Somaan Mirza

Registration number 100318760

2025

Educating Humans on Their Risk of Being Socially Engineered

Supervised by Dr. Riaz Ahmed Shaikh

University of East Anglia
Faculty of Science
School of Computing Sciences

Abstract

This project developed SecurityQuest, a gamified educational platform addressing human vulnerability to social engineering attacks. With 74% of security breaches involving human factors, traditional passive training fails to develop practical threat recognition skills. SecurityQuest transforms this through gamification elements including achievement systems, leaderboards, and personalised Artificial Intelligence feedback. The platform utilises a PostgreSQL-Express-React-Node.js technology stack with integrated large language model capabilities, featuring three progressive difficulty levels covering phishing, vishing, smishing, and spear-phishing scenarios with robust password hashing authentication. Development employed user-centred design methodology with unit and integration testing. Testing achieved 100% pass rates across 326 test cases. User evaluation with six participants over two weeks showed 52.2% improvements in cybersecurity confidence. Behavioural assessment revealed 100% of participants correctly identified phishing responses after intervention compared to 66.6% before, and 83.3% accuracy identifying threats across multiple communication channels. SecurityQuest demonstrates that gamification can transform security awareness from passive consumption into active skill development, creating measurable improvements in social engineering threat recognition.

Acknowledgements

To my father - you are the cornerstone of everything I am. Through all the ebbs and flows of life, your unwavering support has been my foundation. You have always had my back in every battle I have faced. The respect I hold for you holds deeper than words - you shaped the trajectory of my life by showing me what it means to be a man of principle, honour, and relentless determination. This achievement exists because of your example.

To my supervisor, Dr. Riaz Ahmed Shaikh, whose expertise and guidance shaped this work into something meaningful. Your belief in SecurityQuest was instrumental in bringing this project to life.

The man I am today exists because of the people who chose to invest in me when I was still becoming. That investment has returned dividends not just in academic achievement, but in the knowledge that I can face any challenge, because I do not face it alone.

Contents

1. Introduction

In today's digital landscape, cybersecurity threats increasingly target what security professionals regard as the most vulnerable system component: the human element. Social engineering, the psychological manipulation of individuals to divulge confidential information or perform compromising actions, has emerged as one of the most persistent and effective attack vectors. While organisations invest substantially in technical defences such as firewalls, encryption, and intrusion detection systems, these provide limited protection against attacks exploiting human psychology rather than technical vulnerabilities.

This project addresses this critical gap through developing SecurityQuest, an interactive educational platform that transforms how users learn to recognise and defend against social engineering attacks. This introduction provides an overview of the research context, explores the project motivation, and articulates the objectives that guided platform development.

1.1. Problem Statement

Social engineering represents one of the most significant cybersecurity challenges facing individuals and organisations today. Unlike technical attacks targeting system vulnerabilities, social engineering exploits human psychology through manipulation and deception to gain unauthorised access to confidential information. Despite growing defensive efforts, breaches involving the human element continue to dominate cybersecurity incidents.

The nature of social engineering attacks makes them particularly challenging to defend against. Threat actors manipulate inherent human psychological tendencies—trust, helpfulness, fear, and authority bias—that are essential for normal social functioning. Gragg emphasises that social engineering attacks require minimal technical effort to execute successfully, highlighting the persistent effectiveness of psychological manipulation in bypassing security measures (Gragg, 2002).

As social engineering grows more sophisticated, defensive approaches have not evolved correspondingly. Traditional security training relies on passive learning through presentations, documentation, and workshops that prioritise information delivery over skill development. These approaches communicate security concepts but rarely translate into

practical recognition and response capabilities.
Analysis of existing training methods revealed fundamental limitations:

Engagement deficit: Traditional methods fail to maintain attention and motivation, resulting in minimal knowledge retention and behavioural change.

Contextual limitations: Generic guidelines lack contextual relevance for recognising real-world social engineering attempts.

Practical application gap: Theoretical knowledge rarely translates into practical recognition skills without applied practice.

Reinforcement absence: Infrequent training sessions fail to reinforce lasting security behaviours.

1.2. Main Objectives

SecurityQuest's objective is to mitigate these limitations by developing an innovative educational platform that equips users with knowledge and skills to recognise and defend against social engineering attacks through interactive learning. Unlike traditional methods, this platform actively engages users and develops critical thinking skills, provides interactive scenarios simulating real-world social engineering attempts, offers immediate feedback and learning opportunities, and creates a dynamic, motivating environment encouraging skill development.

Rather than simply informing users about threats, the platform engages them in developing critical thinking patterns and behavioural responses to recognise manipulation attempts. This shift from passive knowledge transfer to active skill building defines the project's core innovation.

1.3. Specific Objectives

To achieve these aims, the following S.M.A.R.T. objectives were established:

Technical Objectives: Develop a secure user authentication system incorporating email verification and CAPTCHA implementation, alongside a robust Node.js backend with PostgreSQL database enabling real-time progress tracking. The platform requires a responsive user interface accessible across multiple devices while implementing comprehensive data protection measures ensuring legislative compliance.

Educational Objectives: Develop a three-tier learning progression spanning begin-

ner to advanced levels, creating realistic scenarios covering major attack vectors (phishing, vishing, baiting, spear-phishing). Educational content will maintain a 70% pass-rate threshold with real-time feedback mechanisms integrated into quiz interactions.

Gamification Objectives: Implement a point-based rewards system featuring minimum five achievement types, creating a competitive leaderboard displaying top 10 users with dynamic scoring algorithms. Achievement badges will enhance user engagement while streak-based incentives maintain interaction patterns, solidifying defence practices through sustained platform usage.

These objectives formed the structural development framework, establishing clear deliverables while ensuring educational focus. Each objective supports the platform's core innovation: transforming passive security awareness into active skill building.

1.4. Motivation and Purpose

The importance of addressing social engineering challenges grows as digital technology becomes increasingly integrated into daily life. Despite investments in security training and awareness programmes, organisations worldwide continue facing persistent social engineering attacks. The financial impact is substantial, with the FBI's Internet Crime Complaint Center reporting approximately 300,000 phishing complaints in 2023 (FBI, 2023), aggregating billions in international losses.

This report documents SecurityQuest's research, design, and implementation in addressing human cybersecurity vulnerability, demonstrates effective application of gamification principles in security awareness training, and presents user testing and platform evaluation findings. The concluding analysis assesses the overall effectiveness in addressing social engineering vulnerabilities while providing foundations for future interactive security education work.

2. Background

This section establishes the foundation for understanding the project's significance and approach. It examines the current social engineering threat landscape, explores directly related applications that have attempted to address these challenges, analyses gamification principles to security education, and identifies key needs and opportunities that informed the project's objectives and design decisions.

2.1. The Social Engineering Threat Landscape

Social engineering exploits human psychology rather than technical vulnerabilities in cyber attacks. IBM defines these attacks as manipulative techniques that lead victims to unknowingly compromise their security through information disclosure or actions that create system vulnerabilities (IBM, 2024). The effectiveness of these approaches stems from their ability to bypass technological defences by targeting human behaviour.

The significance of this threat is evidenced in the 2024 Verizon Data Breach Investigations Report, which reveals that nearly three-quarters of all security breaches involve human factors (Verizon, 2024). This statistic likely understates the problem, as many social engineering attacks go unreported or undetected. Despite growing awareness of social engineering as a threat vector, its prevalence shows concerning growth patterns with ENISA documenting a dramatic spike from approximately 15 incidents in March 2023 to nearly 100 incidents by May 2023, representing more than a 500% increase in just two months (ENISA, 2023). While fluctuations occur, the overall trajectory shows social engineering incidents in mid-2023 being significantly higher than the previous year, suggesting inadequacies in current defence approaches despite increased security awareness initiatives.

Furthermore, research has revealed organisational contexts in which also present similar vulnerabilities. Research by Hijji and Alam found that approximately 79% of organisations experienced successful social engineering attacks annually (Hijji & Alam, 2021). This striking disconnect between training implementation and security outcomes raises critical questions about the efficacy of established awareness approaches, and infers from these statistics that traditional corporate security training appears to fail, merely complying to legislation requirements, with no desire for effective development.

Individual users face different but equally significant challenges. As critical services increasingly migrate online, the population vulnerable to these attacks continues to expand. The Federal Trade Commission reported consumer fraud losses exceeding $8.8 billion in 2022, with imposter scams representing the most prevalent form of social engineering attack (FTC, 2023). This continues to insinuate the impact social engineering is having on society: while organisations may have training measures, individuals typically won't receive this guidance against these attacks.

Social engineers employ distinct psychological tactics and possess specific qualities that enable successful manipulation. Gragg identifies that these attackers exploit human

tendencies including trust, helpfulness, and authority bias (Gragg, 2002) - characteristics otherwise essential for normal-functioning society. This psychological dimension shows disparity in social engineering from other cybersecurity threats, necessitating the urgency for proper educational tools that address these challenges.

2.2. Related Work

Google's Phishing Quiz is an accessible interactive security education tool. The platform presents users with simulated emails and challenges them to identify legitimate versus fraudulent communications. While research indicates that email-based attacks serve as the initial vector for over 91% of successful system breaches (Alkhalil et al., 2021), significant limitations exist. The platforms narrow focus ignores other threat vectors including Social Media Attacks, Smishing Attacks, Vishing Attacks and USB-based Attacks that Alkhalil et al., (2021) attributes to the multi-dimensional vector space of attacks social engineering encompass. This targeted approach may leave users prepared for email threats but vulnerable to other acclaimed attack methods. Furthermore, its linear system approach lacks the adaptive elements necessary to accommodate the diverse knowledge levels, potentially creating engagement challenges for both novice and advanced users.

The Anti-Phishing Working Group's educational resource demonstrates the opposite limitation: detailed coverage but minimal interactivity. Their materials thoroughly document diverse attack methodologies but rely primarily on passive learning approaches that Halm (2015) finds inadequate for developing practical recognition capabilities. This approach demonstrates a common misconception in security education tools - that information alone suffices for developing effective defence capabilities.

Critical comparative analysis of these existing solutions reveal distinct strengths and weaknesses. Google's Phishing Quiz excels in accessibility and immediate user engagement through its intuitive interface and realistic scenarios, user's actively practice identification skills rather than merely reading them. However, this approach limits itself in covering multiple attack vectors, potentially creating what Alkhalil et al., (2021) describes as misplaced security confidence in users who master email threat identification while remaining vulnerable to other attack methodologies. Conversely, Anti-Phishing Working Group's resources offer the diverse attack vector coverage but disregards the engagement methods necessary for effective skill development. Their use of

extensive documentation does gives the theoretical understanding but fails to develop the practical recognition skills that Halm (2015) demonstrates are essential for effective defence. This analysis reveals a trade-off pattern in these related solutions: as coverage increases, engagement decreases, and vice-versa. Neither solution successfully integrates both dimensions - reinstating the gap in the current security education landscape that has directly informed this project's approach.

2.3. Gamification in Security Education

Gamification, the application of game elements in non-game contexts offers great potential for security education.

When implemented correctly, gamification offers significant advantages in social engineering education. The creation of safe practice environments allows users to develop practical threat recognition capabilities through experimental learning rather than theoretical knowledge acquisition (Barney, 2023). This approach directly addresses the limitation of traditional security training: the gap between knowing about threats and recognising them in real-world contexts.

Research shows the foundations of gamification align well with security education requirements. Deterding establishes that achievement systems satisfy competence needs by providing clear goals and recognition of task completion (Deterding et al., 2011), addressing the motivation challenges that limit traditional training effectiveness. Zhang and Muhammad demonstrated that thoughtful visual design significantly impacts user engagement in educational platforms (Zhang and Muhammad (2024), finding that visual design significantly impacts user engagement and learning retention.

However, many platforms utilising gamification emphasise competition over mastery, which may result in lower effectiveness, therefore it is paramount not to create a tool that centralises on gamification elements as opposed to the security skills development. Similarly, Halm (2015) identifies that inadequately designed reward structures may create external motivation dependencies that reduce intrinsic learning interest, reducing engagement once the novelty fades. These potential limitations highlight the importance of thoughtful gamification design that balances game mechanics with learning objectives.

2.4. Rationale

The critical analysis of current security education approaches revealed deficiencies that inspired this project's development. The false dichotomy between content and engaging delivery in existing platforms suggested a need for an integrated approach rather than accepting the current limited trade-off. Similarly, the disparity between training implementation and security outcomes indicated that different educational methods were needed, not merely refinements of existing approaches.

These findings, alongside the identified neglect of psychological manipulation awareness and the problematic implementation of gamification principles, pointed toward a significant gap in the current security education landscape. This project recognised these issues as symptoms of a more problem: the failure to approach social engineering education as a practical skill development process rather than a knowledge transfer exercise.

The project therefore aimed to develop a solution that would surpass these limitations through a refined educational tool integrating comprehensive coverage with engaging delivery, practical skill development with psychological awareness, and thoughtful gamification with broad accessibility. This approach represented not an incremental improvement but a re-conceptualisation of how security education can effectively address the human vulnerability in cybersecurity.

3. Preparation

This section outlines the preparation undertaken before implementation began. The phase established methodological foundations, design considerations of core components, and developed an experimental framework to ensure the project would effectively address social engineering vulnerability.

3.1. Methodology

3.1.1. Research Approach

The preparation adopted a user-centred methodology that placed learners at the core of all design decisions. This approach drew from reviewed research demonstrating that effective security training must address psychological factors rather than focusing solely

on technical knowledge (Gragg, 2002). As shown in Appendix A, Figure 1, the Learning Engagement Loop was designed to create a continuous cycle of educational interaction comprising of eight stages that integrate learning modules, LLM-driven feedback, and gamification elements.

The Learning Engagement Loop begins with quiz completion, followed by error analysis and LLM processing of incorrect answers, which generates personalised learning feedback. This facilitates knowledge application, leading to security skill growth, achievement unlocking, and eventually access to advanced challenges—creating a continuous improvement cycle centred around SecurityQuest. This model specifically addresses the psychological aspects of learning by providing immediate, contextually relevant feedback at the moment users are most receptive to corrective information.

The combined approach incorporated qualitative analysis of cybersecurity education platforms to identify engagement mechanisms, alongside quantitative frameworks for measuring learning outcomes and user interaction. This method reflected findings from the literature review that effective social engineering education must address both knowledge acquisition and behavioural change (Hijji & Alam, 2021). The user-centred methodology was selected over alternatives such as technical-centred approaches based on evidence that social engineering specifically exploits human psychology over technology defences. As discussed in my literature review, traditional security education approaches often fail because they prioritise technical content over psychological factors, resulting in poor engagement and limited behavioural change (Mirza, 2024). The methodology specifically addresses this gap by focusing on how users interact and internalise security concepts through active engagement rather than passive consumption.

3.1.2. Ethics and Participant Considerations

The ethics application was submitted to the University of East Anglia's Faculty of Science Research Ethics Subcommittee (SCI S-REC) and received approval on March 18, 2025. Key ethical considerations addressed in the application included informed consent procedures ensuring participants fully understood all aspects of their involvement, data protection protocols for secure handling of personal information, and a risk assessment to identify and mitigate potential participant concerns. These considerations were integrated into a holistic approach to ensure participant welfare throughout their use of the platform.

