Cool Cyber Games

Gamified Cyber Security Learning Platform

Team members

Anthony Clayton - aclayton2023@my.fit.edu

Matthew Goembel - mgoembel1@gmail.com

Ben Allerton - ballerton2020@my.fit.edu

Ludendorf Brice - lbrice2018@my.fit.edu

Faculty advisor

Dr. Sneha Sudhakaran - ssudhakaran@fit.edu

Client

Dr. Sneha Sudhakaran - Florida Institute of Technology

Meeting(s) with the Client for developing this Plan

*Meetings take place weekly on Wednesdays with the entire team

1. 08/27/2025 - Discussed project plan, goals for this semester, and how we plan to finish.

Project Goals

Provide an interactive and effective platform to teach cybersecurity to adult users (18+). Build user awareness, practical skills, and resilience against cyber threats. Allow compatibility with all the most commonly used operating systems and languages. Offer hands-on experience with simulations of real-world scenarios to build real-world technical skills.

Project Motivation

Lack of Accessibility and Usability: Existing cybersecurity learning platforms often use overly technical language, intimidating interfaces, or require prior knowledge, discouraging especially inexperienced audiences from engaging in cybersecurity education.

Absence of Gamification and Fun: Cybersecurity is often presented in a dry, textbook-like manner. Without gamification and interactivity, users struggle to stay motivated, and learning

outcomes are limited.

Fear and Intimidation Around Cybersecurity Concepts: Many users find complex cybersecurity concepts abstract and intimidating. Instead of avoiding the topic entirely, a guided approach can help users feel confident and capable.

Lack of hands-on opportunities: Without practical exposure, users cannot effectively apply cybersecurity principles in real-life scenarios.

Approach (key features of the system):

Interactive tutorials and quizzes: Users can engage with interactive tutorials that guide them through essential cybersecurity concepts in a step-by-step manner. The users can then apply their knowledge through quizzes at the end of each tutorial, which test their understanding using multiple-choice, drag-and-drop, and scenario-based questions. This feature helps users actively participate in the learning process and strengthens their ability to retain the material.

Skill-specific modules with gamification: The users can access modules with games tailored to their desired skill to learn. Users can interact with gamified elements, such as simulated cybersecurity challenges. This gamification enhances the user experience and keeps users engaged while helping them master cybersecurity skills.

Real-world cybersecurity challenges: Users can learn by playing in real-world cybersecurity simulations, where they simulate defending against cyberattacks like phishing, malware, and social engineering. The users can practice applying cybersecurity strategies in scenarios that mirror real-life situations, helping them develop practical, hands-on skills. Additionally, users can potentially analyze case studies of cyber incidents, gaining insights into how cybersecurity threats evolve and how to protect against them.

Progress tracking and certifications: The users can track their learning progress by monitoring their completion of tutorials and quizzes. Users can earn badges, achievements, and certifications as they progress through different levels of content, completing specific milestones or advanced courses, providing them with a tangible way to measure their learning success and motivation to continue improving their cybersecurity knowledge.

Algorithms and tools (libraries/api/frameworks/languages) for the key features

Game Development: Unity, Godot

Front-End: HTML, CSS, JavaScript, React.js.

Back-End: Python, Node.js, Java, Typescript

Database & Storage: PostgreSQL, MySQL, MongoDB.

Hosting & Deployment: GitHub Pages, ExpressJS, Render.

Testing & Debugging: Selenium, JUnit, Mockito

Authentication: OAuth 2.0, JWT (JSON Web Tokens)

Collaboration & Organization: GitHub (version control), Jira (task management), Discord, Text Message, Email (communication), weekly meetings(in-person).

Novel features: Discuss which features/functionalities are novel and why.

Gamify Cyber Security Content: While users could learn about cyber security concepts through YouTube videos or articles, gamifying that experience could potentially be a better learning style as some users may find it less intimidating and more hands-on.

Progress Tracking and Certifications: Many platforms offer courses, but progress tracking and certifications that reflect user achievements in a gamified and interactive environment are less common. Tracking progress across different skill levels and offering certificates upon completion creates motivation and gives users something tangible to show for their efforts.

Adaptive Learning Paths: Based on a learner's progress and performance, the platform could automatically adjust difficulty levels or give personalized learning suggestions, ensuring the content is always appropriately challenging.

