Interactive Web-based Learning Platform for Enhancing Cybersecurity Awareness in Safe Software Downloading and Installation

Rohith Reddy Allam
Department of Computer Science
Southeast Missouri State University
Cape Girardeau, Missouri, USA

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

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

ABSTRACT
This project presents the design, development, and implementation of an educational website focused on improving user awareness of cybersecurity risks associated with downloading and installing software from the internet. The platform delivers a structured sequence of ten interactive lessons that guide users through key concepts related to identifying legitimate software sources, recognizing and avoiding malware and viruses, and performing secure software installation practices. Each lesson is followed by a short quiz designed to reinforce learning and provide immediate feedback, enabling users to assess their understanding and progress in real time. The instructional framework is cumulative, with each lesson building upon prior material to progressively develop core competencies in safe software acquisition and installation. The integration of formative assessments throughout the learning process supports active engagement and enhances knowledge retention. The website employs a simple, intuitive, and user-centered interface designed for accessibility across both desktop and mobile devices. Interactive features, including quizzes, navigation controls, clickable elements, and automated feedback mechanisms, contribute to an engaging and experiential learning environment. Overall, the project demonstrates a scalable and extensible web-based educational solution that combines instructional design principles with best practices in web development. The resulting platform provides a foundation for future expansion, allowing for the addition of new topics, multimedia content, and enhanced learning modules.

Keywords
Cybersecurity awareness, educational website, interactive learning, malware prevention, secure software installation

1. INTRODUCTION
The widespread use of internet-based software distribution has increased users' exposure to cybersecurity risks, particularly during the download and installation of applications. Many users unintentionally install malicious software due to a lack of awareness regarding secure software sources, unsafe installation practices, and deceptive applications. This highlights the need for effective educational approaches that can improve user understanding of cybersecurity threats in a structured and interactive manner.

This paper presents the design, development, and implementation of an educational website aimed at improving user awareness of cybersecurity risks associated with downloading and installing software. The platform delivers a sequence of ten interactive lessons supported by quizzes and immediate feedback mechanisms to reinforce learning and assess user progress. The learning structure is cumulative, enabling gradual development of key competencies in identifying legitimate software sources, avoiding malware, and applying secure installation practices.

The proposed system integrates instructional design principles with interactive web technologies to create an accessible and engaging learning environment. By combining formative assessments with user-centered design, the platform aims to enhance knowledge retention and promote active learning. Overall, the system provides a scalable solution for cybersecurity education that can be extended with additional modules and content in future work.

2. LITERATURE REVIEW
This paper studies the security risks in application installers and online software repositories, which many users trust for downloading software. The authors analyzed around 2,900 programs from five popular repositories over a period of time and found that these platforms are very dynamic, meaning the same application can have different installer versions or change frequently. Many installers are not simple and often download additional files from the internet during installation, which increases the risk of attacks like man-in-the-middle where malicious content can be inserted. The study also shows that some installers include extra components like adware or tracking features without clearly informing users. One important finding is that about 30% of the applications were flagged as malicious by at least one antivirus, showing that even trusted repositories can distribute unsafe software. Overall, the paper highlights that downloading software is not always safe and users should be more careful instead of blindly trusting these sources [1].

The paper examines the threats to security and privacy that come from the pre-installed applications on Android devices. Such applications were not downloaded by the user, yet they are installed on the smartphone. In order to provide proof, the authors gathered a huge data set of applications from different countries' real users. More than 200 app stores from the Android ecosystem were used for analysis. It turns out that there are numerous pre-installed applications in smartphones which lack transparency and use third-party services. Among these pre-installed applications, many applications use personal data of the users for advertising purposes. These include sensitive information about location, phone numbers, contacts, device identifiers, and user's activity. Moreover, in some cases, malware and backdoors were detected. Also, pre-installed applications sometimes had unnecessary permissions which gave them access to the core system files. Furthermore, another problem with pre-installed applications is the fact that many of them are not present in Google Play, thus, they lack regular updates. Overall, this

research proves that smartphones can have applications that are unsafe even prior to installation. [2].

This paper gives an overview of smartphone security by explaining different types of vulnerabilities, malware, and attacks that affect mobile devices. It primarily deals with the ease of infection of smartphones, including those based on the Android system, due to users downloading applications without first ensuring their safety from threats such as viruses, spyware, trojans, and rootkits. These can secretly be installed in the background and may collect sensitive data about the user, such as messages, geographical position, or bank account details. Other forms of attack discussed by the authors include phishing attacks, social engineering attacks, or man-in-the-middle attacks, which could take place during app use or communication through the app. A critical aspect pointed out by the authors includes the deceptive nature of malicious apps, making it difficult for users to detect any threat. Even despite security measures in place in smartphone software, including permissions and sandboxing, it appears these are not sufficient safeguards against malware infections. The paper also proposes some preventative measures against attacks, including installing applications from reliable vendors and updating mobiles regularly [3].