Methodology	Key Advantages	Key Limitations	Fit for Project
Gamification	High engagement, active participation, real-time feedback	Requires balanced design	Excellent
Blended Learning	Combines theory with practice	Inconsistent engagement, requires self-discipline	Moderate
Flipped Learning	Builds foundational knowledge before application	Success dependent on user commitment	Limited

Table 1: Comparison of Methodologies

The Participant Information Sheet (see Appendix B, Figure 13) was designed following established ethical guidelines for human-subject research in educational technology. This document specified that participation would require approximately 2-3 hours over a two-week period and emphasised both educational benefits and data handling procedures.

The ethics application specifically addressed the need for participants to engage with simulated social engineering scenarios while ensuring no actual security risks would be presented. This approach aligns with established practice in cybersecurity education research (IBM, 2024), where controlled exposure to threat scenarios improved recognition without creating vulnerability.

3.1.3. Gamification as Educational Methodology

Following critical analysis of several educational methodologies (see Section 2 of Literature Review), gamification was selected as the primary educational framework. This decision was based from comparative analysis of three educational approaches:

The gamification methodology follows Deterding et al.'s (2011) framework which defines gamification as the application of game design elements with non-gaming environments. As illustrated in Appendix A, Figure 4, the Gamification Framework was de-

signed to support social engineering awareness objectives through four interconnected components: Point System, Progress Tracking, Achievement System, and Social Elements.

This was judged superior to alternatives because it directly addresses the engagement challenges identified in security education literature (Google Books, 2014), provides immediate feedback crucial for learning threat recognition, creates sustained motivation through achievement systems, and aligns with research showing positive correlations between engagement and learning effectiveness (Halm, 2015). These elements together create a structure that enhances both immediate engagement and long-term knowledge retention.

Furthermore, analysis of existing security education platforms demonstrated that gamification elements substantially improved user engagement compared to traditional methods. The implementation focused on four key gamification components as visualised in Appendix B, Figures B.6 and B.7: achievement systems including badges and trophies for milestone completion, point-based scoring that quantifies performance and progress, visual progress tracking to reinforce advancement through difficulty levels, and social comparison via leaderboards to foster community engagement. These components work synergistically to create multiple motivational pathways for different learner preferences.

This methodology transformed the theory from the literature review into practical engagement mechanics to drive behavioural learning. The system specifically addressed the challenge highlighted by Lenaerts-Bergmans (2023) that security must create lasting behavioural changes rather than merely transmitting information.

3.2. Design

3.2.1. System Architecture

The system architecture was designed following a three-tier model, integrating a PERN stack (PostgreSQL, Express, React, and Node.js) with an embedded LLM component to deliver personalised learning feedback. As illustrated in Appendix A, Figure 2, this architecture creates clear separation between client-side presentation, server-side logic, database components, and the LLM processing layer.

PERN Stack Selection

During preparation, two technology stacks were evaluated, with particular focus on

comparing PERN with MERN (MongoDB, Express, React, Node.js). The PERN stack was selected based on several considerations:

- **Handling High Traffic and Transaction Volumes:** PostgreSQL demonstrates superior performance in managing high-volume traffic scenarios, making it well-suited for educational platforms that may experience variable user loads. This performance characteristic was essential for anticipated user engagement patterns, particularly during periods of high usage when multiple users might simultaneously access quizzes, submit responses, and unlock achievements.
- **Data Integrity and ACID Compliance:** Security education data requires strong data consistency, particularly for achievement tracking and user progression. PostgreSQL has been ACID-compliant since 2001, ensuring data consistency and reliability through robust transaction support, making it well-suited for maintaining the integrity of educational progress data where accurate tracking of user performance is critical.
- **Relational Data Structure:** The project required complex relationships between users, quizzes, answers, and achievements that aligned with PostgreSQL's relational model rather than MongoDB's document-orientated approach

The evaluation determined that while MongoDB would offer faster prototyping and greater schema flexibility, the structured nature of the data relationships and transaction requirements made PostgreSQL's relational approach a better architectural fit for long-term development and maintenance.

Architecture Components

The architecture comprises four primary layers, each with distinct responsibilities:

- **Server-Side Layer (Node.js + Express):** Manages API endpoints, business logic, security protocols, and external services integration. Express provides middleware for authentication, CORS handling, and request processing
- **Ollama LLM Layer:** Integrates a locally hosted large language model through the Ollama Service Component, providing personalised feedback based on user quiz performance. This layer performs context-aware analysis of incorrect answers and generates tailored learning recommendations without requiring external API dependencies

- **Database Layer:** Stores structured data across multiple tables including users, quiz content, activity tracking, gamification metrics, and authentication tokens. The relational model facilitates complex queries required for leaderboards, streak calculations, and achievement tracking

LLM Integration Architecture

A key architectural innovation is the integration of a locally-hosted large language model (LLM) for personalised learning feedback, This component analyses user responses to generate recommendation to address knowledge gaps. The LLM integration follows several design principles:

- **Local Processing:** All LLM operations occur on the local machine without transmitting data to external services, enhancing privacy and reducing latency
- **Contextual Awareness:** The system utilises LLM prompts with question metadata, user performance history, and question-specific details to generate highly-relevant feedback for users
- **Structured Responses:** Responses are formatted as JSON objects, allowing consistent parsing and presentation in the UI

The implementation flow begins when a user completes a quiz with incorrect answers. The Ollama Feedback Component in the React layer captures these errors and passes them to the Ollama Service. This service then communicates with the locally hosted Ollama server, which processes the information through prompt templates specifically trained for educational feedback. The resulting personalised recommendations are then presented to the user through an interactive interface that highlights key points, practical tips, and relevant resources.

This architecture supports the educational and gamification objectives by complementing achievement mechanisms with deeper, context-aware learning guidance that helps users overcome specific awareness challenges.

3.2.2. Database Schema Design

The preliminary database schema was designed to effectively support both educational content and user interaction tracking. Appendix A, Figure 3 presents the initial Entity Relationship Diagram showing the current table structure (Users, Quiz_completions,

Quiz_answers, User_logins, Remember_tokens, and Password_reset) with appropriate relationships and constraints. The schema addressed three primary data management challenges:

- **Security:** Isolating authentication data from learning progress metrics
- **Performance:** Optimising for frequent score updates and progress tracking
- **Analytics:** Facilitating detailed analysis of user learning patterns

The design included:

- **Users table** storing account information and gamification metrics like login_streak, and quiz_streak
- **Quiz_completions table** recording attempt details and performance metrics
- **Quiz_answers table** tracking individual responses for detailed learning analytics
- **Users_logins table** maintaining login history for streak calculations
- **Remember_tokens and Password_reset** tables handling authentication security

This schema was developed based on analysis of similar educational platforms and established database normalisation principles. The design supported the gamification structure by storing achievement progress metrics, streak counters, and detailed performance history.

3.2.3. User Interface Design

The user interface design followed a comprehensive approach that integrated visual design principles with functional considerations. As previously documented in the progress report (Mirza, 2024), the design phase involved developing both a visual design system and a set of wireframes to guide implementation.

Visual Design System

At the outset of the design process, a visual design system was created to ensure consistency and enhance the user experience (see Appendix B, Figure 14). This system defined the key visual representation that would inform all interface elements.

The colour palette was carefully constructed to serve both functional and psychological

purposes. Primary colors included dark navy blue (`#1F2937`) for navigation elements to create visual anchor points, blue (`#3B82F6`) for interactive elements to encourage engagement, and white (`#FFFFFF`) for content areas to increase readability. Achievement status was visually communicated through gold (`#F6B95C`) for unlocked achievements, bronze (`#B87333`) for special accomplishments, and silver/grey (`#C0C0C0`) for locked achievements, creating a clear visual hierarchy between gamification elements. Feedback states were color-coded (green for correct answers, red for incorrect, blue for information) to provide immediate visual reinforcement during learning activities.

A typographic system established a clear information hierarchy through page headers (20px), section headers (18px), card headers (16px), body text (12px), and small text (8px). This hierarchy ensured content was appropriately emphasized based on its importance. Consistent component styles were defined for buttons and card elements to ensure interface predictability across the platform.

The visual design system was developed with considerations for accessibility guidelines, ensuring sufficient contrast and readable text sizes. This approach aligns with research by Zhang and Muhammad, noting that thoughtful visual design significantly impacts user engagement in educational platforms (Zhang & Muhammad, 2024).

Wireframe Development

Building upon the visual design system, a set of Lo-Fi wireframes was created to define the structure and layout of key interfaces. The wireframes were specifically used in the preparation stage to show information display and user flow, as opposed to detailed aesthetics - this approach aligns with Nielson's Usability Heuristics where it is recommended to separate structural and visual concerns during the early design phases (Nielson, 2020).

Key interfaces included authentication screens for login and registration (Appendix B, Figures 6-7), a main dashboard presenting personalised messages and progress visualisations (Appendix B, Figure 8), learning components including difficulty selection, quiz interface, and results screens (Appendix B, Figures 9-10), and gamification elements displaying achievements and statistics (Appendix B, Figures 11-12). These wireframes collectively mapped the user journey through the platform, ensuring a coherent experience that supported both educational and motivational objectives.

The combination of the visual design system and wireframes created a strong foundation for implementation. It ensured that the educational and gamification objectives would be supported by an interface that was both visually and functionally effective.

3.3. Educational Content Structure

The educational content was designed to ensure progressive learning experiences across the three difficulty levels. Appendix A, Figure 5 illustrates the Quiz Progression Model showing how content was organised across three tiers:

- **Beginner Level:**
  - Phishing Basics
  - Email Red Flags
  - Multiple Choice Questions
  - Focus on Recognition
- **Intermediate Level:**
  - Advanced Phishing
  - Interactive Scenarios
  - Website Phishing Identification
- **Advanced Level:**
  - Targeted Attacks (Spear Phishing)
  - Social Media Manipulation
  - Complex Threat Identification

This progressive structure was designed based on learning theories that emphasise building foundational knowledge before introducing complex concepts. The content progression reflects FBI's Internet Crime Report Data (2023), highlighting the most common vectors, which construed phishing as being the most prevalent in the threat landscape. Each difficulty level targets specific learning outcomes ranging from basic phishing recognition at the beginner level to sophisticated defence identification at the advanced level, including other key skills to ensure users gain a holistic view on defending against the evolving cyberspace social engineering warps itself in.

3.4. Evaluation Framework

The experimental plan establishes a framework to evaluate both the technical robustness and educational effectiveness of the platform. This dual-focus acknowledges that successful security education requires not only functional systems but also demonstrable learning outcomes.

3.4.1. Testing strategy

The evaluation strategy integrates technical validation with educational assessment through complementing methodologies. Technical testing will verify system functionality through frontend component testing, backend validation, and integration verification. This will ensure that all platform components function reliably before user interaction begins, with particular focus on authentication security, quiz functionality, and the achievement systems. Simultaneously, educational effectiveness will be assessed through a pre-test/post-test design (See Appendix B, Figure 15-16) measuring knowledge acquisition and behaviour change. This will address the research question of whether the gamified methodology improves users' ability to recognise and respond to social engineering threats.

3.4.2. User Testing Strategy

The user testing component will follow a intervention approach with 5-10 community participants representing varied technical backgrounds. This population selection will enable analysis of how prior experience affects learning outcomes in security education. Participants will engage over a two-week period with recommended daily usage of 5-10 minutes – a duration selected to balance skill development with sustained engagement.

Assessment instruments include pre-intervention and post-intervention surveys measuring security awareness across multiple dimensions. The pre-intervention survey will establish baseline knowledge through confidence ratings and scenario responses, while the post-intervention survey will evaluate improvements using parallel but non-identical questions to prevent testing effects. The post-survey will also include open-ended questions gathering qualitative feedback on user experience, platform strengths and limitations, most valuable learning elements, and suggested improvements. These qualitative responses will provide deeper insights into user engagement and satisfaction beyond

quantitative metrics. Platform analytics will automatically track engagement patterns throughout the intervention period without requiring participant action, complementing self-reported measures with objective interaction data.

3.5. Data Collection

Data collection balances comprehensive measurement with participant convenience and ethical considerations. The platform will automatically record interaction data, quiz performance, and achievement progression. Survey instruments at pre and post-intervention stages will gather self-reported measures and scenario responses, with each instrument designed for completion in 10-15 minutes. All data collection follows approved ethics protocols, with participants fully informed about data usage. Personal identifiers will be separated from performance data to ensure confidentiality while maintaining the ability to match pre/post results for individual learning analysis. This creates a structured framework for evaluating both technical functionality and educational effectiveness, establishing clear methodologies for assessing impact on social engineering awareness.

4. Implementation

4.1. Frontend Implementation

The frontend implementation combines modern web technologies with gamification principles to create an engaging learning platform for social engineering awareness. This section explores how the architectural decisions, component implementations, and interface designs work together to deliver an effective educational experience.

4.1.1. Component Architecture

The application follows a structured component architecture built with React, emphasising modularity and clear separations of concerns, which facilitates ongoing feature development and maintenance. The component architecture diagram (see Appendix C, Figure 16) illustrates the relationship between key frontend components.

The architecture employs strategic design patterns to enhance the application's functionality. ThemeProvider implements a context-provider to manage theme states across the entire application, enabling seamless light/dark mode switching without prop drilling

- a practice that Dodds identifies as problematic in respect to application maintenance and component complexity (Dodd, 2018). This approach enhances both accessibility and user satisfaction. Security is enforced through the ProtectedRoute component, which validates authentication before allowing access to private routes, creating a secure learning environment while maintaining a smooth user experience.

4.1.2. Core Component Implementation

Authentication Components Authentication forms the security foundation, implemented through components that balance security with usability - critical for an educational platform focused on security awareness.

The login component handles user authentication with features including real-time validation feedback, password visibility toggling, and "Remember Me" functionality as shown in Appendix C, Figure 17. Security is enhanced through proper credential handling and validation, while maintaining user experience through visual indicators and interactive elements.

The registration component implements robust user registration with password strength validation as shown in Appendix C, Figure 18. This feature not only enhances security but also teaches users best practices through immediate visual feedback. The user interface also provides character counters and validation indicators that educate users about proper password creation.

Security is further enhanced through CAPTCHA integration (Appendix C, Figure 20) that not only prevents automated attacks but also serves as an educational element, demonstrating security verification concepts in practice. The implementation challenges users to identify specific systems in images, providing hands-on experience with security verification systems.

The authentication components integrate with the achievement system through streak tracking, connecting security practices with gamification elements. These components work together to create a secure yet educational experience, implementing best practices in web security whilst teaching users through direct interaction. The password requirements visualisation, character limits, and CAPTCHA verification all serve dual purposes: enhancing application security while providing practical security education.

The frontend authentication demonstrates how security principles are integrated directly into the user experience, making the learning process begin from the moment

the user interacts with the platform. This approach reinforces the educational mission by providing hands-on experience with security concepts from the initial login process through all interactions.

Dashboard Component The Dashboard implements a user interface that combines progress monitoring, learning modules, achievements, and recent activity into one cohesive page. The component structure prioritises both educational elements and gamification features, providing users with a complete view of their security awareness journey.

A weighted progress calculation algorithm ensures accurate representation of skill development across different difficulty levels (see Appendix C, Figure 24). This calculation provides users with an accurate reflection of their security knowledge growth, encouraging continued engagement by giving more weight to advanced content. Upon login, the dashboard implements an achievement notification system that celebrates user milestones in animated pop-ups, as shown in the top right corner of Appendix C, Figure 23. This creates immediate reinforcement when security milestones are reached.

The module selection interface uses colour-coding and progress indicators to guide users toward content based on their current knowledge level (see Appendix C, Figure 25). This creates a structured learning path that gradually increases in complexity - essential for security education as it ensures users master foundational concepts before advancing to more complex topics.

