A Gamified Web-based Approach to Improving Phishing and Social Engineering Attack Recognition Skills

Deepak Galinki
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Suhair Amer
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

Ridam Kaphle
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

ABSTRACT
This paper presents an interactive web-based educational application designed to teach users to identify and recognize social engineering attacks, phishing emails, and scam tactics. The system provides interactive lessons, quizzes, and scenario-based challenges that help users develop critical cybersecurity awareness skills. Through gamified learning experiences, users learn to spot red flags in emails, recognize common scam patterns, and understand the psychological tactics used by attackers. The application aims to significantly improve user awareness and ability to defend against social engineering threats.

Keywords
Cybersecurity awareness, social engineering, phishing detection, scam identification, interactive learning, web-based application, gamified education, cybersecurity training

1. LITERATURE REVIEW
Salama et al. provides an overview of social engineering attacks and their growing prevalence in cybersecurity. It highlights that attackers primarily exploit human behavior rather than technical vulnerabilities through techniques such as phishing, spear phishing, baiting, and scareware. The study also discusses the stages of social engineering attacks and outlines preventive measures, including strong passwords, multi-factor authentication (MFA), cautious handling of suspicious links, and employee awareness training [1]. Suleimanov et al. study models social engineering attacks using social graphs derived from employee communication networks. By analyzing user relationships and estimating attack propagation probabilities, the proposed framework identifies vulnerable attack paths and supports organizational risk assessment. The research also introduces an analyzer system for graph construction and attack-path evaluation [2]. Gupta et al. focuses on phishing attacks as a major form of social engineering. It examines phishing techniques such as email spoofing, fake websites, Trojan horses, and fraudulent social media accounts. The study reviews detection and prevention mechanisms, including spam filters, link analysis, antivirus software, and awareness programs, while comparing several phishing detection approaches [3]. Bishnoi et al. presents a detailed review of social engineering attacks, with particular emphasis on reverse social engineering. It discusses various attack methods, including phishing, vishing, baiting, pretexting, and tailgating, and explains how attackers manipulate trust to obtain sensitive information. The study highlights awareness, training, and security policies as key defensive measures [4]. Khlobystova et al. investigates multistep social engineering attacks and their propagation across interconnected users. Using social graph models and probabilistic analysis, the study evaluates how attacks spread through networks and identifies critical points that may facilitate attack success, thereby supporting improved security planning [5].
Lungu et al. examines computational social engineering attacks on social media platforms. It proposes a hybrid detection framework that combines deep learning techniques with psychological profiling to identify malicious activities. Experimental results indicate improved detection accuracy and reduced false positives compared to traditional methods [6]. Oveh and Aziken explores the human factors associated with social engineering attacks through interviews and behavioral analysis. The findings reveal that insufficient security awareness, poor security habits, and weak personal security practices increase vulnerability to attacks such as phishing, vishing, and baiting [7]. Longtchi et al. investigates the psychological foundations of social engineering attacks. It identifies human traits such as trust, fear, curiosity, and obedience to authority as common targets of attackers. The study argues that integrating psychological understanding with technical security mechanisms is essential for effective defense [8]. Kulkarni and Nath analyze user vulnerability to social engineering attacks, particularly following the increase in online activity after COVID-19. Survey results indicate that education level, age, and organizational cybersecurity support significantly influence users' ability to recognize and resist phishing attacks [9]. Kano and Nakajima examine the role of trust in social engineering attacks on social networking platforms. Experimental findings show that community feedback and warning messages significantly reduce users' trust in suspicious content, suggesting that social influence can be leveraged to mitigate attack success [10]. Mouton et al. proposes the Social Engineering Attack Detection Model (SEADM), a structured framework based on a finite state machine. The model guides users through verification and validation processes before responding to requests, thereby reducing susceptibility to social engineering attacks across different communication channels [11]. Wang and Horkoff research focuses on assessing the likelihood and impact of social engineering attacks by incorporating human behavioral factors into probabilistic models. The proposed framework represents attacks as state changes in human behavior, enabling more accurate estimation of attack success probabilities [12]. Alahmed et al. investigates the impact of artificial intelligence on social engineering attacks. It demonstrates how AI-generated phishing messages, deepfakes, and personalized content increase attack effectiveness and scalability. The study also highlights the potential of AI-based detection systems, while emphasizing the continued importance of user awareness [13]. Arabia-Obedoza et al. provides a comprehensive analysis of social engineering attack patterns, impacts, and countermeasures. It discusses common attack types, including phishing, vishing, smishing, and impersonation, and describes the typical attack lifecycle consisting of information gathering, trust building, exploitation, and data misuse [14]. Syafitri et al. systematic review examines

various models, frameworks, and strategies for preventing social engineering attacks. The study concludes that no single defense mechanism is sufficient and recommends a combination of awareness programs, security policies, behavioral analysis, and technical controls to effectively reduce social engineering risks [15].

2. DESIGN

2.1 Design Idea

The webpage combines educational content with interactive, gamified challenges to create an engaging cybersecurity awareness training tool. The design philosophy centers on learning-by-doing: users don't just read about phishing tactics; they actively identify scams in realistic scenarios.

