Cool Cyber Games

Interactive Web Application for Teaching Cybersecurity

coolcybergames.com

Team: Ben Allerton, Ludendorf Brice, Anthony Clayton, Matthew Goembel

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Goals and Motivation

Goals:

- Provide an interactive and effective platform to teach cyber security 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.
- Foster a smooth learning experience with user-centered content.

Goals and Motivation

Motivation:

- **Lack of Accessibility and Usability**: Existing cyber security learning have not so beginner-friendly, and have intimidating interfaces, discouraging new and inexperienced users from engaging.

- **Absence of Gamification and hands-on opportunities**: Cybersecurity is often presented in a dry, textbook-like manner. Gamification and interactive practical exposure help users to stay motivated.

- **Age-Appropriate Content**: Current platforms leaves adult users feeling the content is either oversimplified or overly technical.

- **Lack of Global Reach and Multilingual Support**: Most existing platforms focus on English-speaking audiences.

Approach

Interactive tutorials and quizzes

- Users will go through interactive tutorials that guide them through learning the essential cybersecurity concepts in a step-by-step manner.
- The users will apply knowledge through quizzes/labs at the end of each tutorial.

Modules and Levels with gamification.

- The users will go through gamified modules tailored to a specific topic.
- Users will interact with gamified elements, such as interactive characters, environments, challenges, etc.
- Users will receive real-time, detailed feedback and clear progress metrics.

Approach (cont.)

Real-world cybersecurity challenges.

- Users will learn by playing in real-world cybersecurity simulations, where for example they simulate defending against cyber attacks like phishing, malware, etc.

- The users can practice applying cybersecurity strategies in scenarios that mirror real-life situations, gaining insights into how cybersecurity threats evolve and how to protect against them.

Progress tracking, certifications, and leaderboards.

- The users will have their progress automatically tracked, like completion of tutorials, modules, quizzes, and be able to view it in a simple dashboard.

- Users can earn achievements, and certifications as they progress through different levels of content, and a real time leaderboard will be kept scoring to defined logic.

Approach (cont.)

Support for multiple languages.

- Users can access the platform in multiple languages
- The users can interact with all tutorials, quizzes, and challenges in linguistically relevant terms.

Novel Features

Multilingual Support/Cross-Platform Compatibility

- Current apps struggle with cross-platform compatibility or lack multi-language support.
- Our platform will be fully compatible with the most commonly used operating systems and supports most prevalent languages.

Gamification

- Gamified elements like fun changes, leaderboards, progress milestones and more, keeping user motivated and gaining building practical experience that is missing in other platforms.

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY

Novel Features (cont.)

Progress Tracking, Achievements, and Feedback

- Give user real-time feedback on their progress, affirming them, helping them, or suggesting net steps, and save their progress automatically

- Offering certificates/badges upon completion of challenges

Interactive Hands-On Learning

- Introduce dynamic, hands-on labs, interactive simulations, and scenario-based challenges.

- Navigation is straightforward.

Technical Challenges

Defining the Structure of Real-World Cybersecurity Simulations:

- Create engaging, realistic threat scenarios while balancing complexity for diverse users

Implementing Multilingual Support and Localization:

- Ensure accurate translations, choose the right tools, and update content without disrupting the user experience.

Technical Challenges(cont.)

Developing Frontend and Backend Architecture:

- Using React.js, Python, and JavaScript while designing efficient APIs for progress tracking and gamification.
- Seamlessly integrating various tools and algorithms.

Hosting and Storage:

- Choosing cost-effective hosting for performance and scalability
- Optimizing database storage for user progress, game results, and authentication.

Current Tools and Algorithms

Frontend Development

- HTML, CSS, Javascript, React.js
- Phaser.js, GDScript(godot)