The dashboard also tracks achievement progress and displays recently unlocked achievements (see Appendix C, Figure 26). The activity timeline component provides a chronological view of user interactions with the platform, highlighting recent quiz completions and tracking performance improvements over time. This feature helps users track their learning journey and identify areas for improvement.

The statistics visualisation transforms raw quiz data into analytical insights, helping users identify their strengths and weaknesses in social engineering awareness. This approach directs users toward the security topics where they need the most improvement.

The dashboard demonstrates effective integration of educational content with gamification elements, creating a compelling user experience that encourages continued engagement. Through its comprehensive visualisations of user progress and achievements, the dashboard transforms abstract security concepts into a tangible learning journey with clear milestones and rewards.

Leaderboard Component The Leaderboard component utilises a competitive element that drives sustained engagement through social comparison - a key gamification

technique that encourages continued learning beyond initial interest. This section examines the implementation of the leaderboard and its role in maintaining user motivation.

The Leaderboard features visualise indicators for top performers, creating aspirational targets that motivate continued engagement (see Appendix C, Figure 28). The implementation differentiates top-ranked users with special icons and styling, while providing detailed performance metrics including scores, accuracy rates, and streak counts (see Appendix C, Figure 29).

The scoring algorithm ensures fair comparisons across different challenge types and difficulty levels (see Appendix C, Figure 30). This sophisticated approach normalises scores from various quiz formats to create consistent ranking criteria.

The streak calculation functionality for both login and quiz completion rewards consistency - a critical factor in building lasting awareness habits (see Appendix C, Figure 31). This dual-streak system encourages both regular platform engagement and continuous learning.

The leaderboard also includes educational information about scoring mechanisms and streak calculations, ensuring users understand how performance is measured (see Appendix C, figure 32).

The leaderboard component demonstrates how competitive elements can enhance security education by creating social motivation through comparison. This focus on habit formation through gamification supports the goal of creating lasting security behaviour changes. By highlighting both accuracy (quality of learning) and streaks (consistency of engagement), the leaderboard encourages both thorough understanding and regular practice - deemed essential for developing effective security habits.

Statistics Component The Statistics component provides detailed analytics that help users identify their strengths and weaknesses in their social engineering journey. This section examines the implementation of the statistics interface and how it transforms raw quiz data into meaningful learning insights.

The statistics implementation utilises a tabbed interface to organise analytics into three categories - overview, quiz details, and streaks - each providing different insights into the user's learning journey (see Appendix C, Figure 33).

The difficulty-specific analysis component (Appendix C, Figure 35) enables targeted improvements by highlighting performance patterns across different security topics. By separating analytics by difficulty level, users can easily identify which areas of social engineering awareness require additional focus (see Appendix C, Figure 36).

The streak tracking functionality (Appendix C, Figure 37) incentivises engagement by visualising login and quiz completion patterns. This feature supports the gamification objectives by promoting regular interaction with the content (see Appendix C, Figure 38).

The statistics component also implements an interactive score history visualisation that shows performance trends over time, with the pass threshold clearly indicated (Appendix C, Figure 39). This implementation used SVG-based charting code to ensure correct rendering.

The statistics component transforms raw quiz data into meaningful analytics that guide the user's learning journey. By providing insights into performance patterns, identifying improvement areas, and incentivising consistent engagement, this feature helps users develop effective security habits. This implementation balances clear visualisation of analytics to make data immediately actionable, directing users to where they need improvement.

Achievement System Component The Achievement system reinforces security awareness through visual recognition of user progress, implementing gamification elements to enhance learning engagement.

The achievement interface organises accomplishments by category, status, and rarity, creating a clear progression path for users (see Appendix C, Figure 40). The dashboard displays progress metrics and includes category filters that help users identify learning objectives (see Appendix C, Figure 41). The visual representation emphasises both current progress and remaining challenges.

The achievement service handles detection, progress tracking, and notification management (see Appendix C, Figures 42-43). The service implements a primary-secondary persistence strategy ensuring it preserves progress even during API failures, maintaining consistent user recognition (see Appendix C, Figure 43).

The notification system creates a satisfying unlocking experience through CSS transitions and icon animations (see Appendix C, Figure 44). For multiple simultaneous unlocks, the system groups them into a consolidated notification to prevent UI clutter.

The system features achievements across multiple categories, from engagement-based rewards like "Dedicated User" to performance-based recognitions such as "Perfect Score" (see Appendix C, Figure 45). Different rarities create a progression hierarchy that maintains motivation at all skill levels, reinforcing both security awareness and habit formation for long-term behaviour changes.

4.1.3. Quiz Component System

The quiz system serves as the primary educational mechanism for social engineering awareness, implementing varied question formats and interactive detection challenges tailored to different skill levels.

The quiz module follows a structured flow pattern with three primary components: difficulty selection, interactive questions, and comprehensive results. This architecture supports a progressive learning approach where users can select appropriate challenge levels based on their current knowledge (see Appendix C, Figure 46).

The quiz system implements four specialised question types, each addressing different social engineering vectors that users may encounter (see Appendix C, Figure 47). The email phishing component has interactive suspicious element detection through clickable elements that users identify as potential security threats (see Appendix C, Figure 48). The SMS-based smishing component simulates mobile messaging threats, teaching users to identify suspicious links and social engineering tactics delivered through text messages (see Appendix C, Figure 49). The voice-based vishing component presents transcript analysis challenges where users identify manipulation techniques in recorded conversations (see Appendix C, Figure 50).

The quiz system's implementation demonstrates the approach to security education through interactive challenges, addressing multiple attack vectors across the social-engineering cyber space. By simulating diverse threats across multiple media formats, users develop practical skills for identifying malicious attempts, reinforced by both challenge completion and achievement recognition (see Appendix C, Figure 51).

The Quiz Component forms the educational core, delivering interactive learning experiences through diverse social engineering scenarios. This subsystem employs a modular architecture that adapts to different difficulty levels while maintaining consistent scoring and feedback mechanisms.

4.2. UI Design System Implementation

The user interface focused on creating an accessible, responsive, and visually engaging experience that reinforces the educational objectives whilst maintaining consistent design principles. This section details how the visual design was translated into functional interfaces and how UI elements were implemented to enhance the learning experience.

System Design Implementation The visual design system described in the preparation section was implemented through a structured CSS architecture. The base styling is defined through global CSS variables and element-specific rules as shown in Appendix C, Figure 52.

Interactive elements were implemented with consistent hover and focus states to enhance usability and provide clear visual feedback. For example, buttons implemented specific visual treatments to indicate their interactive nature, as shown in Appendix C, figure 53.

Accessibility Considerations The platform incorporates accessibility features to improve the application's usability, particularly through its theming system and interface design:

- **ARIA labels:** The theme toggle button includes proper ARIA labelling to improve screen reader compatibility (see Appendix C, Figure 54)
- **Colour Contrast:** The theme system provides contrast ratios between text and background colours in both light and dark modes (see Appendix C, Figure 55)
- **System preference detection:** The application respects users' system-level theme preferences (see Appendix C, Figure 56)

Responsive Design Implementation The platform ensured responsive design across different device sizes. The application employs a multi-tiered strategy using CSS media queries as shown in Appendix C, Figure 58.

The responsive layout implementation includes a grid-based structure for content organisation, as demonstrated in the dashboard's grid layout (see Appendix C, Figure 59). This approach ensures proper content hierarchy and improves the visual organisation of elements across different screen sizes, allowing components like progress metrics, achievement tracking, and learning modules to maintain their visual relationships while adapting to various device dimensions.

4.2.1. Navigation Implementation

The sidebar navigation is the main way users navigate through the platform. It had a strong focus on both accessibility and visual hierarchy to guide users through the various interfaces (see Appendix C, Figure 57 to view the CSS and any interface snippet to view its UI).

The responsive navigation bar includes text to match each page, alongside icons to ensure effective communication and enhance visual appeal, regardless of screen size. The navigation includes a collapsible mobile menu, which uses a hamburger icon to conserve screen space whilst maintaining accessibility. The interface was implemented to enable usage across device sizes, using flexbox and CSS grid layouts to ensure users could interact regardless of device. Media queries define breakpoints at 480px, 768px, and 1024px, with component layouts adjusting dynamically (see Appendix C, Figure 60).

4.2.2. Feedback Mechanism

To enhance educational value, several feedback mechanisms were included to provide users with immediate responses to their actions. The toast notification system shown in Appendix C, Figure 61 provides contextual feedback for user actions.

Progress indicators were added to provide visual feedback on user advancement, as shown in the code snippet in Appendix C, Figure 62.

4.3. Backend Implementation

The backend implementation provides the core infrastructure that supports the educational and gamification objectives of the platform. This section focuses on key components that enable functionality of the application.

4.3.1. API Development and Endpoints

The platform implements security measures for its API architecture, including Cross Origin Resource Sharing (CORS) configuration to prevent unauthorised access from external domains (see Appendix, C, Figure 63). This configuration enhances application security by controlling which domains can interact with the API, preventing cross-site request forgery while allowing legitimate client request from the frontend application.

The system uses a RESTful API architecture with Express.js with endpoints organised into logical categories.

Key API endpoints include:

- **Authentication endpoints:** /api/login, /api/register, /api/forgot/password, /api/reset-password

- **Quiz endpoints:** /api/quiz/questions/:difficulty, /api/quiz/complete
- **User progress endpoints:** /api/users/:userId/achievements, /api/users/:userId/threat-performance
- **Leaderboard endpoints:** /api/users, /api/users/quiz-history

Each endpoint incorporates consistent error handling patterns (see Appendix C, Figure 64).

4.3.2. Database Integration

The system uses PostgreSQL for data persistence, using a schema that supports both educational content and achievement tracking as shown in Appendix c, Figure 65.

The database connection is managed through a connection pool for optimal performance (see Appendix C, Figure 66). The database also enables sophisticated queries that power the analytics features (see Appendix C, Figure 67).

4.3.3. Authentication and Security Implementation

The platform implements a robust authentication system using bcrypt, a powerful cryptographic hashing algorithm designed specifically for password security. As shown in Appendix C, Figure 68, bcrypt provides significant advantages over standard hashing algorithms by using a salt and work factor that protects against rainbow table attacks and brute force attacks - keeping users safe.

The bcrypt implementation uses a work factor of 10, providing an optimal balance between security and performance (see Appendix C, Figure 69). During login, bcrypt compares password using timing-safe verification to prevent timing attacks (see Appendix C, Figure 70). The system also implements a secure "Remember Me" functionality using cryptographically secure tokens (see Appendix C, Figure 71).

4.3.4. LLM integration for Personalised Feedback

A key feature is the integration of a locally-hosted large language model for personalised learning feedback. The implementation uses Ollama, a lightweight inference server for running open-source models locally (see Appendix C, Figure 72).

The personalised feedback is displayed to users through an interactive interface that highlights key learning points and practical recommendations, in effect, a user should understand where they went wrong, to then correct themselves after they re-attempt (see Appendix C, Figure 73).

The backend provides the foundation for the educational objectives, delivering secure authentication through bcrypt password hashing, sophisticated data management, and personalised learning experiences through the integration of local LLM technology.

5. Testing

5.1. Methodology

The testing strategy employed a comprehensive approach that addressed technical reliability and educational effectiveness. As an application designed to educate users about social engineering risks, it was essential to validate not only the software functionality but also its pedagogical impact. Testing was conducted across four distinct dimensions:

- **Frontend unit testing:** Validating individual components and their interactions
- **Backend unit testing:** Ensuring server-side logic and database operations functioned as intended
- **Integration testing:** Verifying operations between frontend and backend components
- **User acceptance testing:** Evaluating the educational effectiveness and user experience

This multi-faceted approach established confidence in both technical and educational value. The testing methodology followed industry best practices, with automated tests providing validation during development and user testing confirming real-world effectiveness.

5.2. Frontend Unit Testing

Frontend testing was implemented using Jest and React Testing Libraries, focusing on component functionality, user interactions, and state management. The test suite in-

cluded 66 individual tests across 15 test suites, with a pass rate of 100%. Table 2 provides a breakdown of the test suite organisation, highlighting the coverage across the application's frontend components.

Category	**Test Suites**	**Test Cases**	**Pass Rate**	**Key Focus Areas**
UI Components	8	81	100%	Display Rendering, User Interaction, Authentication Flow, Security Verification, Achievement Notifications
Pages	5	48	100%	Authentication, Navigation, Data Display, State Management, Error Handling
Services	2	37	100%	Achievement Tracking, LLM Integration, Feedback Generation, API Integration
Overall	**15**	**166**	**100%**	

Table 2: Frontend Unit Testing Summary

The test coverage was distributed to ensure thorough validation of security-focused components (CAPTCHA, authentication), educational elements (quiz questions, personalised feedback), and gamification features (achievements, leaderboard).

Frontend testing revealed that all core functionalities were working as expected with no failures across any test cases. The 100% pass rate confirmed reliable implementation across all components.

Key validated functionalities included:

- Authentication security with password validation, CAPTCHA implementation, and session management
- Achievement tracking and notification systems for gamification elements
- Adaptive feedback generation through the Ollama LLM service

- Responsive UI elements with proper state management across different device sizes

5.3. Backend Unit Testing

Backend unit testing employed a comprehensive test suite to ensure reliability and security of server-side components. While frontend testing focused on user interface and interaction, backend testing targeted the core logic, database operations, authentication flows, and API endpoints - critical infrastructure that supports the application's functionality. Backend testing comprised 94 individual tests across 37 test suites, achieving a 100% pass rate. Tests were strategically organised to verify functionality across all components, Table 3 shows the breakdown:

Category	**Test Suites**	**Test Cases**	**Pass Rate**	**Key Focus Areas**
Authentication	12	26	100%	Login/Registration, Password Security, Token Management, Remember Me, Password Reset Flow
Achievement System	3	8	100%	Achievement Unlocking, Progress Tracking, Edge Case Handling
Quiz System	5	22	100%	Quiz Completion, Performance Analytics, Question Validation
User Management	6	18	100%	Streak Calculation, Profile Management, Leaderboard Functionality
Security & Infrastructure	11	20	100%	CORS Configuration, Error Handling, Database Connection, SQL Injection Prevention
Overall	**37**	**94**	**100%**	

Table 3: Backend Unit Testing Summary

The testing regime verified critical aspects of the system:

- **Authentication Security:** Tests confirmed that all password handling properly utilised bcrypt with appropriate work factors (value: 10) for optimal security-performance balance. Tokens were properly generated with cryptographically secure methods and included appropriate expiration handling
- **Data Validation:** Edge case testing validated input handling in cases where quiz submission processing manages and stores multiple data formats correctly
- **Database Interaction Security:** SQL injection tests confirmed that all database queries are correctly parameterised, preventing malicious attacks crafted by SQL queries
- **Error Handling:** Testing verified standardised error handling across all API endpoints, ensuring predictable client-side error management
- **Transaction Integrity:** Streak calculation testing confirmed proper handling of edge cases, including same-day quiz submissions and streak reset logic based on user activity patterns

5.4. Integration Testing

Integration testing was implemented to evaluate how different systems interact with each other, complementing the unit tests by focusing on cross-component functionality. This approach helped identify issues that may not have been apparent when testing components in isolation.