This paper outlines how people use software updates and their reasons for not installing them. Updates are essential since they resolve bugs and security problems; however, many individuals still procrastinate and skip updates due to factors such as time consumption, interruption, or unforeseen occurrences. The study involved a survey in which participants described their update experiences, and the result is that software updates generally go through six phases: awareness, decision-making, preparation, installation, problem-solving, and consequences. This paper emphasizes that users have reservations about updates, including risks of data loss, performance slowdowns, bugs, or incompatible systems. Additionally, this paper reveals that many users depend on feedback or previous encounters when making update decisions. While automatic updates can improve security, they can also reduce user trust if they happen without control. Overall, the study suggests that developers should make updates clearer, less disruptive, and provide better recovery options so users feel more confident installing them [4].

This paper presents a systematic study on detecting malicious applications in both official and third-party Android app markets using a system called Droid Ranger. According to the authors, due to an increasing number of smartphone users, the availability of app stores allows downloading any application; however, this makes people more prone to download malicious software. The methods of detecting malicious applications include behavioral permission-based approach and heuristic techniques that help identify malicious software. In their research, the authors analyzed more than 200,000 mobile applications downloaded from different markets and detected more than 200 applications that can be regarded as malicious software. Thus, the results of the research prove that the existence of malicious software is not limited to alternative markets only; however, the infection level of the latter is higher than that of the official ones, which makes them less secure. Additionally, the authors discovered several instances of bypassing anti-virus systems as well as newly developed forms of malware, which means that people who use Android devices should be extremely careful when downloading any applications from unofficial markets [5].

The paper investigates the process of developing and testing Android malware via publicly available platforms such as VirusTotal. The findings suggest that attackers make use of these services for developing malware in order to evaluate whether it successfully evades antivirus detection. According to the researchers, malware developers upload several versions of their applications in order to test their effectiveness of bypassing antivirus software and further develop better evasion techniques for further distribution. In order to detect these activities, the authors have developed a system known as AMD hunter, which was able to find several indicators that help determine whether the submitted applications belong to malware development. As a result, the researchers managed to identify several hundred malware samples in various countries, most of which were previously unknown to them. However, some of the identified samples failed to be detected by antivirus engines at the point of submission [6].

This paper proposes a recommendation system for mobile apps that takes into consideration popularity as well as the possible security threats. This paper points out the fact that current recommendation systems for mobile applications take popularity as the only determinant of an app, including the number of times the application has been downloaded and rated, while ignoring the aspect of privacy and security threats, which are vital in ensuring that the users make the right decisions. In this regard, the study focuses on addressing the issue through analysis of the permissions given by the apps, as the permission can result in privacy threats. They introduce a method to calculate risk scores using an app-permission bipartite graph and regularization techniques, which helps identify how risky an app is without needing source code analysis. The system then recommends apps by balancing popularity and security using a portfolio-based approach, allowing users to choose based on their security preferences. The paper also presents an efficient framework with offline risk computation and fast online recommendations, and experiments on a large dataset from Google Play show that the approach improves both safety and recommendation quality [7].

This paper studies different personal safety apps and tries to understand if they really help users stay safe or not. The authors analyzed many apps from app stores and found that most of them mainly use features like location tracking, panic alarms, and sending alerts to contacts. However, they found that most apps only help during or after a dangerous situation, instead of actually preventing it from happening. For example, many apps send location to others only after the user is already in danger, which means it may not stop the crime. The paper also explains that these apps can sometimes give users a false sense of safety, making them believe they are protected when they are not. Very few apps provide proper guidance or education to help users avoid risky situations in the first place. Overall, the study shows that while these apps may reduce fear, they do not always reduce real risk, so users should not fully depend on them for safety [8].

This paper focuses on the problem of overprivileged permissions in Android applications and how they can lead to privacy risks for users during app installation. It explains that many apps request permissions that are not actually required for their functionality, which increases the risk of sensitive data exposure such as location, contacts, and messages. Since most users do not fully understand permission requests, they often allow all permissions without careful consideration. To address this issue, the authors propose a tool called Droidtector that uses data mining techniques to analyze permission patterns across similar categories of apps and identify unnecessary permissions. The study shows that many apps contain overprivileged permissions, indicating that this is a common problem. The paper also highlights that existing detection methods are limited due to code encryption and system updates, while the proposed approach

works without needing source code analysis. Overall, the study emphasizes that excessive permission requests are a major security concern and improving permission analysis can help make software downloading and installation safer for users [9].

This paper explains how smartphone apps can be used to collect both survey data and passive data like location, activity, and phone usage, but it also highlights important concerns about privacy and consent. The study shows that although only about 16% of invited participants installed the app, most of those who did were willing to share large amounts of personal data. Interestingly, participants did not strongly differentiate between different types of data being collected, and many gave consent to almost all data types. The research also found that very few participants changed or revoked their consent after agreeing. However, a major issue identified is that many users do not fully read or understand the consent information provided, even when detailed explanations are given. This raises questions about whether consent is truly "informed." Overall, the paper shows that while smartphone-based data collection has strong potential for research, there are still challenges in ensuring user awareness, privacy, and ethical data usage [10].