Backend & API

- Java, Python (FastAPI)
- Node.js

Database & Storage

- PostgreSQL
- MongoDB

Security & Hosting

- OAuth, JWT, Bcrypt
- AWS/GCP

Algorithms

- in progress

Milestone One: (Feb 24) Initial Setup and Technical Evaluation

1. **Compare and Select Technical Tools**:
   - Evaluate front-end frameworks (React.js, Angular, etc.).
   - Compare back-end frameworks (Node.js, expressJS, etc.).
   - Decide on database options (PostgreSQL, MongoDB, etc.).
   - Authentication: Evaluate OAuth 2.0, JWT (JSON Web Tokens), bcrypt.
   - Assess gamification tools (GoDot, Phaser, GamifyJS).
2. **"Hello World" Demos**:
   - Frontend: Create a functional webpage.
   - Backend: Set up a simple server and database with a test API endpoint.
   - Gamification: Develop a basic working game in Phaser or GoDot.
   - Authentication: Demonstrate basic login functionality.
3. **Resolve Technical Challenges**:
   - Finalize frameworks for real-world simulations and gamification.
   - Choose localization tools to implement ADA-compliant, multilingual support.
   - Identify tools for seamless integration between the frontend, backend, and games.
   - Address integration of authentication tools with the front end and back end.
4. **Select Collaboration Tools**:
   - Compare and finalize tools like GitHub, Jira, and Discord for task management and communication.
5. **Draft Initial Documents**:
   - Requirement Document: Define core features, technical specifications, and target users.
   - Design Document: Outline system architecture and workflows.
   - Test Plan: Develop a preliminary testing strategy for unit, integration, and usability testing.

Milestone Two: (Mar 26) Feature Implementation and Testing

1. **Implement, Test, and Demo Core Features**:
   - Frontend: Develop interactive tutorials and quizzes for users(adults:18+).
   - Backend: Implement APIs for user authentication, progress tracking, and quiz scoring.
   - Gamification: Integrate age-appropriate games for users into the front end.
   - Authentication system: User login, registration, and secure password handling. Integrate with Google (e.g. login using a Google account).
2. **Address Localization and Accessibility**:
   - Add multi-language support for at least two languages.
   - Ensure the platform complies with ADA accessibility standards.
3. **Progress Tracking**:
   - Develop a progress-tracking system that connects the backend to the front end.
   - Test user accounts for data persistence and security.
4. **Validate System Performance**:
   - Perform load testing on APIs and game components.
   - Debug and resolve latency issues in real-time simulations.
5. **Documentation**:
   - Update Requirements and Design Documents with finalized features.
   - Refine the Test Plan based on new features and performance data.

Milestone Three: (Apr 21) Final Implementation and Launch Prep

1. **Implement, Test, and Demo Advanced Features**:
   - Add remaining user modules and levels with tailored tutorials and games.
   - Complete real-world cybersecurity simulations (phishing, malware defense, etc.).
   - Finalize certifications and badge system.
   - Refine authentication: Ensure robust security and user account management.
2. **Integrate Multilingual**:
   - Extend language support to more languages.
3. **Deployment Preparation**:
   - Configure hosting platform (AWS, Azure, or GitHub Pages, ExpressJS, Render).
   - Set up HTTPS for secure connections.
4. **Testing and QA**:
   - Conduct end-to-end testing for all modules.
   - Perform user testing to gather feedback on usability and accessibility.
5. **Finalize Documentation**:
   - Submit complete Requirements, Design, and Test Documents.
   - Prepare user manuals and deployment guides for the platform.
6. **Prepare for Client Presentation**:
   - Develop a walkthrough demo showcasing key features.
   - Collect usage data and metrics for the client report.

Task Matrix: Milestone 1

Task	Anthony	Matthew	Ludendorf	Ben
Compare and select Technical Tools	Backend, Database	Gamification	Authentication	Frontend
"hello world" demos	Set up a simple server and database with a test API endpoint	Develop a basic working game in Phaser or Unity	Demonstrate basic login functionality	Create a functional webpage
Resolve Technical Challenges	Developing Database and Backend Architecture	Defining the Structure of Real-World Cybersecurity Simulations	Implementing Secure Authentication and Authorization, Designing Effective Cybersecurity Education Content	Implementing Multilingual Support and Localization, Develop Frontend Architecture
Compare/select Collaboration Tools	Task calendar, programs	documentation, presentations, communication		
Requirement Document	Write 20%	Write 50%	Write 20%	Write 10%
Design Document	Write 40%	Write 20%	Write 20%	Write 20%
Test Plan	Write 30%	Write 20%	Write 20%	Write 30%

Questions?

FLORIDA INSTITUTE OF TECHNOLOGY

AD ASTRA PER SCIENTIAM

★ 1958 ★

FLORIDA TECH
FLORIDA'S STEM UNIVERSITY