Testing Methodology The integration tests employed Jest with Supertest for HTTP request simulation. Each test suite included setup and teardown approaches to create controlled testing environments with consistent test data. The tests were designed to validate complete workflows from frontend user interactions through API endpoints to database operations. The integration tests comprised 5 test suites containing 66 total tests, achieving a 100% pass rate across all system components. Table 4 provides a breakdown of the integration tests:

Category	**Test Suites**	**Test Cases**	**Pass Rate**	**Key Focus Areas**
Authentication	1	16	100%	Login/Registration, Password Security, Token Management, Remember Me, Password Reset Flow
Achievement System	1	15	100%	Achievement Unlocking, Progress Tracking, Edge Case Handling
Quiz System	1	14	100%	Quiz Completion, Performance Analytics, Question Validation
User Management	1	12	100%	Streak Calculation, Profile Management, Leaderboard Functionality
Security & Infrastructure	1	9	100%	CORS Configuration, Error Handling, Database Connection, SQL Injection Prevention
Overall	**5**	**66**	**100%**	

Table 4: Integration Testing Summary

Each integration test validated its respective area of functionality. The tests confirmed critical aspects of system behaviour:

- **Authentication:** Tests validated complete user journey from registration through login to password reset, confirming proper credential handling, token management, and security protections
- **Achievement System:** Tests verified that achievements were properly tracked and unlocked based on user activity patterns, including login streaks, quiz completions, and performance milestones
- **Leaderboard:** Tests confirmed proper user ranking, score calculations, and streak tracking logic across different quiz types and difficulty levels
- **Statistics:** Tests validated user performance analytics, quiz history tracking, and login pattern monitoring

- **Ollama AI Service:** Tests verified the AI feedback generation system, question analysis capabilities, and response formatting for personalised learning feedback

The completion of all 66 integration tests with a 100% pass rate confirms robust interaction between components. Outputs to each section of testing can be found in Section C of the Appendix, Figure 74.

5.5. User Testing

Following the technical validation through unit and integration testing, user testing was conducted to evaluate the platform's educational effectiveness and user experience. This phase directly implemented the gamification assessment outlined in 3.4 of the preparation section, determining whether the platform achieved its primary objective: improving users' ability to recognise and respond to social engineering threats through gamified learning.

5.5.1. User Testing Methodology

User testing employed a pre/post survey structure to measure changes in cybersecurity awareness, knowledge, and behaviour. 6 participants were recruited, with testing conducted across three distinct phases:

- **Pre-intervention assessment:** Initial survey measuring demographic information baseline cybersecurity knowledge, and responses to social engineering scenarios
- **Platform interaction:** 10-day usage period engaging with educational content and gamification features
- **Post-intervention assessment:** Follow-up survey with new scenarios and detailed platform evaluation questions

To ensure multi-device accessibility as promised in the participant information sheet while working within local hosting constraints, testing sessions were conducted on the researcher's development machine with participants who accessed the application directly for desktop evaluation and used browser developer tools for mobile interface testing. This approach maintained the device compatibility while providing controlled conditions for consistent data collection. Participant demographics represented a diverse

range of ages and technical proficiency levels, with 66.7% having no prior cybersecurity training. Demographic distribution included participants primarily in the 18-24 age range and most participants identified having beginner-level technical proficiency, making them an ideal target audience for testing the educational impact of the platform.

The comprehensive user testing results and analysis of educational effectiveness are presented in the Discussion and Evaluation section, where they are evaluated against the project's core objectives.

6. Discussion and Evaluation

This section evaluates the platform's effectiveness in addressing social engineering vulnerabilities through gamified education, analysing outcomes against objectives and examining implementation challenges.

6.1. Achievement of Technical Objectives

The technical implementation successfully achieved all core objectives. Unit testing confirmed full functionality of the authentication system with 26 test cases verifying bcrypt implementation, token management, and security protocols (Table 3, Backend Testing Summary). Integration testing validated these components with a 100% pass rate across 16 authentication test cases. Email verification was also successfully implemented using SendGrid's cloud-based SMTP service, completing the comprehensive security authentication flow.

The database implementation effectively supported the application's functionality with all 94 backend unit tests passing. The UI delivered responsive design across multiple device sizes, though user testing identified specific improvements needed for mobile interaction patterns especially in quiz scenarios where 33.3% of participants reported difficulty with selection precision.

6.2. Educational Effectiveness and Knowledge Acquisition

The platform's educational objectives were thoroughly validated through comprehensive user testing. The pre and post-intervention surveys revealed substantial improvement in participants' self-reported cybersecurity knowledge across all measured dimen-

sions. As shown in Appendix C, Figure 75, confidence levels increased by an average of 52.2% across all categories, with the most significant improvement observed in social engineering attack recognition (from 2.17 to 4.00, an 84.3% increase).

The post-intervention assessment evaluated participants' mastery of specific social engineering concepts. Results indicated strong understanding of urgent language tactics (4.50/5.00) and the distinction between phishing and spear-phishing (4.33-5.00), as illustrated in Appendix C, Figure 76. These results demonstrate that the three-tier difficulty system effectively facilitated progressive learning, with participants mastering foundational concepts before advancing to complex threat identification.

The attack vector simulations correctly covered all planned threat types, with 83.3% of participants correctly identifying threats across different communication channels compared to baseline measurements below 50% (Appendix C, Figure 76). This multi-vector approach represents a significant advancement over existing platforms like Google's Phishing Quiz, which focuses solely on email threats.

6.3. Security Behaviour Change Analysis

Beyond knowledge acquisition, the study measured actual security behaviour changes through scenario-based assessments - the ultimate measure of educational effectiveness. Participants showed substantial improvement in their responses to simulated social engineering scenarios, as shown in Appendix C, Figure 77.

Most notably, after using the platform:

- 100% of participants correctly identified checking the sender's email address as the first action when receiving suspicious attachments (up from 66.7%)
- 100% showed appropriate caution regarding password sharing over the phone (up from 83.3%)
- 83.3% correctly identified proper verification procedures for suspicious requests (up from 50.0%)

The post-survey included additional security scenarios not present in the pre-survey to test knowledge transfer to new contexts. In these scenarios:

- 100% of participants correctly identified the proper response to suspicious SMS messages from spoofing banking institutions

- 83.3% correctly handled unsolicited document requests through official work channels
- 66.7% selected appropriate network security practices when working remotely

These results suggest not only improved knowledge but actual behavioural adaptation to security threats, addressing the fundamental gap identified in traditional security training approaches that fail to translate theoretical knowledge into practical defensive behaviours.

6.4. Gamification Effectiveness and User Engagement

A key objective of this research was to evaluate the effectiveness of gamification elements in maintaining user engagement and enhancing learning outcomes. The results demonstrated strong impact of competitive features on learning motivation, with 100% of participants reporting that gamification elements either somewhat (66.7%) or significantly (33.3%) enhanced their learning experience.

As shown in Appendix C, Figure 79, the leaderboard emerged as the most motivating feature (50%) with achievement badges, point scoring, and daily streaks each motivating 16.7% of participants. The leaderboard's impact was further confirmed by usage patterns, with 88.3% checking their leaderboard position frequently. System logs confirmed consistent interaction with these elements, with achievement unlocks correlating with increased session duration.

Qualitative feedback reinforced these quantitative findings: *"I really enjoyed the competitive aspect with the leaderboard - seeing my progress compared to others kept me motivated to improve. The daily streaks feature made me want to come back each day to maintain my position" "The daily streaks feature encouraged me to practice consistently, and I actually looked forward to maintaining my streak. The notifications were helpful reminders without being too intrusive"*

This validates the theoretical foundation for gamification in security education, demonstrating that competitive elements can enhance learning outcomes when properly integrated with educational content.

6.5. Platform Impact and User Experience

Overall, participants reported substantial educational benefit from using the platform. As shown in Appendix C, Figure 78, 83.4% of participants reported either significant improvement or complete transformation of their cybersecurity understanding, with no participants indicating minimal learning.

Qualitative responses provided specific insights into participants' learning experiences. When asked about the most important concept they learned, participants highlighted: *"That urgency and pressure tactics are almost always indicators of social engineering attacks" "The importance of checking the actual URL in the address bar rather than just looking at website design. I was surprised by how convincing some of the fake websites looked until I examined the URLs" "I learned to always verify suspicious requests through official channels rather than responding directly to the message"*

These responses align with the quantitative data showing strongest mastery in urgent language tactics (4.50/5.00) and suspicious link identification (4.00/5.00), demonstrating effective integration between theoretical learning and practical application.

Participants provided comprehensive feedback on the platform's interface and user experience. Positive responses highlighted design elements: "I appreciated how the platform showed detailed explanations after each question rather than just right/wrong feedback. The explanations really helped me understand why certain elements were suspicious" "The visual design was engaging and professional. The security-themed interface made learning about cybersecurity feel more immersive, and I appreciated how clean and intuitive the navigation was" "I liked how when I got an answer incorrect, the AI feature gave me more detailed information about the concepts surrounding the question. It really helped me to think about my answers next time round"

6.6. Implementation Challenges and Solutions

Several technical challenges required innovative solutions. The LLM integration produced inconsistent educational feedback during early testing phases. This was resolved through structured JSON response formatting and context-aware prompt templates, which led to the success of the 37 service tests passing. Mobile responsiveness testing highlighted challenges with detection precision in phishing simulations, an issue specifically noted by test participants that remains unresolved in the current implementation.

User feedback also identified areas for improvement: "Make the advanced email

phishing questions easier to interact with - it was quite hard to select the different answers precisely. The clickable areas for identifying suspicious elements were too small on my screen" "While the toggle feature was good depending on the time of day I was using it, the light-mode was too harsh on my eyes, also it didn't really work across all the pages as I thought it would have" "Include a search function to quickly find specific security topics or review past learning materials. When I wanted to revisit a particular concept, it was difficult to locate the specific content"

6.7. Critical Limitations

The most critical limitation was the absence of public hosting capabilities. Without deployment to platforms like Heroku, user testing relied on locally deployed instances that restricted access and limited engagement analysis. This constraint directly affected the evaluation of social comparison effects that theoretical research had identified as beneficial (Halm, 2015).

The evaluation involved only six participants, limiting the statistical significance of quantitative findings despite clear improvement patterns. Testing logs showed consistent engagement patterns and learning progress across all participants, but a larger sample would have provided more robust validation of these effects across diverse demographics.

User testing highlighted specific content limitations, particularly the absence of industry-standard scenarios that would enhance relevance for specialised users. While the platform successfully implemented all planned attack vectors, 33.3% of participants requested specific content customisation in their feedback responses: "Include more industry-specific phishing examples (healthcare, finance, education) since threats often target particular sectors. Also, more frequent updates to reflect current phishing trends would be valuable"

6.8. Comparative Effectiveness Analysis

When compared to existing platforms, the platform demonstrates distinct advantages. Unlike Google Phishing Quiz with its singular focus on email threats, the multi-vector approach resulted in improved threat recognition, with participants achieving 83.3% correct identification of SMS-based threats despite no prior exposure to this vector.

The platform successfully balanced engagement with educational content, overcoming trade-offs observed in existing solutions. This was evidenced by consistently high metrics throughout the testing period, with 71.4% of participants maintaining unbroken usage patterns whilst also showing improved performance across increasingly complex security scenarios.

Participants compared the platform to "Duolingo for cybersecurity", highlighting the successful integration of gamification principles with security education. This comparison suggests the platform achieved its core innovation: transforming passive security awareness into active skill building through engaging, interactive experiences.

7. Conclusion and Future Work

This project developed SecurityQuest, a gamified educational platform addressing human vulnerability to social engineering attacks. Through interactive scenarios, achievement systems, and competitive elements, the platform transformed passive awareness into active skill development.

The platform successfully addressed fundamental limitations in traditional training. Testing confirmed it converted passive awareness into interactive experience maintaining engagement, with 71.4% of participants sustaining consistent usage. Beyond theoretical knowledge, it developed practical recognition skills through realistic scenario simulation, resulting in measurable behaviour changes—100% of participants correctly identified proper responses to phishing attempts after intervention compared to 66.7% beforehand.

The LLM-powered feedback system represents significant innovation in cybersecurity education, providing contextualised explanations tailored to specific user errors. Unlike traditional training where passive information precedes testing—often leading to poor retention—this approach reverses the paradigm by delivering personalised feedback after users make mistakes. This "learn-by-doing" method allows users to attempt scenarios before receiving AI-generated feedback addressing their misunderstandings. Testing verified system effectiveness through successful integration and positive user feedback, with participants highlighting how AI explanations enhanced understanding. One user noted: "I liked how when I got an answer incorrect, the AI feature gave me more detailed information about the concepts." This just-in-time learning creates stronger connections between concepts and practical application by providing targeted

explanations when users are most receptive to corrective information.

The core innovation treats social engineering defence as skill development rather than information memorisation. This fundamental shift addresses psychological aspects of threat recognition through practical experience. Post-intervention testing revealed participants showing dramatic improvements in recognising manipulation tactics across contexts—skills traditional information-focused training fails to develop.

Several limitations suggest improvements. Public cloud hosting would enable broader accessibility and comprehensive user testing with statistically significant samples. Future development should include customisable content modules targeting specific sectors with industry-relevant scenarios, addressing common user feedback. Enhanced social features and administrative capabilities would extend deployment functionality, while full mobile optimisation would address identified usability issues.

This project demonstrates that effective cybersecurity education must address psychological foundations of social engineering rather than focusing solely on technical defences. By transforming passive awareness into interactive skill development through gamification, the platform achieved significant improvements in both knowledge and behaviours. As social engineering continues evolving as primary attack vector, educational approaches developing practical recognition skills through engaging interactions remain crucial for creating resilient defensive behaviours.

