Insecure by Design: A Hands-On Learning Platform for Web Security Training

Project Proposal

CSIS 4495-003 | Applied Research Project

Winter 2026

Instructor: Arasanipalai Kandhadai, Padmapriya

Allwyn Mascarenhas | 300381351 | Team Leader

Thilan Udahage Don | 300398798

Table of Contents

Introduction

Web application security has become a critical area of study as modern systems increasingly rely on complex, interconnected web technologies. Despite the availability of extensive documentation and theoretical resources, many learners struggle to understand how vulnerabilities emerge within real application workflows without the opportunity to experiment in a safe, controlled environment. Practical, hands-on exposure is essential for developing the ability to recognize, exploit, and mitigate security flaws, skills that are foundational in cybersecurity, software development, and digital forensics.

Existing vulnerable training platforms such as DVWA, WebGoat, and OWASP Juice Shop provide valuable learning opportunities; however, they often present limitations. Many rely on outdated architectures, lack structured guidance, or do not reflect the design patterns and frameworks commonly used in contemporary enterprise applications. As a result, learners may find it difficult to transfer their knowledge to real-world development environments. Current research emphasizes the importance of experiential learning in cybersecurity education, yet there remains a gap in accessible, framework-based platforms that integrate vulnerabilities directly into realistic application features.

Industry research further highlights the urgency of improving practical cybersecurity training. According to Fortinet's 2024 Cybersecurity Skills Gap Report, 58% of IT decision-makers identify a lack of hands-on cybersecurity skills and training among staff as the leading cause of security breaches. This reinforces the need for learning platforms that allow users to experiment with real vulnerabilities in a safe, controlled environment—an area where many existing tools fall short.

This project aims to address that gap by designing and implementing an intentionally vulnerable web-based learning platform built using Spring Boot MVC, a modern and widely adopted Java framework. The research explores the following questions:

- How can intentional vulnerabilities be embedded into a modern Spring Boot application in a way that is realistic, safe, and educational?
- Does guided exploration within a functional application improve learners' understanding of web vulnerabilities compared to static examples or documentation?
- What types of vulnerabilities are most effective for teaching foundational security concepts?

The initial hypothesis is that learners who interact with vulnerabilities embedded in realistic workflows will develop a deeper understanding of how security flaws arise and how to mitigate them. The project assumes that a structured, modern, and approachable platform will increase engagement, reduce learning barriers, and improve knowledge retention. By providing a controlled environment for experimentation, the platform is expected to support practical skill development while reinforcing secure coding practices.

The anticipated benefit of this research is the creation of a reusable educational tool that bridges the gap between theory and practice. The platform can support cybersecurity training, academic coursework, and independent learning by offering a modern, framework-aligned environment for vulnerability discovery and analysis. Through this applied research, the project contributes to improving accessibility, relevance, and effectiveness in web security education.

Proposed Research Project

Research Design, Objectives, and Methodology

This project adopts a design-and-implementation research methodology, which is commonly used in applied cybersecurity studies where the goal is to build, test, and evaluate a functional system. The research focuses on creating an intentionally vulnerable web application 'Insecure by Design' to investigate how embedded vulnerabilities within realistic workflows can enhance learning outcomes in web security training.

The primary objectives of the research are:

- To design and implement a modern, framework-based web application that intentionally incorporates common security vulnerabilities.
- To evaluate how learners interact with these vulnerabilities and whether guided exploration improves understanding of secure coding practices.
- To analyze which types of vulnerabilities are most effective for teaching foundational security concepts.
- To contribute a reusable, structured learning platform that aligns with contemporary development practices.

The methodology is justified by existing literature and industry practice. Prior studies on cybersecurity education emphasize that experiential learning where learners actively experiment with vulnerabilities significantly improves comprehension compared to passive reading or theoretical exercises. Platforms such as DVWA, WebGoat, and OWASP Juice Shop

demonstrate the value of hands-on learning, but they often lack structured guidance or do not reflect modern frameworks like Spring Boot. Drawing from coursework in web development, cybersecurity, and software engineering, this project applies those principles to create a more accessible, framework-aligned learning environment.

The research will follow these methodological steps:

1. System Design: Define the architecture, user flows, and vulnerability placement strategy.
2. Implementation: Build the application using Spring Boot MVC, embedding vulnerabilities into realistic features such as authentication, input handling, and file uploads.
3. Testing and Validation: Evaluate whether vulnerabilities behave as intended and ensure the environment remains safe for controlled use.
4. Analysis: Observe how vulnerabilities can be discovered, exploited, and mitigated, and assess the educational value of the platform.