This paper provides a comprehensive review of the available techniques for identifying software supply chain attacks, concentrating specifically on malicious packages within open-source communities. The importance of third-party software packages in today's software development is emphasized, as this dependency increases the chances of introducing malicious code by means of a compromised or imposter package. The literature is reviewed in several research articles and classification of various detection techniques is made into rules and heuristics-based detection, typo-squatting detection, differential analysis, machine learning-based detection, anomaly-based detection, and clustering-based detection. The paper also discusses key challenges, including high false positive rates, lack of malicious data for proper testing, and difficulty in scaling these methods for real-world use. It also shows that no single approach is fully effective on its own, as each method has its own limitations. Overall, the paper suggests that combining multiple detection techniques is more effective and that these methods should be used to support manual review rather than completely replace it [11].

This paper studies the rise of fake antivirus (Fake AV) attacks on the web and how they trick users into paying money for fake security software. It explains that Fake AV uses social engineering by showing fake warnings that a computer is infected and then asking users to download or pay for a solution. The study analyzed a large dataset of web pages and found that Fake AV accounts for about 15% of all malware, showing that it is a growing threat. The paper also shows that these attacks spread widely through ads, spam, and popular search keywords, making them reach more users. Another important finding is that Fake AV domains change quickly and rotate frequently to avoid detection, with some lasting less than a few hours. The paper highlights that most attacks rely on user interaction rather than automatic downloads, meaning users are tricked into installing the malware themselves. Overall, the study shows that Fake AV is increasing rapidly, uses clever distribution strategies, and remains a serious security problem that requires better detection and user awareness [12].

The current research is focused on analyzing the perception of the users of computer security warnings and finding out why computer security warnings do not influence people's behavior properly. The reason for that is human behavior since it is difficult for the users to follow security recommendations, ignoring warnings because of lack of knowledge and motivation, or even improper design of these notifications. In order to explain this phenomenon, the models that will be used in this study include C-HIP and the human-in-the-loop. These models present the process of human processing of warning messages through attention, comprehension, motivation, and behavior stages. The problems related to computer security warnings also cover such topics as warning fatigue when the repeated warnings stop being effective. Moreover, mental models will be considered within this analysis demonstrating the difference in interpreting computer security warnings by users depending on their experience and leading to some wrong decisions on the part of novice users. The effectiveness of warnings will also be discussed [13].

This paper presents a comprehensive study of Android malware by analyzing a large dataset of 1,260 samples across 49 malware families to understand their behavior, evolution, and detection challenges. The study finds that a majority of malware (86%) spreads through repackaged applications, where legitimate apps are modified to include malicious payloads, highlighting a major weakness in app marketplaces. It categorizes malware based on installation methods such as repackaging, update attacks, and drive-by downloads, and explains how these techniques rely heavily on social engineering. The paper also examines activation mechanisms, showing that malware commonly uses system events like boot completion or SMS receipt to trigger malicious actions. In terms of payloads, it identifies key threats including privilege escalation using root exploits, remote control through command-and-control servers (affecting over 90% of samples), financial fraud via premium SMS services, and theft of sensitive user data such as messages and account details. The study further demonstrates that Android malware is rapidly evolving, using techniques like encryption, obfuscation, and dynamic code loading to evade detection. Experimental results reveal that existing mobile antivirus solutions are largely ineffective, with detection rates ranging from only 20.2% to 79.6%, emphasizing the urgent need for more advanced and adaptive security solutions [14].

This paper studies how Android permissions can help users understand the risks and benefits before installing an application but also shows that the current system is not very effective. It explains that many apps request multiple permissions, including access to sensitive data like location, contacts, and messages, which can lead to privacy and financial risks if misused. However, since most apps request similar permissions, users often ignore the warnings and install apps without carefully checking them. The paper points out that this happens because permission messages are repetitive and do not clearly show whether the risk is justified based on the app's purpose. To improve this, the authors propose using "risk signals" that compare an app's permissions with other apps in the same category, so users can better understand if something is unusual or risky. The study shows that many malicious apps request uncommon or unnecessary permissions, which can be used to identify them. Overall, the paper highlights that understanding permissions is important for safe downloading and installation, but current systems need improvement to help users make better decisions [15].

3. DESIGN

3.1 Design Goals

The primary objective of this project is to develop an engaging, user-centered learning platform that educates users on safe

software downloading and installation practices. The website is designed to provide structured access to instructional content, offering step-by-step guidance supported by interactive learning activities. Users are encouraged to actively engage with the material through short tutorials and quiz-based assessments embedded within each lesson.

A key design consideration is the reduction of cognitive load. To achieve this, the interface priorities simplicity and clarity, enabling users to navigate effortlessly between the homepage, lesson modules, and quizzes. Careful attention has been given to layout structure, typography, and navigation design to ensure a clean, uncluttered environment that supports focused learning without overwhelming the user.