References

Alkhali, Z., Hewage, C., Nawaf, L. and Khan, I. (2021). *'Phishing Attacks: A Recent Comprehensive Study and a New Anatomy'*. Available at: *https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2021.563060/full*

Barney, N. (2023). *What is gamification? How it works and how to use it*. Available at: *https://www.techtarget.com/searchhrsoftware/definition/gamification*

Deterding, S., Dixon, D., Khaled, R. and Nacke, L. (2011). *From game design elements to gamefulness: defining "gamification"*. Available at: *https://dl.acm.org/doi/10.1145/2181037.2181040*

Dodds, K. C. (2018). *Prop Drilling*. Available at: *https://kentcdodds.com/blog/prop-drilling*

Easterly, J. and Fanning, T. (2023). *The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years*. Available at: *https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years*

ENISA (2023). *ENISA Threat Landscape 2023*. Available at: *https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023*

Federal Bureau of Investigation (2023). *Internet Crime Report*. Available at: *https://www.ic3.gov/AnnualReport/Reports/2023_IC3Report.pdf*

Federal Trade Commission (2023). *Consumer Sentinel Network 2024*. Available at: *https://www.ftc.gov/system/files/ftc_gov/pdf/CSN-Annual-Data-Book-2023.pdf*

Google Books (2014). *Gamification and Education: A Literature Review*. Available at: *https://books.google.co.uk/books?hl=en&lr=&id=IedEBQAAQBAJ&oi=fnd&pg=PA50&dq=gamification+in+education&ots=bHWm1X_IYY&sig=kcTiRwTTQNdAq4chluhU0fthsT0&redir_esc=y#v=onepage&q=gamification%20in%20education&f=false*

Gragg, D. (2002). *A Multi-Level Defense Against Social Engineering*. Available at: *http://taupe.free.fr/book/psycho/social%20engineering/Social%20Engineering%20-%20Sans%20Institute%20-%20Multi%20Level%20Defense%20Against%20Social%20Engineering.pdf*

Halm, D. (2015). *The Impact of Engagement on Student Learning*. Available at: *https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=a1ba4482f44cc16ac81e1c4cdbb6211c*

Hijji, M. and Alam, G. (2021). *A Multivocal Literature Review on Growing Social Engineering Based Cyber-Attacks/Threats During the COVID-19 Pandemic: Chal-*

lenges and Prospective Solutions. Available at: *https://ieeexplore.ieee.org/document/9312039*

IBM (2024). *What is Social Engineering?*. Available at: *https://www.ibm.com/topics/social-engineering*

Jigsaw (2024). *Phishing Training Simulation. Google Security Training Platform*. Available at: *https://phishingquiz.withgoogle.com*

Lenaerts-Bergmans, B. (2023). *10 Types of Social Engineering Attacks and how to prevent them*. Available at: *https://www.crowdstrike.com/en-us/cybersecurity-101/social-engineering/types-of-social-engineering-attacks/*

Mirza, S, (2024). *Progress Report: Educating Humans on Their risk of Being Socially Engineered*. University of East Anglia

NCSC (2020). *What is cybersecurity?*. Available at: *https://www.ncsc.gov.uk/section/about-ncsc/what-is-cyber-security*

Nielson, J (2020). *10 Usability Heuristics for User Interface Design*. Available at: *https://www.nngroup.com/articles/ten-usability-heuristics/*

Verizon Enterprise Solutions (2024). *2024 Data Breach Investigations Report*. Available at: *https://www.verizon.com/business/resources/T6d1/reports/2024-dbir-data-breach-investigations-report.pdf*

Zhang, Y. and Muhammad (2024). *Integrating Aesthetic Theory into the DesignofImmersive Exhibitions for Data Imaging*. Available at: *https://www.jisem-journal.com/download/integrating-aesthetic-theory-into-the-design-of-immersive-exhibitions-for-data-imaging-15203.pdf*

A. System Design and Architecture

A.1. Learning Engagement Loop - SecurityQuest

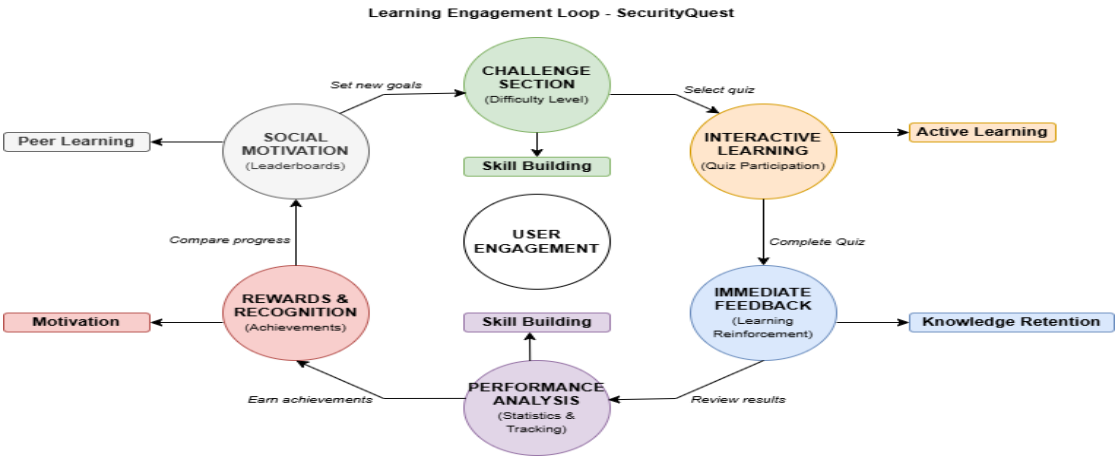


Figure 1: Learning engagement loop demonstrating the cyclical relationship between the various components in SecurityQuest

A.2. System Architecture - SecurityQuest

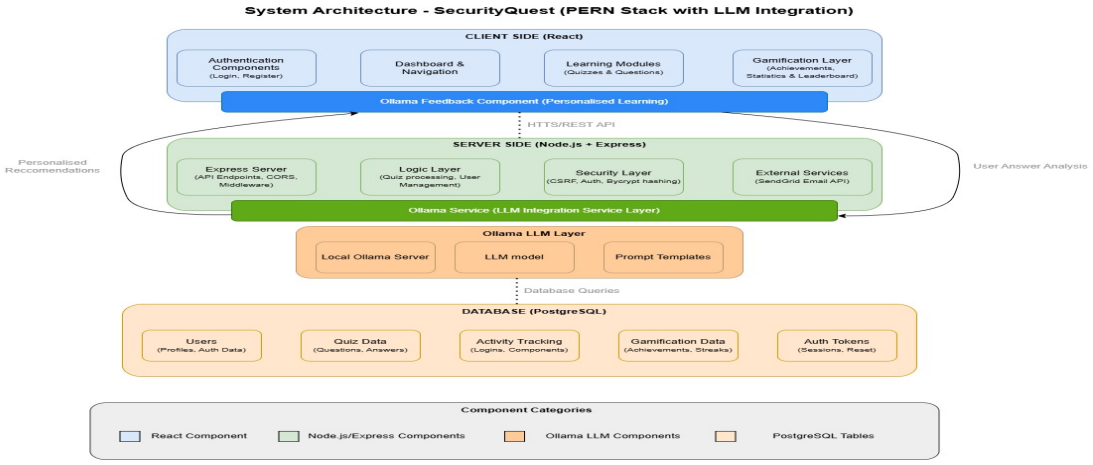


Figure 2: Three-tier system architecture showing the separation between client-side React components, server-side Node.js/Express implementation, and PostgreSQL database

A.3. Entity Relationship Model - SecurityQuest

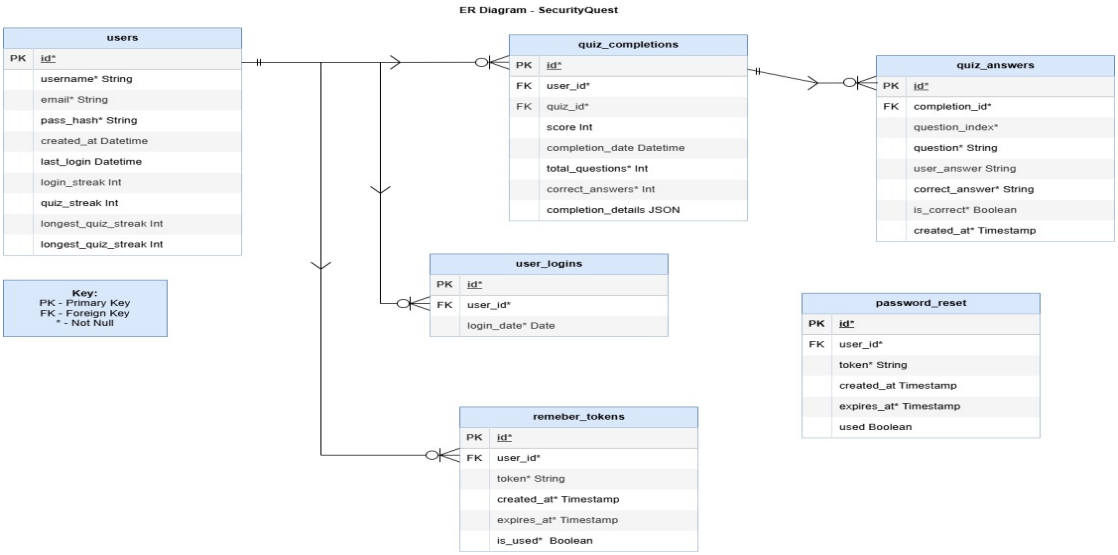


Figure 3: Database schema showing the relationships between user data, quiz completions, answer tracking, and authentication components

A.4. Gamification Framework - SecurityQuest

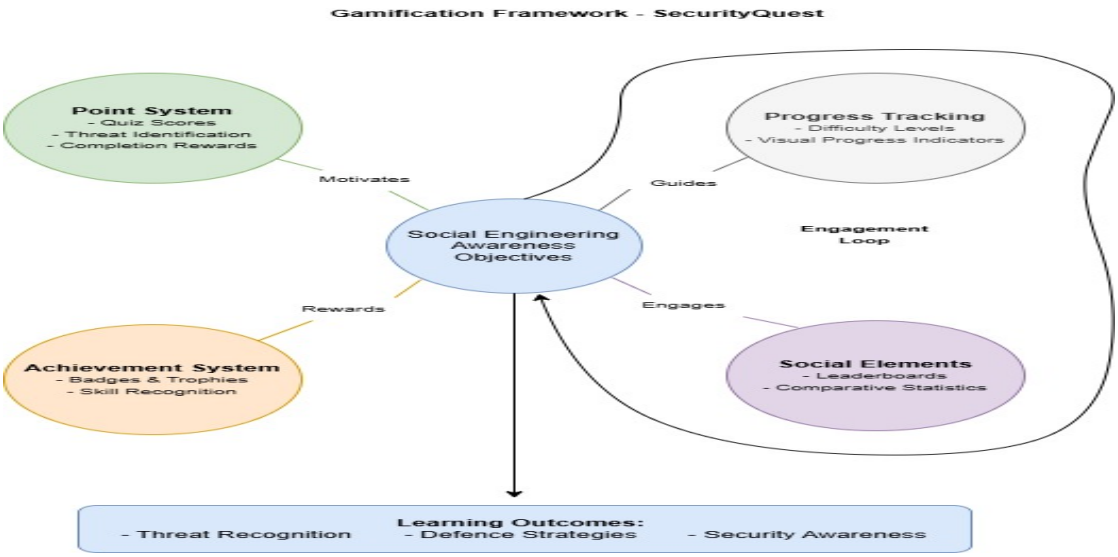


Figure 4: Conceptual framework illustrating how point systems, progress tracking, achievement systems, and social elements that support the main objectives

A.5. Quiz Progression Model - SecurityQuest

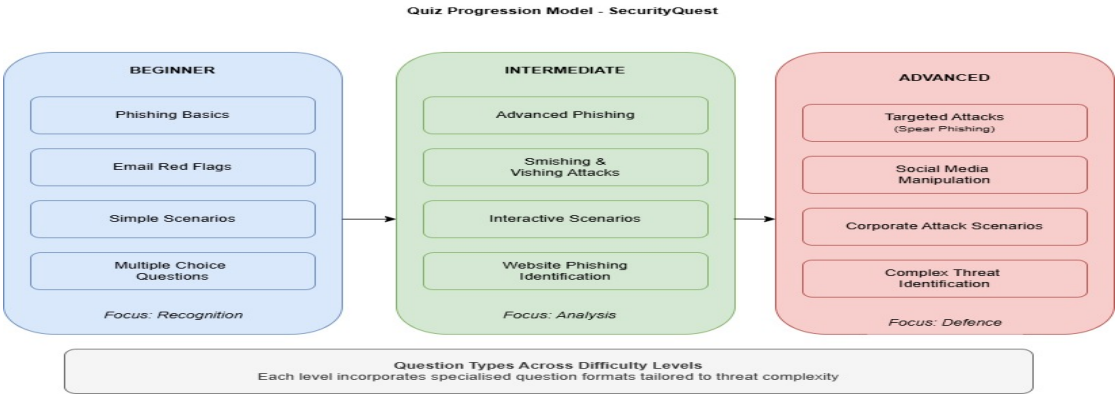


Figure 5: Educational content structure showing progression from beginner to advanced levels with corresponding focus areas and question formats

B. Appendix B: User Interface Design and Research Materials

B.1. Application Wireframes

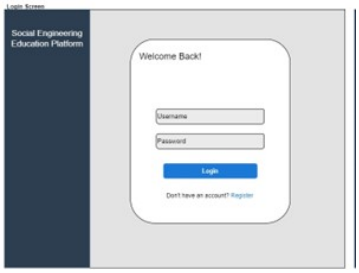


Figure 6: Login Interface

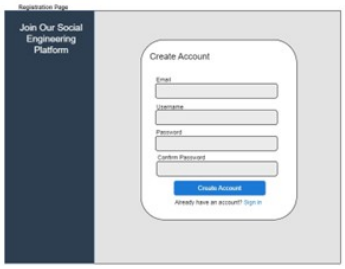


Figure 7: Registration Interface

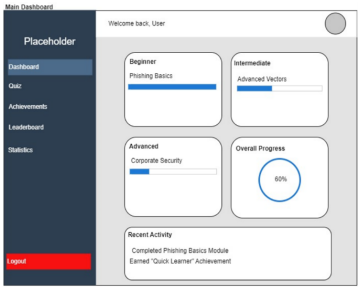


Figure 8: Main Dashboard

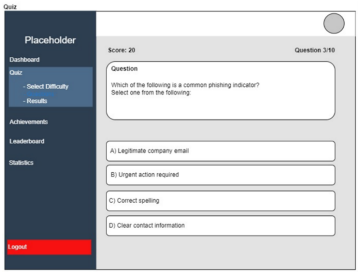


Figure 9: Quiz Interface

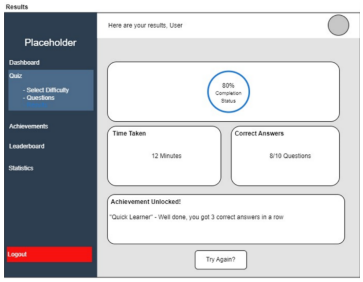


Figure 10: Results Interface

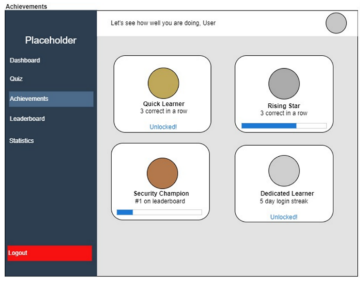


Figure 11: Achievements Interface

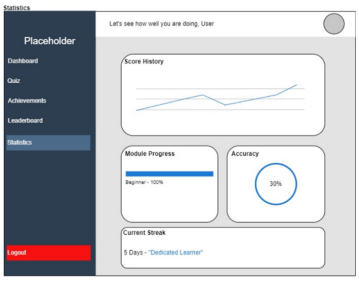


Figure 12: Statistics Interface

B.2. Research Documentation

Mr Somaan Mirza
Student - Main Researcher

13 March 2025

Faculty of Science
School of Computing Sciences

University of East Anglia
Norwich Research Park
Norwich NR4 7TJ
United Kingdom

Email: Somaan.Mirza@uea.ac.uk
Tel: 07443875073
Web: www.uea.ac.uk

Educating Humans on Their Risk of Being Socially Engineered

PARTICIPANT INFORMATION SHEET

(1) What is this study about?
You are invited to take part in a research study about improving cybersecurity awareness through interactive learning. This study will evaluate a gamified educational platform designed to help recognise and defend against social engineering attacks - a common cyber threat that targets human psychology rather than technical vulnerabilities You have been invited to participate in this study because Because we are seeking individuals who use technology in their daily lives. Your experience with regular internet and device usage makes you an ideal participant to help us understand how effective the platform can teach cybersecurity awareness. Your feedback will be valuable in developing better security education tools, regardless of your current technical background. This Participant Information Sheet tells you about the research study. Knowing what is involved will help you decide if you want to take part in the study. Please read this sheet carefully and ask questions about anything that you don't understand or want to know more about.

Participation in this research study is voluntary. By giving consent to take part in this study you are telling us that you:

- ✓ Understand what you have read.
- ✓ Agree to take part in the research study as outlined below.
- ✓ Agree to the use of your personal information as described.
- ✓ You have received a copy of this Participant Information Sheet to keep.