Technical Challenges

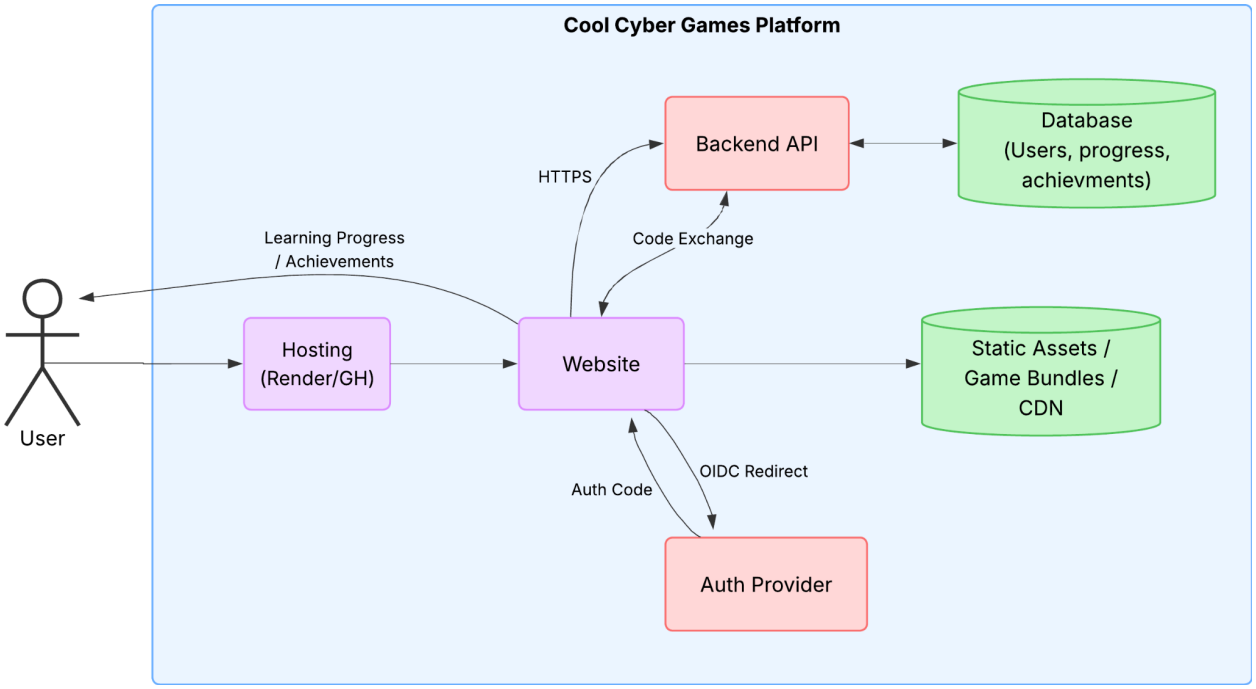
Defining the Structure of Real-World Cybersecurity Simulations: Designing realistic yet engaging scenarios for threats like phishing and malware is challenging, especially in balancing complexity for diverse audiences. We must identify the best tools and frameworks and the methods for maintaining simplicity and efficiency.

Implementing Secure Authentication and Authorization: Designing a secure and user-friendly authentication system that handles third-party login providers (e.g., Google).

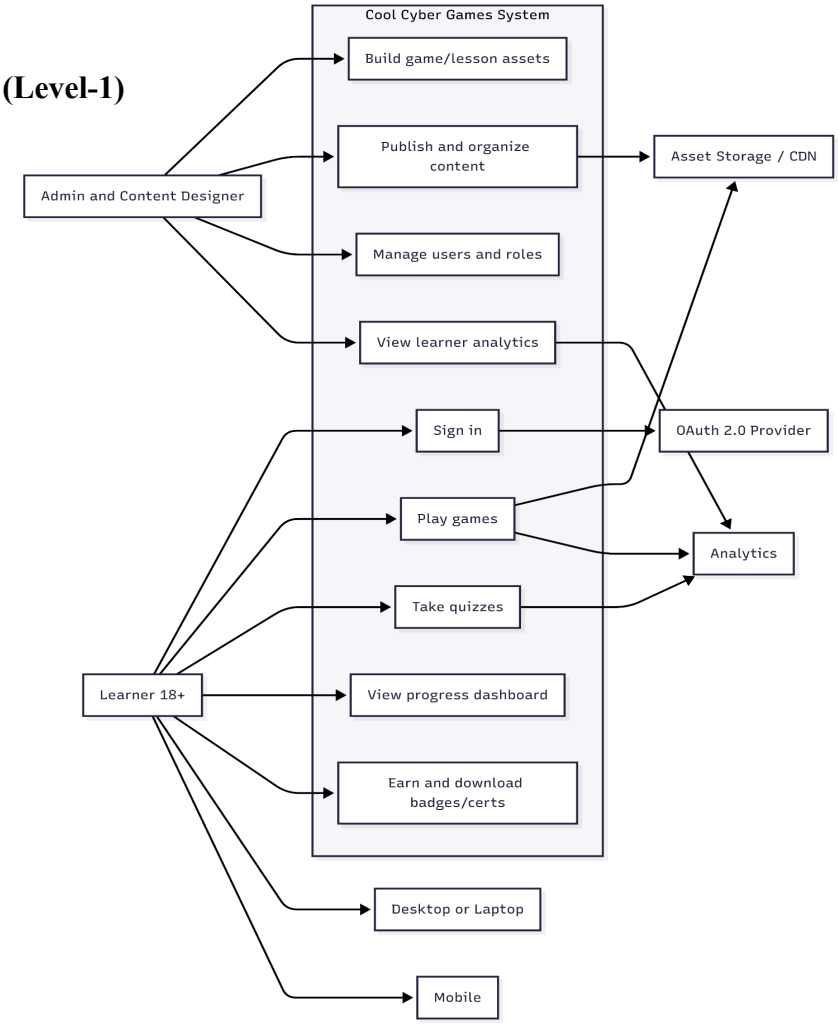
Designing Effective Cybersecurity Education Content: Developing interactive, engaging, and skill-specific lessons involves technical challenges such as integrating dynamic educational content into gamified systems and ensuring compatibility with various devices and operating systems. Additionally, implementing tools to track user comprehension and progression across diverse topics poses technical hurdles.