In addition, the platform aims to deliver an engaging and visually appealing experience. A modern aesthetic approach has been adopted, incorporating controlled use of color and visual hierarchy to maintain user interest while preserving credibility and readability. The overall visual design reinforces the educational message by presenting the platform as both approachable and informative, while emphasizing the importance of safe online behavior.

3.2 Main User Requirements

Users of the system require access to clear and reliable information regarding the risks associated with downloading files that may contain malware, spyware, or viruses. They also require practical guidance on how to identify trusted software publishers and legitimate download sources. In addition, users benefit from hands-on learning opportunities that allow them to practice identifying unsafe downloads in a controlled environment.

The content is organized into a structured educational framework consisting of concise, self-contained lessons. The interface is designed to be intuitive and efficient, enabling users to locate and access required information quickly. Immediate feedback mechanisms, such as “Correct” and “Try Again” responses, are implemented to reinforce learning outcomes. Furthermore, the platform is required to function effectively across multiple devices, ensuring accessibility and usability in different contexts.

3.3 System Functionalities

The system includes a homepage that introduces the purpose of the platform and provides navigation to the learning modules. The lessons section consists of ten structured modules, each focusing on a specific aspect of safe software downloading. Within these modules, approximately thirty lesson pages present explanations, examples, expected outcomes, and supporting visual material.

Each module concludes with a quiz designed to evaluate user understanding. The quizzes provide immediate feedback for each response, reinforcing correct knowledge and addressing misunderstandings. A cumulative scoring system summarizes user performance across all quizzes. Consistent navigation links allow users to move freely between lessons, quizzes, and the homepage. An additional About page provides an overview of the platform’s purpose and scope. The system is fully responsive, adapting its layout dynamically for desktop computers, tablets, and smartphones to ensure a consistent user experience across devices.

3.4 Design Approach and Structure

The overall design emphasizes clarity, usability, and visual consistency. The instructional model follows a structured, repetitive learning cycle in which users progress from lesson content to quiz assessment, followed by scoring and progression to the next module. This micro-learning approach supports incremental knowledge acquisition by presenting information in manageable segments.

Lessons are structured using headings, bullet points, examples, cautionary notes, and supporting visuals where appropriate. This format is intended to highlight key concepts while illustrating real-world risks associated with unsafe downloads.

JavaScript is used to enhance interactivity within quizzes, enabling one-question-at-a-time presentation, immediate feedback, and final score calculation. This supports active learning and improves user engagement.

3.5 Design Principles

The system follows several core design principles:

- Content is divided into short, focused lessons to reduce cognitive overload.
- A consistent visual hierarchy is maintained through uniform spacing, typography, and color usage.
- Navigation is designed for clarity, ensuring users always understand their current location and available actions.
- A mobile-first approach ensures usability across smaller devices, with optimized touch interactions.
- Accessibility considerations include high-contrast text, descriptive alternative text for images, and keyboard navigation support.
- Interactive feedback is designed to be positive and encouraging to reinforce learning outcomes.
- Security awareness is embedded through the use of real-world examples demonstrating safe and unsafe online behaviors.

3.6 Design Challenges

Several functional and interface-related challenges were encountered during development. On the functional side, managing quiz state, including answer tracking, scoring, and feedback generation, required careful implementation in JavaScript to ensure reliability. The dynamic creation of lesson and quiz modules was necessary to allow future scalability and content updates.

Responsive design adjustments were required to optimize usability on smaller screens, particularly for quiz and lesson layouts. Input validation was also essential to prevent unintended or invalid quiz submissions.

From an interface perspective, maintaining consistency across all ten lessons required strict adherence to established design guidelines. Content complexity was addressed by simplifying lesson text to maintain user engagement. Additionally, quiz elements were optimized for readability and usability on mobile devices, with appropriately sized buttons and clear navigation pathways to ensure users could easily access learning materials.

4. IMPLEMENTATION

4.1 Tools

The website “Safe Downloading & Software Installation" was created using different web technologies and development tools to be an exploratory, responsive, and approachable location for users to learn.

- HTML5: Web pages are structured using HTML5, which serves as the foundation for the Home, Lessons, Quiz, and Results pages. All content for the webpage is arranged into a series of individual headings, sections, buttons, and links.
- CSS3: CSS3 was used to design the visual aspects of the user interface in the form of a clean, contemporary design. Through the use of various design styles, CSS3 determines how everything appears visually on your desktop or mobile device, including how items are spaced apart, the colors that are used, the size of the font, and how everything resizes properly using flexbox and grid techniques according to the size of your screen.
- JavaScript: JavaScript provided the interaction capabilities for the site. Each time you click a button, scroll a page down or hover over an image, JavaScript is in action and controlling everything about that experience.
- Visual Studio Code: VS Code was our primary development environment. It offers features such as syntax highlighting, integration with third-party libraries for HTML/CSS/JS, live preview, and the ability to debug any issues with your code quickly.
- Web Browsers (Chrome/Edge): Browsers were used during development to confirm that the layout of your site was working correctly, that it was functional across all device sizes, and that all of your JavaScript code was functioning correctly. Using Developer Tools, developers can inspect the site for errors, see if the design is responsive, and check the quiz logic for accuracy.
- GitHub Pages (or Netlify): After the completion of the site, it was deployed using Github Pages or Netlify.