(2) Who is running the study?
The study is being carried out by the following student - Mr Somaan Mirza - who is conducting this study as the basis for the degree of Business Information Systems at the University of East Anglia.
This will take place under the supervision of Dr Riaz Ahmed Shaikh (riaz-ahmed.shaikh@uea.ac.uk).

Figure 13: Ethics Documentation

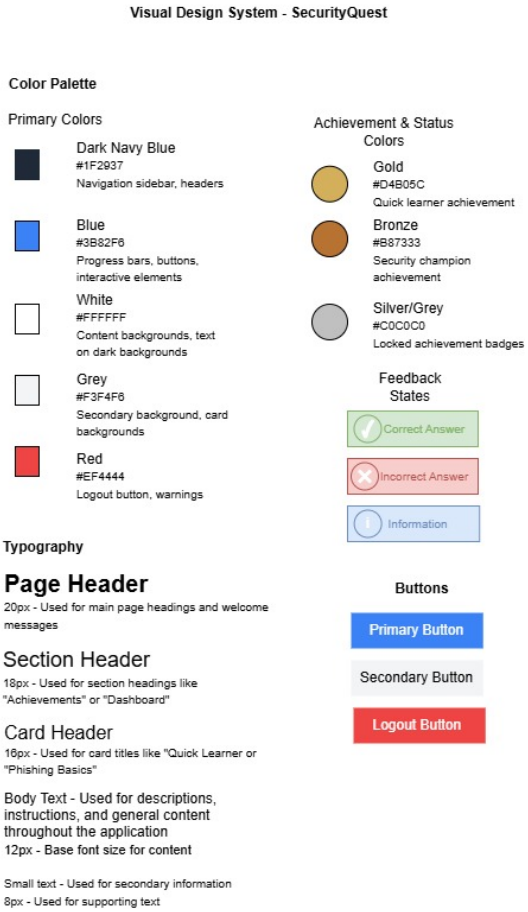


Figure 14: UI Design

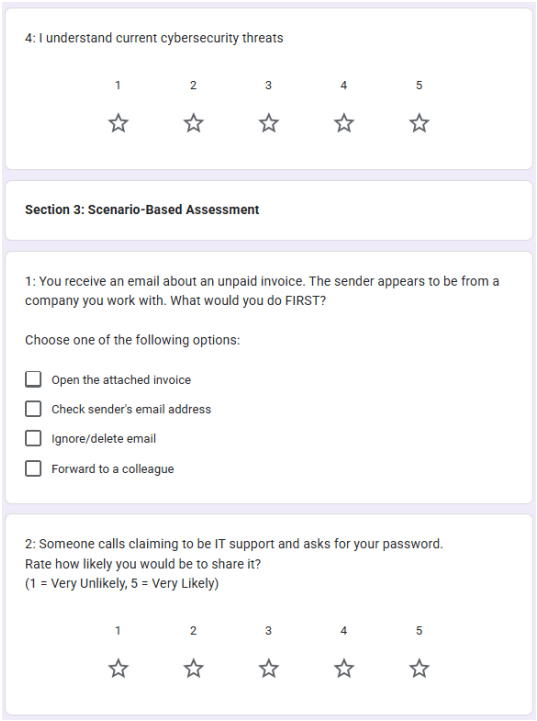


Figure 15: Pre-survey

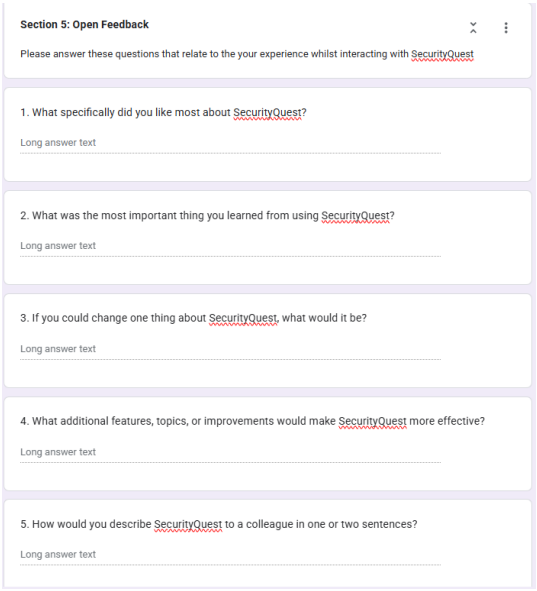


Figure 16: Post-survey

C. Appendix C: Code Snippets and Final UX/UI

C.1. Component Architecture

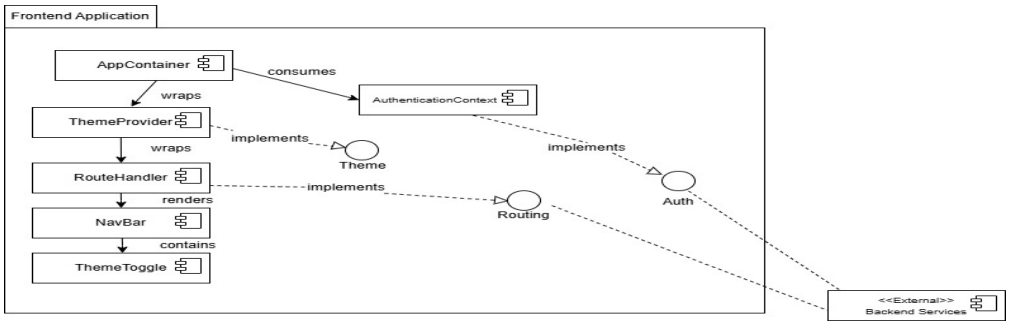


Figure 16: Component Architecture Diagram

C.2. Authentication Components

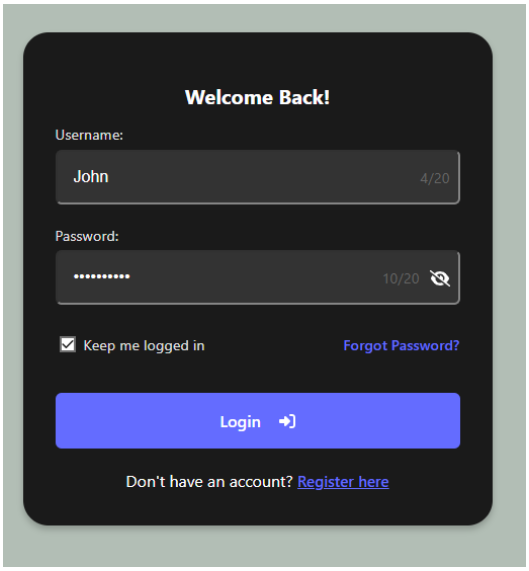


Figure 17: Login interface

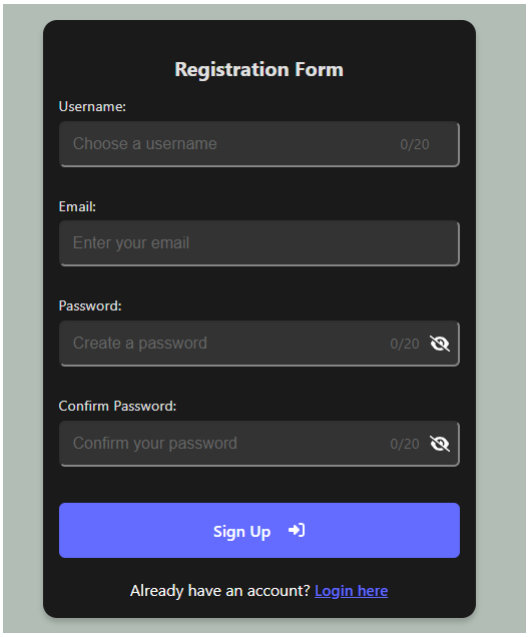
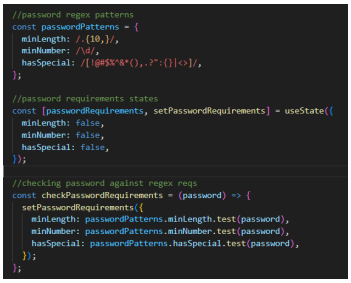


Figure 18: Registration interface



```
//password regex patterns
const passwordPatterns = {
  minLength: /.{10,}/,
  minNumber: /\d/,
  hasSpecial: /[!@#$%^&*(),.?":{}|<>]/,
};

//password requirements states
const [passwordRequirements, setPasswordRequirements] = useState({
  minLength: false,
  minNumber: false,
  hasSpecial: false,
});

//checking password against regex reqs
const checkPasswordRequirements = (password) => {
  setPasswordRequirements({
    minLength: passwordPatterns.minLength.test(password),
    minNumber: passwordPatterns.minNumber.test(password),
    hasSpecial: passwordPatterns.hasSpecial.test(password),
  });
};
```

Figure 19: Password validation

Figure 20: CAPTCHA verification

```
const handleSubmit = async (e) => {
  e.preventDefault();

  if (!formData.username || !formData.password) {
    toast.error("Please fill in all fields");
    return;
  }

  setIsLoading(true);

  try {
    // API call with credentials
    console.log("Attempting login with:", {
      username: formData.username,
      remember_me: rememberMe,
    });
    const response = await fetch("http://localhost:5000/api/login", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
      },
      credentials: "include",
      body: JSON.stringify({
        username: formData.username,
        password: formData.password,
        remember_me: rememberMe,
      }),
    });

    // Checks response before setting timeout
    const data = await response.json();
    console.log("server response:", data);

    if (!response.ok) {
      throw new Error(data.error || "Login Failed");
    }
```

Figure 21: Authentication logic

```
const handleVerify = (e) => {
  e.preventDefault();

  const currentCorrectCells = correctCellsMap[currentImageIndex];
  const isCorrect =
    selectedCells.size === currentCorrectCells.size &&
    [...selectedCells].every((cell) => currentCorrectCells.has(cell));

  if (isCorrect) {
    // Set authentication in session storage
    sessionStorage.setItem("isAuthenticated", "true");
    sessionStorage.setItem(
      "username",
      sessionStorage.getItem("username") || "User"
    );

    // Redirect to dashboard immediately
    window.location.href = ROUTES.DASHBOARD;

    // Also call the onSuccess callback if needed
    if (onSuccess) onSuccess();
  } else {
    toast.error("Incorrect selection. Please try again.", {
      position: "top-center",
      autoClose: 500,
      hideProgressBar: false,
      closeOnClick: true,
      pauseOnHover: true,
      draggable: true,
      theme: "dark",
    });
    setSelectedCells(new Set());
    getRandomImage(); // new image on incorrect instance
  }
};
```

Figure 22: CAPTCHA verification logic

C.3. Dashboard and Progress Components

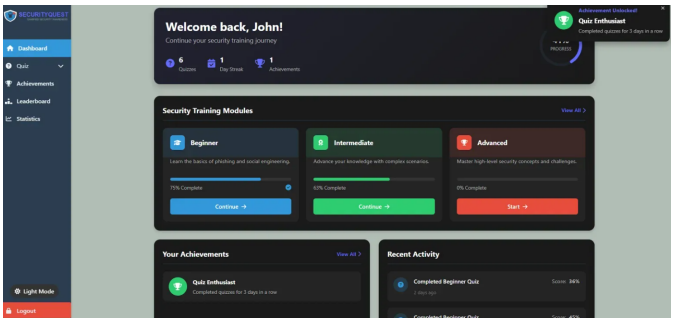


Figure 23: Dashboard interface

```
// Calculate overall progress based on completed quizzes and achievements
const calculateProgress = () => {
  if (!userData) return 0;

  // Calculate based on module progress
  const beginnerProgress = calculateModuleProgress("beginner");
  const intermediateProgress = calculateModuleProgress("intermediate");
  const advancedProgress = calculateModuleProgress("advanced");

  // Weight the progress (beginner: 30%, intermediate: 30%, advanced: 40%)
  const weightedProgress =
    beginnerProgress * 0.3 +
    intermediateProgress * 0.3 +
    advancedProgress * 0.4;

  return Math.round(weightedProgress) || 0;
};
```

Figure 24: Progress calculation

```
// Difficulty info
const difficultyInfo = [
  {
    id: 1,
    name: "Beginner",
    progress: getBeginner(),
    icon: faGraduationCap,
    color: "#3498db",
    description: "Learn the basics of phishing and social engineering.",
  },
  {
    id: 2,
    name: "Intermediate",
    progress: getIntermediate(),
    icon: faAward,
    color: "#2ecc71",
    description: "Advance your knowledge with complex scenarios.",
  },
  {
    id: 3,
    name: "Advanced",
    progress: getAdvanced(),
    icon: faTrophy,
    color: "#e74c3c",
    description: "Master high-level security concepts and challenges.",
  },
];
```

Figure 25: Module configuration

```
// Check for new achievements - simplified to only handle top notification
const checkForNewAchievements = async () => {
  try {
    console.log("Checking for new achievements...");

    // Check for streak-based achievements
    const streakAchievements =
      await AchievementService.checkStreakAchievements(userId);
    console.log("Found streak achievements:", streakAchievements);

    // Check for new achievements from API
    const newAchievements = await AchievementService.checkForNewAchievements(
      userId
    );
    console.log("New achievements from API:", newAchievements);

    // Don't set notifications here - let the main useEffect handle it
    return newAchievements;
  } catch (err) {
    console.error("Error checking achievements:", err);
    return [];
  }
};
```

Figure 26: Achievement tracking

```
const handleCaptchaSuccess = async () => {
  // Store user information in session storage
  sessionStorage.setItem("isAuthenticated", "true");
  sessionStorage.setItem("username", formData.username);

  let userId = "1";
  if (loginData && loginData.user && loginData.user.id) {
    userId = loginData.user.id.toString();
  } else {
    console.warn("UserId not found in login response, using default");
  }

  sessionStorage.setItem("userId", userId);

    if (streakResponse.ok) {
      const streakData = await streakResponse.json();
      console.log("Login streak updated via API:", streakData.login_streak);
      streakUpdated = true;
    }
  } catch (apiError) {
    console.warn("API streak update failed:", apiError);
  }

  // If API fails, update locally for demo purposes
  if (!streakUpdated) {
    const currentStreak =
      parseInt(localStorage.getItem(`login_streak_${userId}`)) || 0;
    const newStreak = currentStreak + 1;
    localStorage.setItem(`login_streak_${userId}`, newStreak);
    console.log("Login streak updated locally:", newStreak);
  }

  // Check for achievements after login
  console.log("Checking for achievements after login");
  const achievements = await AchievementService.checkStreakAchievements(userId);

  if (achievements && achievements.length > 0) {
    console.log("New achievements unlocked:", achievements);
    achievements.forEach((achievement) => {
      // Queue achievements for notification
      AchievementService.queueAchievement(achievement);
    });
  }
} catch (error) {
  console.error("Error checking achievements:", error);
```

Figure 27: Achievement integration

C.4. Leaderboard Components

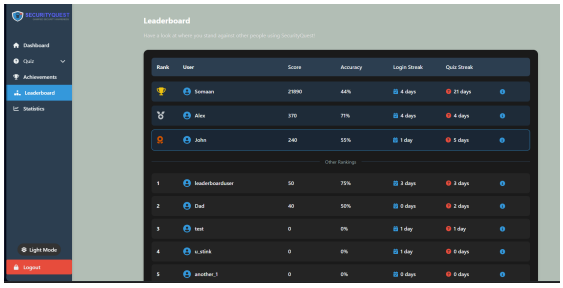


Figure 28: Leaderboard interface

```jsx
// Icons based on ranking position
const getRankIcon = (position) => {
  switch (position) {
    case 1:
      return <FontAwesomeIcon icon={faTrophy} className="rank-icon gold" />;
    case 2:
      return <FontAwesomeIcon icon={faMedal} className="rank-icon silver" />;
    case 3:
      return <FontAwesomeIcon icon={faAward} className="rank-icon bronze" />;
    default:
      return <span className="rank-number">{position}</span>;
  }
};
```