The application uses a lesson-based structure:

1. Lesson Content - Theory, examples, and red flags
2. Interactive Challenges - Quizzes, drag-and-drop matching, hotspot identification
3. Immediate Feedback - Users learn the correct answers immediately
4. Progress Tracking - Visual indicators show completion status

2.2 Design Goals

- Educational Effectiveness: Provide comprehensive, engaging lessons on social engineering and phishing tactics
- User Engagement: Create an interactive, gamified learning experience that maintains user interest
- Practical Application: Include realistic email scenarios and social engineering examples for hands-on learning
- Accessibility: Ensure the application is user-friendly for individuals with varying cybersecurity knowledge levels
- Knowledge Retention: Implement quizzes and interactive challenges to reinforce learning concepts
- Clear Feedback: Provide immediate, constructive feedback on user responses to quizzes and scenarios
- Evaluation Support: Include tools to collect quantitative and qualitative feedback from users
- Visual Appeal: Design an attractive, modern interface that encourages repeated use

2.3 Design Considerations

- User Experience Hierarchy: Clear navigation between lessons, Sticky sidebar for quick access to lesson list, Visual indicators of progress and active lessons, and Minimalist, distraction-free interface
- Visual Design: Dark theme with teal accent colors for reduced eye strain, Clear typography with proper contrast ratios (WCAG compliant), Consistent spacing and alignment following a grid system, and Interactive elements provide clear visual feedback
- Interaction Patterns: Single-click interaction for most tasks (minimize cognitive load), Immediate feedback (0.2-0.5 second response time), Hover states for interactive elements, and Clear disabled states for unavailable actions
- Learning Psychology: Spaced repetition through multiple lessons, Varied question types to maintain engagement, Immediate positive/negative feedback, and Clear explanations of why answers are correct or incorrect

2.4 Design Issues

Next are programming component issues:

1. State Management:
- Challenge: Managing user progress, quiz answers, and lesson completion across page reloads
- Solution: Implement localStorage to persist state; consider backend database for production
2. Dynamic Content Rendering
- Challenge: Efficiently rendering different lesson content types (text, examples, quizzes, images)
- Solution: Use component-based architecture with template rendering
3. Drag-and-Drop Functionality
- Challenge: Cross-browser compatibility for HTML5 drag-and-drop API
- Solution: Test thoroughly on Chrome, Firefox, Safari; use fallback implementations if needed
4. Quiz Logic Complexity
- Challenge: Supporting multiple question types (multiple choice, drag-and-drop, hotspot, matching)
- Solution: Create abstracted quiz component that handles different answer types polymorphically
5. Performance Optimization:
- Challenge: Keeping file size small while including all educational content
- Solution: Minify CSS/JS, optimize images, lazy-load lesson content

Nex are interface component issues:

1. Visual Clarity of Phishing Examples
- Challenge: Showing phishing emails without being too realistic and actually deceiving users
- Solution: Add clear context labels, use boxes to highlight suspicious elements, provide annotations
2. Mobile Responsiveness
- Challenge: Lesson sidebar and email examples must display well on small screens
- Solution: Implement responsive grid; convert to single-column layout on mobile; ensure touch-friendly buttons
3. Accessibility
- Challenge: Color-blind users must still see red flags in email examples; keyboard users must navigate all content
- Solution: Use patterns in addition to colors; ensure all interactive elements are keyboard-accessible (tab order, Enter/Space support)
4. Cognitive Load
- Challenge: Too much information on screen causes cognitive overload
- Solution: Progressive disclosure; show hints only on hover; break lessons into smaller sections
5. Feedback Clarity
- Challenge: Users must clearly understand why their answer was wrong
- Solution: Provide detailed explanation after quiz submission; highlight correct elements; offer learning path for improvement

3. IMPLEMENTATION

This project's implementation choices were driven by three priorities: clarity (easy to read and maintain), portability (runs in any modern browser), and interactivity (responsive, low-friction user experience).

3.1 Tools

- Programming Language: HTML5, CSS3, JavaScript (ES6).
- Frameworks/Libraries: Vanilla JavaScript (no external frameworks, for simplicity and performance)

- Development Environment: Visual Studio Code, Git version control
- Browser Compatibility: Chrome, Firefox, Safari, Edge (modern versions)
- Testing Tools: Browser DevTools, Jest (for unit tests, optional)
- Design Tools: Figma (for wireframes and design mockups)
- Version Control: Git/GitHub
- Hosting: Local file system (for submission), can be deployed to GitHub Pages or any static host
- Data Storage: Browser localStorage API
- Icons/Assets: CSS-generated shapes and gradients (no external icon libraries)

3.2 Design and Coding Decisions

The following explains the main design and coding decisions, the rationale behind them, and short descriptions of how each part of the code works.

1. Architecture & file organization
- Decision: Keep a single-page application (SPA) style structure using plain HTML, a single CSS file for theming, and a single JavaScript file for behavior.
- Rationale: An SPA with vanilla JS is lightweight, easy to host (GitHub Pages / Netlify), and simple for reviewers to run without a build step.
- How it maps to code: index.html holds the UI skeleton and placeholders (content, #lessonList, #nameBanner). style.css holds all visual variables and rules. app.js encapsulates all behavior (data, rendering, event handlers).