Data Collection and Analysis

Data collection will consist of:

- Application behavior during testing
- User interaction with vulnerable features
- Observations from vulnerability discovery exercises

Analysis will focus on whether vulnerabilities can be identified and understood through guided exploration.

Technologies Used

The project will use a modern, industry-relevant technology stack to ensure realism and practical value:

Programming Languages / Frameworks

- Java
- Spring Boot MVC
- Spring Security (for demonstrating misconfigurations)
- Maven for dependency management

Database

- H2 (for rapid development and testing)

- MySQL (optional for production-like scenarios)

Frontend

- Thymeleaf
- HTML, CSS, JavaScript

Backend

- Spring Boot REST controllers
- Service and repository layers following MVC architecture

This technology stack mirrors real-world enterprise applications, making the learning experience more relevant and transferable.

Expected Results and Practical Contributions

The expected outcome of this research is a fully functional, intentionally vulnerable web application that supports hands-on learning of web security concepts. The platform will include multiple embedded vulnerabilities, such as SQL injection, XSS, broken authentication, insecure direct object references, CSRF weaknesses, and file upload flaws each integrated into realistic workflows.

The project is expected to:

- Provide a structured, modern environment for learning web vulnerabilities.
- Improve learners' understanding of how vulnerabilities arise in real applications.
- Demonstrate the effectiveness of guided exploration as a teaching method.
- Offer a reusable tool for future cybersecurity courses, workshops, and self-study.
- Contribute to bridging the gap between theoretical security knowledge and practical application.

By combining applied research with modern development practices, the project delivers both academic value and practical impact, supporting the broader goal of improving cybersecurity education.

Project Planning and Timeline

Project Timeline

Week	Milestones	Deliverables
Week 1 — Jan 6	Brainstorm project ideas; Begin preliminary research	Interest Survey (Due Jan 12, 11:59 PM)
Week 2 — Jan 13	Finalize project direction; Begin drafting proposal	
Week 3 — Jan 20	Refine proposal sections; Create initial GitHub repo structure	
Week 4 — Jan 27	Submit project proposal; Finalize architecture outline	Project Proposal (Due Jan 26, 11:59 PM)
Week 5 — Feb 3	Set up Spring Boot skeleton; Implement basic UI templates; Configure H2 database	
Week 6 — Feb 10	Implement first vulnerability (SQL Injection); Begin authentication module	Progress Report + Research/Implementation 1 (Due Feb 9, 11:59 PM)
Week 7 — Feb 24	Implement additional vulnerabilities (Reflected XSS, Broken Auth); Conduct internal testing; Record midterm demo video	Midterm Report + Video Archive + Working Implementation (Due Feb 23, 11:59 PM)

Week 8 — Mar 3	Implement Stored XSS; Add guided learning hints	Progress Report + Research/Implementation 2 (Due Mar 2, 11:59 PM)
Week 9 — Mar 10	Implement CSRF vulnerability; Improve UI/UX	Progress Report + Research/Implementation 3 (Due Mar 9, 11:59 PM)
Week 10 — Mar 17	Implement IDOR and File Upload vulnerability; Begin writing vulnerability walkthroughs	Progress Report + Research/Implementation 4 (Due Mar 16, 11:59 PM)
Week 11 — Mar 24	Full system testing; Finalize all vulnerabilities; Begin drafting final report	Progress Report + Research/Implementation 5 (Due Mar 23, 11:59 PM)
Week 12 — Mar 31	Finalize documentation; Prepare final presentation slides; Clean and organize GitHub repo	
Week 13 — Apr 7	Deliver final defense presentation; Submit final report and implementation	Final Report + Implementation (Due Apr 8, 11:59 PM)

Team Responsibilities

Allwyn Mascarenhas (Team Lead)