Figure 29: Ranking visualisation

```jsx
function normaliseScore(quiz) {
  if (quiz.earnedPoints !== undefined && quiz.totalPoints !== undefined) {
    return Math.min(100, (quiz.earnedPoints / quiz.totalPoints) * 100);
  } else {
    return Math.min(100, quiz.score);
  }
}

function getBeginner() {
  const beginnerQuizzes = quizHistory.filter((q) => q.quiz_id === 1);
  if (beginnerQuizzes.length === 0) return 0;

  const normalizedScores = beginnerQuizzes.map((quiz) =>
    normaliseScore(quiz)
  );
  return Math.max(...normalisedScores);
}

function getIntermediate() {
  const intermediateQuizzes = quizHistory.filter((q) => q.quiz_id === 2);
  if (intermediateQuizzes.length === 0) return 0;

  const normalisedScores = intermediateQuizzes.map((quiz) =>
    normaliseScore(quiz)
  );
  return Math.max(...normalisedScores);
}

function getAdvanced() {
  const advancedQuizzes = quizHistory.filter((q) => q.quiz_id === 3);
  if (advancedQuizzes.length === 0) return 0;

  const normalisedScores = advancedQuizzes.map((quiz) =>
    normaliseScore(quiz)
  );
  return Math.max(...normalisedScores);
}
```

Figure 30: Cross-quiz scoring

```jsx
// Format relative time (e.g., "2 days ago")
const formatTimeAgo = (timestamp) => {
  if (!timestamp) return "Never";

  const date = new Date(timestamp);
  const now = new Date();
  const diffTime = Math.abs(now - date);
  const diffDays = Math.floor(diffTime / (1000 * 60 * 60 * 24));

  if (diffDays === 0) {
    return "Today";
  } else if (diffDays === 1) {
    return "Yesterday";
  } else if (diffDays < 7) {
    return `${diffDays} days ago`;
  } else if (diffDays < 30) {
    const weeks = Math.floor(diffDays / 7);
    return `${weeks} ${weeks === 1 ? "week" : "weeks"} ago`;
  } else {
    return date.toLocaleDateString();
  }
};
```

Figure 31: Streak formatting

```jsx
{/* Scoring information card */}
<div className="streak-info-card">
  <h3>
    <FontAwesomeIcon icon={faInfoCircle} /> Scoring Information
  </h3>
  <p>The leaderboard displays two types of streaks:</p>
  <ul>
    <li>
      <strong>Login Streak:</strong> Increases each day you log in,
      resets after 24 hours of inactivity
    </li>
    <li>
      <strong>Quiz Streak:</strong> Increases each day you complete at
      least one quiz, resets after 24 hours of inactivity
    </li>
  </ul>
  <p>Your score is based on:</p>
  <ul>
    <li>Every correct question is worth 10 points</li>
    <li>
      Your score is the sum of all points earned across all quiz
      completions
    </li>
    <li>Accuracy shows your average score across all quizzes</li>
  </ul>
</div>
```

Figure 32: Scoring explanation

C.5. Statistics Components

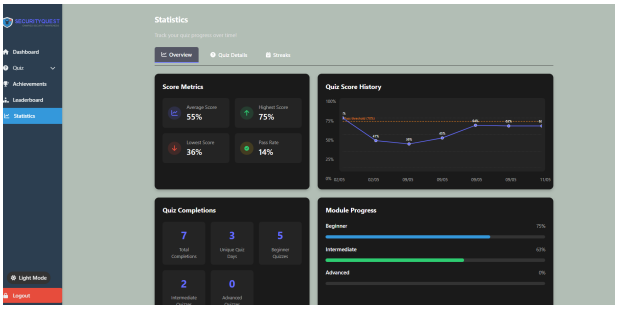


Figure 33: Statistics dashboard

Figure 34: Analytics navigation

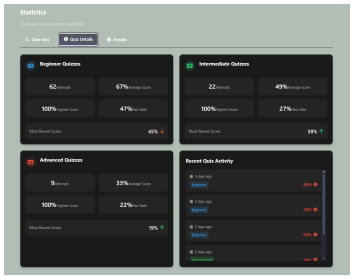


Figure 35: Analytics detail view

Figure 36: Metrics algorithm

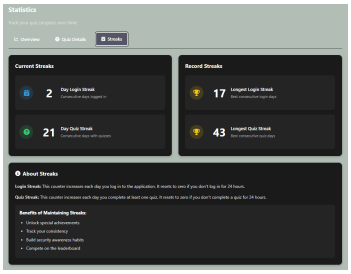


Figure 37: Consistency tracking

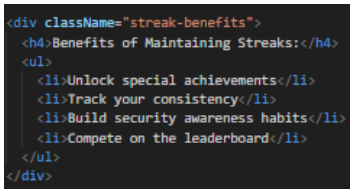


Figure 38: Habit consistency

Figure 39: Trend visualisation

C.6. Achievement System Components

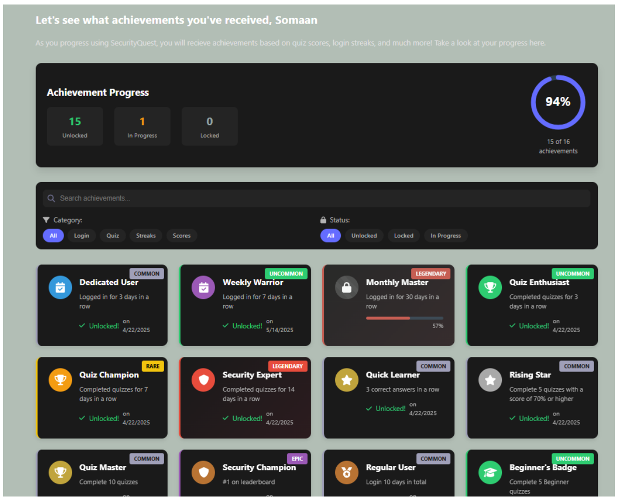

Let's see what achievements you've received, Somaan
As you progress using SecurityQuest, you will recieve achievements based on quiz scores, login streaks, and much more! Take a look at your progress here.
Achievement Progress
15
Unlocked
1
In Progress
0
Locked
94%
15 of 16 achievements
Search achievements...
Category:
All
Login
Quiz
Streaks
Scores
Status:
All
Unlocked
Locked
In Progress
COMMON
Dedicated User
Logged in for 3 days in a row
Unlocked! on 4/22/2025
UNCOMMON
Weekly Warrior
Logged in for 7 days in a row
Unlocked! on 5/14/2025
LEGENDARY
Monthly Master
Logged in for 30 days in a row
57%
UNCOMMON
Quiz Enthusiast
Completed quizzes for 3 days in a row
Unlocked! on 4/22/2025
RARE
Quiz Champion
Completed quizzes for 7 days in a row
Unlocked! on 4/22/2025
LEGENDARY
Security Expert
Completed quizzes for 14 days in a row
Unlocked! on 4/22/2025
COMMON
Quick Learner
3 correct answers in a row
Unlocked! on 4/22/2025
COMMON
Rising Star
Complete 5 quizzes with a score of 70% or higher
Unlocked! on 4/22/2025
COMMON
Quiz Master
Complete 10 quizzes
EPIC
Security Champion
#1 on leaderboard
COMMON
Regular User
Login 10 days in total
UNCOMMON
Beginner's Badge
Complete 5 Beginner quizzes

Figure 40: Achievement dashboard


const mapAchievementsForDisplay = (apiAchievements) => {
// Make sure we're dealing with an array
if (!Array.isArray(apiAchievements)) {
console.error(
"Expected achievements to be an array, got:",
typeof apiAchievements
);
return [];
}
// Set rarity and tier based on title keywords
if (title === "Security Expert" || title === "Monthly Master") {
rarity = "legendary";
tier = 4;
} else if (title === "Security Champion" || title === "Perfect Score") {
rarity = "epic";
tier = 3;
} else if (
title === "Quiz Champion" ||
title === "Email Vigilance" ||
title === "Advanced Defender"
) {
rarity = "rare";
tier = 3;
} else if (
title === "Weekly Warrior" ||
title === "Quiz Enthusiast" ||
title === "Intermediate Guardian" ||
title === "Beginner's Badge"
) {
rarity = "uncommon";
tier = 2;
}

Figure 41: Rarity categorisation

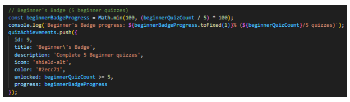

// Beginner's Badge (5 beginner quizzes)
const beginnerBadgeProgress = Math.min(100, (beginnerQuizCount / 5) * 100);
console.log(`Beginner's Badge progress: ${beginnerBadgeProgress.toFixed(1)}% (${beginnerQuizCount}/5 quizzes)`);
quizAchievements.push({
id: 9,
title: 'Beginner\'s Badge',
description: 'Complete 5 Beginner quizzes',
icon: 'shield-alt',
color: '#2ecc71',
unlocked: beginnerQuizCount >= 5,
progress: beginnerBadgeProgress
});

Figure 42: Progress tracking


static async checkStreakAchievements(userId) {
try {
// Try to fetch streak data from API first
let userData = {};
let apiSuccess = false;
try {
console.log(`Fetching streak data from API for user ${userId}...`);
const response = await fetch(API_ENDPOINTS.GET_USER_STREAKS.replace(':userId', userId));
if (response.ok) {
const data = await response.json();
userData = data.userData || {};
apiSuccess = true;
console.log('Successfully fetched streak data from API');
console.log('Login streak:', userData.login_streak);
console.log('Longest login streak:', userData.longest_login_streak);
console.log('Quiz streak:', userData.quiz_streak);
console.log('Longest quiz streak:', userData.longest_quiz_streak);
// If API provides a calculated streak, use it for verification
if (userData.calculatedStreak !== undefined) {
console.log(`API calculated streak: ${userData.calculatedStreak}`);
console.log(`Database streak: ${userData.login_streak}`);
if (userData.calculatedStreak !== userData.login_streak) {
console.warn(`Streak mismatch detected! API calculated ${userData.calculatedStreak} but database has ${userData.login_streak}`);
}
}
} else {
throw new Error('Failed to fetch user streaks from API');
}
} catch (error) {
console.warn('Using local streak storage as fallback:', error);
// Fallback to localStorage if API fails
const loginStreak = localStorage.getItem(`login_streak_${userId}`) || 0;
const quizStreak = localStorage.getItem(`quiz_streak_${userId}`) || 0;

Figure 43: API persistence

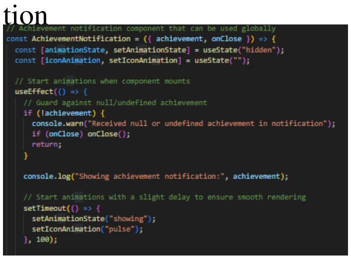

// Achievement notification component that can be used globally
const AchievementNotification = ({ achievement, onClose }) => {
const [animationState, setAnimationState] = useState("hidden");
const [iconAnimation, setIconAnimation] = useState("");
// Start animations when component mounts
useEffect(() => {
// Guard against null/undefined achievement
if (!achievement) {
console.warn("Received null or undefined achievement in notification");
if (onClose) onClose();
return;
}
console.log("Showing achievement notification:", achievement);
// Start animations with a slight delay to ensure smooth rendering
setTimeout(() => {
setAnimationState("showing");
setIconAnimation("pulse");
}, 100);

Figure 44: Animated feedback

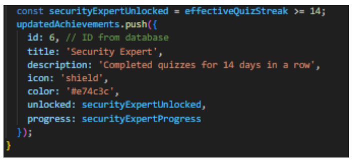

const securityExpertUnlocked = effectiveQuizStreak >= 14;
updatedAchievements.push({
id: 6, // ID from database
title: 'Security Expert',
description: 'Completed quizzes for 14 days in a row',
icon: 'shield',
color: '#e74c3c',
unlocked: securityExpertUnlocked,
progress: securityExpertProgress
});
}

Figure 45: Unlock conditions

C.7. Quiz Component System

```jsx
const Quiz = () => {
  // Get current location to determine if we're on the results page
  const location = useLocation();
  const isResultsPage = location.pathname.includes("/results");

  return (
    // Apply a class to the container based on whether we're in results mode
    <div
      className={`quiz-route-container ${isResultsPage ? "results-mode" : ""}`}
    >
      <Routes>
        <Route path="/" element={<Navigate to="difficulty" replace />} />
        <Route path="difficulty" element={<QuizDifficulty />} />
        <Route path="questions" element={<QuizQuestions />} />
        <Route path="results" element={<QuizResults />} />
      </Routes>
```

Figure 46: Quiz router

```jsx
// Render different question types based on the type property
const renderQuestion = () => {
  const questionType =
    currentQuestion.type || QUIZ_CONFIG.QUESTION_TYPES.MULTIPLE_CHOICE;

  switch (questionType) {
    case QUIZ_CONFIG.QUESTION_TYPES.EMAIL_PHISHING:
      return (
        <EmailPhishingQuestion
          question={currentQuestion}
          onAnswer={handleSpecialQuestionAnswer}
        />
      );

    case QUIZ_CONFIG.QUESTION_TYPES.VISHING:
      return (
        <VishingQuestion
          question={currentQuestion}
          onAnswer={handleSpecialQuestionAnswer}
        />
      );

    case QUIZ_CONFIG.QUESTION_TYPES.SMISHING:
      return (
        <SmishingQuestion
          question={currentQuestion}
          onAnswer={handleSpecialQuestionAnswer}
        />
      );

    case QUIZ_CONFIG.QUESTION_TYPES.WEBSITE_PHISHING:
      return (
        <WebsitePhishingQuestion
          question={currentQuestion}
          onAnswer={handleSpecialQuestionAnswer}
        />
      );

    case QUIZ_CONFIG.QUESTION_TYPES.MULTIPLE_CHOICE:
    default:
```