2. Data model for lessons
- Decision: Represent lessons as a JavaScript array of objects (lessons), each object containing id, title, content, flags, example, quiz[], and optional interactive fields (e.g., dragMatch, mockImageHotspots, mockCompare, timed).
- Rationale: Plain objects are easy to modify, extend, and iterate over. The structure supports multiple question types and activities without complex parsing.
- Implementation note: Each MCQ has { q, choices, a } where a is the index of the correct choice. Drag-match activity includes targets, draggables, and an answers mapping.

3. Rendering strategy
- Decision: Use DOM creation and innerHTML to render lesson views dynamically on demand instead of templating libraries.
- Rationale: Direct DOM rendering avoids dependencies and is straightforward for this project scale. Views are only built when needed (landing, lesson, results, leaderboard), which reduces memory overhead.
- How it works in code: render() decides current view based on contentEl.dataset.view. renderLessonView() builds the lesson UI (example, flags, quiz questions, drag area, hotspots). renderResultView(), renderLeaderboardView() follow similar patterns.

4. User interaction & state
- Decision: Maintain an in-memory state object containing currentLesson, players, currentPlayer, progress, bestScore, and board. Persist per-player data in localStorage.
- Rationale: localStorage provides a simple, persistent, client-side store that works offline and respects privacy (no external server needed). state centralizes run-time data and simplifies rendering logic.
- Key functions: loadCurrentPlayerData(), saveCurrentPlayerData(), addPlayer(), switchToPlayer().

5. Answer tracking & scoring
- Decision: Track answers in an array (answers) matching question indexes; compute score by comparing chosen indexes to q.a. For drag-drop, compare dropped item IDs to expected mapping.
- Rationale: Index-based answers are compact and easy to compare for correctness. Drag/drop uses data-* attributes to store assigned IDs.
- Implementation snippet (logic):
  - MCQ: if (answers[i] === q.a) score++;
  - Drag: if (target.dataset.got === lesson.dragMatch.answers[targetId]) score++;

6. Timed lessons & auto-submit
- Decision: Support optional timed lessons with a countdown and automatic submission on timeout.
- Rationale: Timers add challenge and gamification (e.g., the watering-hole lesson). Auto-submit ensures the lesson ends if the user doesn't click Submit.
- Implementation: setInterval decrement updates a timer element; on timeLeft < 0 the code triggers submit. Active timer cleared on navigation to prevent leaks.

7. Drag & drop implementation
- Decision: Use native HTML5 drag-and-drop events (dragstart, dragover, drop) and dataTransfer to pass IDs.
- Rationale: HTML5 drag-drop is supported across modern browsers and avoids external libraries.
- Notes: Dropped targets store dataset.got and visual state (.filled), making scoring straightforward.

8. Accessibility & keyboard support
- Decision: Make option elements focusable (tabIndex=0) and listen for Enter so keyboard users can select answers.
- Rationale: Improves accessibility and usability for users who navigate via keyboard.
- Further opportunities: Add ARIA attributes for roles and live regions for more complete accessibility (left as future improvement).

9. Badges, gamification and leaderboard
- Decision: Award badges based on progress rules (first lesson, perfect lesson, >=60% on any lesson, completed count thresholds). Leaderboard is a local store of {name, score, when} sorted by score.
- Rationale: Badges and leaderboard increase motivation and make classroom use more engaging while keeping data local and private.
- Implementation: awardBadgesAndSave() computes and persists badges; Leaderboard functions push entries into state.board and save to localStorage.

10. Persistence & privacy
- Decision: Store everything locally (localStorage), keyed per player to isolate progress and avoid server storage.
- Rationale: Keeps student data private, simplifies setup, and avoids backend development.
- Keys used: se_progress_<player>, se_best_<player>, se_badges_<player>, and se_board.

11. Export/report feature
- Decision: Provide a simple TXT export that compiles per-lesson scores and best total, then downloads via a Blob URL.
- Rationale: Allows students to include results in reports or submit offline without server-side export.
- Implementation: exportReport() builds a string, creates a blob, creates a temporary <a> link and triggers click to download.

12. UI theme & styling

- Decision: Implement a theme using CSS variables (:root) so switching accent colors or dark/light themes is straightforward.
- Rationale: Centralized theming makes future tweaks simple (e.g., add a toggle).
- Implementation: The light theme uses --bg, --card, --accent, --muted, --text etc.

13. Error handling & user prompts

- Decision: Use simple alert() and confirm() prompts for small flows (e.g., missing player name, delete confirmation).
- Rationale: Quick, browser-native dialogs are sufficient for a prototype and avoid building custom modal components.

14. Testing & extensibility

- Decision: Code is modular enough to add additional question types (e.g., multi-select), an external DB, or multi-user server later.
- Rationale: Keeps future enhancements feasible without major refactor.
- Notes: To add server-side persistence, swap localStorage calls for API calls in saveCurrentPlayerData() and loadCurrentPlayerData() and add authentication as needed.

15. Limitations & known trade-offs

- No server-side storage: good for privacy & simplicity but prevents cross-device progress sync.
- Accessibility: basic keyboard navigation is present, but ARIA roles and semantic markup can be improved.
- Drag-and-drop UX: HTML5 DnD can be less friendly on mobile — a touch-friendly fallback would improve mobile usability.
- Security of examples: All sample content is static; avoid embedding real credentials or screenshots with real user data.

