their teams to align on tasks and blockers.
2. **Integration Points:**
   * Security and DevSecOps group collaborates with front-end and back-end teams to integrate security practices.
3. **Issue Tracking:**
   * Use tools like Trello or Jira to track tasks, issues, and progress.
   * Assign tasks with clear deadlines and priority levels.

**Team Structure and Size**

The class will be divided into **three specialized groups**, with roles and responsibilities distributed to ensure a balanced workload across the 18 students.

1. **Front-End Development Group (6 Students)**
   * **Role:** Design and implement the user interface and game mechanics.
   * **Responsibilities:**
     + Create an intuitive, responsive UI using frameworks like React or Vue.js.
     + Develop interactive game components aligned with training goals.
     + Ensure accessibility and usability for diverse users.
   * **Roles Within Group:**
     + 1 Group Manager
     + 1 Build Master
     + 1 Risk Manager (Part-Time)
     + 1 Development Lead
     + 2 Developers
2. **Back-End Development Group (7 Students)**
   * **Role:** Develop the application logic, API, and database management.
   * **Responsibilities:**
     + Use frameworks like Node.js, Django, or Flask to handle back-end functionality.
     + Implement secure user authentication and data handling.
     + Set up a scalable database (e.g., PostgreSQL or MongoDB).
   * **Roles Within Group:**
     + 1 Group Manager
     + 1 Build Master
     + 1 Risk Manager (Part-Time)
     + 1 Development Lead
     + 3 Developers
3. **Security and DevSecOps Group (3 Students)**
   * **Role:** Embed security into the development process and ensure the application meets security standards.
   * **Responsibilities:**
     + Implement CI/CD pipelines with tools like GitHub Actions or Jenkins.
     + Conduct security testing using tools like OWASP ZAP, Snyk, and Burp Suite.
     + Automate dependency checks and vulnerability scans.
   * **Roles Within Group:**
     + 1 Group Manager
     + 1 Build Master
     + 1 Risk Manager (Part-Time)
     + 1 Development Lead
     + 1 Security Specialist