4.2 System Implementation and Architecture Design

- The system has been implemented with a modular structure to ensure the site is not very difficult to maintain and grow. All the lessons were developed as individual HTML pages, whereas all quizzes were managed by JavaScript to maintain consistency and minimize redundant codes.
- Questions and answers were stored in JavaScript arrays and that allowed the quiz engine to dynamically load content and to keep track of the score. The event listeners were used to process the interactions with the user, i.e. to choose the answers and to move between the questions. The immediate feedback messages were provided in real time through manipulation of the DOM elements, which were either correct or try again.
- The responsive layout based on CSS Flexbox and CSS Grid was created, which is smooth to screen on mobile and desktop devices. Standardization of colors, types of the buttons, and the spacing among pages was done to ensure design uniformity.
- The navigation was customized by means of basic internal links and explicit buttons. To avoid the use of a backend server, the site structure was maintained lightweight to provide fast loading and transitions.
- The code was split into the following files: HTML, CSS, and JS to maintain the independence of logic, design, and content, which means that one will easily update or add something in the future.

4.3 Interface

The following explains how to use the website step by step. Step 1: Open the browser (Microsoft Edge) and enter the website link [Figure 1].

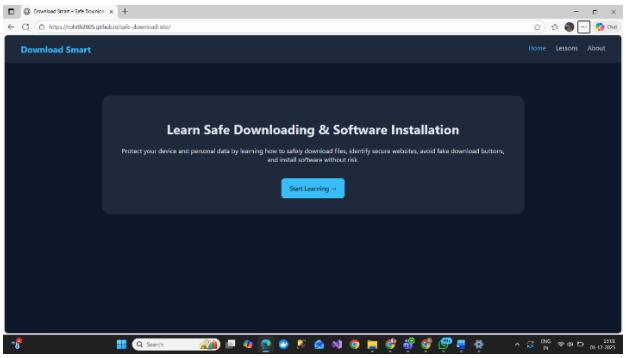

Fig 1: Entering the website link and starting

Step 2: Explore the Home Page [Figure 2]

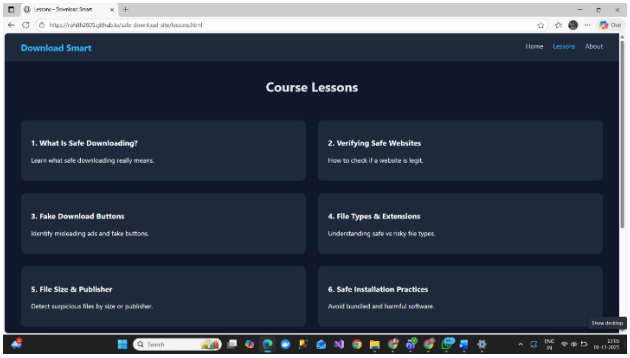

Fig 2: Home Page

Step 3: Navigate to Lessons (By clicking Start Learning) [Figure 3]

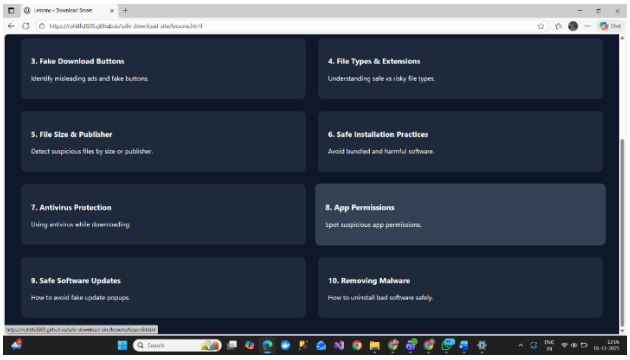

Fig 3: Lessons page

Step 4: Read Lesson Content (By clicking any lesson) [Figure 4]

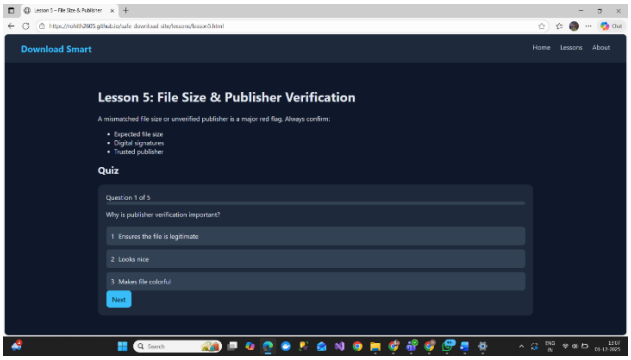

Fig 4: Lesson Content

Step 5: Take the Quiz [Figure 5]