16. Why these decisions were chosen

- Simplicity and reproducibility were prioritized: instructors and students should be able to run the project locally or host it quickly.
- Using native web APIs (DOM, localStorage, Blob, drag-and-drop) reduces dependencies and makes the project robust and easy to grade.
- The structure supports classroom workflows (local leaderboard, per-student profiles, exportable TXT report).

3.3 User interface

The system is designed to teach users how to identify social engineering attacks and phishing emails. The application consists of interactive lessons, quizzes, and practical scenarios that help you develop critical cybersecurity awareness. Next are the steps to use the system:

Step 1: Open the Application. Figure 1. Open the HTML file (spot-the-scam.html) in a modern web browser (Chrome, Firefox, Safari, or Edge)

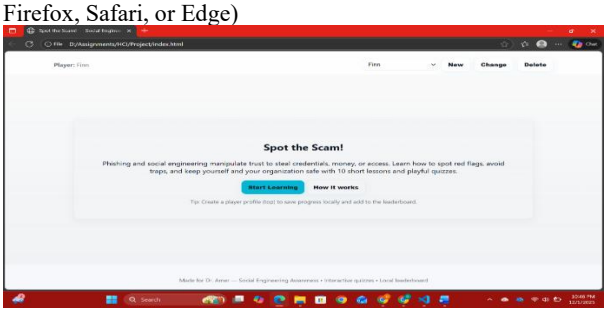

Fig 1 : Opening the HTML file

Figure 2 shows the main interface with the lesson sidebar on the left and content area on the right

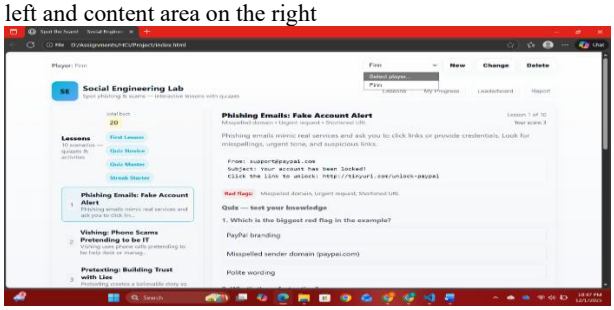

Fig 2: main interface

Step 2: Navigate to First Lesson: In the left sidebar, click on Lesson 1: Phishing Email: Fake Account Alert. The lesson content appears in the main area. Read the introduction and familiarize yourself with the key concepts. Check Figure 3.

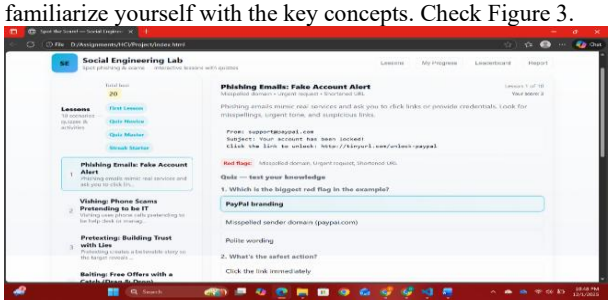

Fig 3: Lesson content

Step 3: Review Lesson Content. Read the lesson title and introduction. Look for highlighted red flags (suspicious elements) marked in red boxes. Check Figure 4.

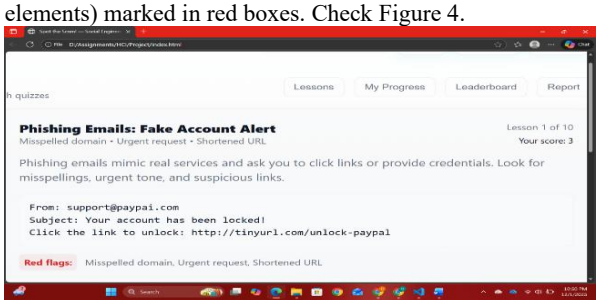

Fig 4: Locating 'red flags'

Step 4: Take the Quiz. Figure 5. After reading the lesson, scroll down to the Quiz section. Read each question carefully. Select your answer by clicking on one of the options. The selected option will highlight. Check Figure 5.

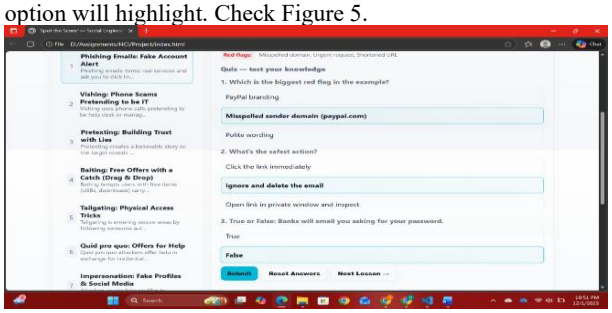

Fig 5: taking a quiz and submitting when completed

Step 5: Submit Your Answer. Figure 6. Click the Submit button. The application immediately shows if you're correct or incorrect: (a) Correct answers appear with a green border. (b) Incorrect answers appear with a red border. The score, percentage and safety tips will be displayed.