Design: system architecture diagram

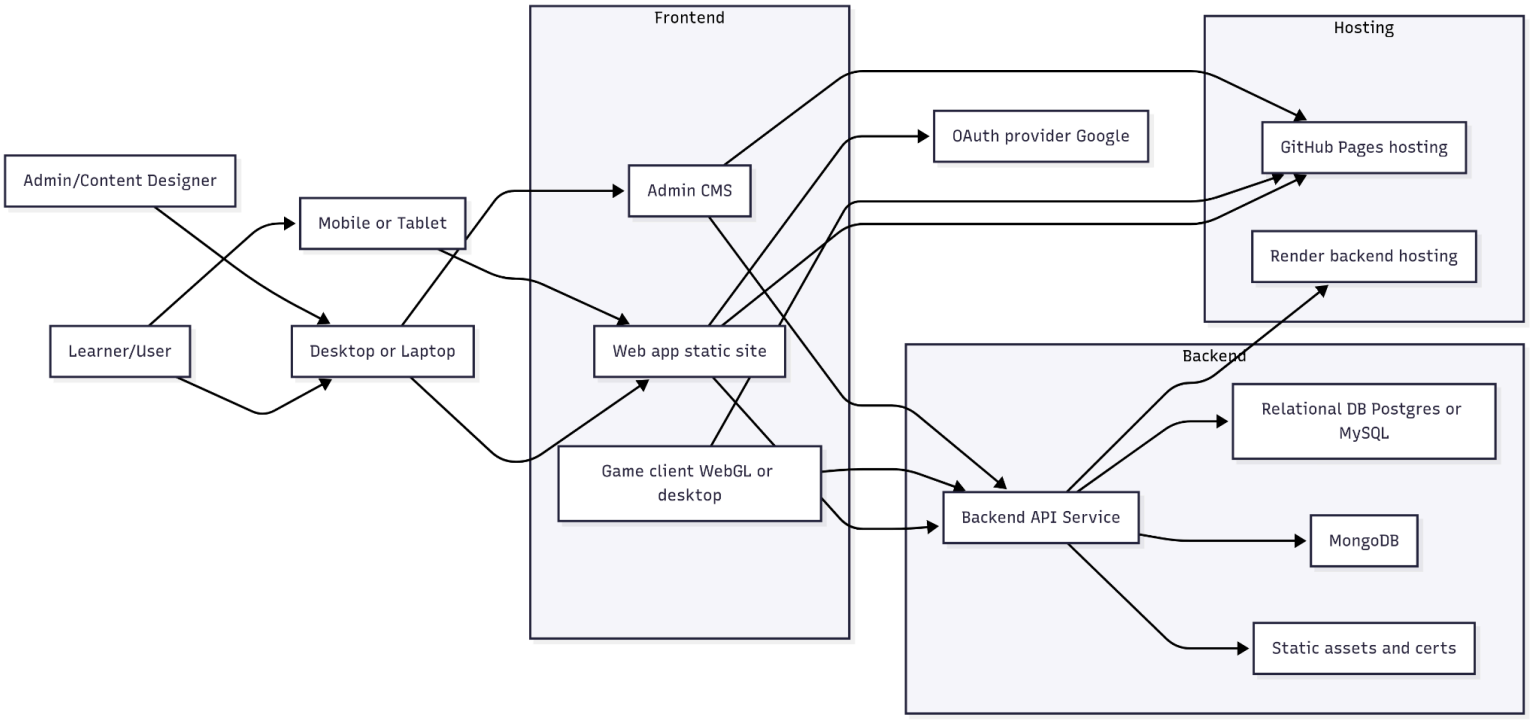
Components/modules



Context (Level-1)



Deployment + Hardware interface



Evaluation:

To evaluate the effectiveness of Cool Cyber Games using both technical metrics and learning outcomes, we could measure:

1. **System Functionality & Performance**
    a. Page load time for website and games ≤ 3 seconds.
    b. Backend API response times (p95 latency < 300ms).
    c. No critical errors or crashes during playtesting.
2. **Usability**
    a. Users can navigate the dashboard, play a game, and view achievements without external guidance.
    b. Target ≥ 80% of testers successfully complete a game module in their first session.