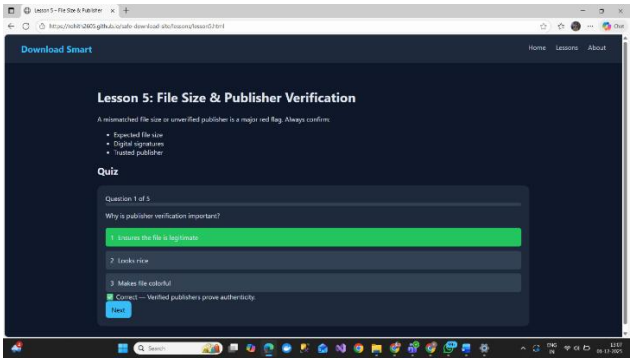

Fig 5: Taking the quiz

Step 6: Review Your Score [Figure 6]

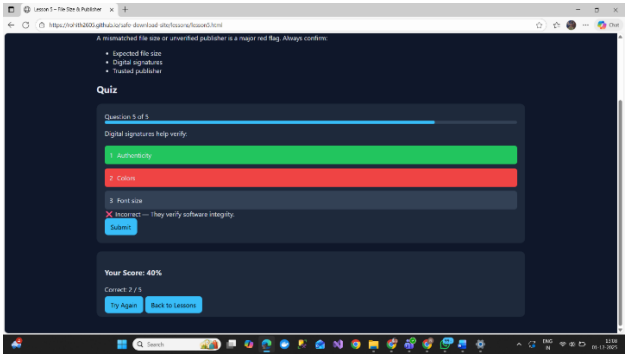

Fig 6: Score page

Step 7: Learn More in the About Page [Figure 7]

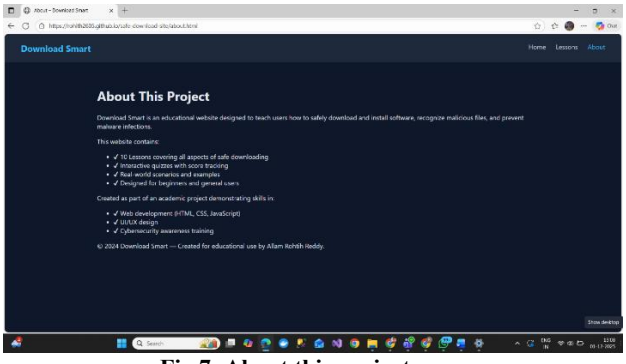

Fig 7: About this project page

5. EVALUATION

5.1 Quantitative Data Analysis of User Feedback

Each user was asked to complete the following questions.
Rate the following from (1:least to 5:best)

1. The site design is pleasing to the eye and user-friendly..
2. The quizzes and lessons are intelligible and not difficult to read.
3. The site loads fast and is functional
4. This site encourages me to educate myself on the safe downloading.
5. The page navigation is very natural and smooth.

The collected quantitative feedback [Figure 8] indicates a generally high level of user satisfaction across all evaluated aspects of the website, with all mean scores ranging between 4.25 and 5.00. This suggests that the platform is perceived as effective, well-designed, and engaging from a user experience perspective.

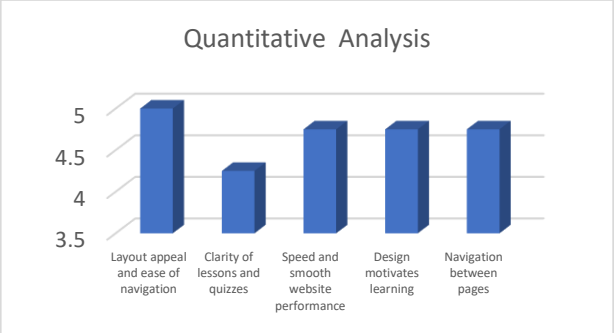

Fig 8: Quantitative analysis

For Layout Appeal and Ease of Navigation the average score was (5.0). This category received the highest possible rating, indicating that users found the website layout highly intuitive and visually appealing. The result suggests that the design structure, including page organization and interface consistency, successfully supports ease of use. Users were able to navigate the platform without confusion, reflecting strong usability principles and effective interface planning.

For Clarity of Lessons and Quizzes the average score was (4.25). This is the lowest scoring category, though still relatively strong. The score suggests that while users generally understood the instructional content and assessment structure, there is room for improvement in how information is presented. This aligns with qualitative feedback indicating that quiz explanations and some lesson content could be clearer or more detailed. Enhancing instructional depth and feedback quality may further improve comprehension.

For Speed and Smooth Website Performance the average score was (4.75). The high rating in this category indicates that users experienced fast loading times and smooth interactions across the platform. This reflects efficient front-end implementation and optimized page structure. It suggests that performance issues are minimal and do not significantly impact the user experience.

For Design Motivates Learning the average score was (4.75). This score demonstrates that the website is effective in engaging users and encouraging continued learning. The combination of interactive quizzes, structured lessons, and visual design elements appears to positively influence user motivation. This supports the effectiveness of the micro-learning and feedback-based instructional approach. For Navigation Between Pages the

average score was (4.75). The strong score indicates that users found it easy to move between different sections of the website, including lessons, quizzes, and other pages. This suggests that the internal linking structure and navigation design are effective. However, when considered alongside qualitative feedback, minor improvements could still be made to make navigation elements more prominent and consistent.