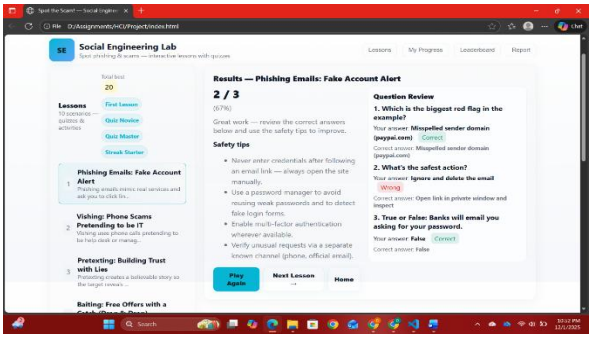

Fig 6:results

Drag-and-Drop Challenge (When Applicable). Figure 7. On certain quizzes, you'll see items you can drag. Drag it to the corresponding category. Release to drop it in place. The interface validates if your match is correct.

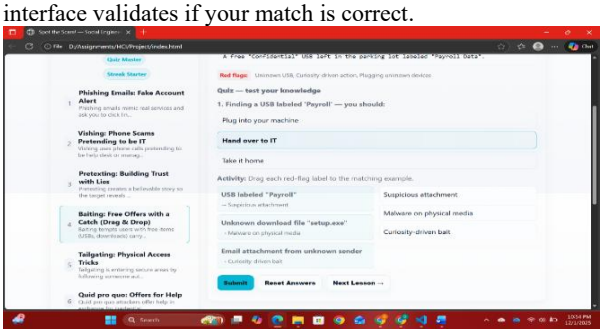

Fig 7: option to drag and drop

Hotspot Email Analysis (When Applicable). Figure 8. See a mock email interface with circular hotspots. Hover over or click each hotspot to reveal suspicious elements. Read the explanation for why each element is a red flag. Try to identify all suspicious elements before moving to the next challenge

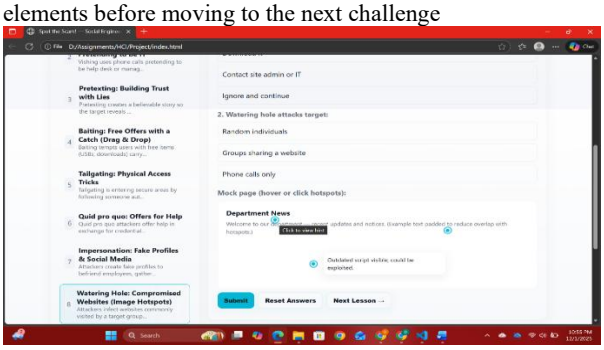

Fig 8: Hotspot Email Analysis

Moving Between Lessons. Figure 9. Click any lesson number in the left sidebar to jump to that lesson. The active lesson is highlighted with a teal border. Your progress is automatically saved. Initially it is in 2nd lesson.

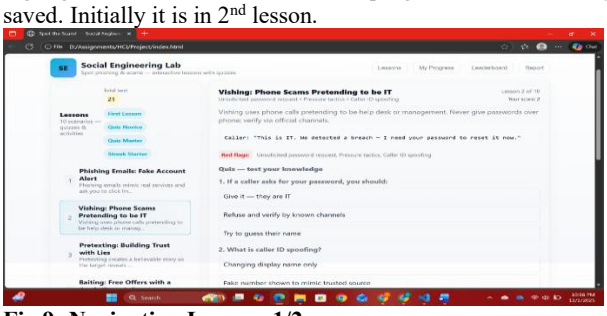

Fig 9: Navigating Lessons 1/2

After clicking on the 4th lesson, it is displayed. Figure 10.

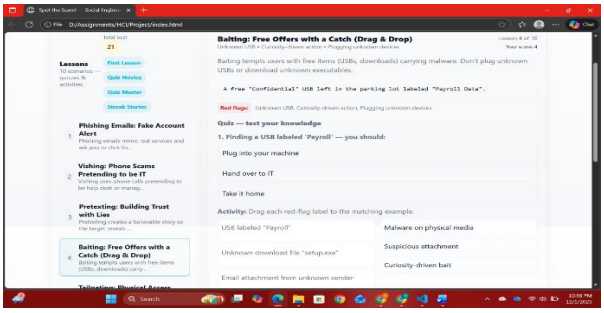

Fig 10: Navigating Lessons 2/2

For tracking progress [Figure 11], the progress indicator at the top right shows: "Lesson 1 of 10" and your score in that particular lesson. A yellow badge appears at the top left shows the maximum score in the quiz

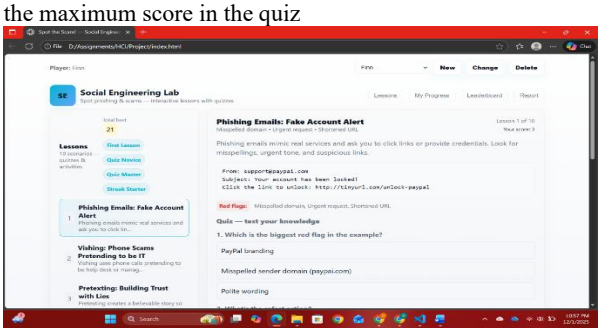

Fig 11: Tracking Progress

Step 6: Finish All Lessons [Figure 12]. Work through all lessons in sequence. Each lesson builds on the previous knowledge. As you progress, challenges become more complex. After completion of all the lessons you can see your progress.