Figure 47: Vector simulation

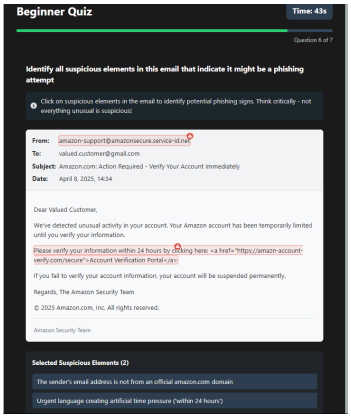


Figure 48: Phishing identification

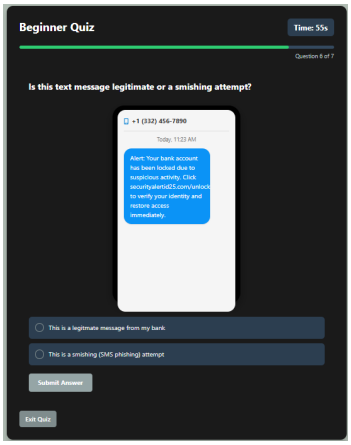


Figure 49: SMS simulation

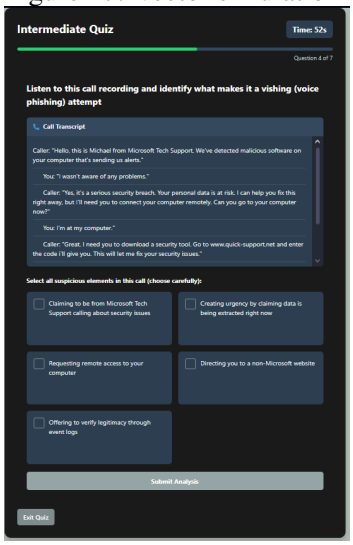


Figure 50: Voice transcript

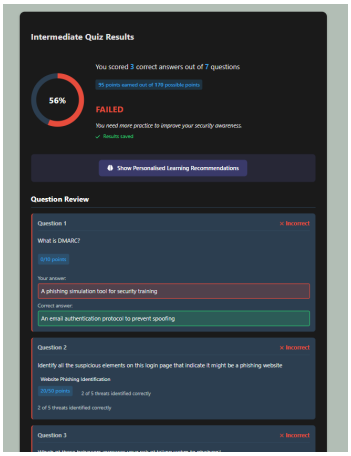


Figure 51: Analytics dashboard

C.8. UI Implementation

```css
:root {
  font-family: Inter, system-ui, Avenir, Helvetica, Arial, sans-serif;
  line-height: 1.5;
  font-weight: 400;
  color-scheme: light dark;
  color: rgba(255, 255, 255, 0.87);
  background-color: #242424;
  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
}

/* ---------- Base Styles ---------- */
a {
  font-weight: 500;
  color: #646cff;
  text-decoration: inherit;
}

a:hover {
  color: #535bf2;
}

body {
  margin: 0;
  background-color: #b2beb5;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
}

h1 {
  font-size: 3.2em;
  line-height: 1.1;
}
```

Figure 52: Typography styling

```css
button {
  border-radius: 8px;
  border: 1px solid transparent;
  padding: 0.6em 1.2em;
  font-size: 1em;
  font-weight: 500;
  font-family: inherit;
  background-color: #1a1a1a;
  cursor: pointer;
  transition: border-color 0.25s;
}

button:hover {
  border-color: #646cff;
}

button:focus,
button:focus-visible {
  outline: 4px auto -webkit-focus-ring-color;
}
```

Figure 53: Button states

```jsx
<button
  onClick={toggleTheme}
  className={`theme-toggle-btn ${isLight ? "light-toggle" : "dark-toggle"}`}
  aria-label={`Switch to ${isLight ? "dark" : "light"} mode`}
>
```

Figure 54: Theme toggle

```css
.light-theme {
  /* Main colors */
  --bg-primary: #f8f9fa;
  --bg-secondary: #ffffff;
  --bg-tertiary: #f0f4f8;
  --text-primary: #2c3e50;
  --text-secondary: #495057;
  --text-tertiary: #6c757d;
  --border-color: #dee2e6;
  --box-shadow: 0 2px 8px rgba(0, 0, 0, 0.05);
```

Figure 55: Colour contrast

```jsx
// Check for system change to lightmode
if (!savedTheme) {
  if (
    window.matchMedia &&
    window.matchMedia("(prefers-color-scheme: light)").matches
  ) {
    return "light";
  }
}

return savedTheme || "dark";
```

Figure 56: Theme detection

```css
/* ---------- Sidebar / Navigation ---------- */
.sidebar {
  width: 200px;
  background-color: #2c3e50;
  color: #fff;
  position: fixed;
  height: 100vh;
  left: 0;
  top: 0;
  display: flex;
  flex-direction: column;
  box-shadow: 2px 0 5px rgba(0, 0, 0, 0.1);
  padding: 0;
  z-index: 100;
}

.sidebar-title {
  font-size: 18px;
  font-weight: bold;
  margin-bottom: 20px;
  text-align: left;
  padding-left: 20px;
  padding-top: 10px;
}

.nav-links-box {
  display: flex;
  flex-direction: column;
  flex: 1;
}

.nav-links-box a {
  padding: 12px 20px;
  color: #ecf0f1;
  text-decoration: none;
  transition: all 0.3s ease;
}

.nav-links-box a:hover {
  background-color: #34495e;
  color: #fff;
}

.nav-links-box a.active {
  background-color: #3498db;
  color: white;
  border-left: 4px solid #2980b9;
}
```

Figure 57: Navigation states

```css
/* ---------- Layout Styles ---------- */
.app-container {
  display: flex;
  min-height: 100vh;
  position: relative;
}

.content-wrapper {
  margin-left: 200px;
  width: calc(100% - 200px);
  min-height: 100vh;
  padding: 20px;
  box-sizing: border-box;
  transition: margin-left 0.3s ease, width 0.3s ease;
  display: flex;
  justify-content: center;
  align-items: center;
}

/* Tablets (768px and below) */
@media (max-width: 768px) {
  /* Layout adjustments */
  .sidebar {
    width: 160px;
  }

  .content-wrapper {
    margin-left: 160px;
    width: calc(100% - 160px);
    padding: 1rem;
  }

  /* Card and grid adjustments */
  .dashboard-grid,
  .statistics-grid {
    grid-template-columns: 1fr;
    gap: 1rem;
  }
}

/* Mobile devices (480px and below) */
@media (max-width: 480px) {
  /* Full width layout */
  .sidebar {
    width: 0;
    transform: translateX(-100%);
    transition: transform 0.3s ease, width 0.3s ease;
    z-index: 950;
  }

  .sidebar.open {
    width: 200px;
    transform: translateX(0);
  }

  .content-wrapper {
    margin-left: 0;
    width: 100%;
    padding: 0.75rem;
    padding-top: 60px; /* Space for mobile menu button */
  }
```

Figure 58: Responsive layout

```css
.dashboard-container,
.statistics-container {
  max-width: 1200px;
  width: 100%;
  margin: 0 auto;
  padding: 2rem;
}

.dashboard-header,
.statistics-header {
  margin-bottom: 2rem;
}

/* Card and grid adjustments */
.dashboard-grid,
.statistics-grid {
  grid-template-columns: 1fr;
  gap: 1rem;
}

.dashboard-card,
.statistics-card {
  padding: 1.25rem;
}

/* Compact components */
.dashboard-container,
.statistics-container {
  padding: 0.75rem;
}

.dashboard-header h2,
.statistics-header h2 {
  font-size: 1.25rem;
  padding-left: 30px; /* Space for menu button */
}
```

Figure 59: Dashboard grid

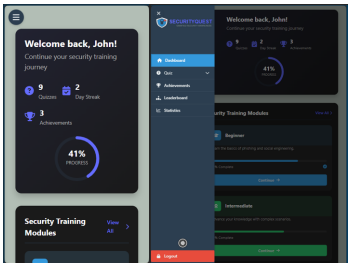


Figure 60: SecurityQuest devices

```css
/* ---------- Toast Styling ---------- */
.custom-toast {
  background-color: #1a1a1a !important;
  color: white !important;
  min-height: 100px !important;
  display: flex !important;
  flex-direction: column !important;
  justify-content: center !important;
  align-items: center !important;
  gap: 1rem !important;
  padding: 1.5rem !important;
}

.custom-toast .Toastify__toast-body {
  text-align: center !important;
  font-size: 1.1rem !important;
  margin-bottom: 1rem !important;
}

.custom-toast button {
  margin: 0 0.5rem !important;
  padding: 0.5rem 1.5rem !important;
  border-radius: 4px !important;
  cursor: pointer !important;
  transition: all 0.2s ease !important;
}

.toast-button-yes {
  background-color: #dc3545 !important;
  color: white !important;
}

.toast-button-yes:hover {
  background-color: #c82333 !important;
}

.toast-button-cancel {
  background-color: #6c757d;
  color: white;
}

.toast-button-cancel:hover {
  background-color: #5c636a;
}
```

Figure 61: Toast notifications

```css
.progress-bar {
  width: 100%;
  height: 10px;
  background-color: #333;
  border-radius: 5px;
  margin-top: 0.5rem;
}

.progress-fill {
  height: 100%;
  background-color: #646cff;
  border-radius: 5px;
  transition: width 0.3s ease;
}

.progress-circle {
  width: 150px;
  height: 150px;
  margin: 1rem auto;
  position: relative;
  background: conic-gradient(#646cff 216deg, #333 0);
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
}

.progress-circle-inner {
  width: 120px;
  height: 120px;
  background: #1a1a1a;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
}

.progress-percentage {
  color: #ffffff;
  font-size: 1.5rem;
  font-weight: bold;
}
```

Figure 62: Progress visualisation

C.9. Backend Implementation

Figure 63: CORS configuration

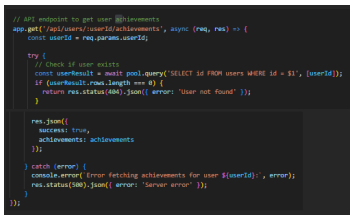


Figure 64: Error handling

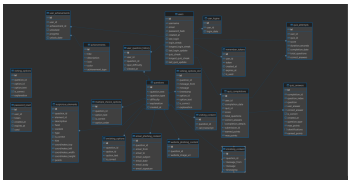


Figure 65: Database schema

Figure 66: Connection pooling

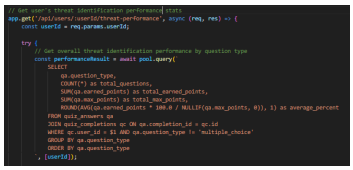


Figure 67: Metrics query

Figure 68: Password hashing

Figure 69: Registration

Figure 70: Login verification

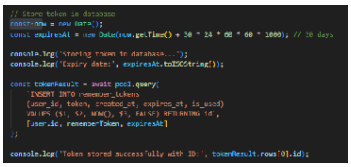


Figure 71: Session tokens

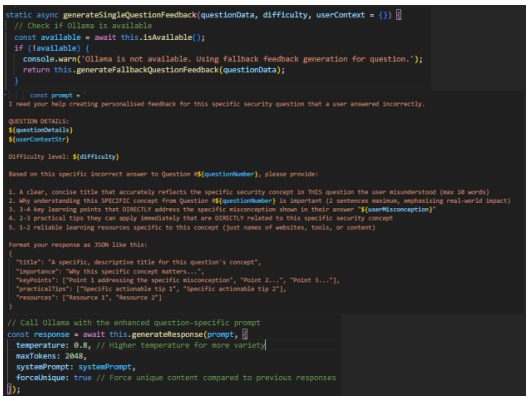


Figure 72: AI feedback service

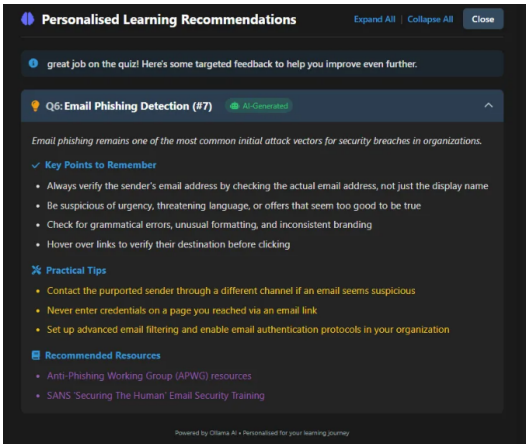


Figure 73: Recommendation interface

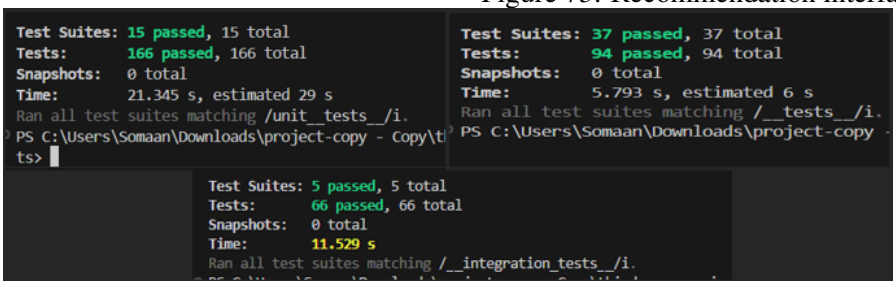


Figure 74: Unit and Integration Testing Results

C.10. Evaluation Results

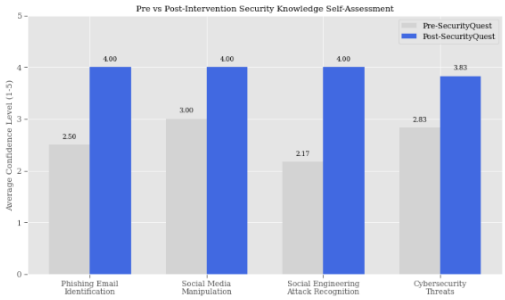


Figure 75: Pre vs Post-Intervention Security Knowledge

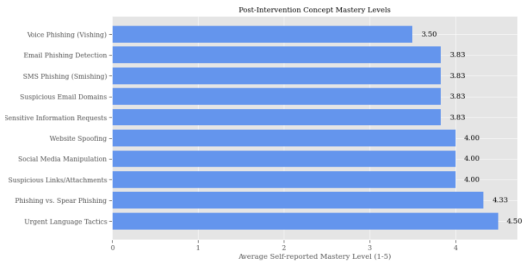


Figure 76: Post-Intervention Concept Mastery

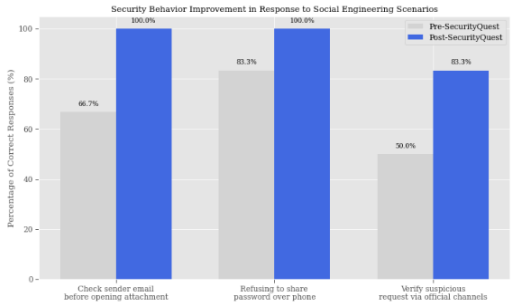


Figure 77: Security Behaviour Improvement

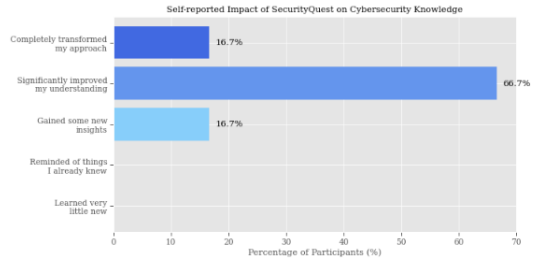


Figure 78: Self-reported Impact

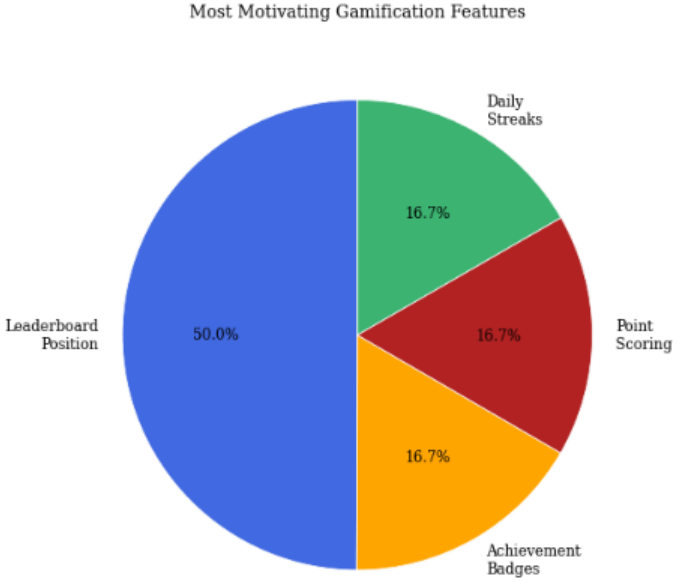


Figure 79: Most Motivating Gamification Feature