Overall, the quantitative results reflect a highly positive user experience, with all categories scoring above 4.0, indicating strong usability, performance, and engagement. The highest satisfaction was observed in layout design and navigation, while the relatively lower score for clarity of lessons and quizzes highlights an area for targeted improvement. In general, the website is highly usable and visually well-structured. Performance and responsiveness are strong and reliable. The learning design is motivating and engaging. The main improvement area is content clarity, particularly in quizzes and instructional explanations.

Qualitative Data Analysis of User Feedback

Each user was asked to complete the following questions.

1. Which aspects did you like in the site?
2. What were your difficulties in any of the challenges you had experienced in navigating it?
3. What page should be improved the most and why?
4. How would you recommend the design or content to be improved?
5. What was your experience with the site (motivated, confused, satisfied, etc)?

Table 1 is a summary of the major themes identified from user feedback that is listed next.

The qualitative responses indicate an overall positive user experience with the website, particularly in relation to its instructional clarity, structured learning approach, and practical relevance to safe software downloading practices. However, the feedback also highlights several usability and design issues that impact navigation, readability, and assessment clarity. For 'Positive User Experience and Strengths', a key strength identified across all users is the clarity and structure of the lesson content. Users consistently reported that the lessons were well-organized, concise, and easy to follow. For example, multiple participants highlighted the effectiveness of short, focused lessons that directly connect theory to practical safe downloading behaviors. This suggests that the micro-learning approach is successful in supporting comprehension and engagement.

Another frequently mentioned strength is the use of real-world examples, particularly those illustrating unsafe downloads, fake buttons, and counterfeit sources. These examples were described as persuasive, clear, and helpful in reinforcing learning. This indicates that contextual and visual comparisons between safe and unsafe practices significantly enhance user understanding. Additionally, users reported positive emotional responses such as feeling "encouraged," "confident," and "satisfied," suggesting that the interactive quizzes and structured progression contribute to motivation and learning validation. The immediate feedback mechanism appears to reinforce understanding effectively.For 'Usability Challenges and Navigation Issues', despite the positive feedback, several usability issues were identified, particularly on mobile devices. Users reported difficulties with small icons, tight text spacing, and low contrast headings, all of which negatively impacted readability and accessibility. These issues suggest that while the layout is functional, it requires further optimization for smaller screens and accessibility standards. Navigation was another recurring concern. One user noted difficulty locating the quiz button after completing lessons, indicating that the transition between lesson content and assessment is not always intuitive. This suggests a need for clearer visual cues and more consistent placement of navigation elements. For 'Areas Requiring Improvement', the Quizzes section was identified as the area most in need of improvement. Users suggested that incorrect answers should include explanations rather than simple feedback, indicating a demand for more formative learning support. Additionally, quiz feedback was described as insufficiently detailed, limiting its educational value. The Lessons page was also highlighted as needing enhancement through the inclusion of more visual aids, such as screenshots of real-world download scenarios. Similarly, the About page was considered too minimal and could benefit from a clearer explanation of the platform's purpose and learning objectives.

For 'Suggested Enhancements', users provided several constructive suggestions for improving the system. A progress tracking feature across the ten lessons was recommended to improve learning visibility and motivation. This indicates a desire for greater awareness of learning progression within the platform. Interface improvements such as enlarging the "Take Quiz" button and improving its visibility were also suggested, reinforcing earlier concerns regarding navigation clarity. Additionally, users recommended the inclusion of downloadable safety checklists and more visual examples of unsafe or counterfeit installers, which would further strengthen practical learning outcomes. For 'Overall User Perception', user responses indicate a generally positive experience with the platform. Most users reported feeling satisfied and engaged, with some expressing increased confidence in identifying safe downloading practices. However, occasional confusion during quizzes and minor accessibility issues suggest that improvements in design clarity, feedback depth, and mobile optimization are necessary to enhance the overall learning experience.

Table 1. Summary of Major Themes Identified from User Feedback

Major Theme	Positive Findings	Identified Issues	Recommended Improvements
Learning Content	Clear, concise, well-structured lessons; practical examples	Limited visual content in some lessons	Add screenshots, visual demonstrations, and additional real-world examples
Learning Approach	Effective micro-learning structure; lessons are focused and manageable	Learning progress is not sufficiently visible	Introduce progress tracking across the ten lessons
Navigation	Generally functional website structure	Quiz button can be difficult to locate; inconsistent navigation cues	Improve navigation flow and use consistent, prominent action buttons
Quizzes and Assessment	Interactive quizzes and immediate feedback support engagement	Limited explanations for incorrect answers	Provide detailed explanations and formative feedback for incorrect responses
Accessibility and Mobile Usability	Overall interface is functional	Small icons, tight spacing, and	Optimize responsive design, typography,

		low-contrast headings affect readability	spacing, contrast, and button sizes
Supporting Information	Core learning purpose is evident	About page provides limited contextual information	Expand the About page to explain the platform's purpose and learning objectives
Practical Application	Real-world unsafe-download examples are effective	Users requested additional practical resources	Provide downloadable safety checklists and more examples of counterfeit/unsafe installers
Overall User Experience	Users generally felt satisfied, encouraged, engaged, and more confident	Minor confusion during quizzes and navigation	Combine interface, accessibility, navigation, and feedback improvements to strengthen the overall experience