3. **Learning Outcomes**

    a. Pre-/post-quiz improvement: average score increases by ≥ 25% after completing a module.

    b. Completion rate of modules ≥ 70% of testers finish at least one full game.

4. **Engagement**

    a. Retention: at least 40% of testers return for a second session.

    b. Feedback survey: average user satisfaction ≥ 4 out of 5 (ease of use, fun, learning value).

5. **Security**

    a. Authentication works consistently with Google login (OAuth 2.0).

    b. All API calls between front- and back-end are protected by secure tokens.

Progress Summary:

Module/feature	Completion %	To do
Website & GUI	50%	-Allowing drop-down menus, mouse-over help messages -User Dashboard -Leaderboard -Account Management -Prettify
Games	50%	-Finish game 2 -Finish game 3 -Finish game 4
Backend	90%	-Test delivery speed, game speed, asset loading etc.
Auth/Security	90%	-Ensure API calls between front and backend are secure
Database	90%	-Store user-specific game data in the DB -Test flown between games, front-end, and DB

User Feedback	75%	-Collect more feedback -Add in-game feedback

Milestone 4 (Sep 29): itemized tasks:

1. Implement, test, and demo three mini-games (Master the Password, File Detective, Web Quest).
2. Connect Malware Maze to backend → frontend (progress tracking and achievements).
3. Build and demo User Dashboard + Leaderboard (fetches from DB).

Milestone 5 (Oct 27): itemized tasks:

4. Finish Game 2, Game 3, and Game 4 final versions with art/UI polish and backend integration.
5. Refine User Dashboard with visual improvements (progress bars, badges).
6. Conduct evaluation study with testers:
    - Collect learning outcome data (pre/post quiz scores, completion rates).
    - Collect usability feedback (navigation, clarity, enjoyment).
7. Begin poster design for Senior Design Showcase.
8. Midpoint client demo: show functional website with multiple games, working dashboard, and first evaluation results.

Milestone 6 (Nov 24): itemized tasks:

9. Implement and test full system integration (all games, dashboard, backend, DB, auth).
10. Conduct final end-to-end testing and demos
11. Run final evaluation: analyze technical metrics (load times, errors) + user learning outcomes.
12. Create User Manual (how to play, navigate, track progress) and Developer Manual (setup, architecture, future work).
13. Record and edit a demo video highlighting system features.
14. Finalize and print poster for Senior Design Showcase.
15. Final presentation to client and faculty advisor.

Task matrix for Milestone 4 (teams with more than one person)

Task	Anthony	Matthew	Ben	Ludendorf
Implement, test & demo *Game: Master the Password*	100%	0%	0%	0%

Implement, test & demo *Game: File Detective*	0%	0%	0%	100%
Implement, test & demo *Game: Web Quest*	0%	0%	100%	0%
Implement & test Backend→Frontend Game connection for Malware Maze	0%	100%	0%	0%
Implement, test & demo Frontend user dashboard and leaderboard	0%	100%	0%	0%

Task 1: Implement, test, and demo Game: Master the Password
Anthony will finalize the gameplay mechanics, polish the UI.. The demo will show users logging in, playing the game, and receiving achievements.

Task 2: Implement, test, and demo Game: File Detective
Ludendorf will build out the puzzle flow and challenge mechanics. This task includes debugging asset loading, connecting it to backend progress tracking, and preparing a working demo of at least one playable scenario.

Task 3: Implement, test, and demo Game: Web Quest
Ben will complete front-end integration and ensure quiz-style challenges work within the game. The deliverable is a functional Web Quest level that tracks completion and updates user progress.

Task 4: Implement & test Backend→Frontend Game Connection for Malware Maze
Matthew will connect the backend API to the front-end for Malware Maze. This includes storing progress data in the database and retrieving it for the user dashboard. The task will demonstrate a full end-to-end flow.

Task 5: Implement, test, and demo Frontend User Dashboard & Leaderboard
Matthew will design and integrate the user dashboard, showing progress, achievements, and leaderboard rankings. This will include API calls to fetch data from the backend and display it in a clean, user-friendly interface.