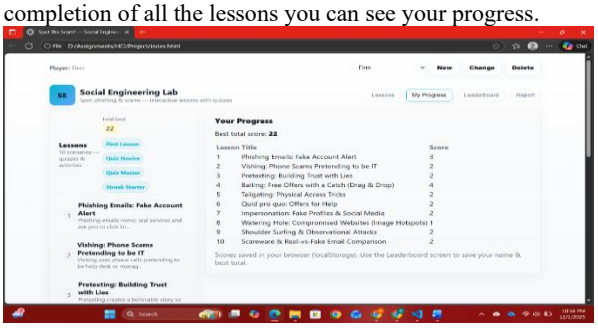

Fig 12: Completing the Learning Path

Step 7: [Figure 13] After completion of quiz you can also check the leaderboard, it compares your score and others score. You can see all the player's best score, date and time.

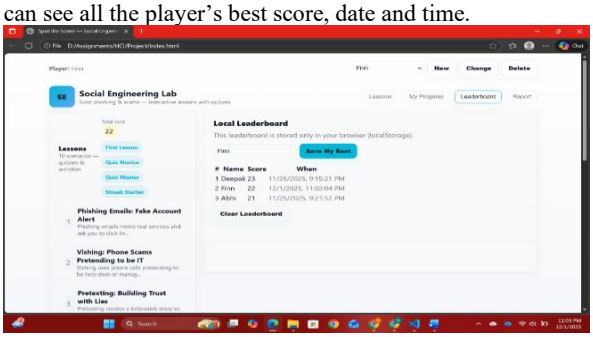

Fig 13: Leaderboard

Step 8: [Figure 14] You can export your score report in txt format. It shows your score lesson by lesson

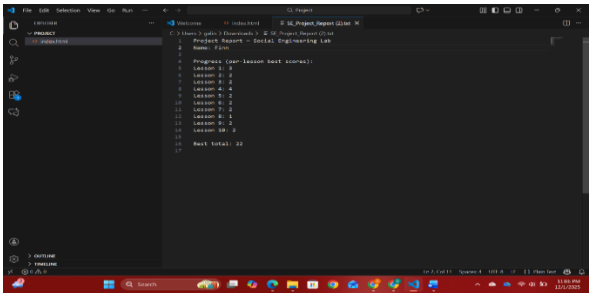

Fig 14: Export scores

4. ANALYSIS OF PRACTICAL AND ETHICAL ISSUES

The implementation of the proposed phishing-awareness application presents several practical and ethical issues that may influence its reliability, usability, accessibility, and educational effectiveness. Addressing these issues is therefore essential to ensure that the application provides consistent and responsible cybersecurity training.

4.1 Analysis of Practical Issues

Browser Compatibility: The use of the HTML5 drag-and-drop API introduces cross-browser compatibility concerns because event handling and interaction behavior may differ across Chrome, Firefox, Safari, and Edge. Such inconsistencies can directly affect task completion and user experience. Cross-browser testing is therefore required, with a text- or selection-based fallback provided where native drag-and-drop functionality is unreliable.

Mobile Touch Interaction. Drag-and-drop interactions designed primarily for mouse input may perform inconsistently on touchscreen devices. This limitation can reduce usability on smartphones and tablets. Touch event handling and a tap-based alternative should therefore be incorporated, followed by validation on representative iOS and Android devices.

Offline Functionality: Offline use presents a dependency challenge when application resources are externally hosted. Failure to retrieve these resources can prevent normal operation without network connectivity. Service-worker caching and self-contained HTML, CSS, and JavaScript resources can reduce this dependency and improve operational resilience in offline environments.

Client-Side Data Storage: Browser localStorage imposes finite storage capacity, typically approximately 5–10 MB per domain, which may become restrictive as evaluation data accumulates. Data compression, removal of obsolete records, and user-controlled data export can mitigate this limitation. Storage monitoring should also be implemented to reduce the risk of unexpected data loss.

4.2 Analysis of Ethical Issues

Realism of Phishing Content: Excessively realistic phishing examples may create a dual-use risk by exposing users to techniques that could potentially be repurposed for malicious activity. Consequently, training content should maintain sufficient realism for recognition-based learning while avoiding unnecessary operational details. Simplified, annotated examples and explicit defensive context provide an appropriate balance between educational effectiveness and misuse prevention.

User Data Privacy: Evaluation activities may inadvertently collect personally identifiable information, creating privacy and data-governance risks. The application should therefore apply data minimization principles by collecting only information necessary for evaluation, using anonymous identifiers, and avoiding unnecessary retention of IP addresses or other identifying data. Clear privacy notices and optional responses to sensitive questions should further strengthen participant protection.

Accessibility and Equity: Limited accessibility may exclude users with disabilities and consequently undermine equitable access to cybersecurity education. Compliance with WCAG 2.1 AA principles, keyboard-accessible interaction, alternative text, appropriate contrast, and compatibility with assistive technologies should therefore be treated as core design requirements rather than optional enhancements. Accessibility testing and user feedback are necessary to validate these measures.