6. CONCLUSION

This project focused on the design and development of a user-centered educational system for safe downloading and software installation. The primary objective was to create an engaging, accessible, and easy-to-use platform that effectively delivers essential cybersecurity awareness content in a structured and practical manner. The implemented system consists of concise, interactive lessons supported by quizzes that reinforce learning through immediate feedback. A responsive design approach was applied to ensure compatibility across both desktop and mobile devices, enhancing accessibility and usability for a wider range of users. Evaluation results, based on both quantitative and qualitative feedback, indicated a high level of user satisfaction. Participants particularly appreciated the intuitive interface, engaging learning content, and overall system performance, which contributed to a positive learning experience. However, several limitations were identified during implementation and evaluation. These include inconsistencies in lesson layout across different screens, reduced readability on smaller devices, and limited clarity in quiz feedback presentation. These challenges highlight areas that require refinement to further improve usability and learning effectiveness.

For future work, the platform can be enhanced by improving interface consistency and optimizing readability across all device sizes. Additional multimedia content such as videos, animations, and real-world case studies can be integrated to enrich the learning experience. Personalization features, such as adaptive learning paths based on user performance, could also be introduced to better accommodate individual learning needs. Furthermore, expanding the curriculum to include broader cybersecurity topics would increase the system's scope and long-term educational value.

7. REFERENCES

[1] M. Botacin, G. Bertão, P. de Geus, A. Grégio, C. Kruegel, and G. Vigna, "On the security of application installers and online software repositories," in *Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2020)*, C. Maurice, L. Bilge, G. Stringhini, and N. Neves, Eds., Lecture Notes in Computer Science, vol. 12223. Cham, Switzerland: Springer, 2020. doi: 10.1007/978-3-030-52683-2_10.

[2] J. Gamba, M. Rashed, A. Razaghpanah, J. Tapiador and N. Vallina-Rodriguez, "An Analysis of Pre-installed Android Software," *2020 IEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 2020, pp. 1039-1055, doi: 10.1109/SP40000.2020.00013.

[3] M. Taleby, Q. Li, M. Rabbani, and A. Raza, "A survey on smartphones security: Software vulnerabilities, malware, and attacks," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 10, 2017. doi: 10.14569/IJACSA.2017.081005.

[4] K. Vaniea and Y. Rashidi, "Tales of software updates: The process of updating software," in *Proc. 2016 CHI Conf. Human Factors in Computing Systems*, pp. 3215–3226, 2016.

[5] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang, "Hey, you, get off of my market: Detecting malicious apps in official and alternative Android markets," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2012.

[6] H. Huang *et al*., "A Large-Scale Study of Android Malware Development Phenomenon on Public Malware Submission and Scanning Platform," in *IEEE Transactions on Big Data*, vol. 7, no. 2, pp. 255-270, 1 June 2021, doi: 10.1109/TBDATA.2018.2790439.

[7] H. Zhu, H. Cao, E. Chen, H. Xiong, and J. Tian, "Mobile app recommendations with security and privacy awareness," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2014, pp. 951–960.

[8] L. Maxwell, A. Sanders, J. Skues, and L. Wise, "A content analysis of personal safety apps: Are they keeping us safe or making us more vulnerable?" *Violence Against Women*, vol. 26, no. 2, pp. 233–248, 2020.

[9] S. Wu and J. Liu, "Overprivileged Permission Detection for Android Applications," *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761572.

[10] F. Kreuter *et al.*, "Collecting survey and smartphone sensor data with an app: Opportunities and challenges around privacy and informed consent," *Social Science Computer Review*, vol. 38, no. 5, pp. 533–549, 2020.

[11] M. Ohm and C. Stuke, "SoK: Practical detection of software supply chain attacks," in *Proceedings of the 18th International Conference on Availability, Reliability and Security (ARES)*, 2023.

[12] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "The Nocebo effect on the web: An analysis of fake anti-virus distribution," in *Proceedings of the 3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2010.

[13] W. Börger and L. Lo Iacono, "User perception and response to computer security warnings," in *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES)*, 2015.

[14] Y. Zhou and X. Jiang, "Dissecting Android Malware: Characterization and Evolution," *2012 IEEE Symposium on Security and Privacy*, San Francisco, CA, USA, 2012, pp. 95-109, doi: 10.1109/SP.2012.16.

[15] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Android permissions: A perspective combining risks and benefits," in *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2012.