- Backend architecture and Spring Boot implementation
- Database design (H2/MySQL)
- Backend-driven vulnerabilities (SQL Injection, Broken Authentication, IDOR)
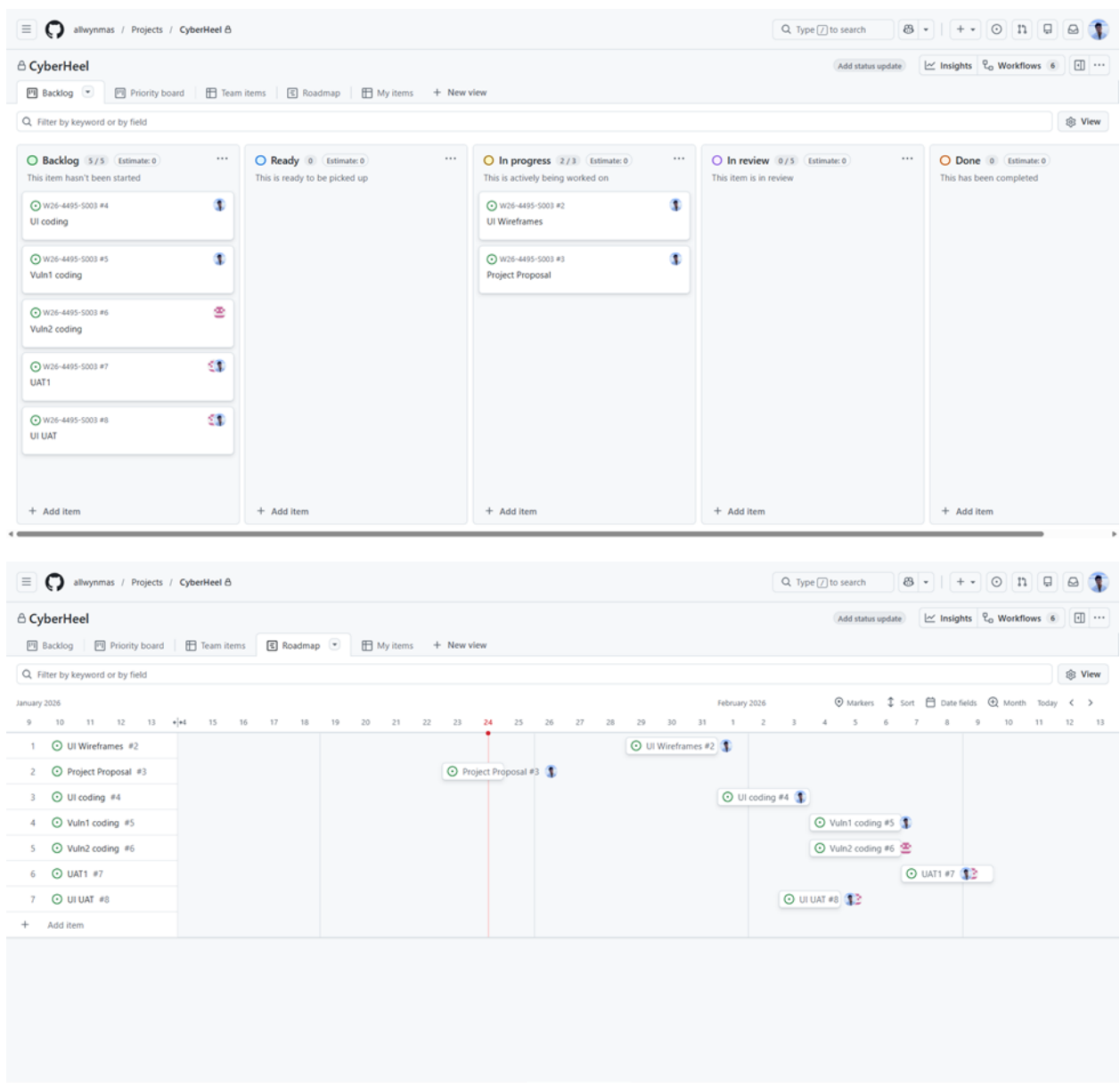- Midterm and final demo coordination

Thilan Udahage Don

- Frontend development (Thymeleaf, HTML, CSS, JS)
- Guided learning flow and UI/UX
- Frontend-related vulnerabilities (XSS, CSRF, File Upload flaws)
- System testing and vulnerability walkthroughs

Both members collaborate on:

Vulnerability research, GitHub repo management and project board, testing, documentation (proposal, midterm report, final report) video recording, and final presentation.

Project Management

The project tasks will be tracked and assigned using GitHub projects.

Team Project Contract

Project: Insecure by Design: A Hands-On Learning Platform for Web Security Training

Course: CSIS 4495 - Applied Research Project

Term: Winter 2026

1. General Commitment

We, the undersigned team members, agree to participate fully in the project to the best of our abilities and complete all work on time.

2. Communication

The project is to be discussed either in person at the college or via WhatsApp online.

3. Meetings

Regular meetings are to be held every Friday 11:00AM and Saturday 11:00AM. The duration of the meetings is 1 hour but can be extended if needed. Additional meetings are allowed.

4. Work Rules

Every team member is responsible for their part of the project. If a team member fails to deliver results on time, they need to inform other team members in advance so that this issue can be addressed.

5. Decision Making

Every decision is to be made unanimously. In case of a conflict and inability to reach a decision, a third party (instructor) can be asked for resolution.