Information Accuracy and Currency: The effectiveness of phishing-awareness training depends on the continued relevance of its instructional content. Because phishing techniques evolve rapidly, static content can become outdated and reduce training effectiveness. Periodic content reviews, monitoring of emerging phishing trends, expert validation, version control, and a documented changelog should therefore be incorporated into the content-management process.

Overall, the analysis indicates that the application's effectiveness is determined not solely by its technical functionality but also by its ability to operate consistently across platforms, protect user data, support diverse users, and deliver accurate and ethically responsible cybersecurity content. These considerations should be incorporated throughout the system's development and maintenance lifecycle.

5. DATA ANALYSIS

This study employs both quantitative and qualitative analysis. Figure 15 lists the questions used for quantitative analysis and each question is assigned a score of 1 to 5, where 1 indicates low value. to evaluate and compare the selected research papers. Figure 16 lists the questions used for qualitative analysis.

Please rate your experience with Spot the Scam! on the following scale:
1 = Strongly Disagree | 2 = Disagree | 3 = Neutral | 4 = Agree | 5 = Strongly Agree

Question	1	2	3	4	5
The lessons were clear and easy to understand	☐	☐	☐	☐	☐
The examples provided were realistic and helpful	☐	☐	☐	☐	☐
The quizzes effectively tested my understanding	☐	☐	☐	☐	☐
The user interface was intuitive and user-friendly	☐	☐	☐	☐	☐
I feel more confident identifying by phishing emails after using this system	☐	☐	☐	☐	☐

Fig 15: Quantitative evaluation form

Please provide detailed responses to the following questions:

1. What aspect of the Spot the Scam! application did you find most valuable? Why?
2. Were there any lessons or concepts that were difficult to understand? Which ones and why?
3. How could the application be improved to enhance your learning experience?
4. Which types of interactive challenges (quizzes, drag-and-drop, hotspots) were most effective for your learning?
5. Would you recommend this application to others for cybersecurity awareness training? Why or why not?

Fig 16: Qualitative evaluation form

5.1 Quantitative Data Analysis

The quantitative results [Figure 17] indicate a high level of user satisfaction with the platform, with all evaluation criteria receiving average scores above 4.0 out of 5.

The highest-rated aspects were lesson clarity and understandability (4.75) and increased confidence in identifying phishing attacks (4.75), suggesting that the platform effectively communicates cybersecurity concepts and enhances users' ability to recognize threats. Participants also rated the realism and usefulness of examples (4.50) and the interface's usability (4.50) positively, demonstrating satisfaction with both the educational content and user experience.

The lowest-rated category, although still highly positive, was quiz effectiveness in testing understanding (4.25). This result

suggests potential opportunities to refine assessment questions to better measure learning outcomes and reinforce key concepts. Overall, the average score across all categories was 4.55 out of 5, indicating strong user acceptance and confirming that the platform successfully delivers engaging and effective cybersecurity awareness training.

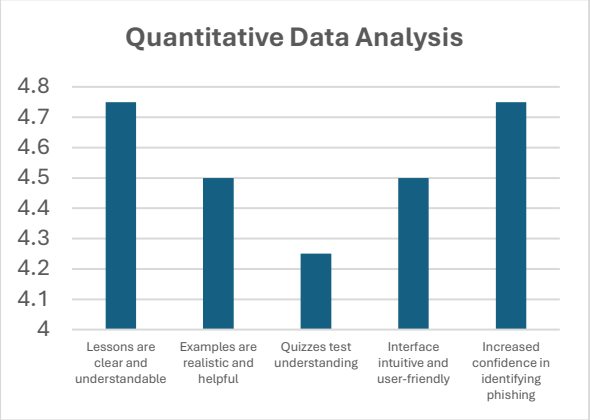

Fig 17: Quantitative data analysis

5.2 Qualitative Data Analysis

The qualitative feedback revealed four main themes: engagement and user experience, educational value, areas for improvement, and recommendation intent. Overall, participants reported strong acceptance of the website platform and found it effective for cybersecurity awareness training. In terms of engagement, users highlighted the interactive and realistic design, particularly drag-and-drop exercises and phishing simulations, which helped maintain attention and improve learning through active participation. The polished interface and realistic scenarios were also noted as enhancing user trust and experience. Regarding educational value, participants reported increased awareness of phishing techniques and improved understanding of attacker strategies and psychology. The structured content and coverage of multiple attack types were seen as effective in supporting learning outcomes. For improvements, users suggested better support for beginners, including simpler explanations, definitions of technical terms, and shorter content segments. They also recommended adding gamification features and more practical guidance such as incident response steps. Finally, all participants expressed willingness to recommend the platform, indicating strong perceived usefulness and potential for organizational deployment. Overall, the findings confirm that the platform is effective, while also highlighting opportunities for improved accessibility and learner support.

6. CONCLUSION AND FUTURE WORK

This study presented the design, implementation, and evaluation of webpage that is an interactive cybersecurity awareness platform aimed at improving users' ability to recognize phishing and social engineering attacks. By combining educational content with interactive activities and realistic scenarios, the platform demonstrated that cybersecurity training can be both engaging and effective.

The project adopted a user-centered, iterative design approach, resulting in an accessible learning tool for users with diverse technical backgrounds. Evaluation results showed high user satisfaction and confirmed the effectiveness of interactive learning methods. The implementation also emphasized modularity, scalability, and accessibility, while addressing practical challenges related to responsiveness, browser compatibility, and data management.

Future enhancements include improved accessibility features, gamification elements, personalized learning pathways, expanded content on emerging threats, backend integration for user tracking, and adaptive learning capabilities. Overall, the website provides a strong foundation for scalable and effective cybersecurity awareness training across educational, corporate, and public-sector environments.

7. REFERENCES

[1] R. Salama, F. Al-Turjman, S. Bhatla and S. P. Yadav, "Social engineering attack types and prevention techniques- A survey," *2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN)*, Ghaziabad, India, 2023, pp. 817-820, doi: 10.1109/CICTN57981.2023.10140957.

[2] A. Suleimanov, M. Abramov and A. Tulupyev, "Modelling of the social engineering attacks based on social graph of employees communications analysis," *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, St. Petersburg, Russia, 2018, pp. 801-805, doi: 10.1109/ICPHYS.2018.8390809.

[3] S. Gupta, A. Singhal and A. Kapoor, "A literature survey on social engineering attacks: Phishing attack," *2016 International Conference on Computing, Communication and Automation (ICCCA)*, Greater Noida, India, 2016, pp. 537-540, doi: 10.1109/CCAA.2016.7813778.

[4] A. Bishnoi, Garv, S. Bishnoi and N. Gupta, "Comprehensive Assessment of Reverse Social Engineering to Understand Social Engineering Attacks," *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2023, pp. 681-685, doi: 10.1109/ICSSIT55814.2023.10061054.

[5] A. O. Khlobystova and A. L. Tulupyev, "Approaches to Modeling Development Scenarios of Multistep Social Engineering Attacks," *2021 IV International Conference on Control in Technical Systems (CTS)*, Saint Petersburg, Russian Federation, 2021, pp. 100-102, doi: 10.1109/CTS53513.2021.9562746.

[6] N. Lungu, L. Barik, A. A. AlRababah, S. Rout, J. R. Saini and S. S. Patra, "Computational Social Engineering Attacks Mitigation in Social Media Platforms," *2025 5th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, Tumkur, India, 2025, pp. 1-5, doi: 10.1109/ICMNWC66779.2025.11354345.

[7] R. O. Oveh and G. O. Aziken, "Mitigating Social Engineering Attack: A Focus on the Weak Human Link," *2022 5th Information Technology for Education and Development (ITED)*, Abuja, Nigeria, 2022, pp. 1-4, doi: 10.1109/ITED56637.2022.10051202.

[8] T. T. Longtchi, R. M. Rodriguez, L. Al-Shawaf, A. Atyabi and S. Xu, "Internet-Based Social Engineering Psychology, Attacks, and Defenses: A Survey," in *Proceedings of the IEEE*, vol. 112, no. 3, pp. 210-246, March 2024, doi: 10.1109/JPROC.2024.3379855

[9] A. V. Kulkarni and S. Nath, "Human Susceptibility to Social Engineering Attacks: an innovative approach to social change," *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, Gwalior,

India, 2024, pp. 1-6, doi: 10.1109/IATMSI60426.2024.10502492.

[10] Y. Kano and T. Nakajima, "Trust Factors of Social Engineering Attacks on Social Networking Services," *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, Nara, Japan, 2021, pp. 25-28, doi: 10.1109/LifeTech52111.2021.9391929.

[11] F. Mouton, A. Nottingham, L. Leenen and H. S. Venter, "Finite State Machine for the Social Engineering Attack Detection Model: SEADM," in *SAIEE Africa Research Journal*, vol. 109, no. 2, pp. 133-148, June 2018, doi: 10.23919/SAIEE.2018.8531953.

[12] T. Li, K. Wang and J. Horkoff, "Towards Effective Assessment for Social Engineering Attacks," *2019 IEEE 27th International Requirements Engineering Conference (RE)*, Jeju, Korea (South), 2019, pp. 392-397, doi: 10.1109/RE.2019.00051.

[13] Y. Alahmed, R. Abadla and M. J. A. Ansari, "Exploring the Potential Implications of AI-generated Content in Social Engineering Attacks," *2024 International Conference on Multimedia Computing, Networking and Applications (MCNA)*, Valencia, Spain, 2024, pp. 64-73, doi: 10.1109/MCNA63144.2024.10703950.

[14] M. R. Arabia-Obedoza, G. Rodriguez, A. Johnston, F. Salahdine and N. Kaabouch, "Social Engineering Attacks A Reconnaissance Synthesis Analysis," *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2020, pp. 0843-0848, doi: 10.1109/UEMCON51285.2020.9298100.

[15] W. Syafitri, Z. Shukur, U. A. Mokhtar, R. Sulaiman and M. A. Ibrahim, "Social Engineering Attacks Prevention: A Systematic Literature Review," in *IEEE Access*, vol. 10, pp. 39325-39343, 2022, doi: 10.1109/ACCESS.2022.3162594.