6. Additional Agreements

For the duration and in relation to the project, all the team members are to address the selected team leader.

Signed:

#	Name	Student ID	Role	Signature
1	Allwyn Mascarenhas	300381351	Team Leader	
2	Thilan Udahage Don	300398798	Team Member	

AI Use Section

Table of AI Tools and Specific Use

AI Tool	Version	Feature Used	Value Add
Chatgpt	5.1 free	Grammar and writing help	Used only as a writing aid to flesh out original ideas
Claude	4.5 free	Prototyping and coding	Used to generate quick mocks before making final commitment to the design.

Appendix

Work Log

Allwyn Mascarenhas

Date	Number of Hours	Description of Work
15th Jan	2	Conducted initial research on OWASP Top 10 vulnerabilities and reviewed documentation for DVWA, WebGoat, and Juice Shop. Compared their learning approaches and identified opportunities for a simpler, framework-based platform.
20th Jan	1	Participated in instructor-led project consultation. Discussed project direction, technology stack, and how to embed vulnerabilities safely within a Spring Boot MVC application.
21st Jan	1.5	Co-drafted the pre-proposal for instructor approval. Summarized research findings and outlined the

Date	Number of Hours	Description of Work
		initial problem definition, research questions, and project scope.
22nd Jan	1	Researched Spring Boot security misconfigurations and common vulnerability patterns (e.g., insecure authentication flows, improper input validation). Documented potential implementation strategies.
23rd Jan	1	Attended scheduled Friday team meeting. Finalized task distribution and reviewed proposal template.
23rd Jan	1.5	Created and organized the GitHub Project Kanban board. Added tasks for proposal writing, research, vulnerability implementation, and documentation. Assigned responsibilities and set up workflow columns.
24th Jan	2	Collaborated with Thilan on drafting the full project proposal. Wrote sections on scope, methodology, and technology stack. Ensured the proposal met the rubric requirements for introduction and research design.
25th Jan	2	Continued refining the proposal independently. Reviewed the draft for clarity, strengthened the methodology section, and added additional justification for the selected technology stack.
26th Jan	1	Reviewed and refined the proposal draft. Added details to the methodology and expected results sections. Submitted the document to Blackboard and GitHub.

Thilan Udahage Don

Date	Number of Hours	Description of Work
20th Jan	1	Participated in the instructor-led project consultation. Discussed feasibility of building an intentionally vulnerable Spring Boot application and clarified expectations for the proposal.

Date	Number of Hours	Description of Work
20th Jan	1	Conducted research on existing vulnerable web applications (DVWA, WebGoat, Juice Shop). Identified gaps such as outdated frameworks and lack of structured guidance. Documented findings for proposal use.
21st Jan	1.5	Co-drafted the pre-proposal for instructor approval. Summarized research findings and outlined the initial problem definition, research questions, and project scope.
22nd Jan	1	Reviewed OWASP Top 10 documentation to determine which vulnerabilities would be most suitable for embedding into a Spring Boot MVC application. Took notes on implementation patterns and potential risks.
23rd Jan	1	Attended scheduled team meeting. Discussed proposal structure, division of responsibilities, and GitHub setup tasks.
23rd Jan	1.5	Created the GitHub repository structure following course requirements. Added folders (ReportsAndDocuments, Implementation, Misc), created worklog.md and deliverables.md, updated README.
24th Jan	2	Attended scheduled Saturday team meeting. Collaborated with Allwyn on drafting the proposal. Focused on introduction, research problem framing, and practical contributions.
25th Jan	2	Reviewed each section for coherence, corrected structural inconsistencies, and ensured that the introduction, methodology, and expected results flowed logically. Added missing justification points and improved the clarity of research questions.
26th Jan	1	Conducted a full review of the proposal draft to ensure consistency between the problem statement, methodology, and timeline. Verified that all sections met rubric requirements. Checked formatting, corrected grammar. Prepared the document for submission.

References

Fortinet. (2024). 2024 Cybersecurity Skills Gap Report. Retrieved January 2026, from
https://www.fortinet.com/content/dam/fortinet/assets/reports/2024-cybersecurity-skills-gap-report.pdf

Mudge, R. (digininja). (n.d.). Damn Vulnerable Web Application (DVWA) [GitHub repository]. Retrieved January 2026, from
https://github.com/digininja/DVWA

OWASP Foundation. (n.d.). OWASP Juice Shop. Retrieved January 2026, from
https://owasp.org/www-project-juice-shop/

OWASP Foundation. (n.d.). OWASP WebGoat. Retrieved January 2026, from
https://owasp.org/www-project-webgoat/