Strengthening Web Application Security through Email Verification and JWT Authentication

Migena KERI[1]

Malvina NIKLEKAJ[2]

Abstract

Information security remains one of the most critical challenges in web application development, especially as users are increasingly exposed to risks such as unauthorized access and poor credential management. This paper addresses these challenges by designing and implementing a secure web application that integrates email address verification and JSON Web Token (JWT) authentication, combined with a user-friendly interface aimed at strengthening security and raising awareness of good credential management practices.

The system is developed using React on the frontend and Django REST Framework on the backend, connected to a SQL database. Key functionalities include user registration with email verification via Mailtrap, role-based access control, and an administrative panel for account management. Functional testing showed that email verification reduces unauthorized logins, JWT provides consistent and secure session management, while the interface contributes to educating users on the importance of secure practices.

[1] Department of Informatics and Technology, Faculty of Engineering, Informatics and Architecture, European University of Tirana, Tiranë, Albania, mkeri@uet.edu.al

[2] Department of Informatics and Technology, Faculty of Engineering, Informatics and Architecture, European University of Tirana, Tiranë, Albania, malvina.niklekaj@uet.edu.al
https://orcid.org/0009-0005-0506-215X

The results of the paper prove that combining modern technologies with secure development practices provides not only data protection, but also practical education for users. The main contribution lies in providing an applicable model for strengthening authentication and improving user behavior. In the future, the system can be expanded with multi-factor authentication, password recovery mechanisms, and real-time security analysis.

Key Words: *Web Application Security, Email Verification, JSON Web Token (JWT), User Authentication, Django REST Framework.*

Introduction

Scope and Context

In the digital age, where information is distributed and accessed massively through computer networks and cloud services, data security has become one of the most important challenges. Traditional authentication mechanisms, mainly based on passwords, often do not provide sufficient protection, as users tend to create and manage weak credentials. This brings great risks, making the system vulnerable to attacks such as phishing, credential stuffing and brute force.

In this context, the need for additional security mechanisms is essential. One of them is the verification of the email address as an additional step before activating the user account, which ensures not only the validity of the identity, but also a reliable communication channel with the user. In parallel, friendly and well-designed interfaces can serve as educational tools that directly affect the improvement of user behavior towards secure practices.

Motivation and Contribution

The main motivation of this study lies in addressing the gap between advanced security technologies and poor user behavior. Although strong protection mechanisms exist, in practice it is often the user who represents the weakest link in the security chain. By building a system that not only provides secure authentication but also educates users on the importance of identity confirmation and correct credential management, this paper aims to contribute to both the academic and practical fields.

The main contribution of the paper lies in providing a simple, applicable and scalable model for web applications, which combines email verification, JWT authentication and a polite interface, demonstrating how these elements can strengthen security and user awareness at the same time.

Research Questions and Hypotheses

Based on the above challenges, two research questions have been raised:

- How does email verification improve authentication security in web applications?
- In what way can the interface and technology contribute to educating users on secure credential management practices?

The relevant hypotheses are:

- H1: Implementing email verification reduces unauthorized logins and strengthens identity control.
- H2: A user-friendly interface, combined with clear instructions on secure practices, positively affects user education and behavior.

Objectives

To address the questions and hypotheses raised, the paper aims to achieve the following objectives:

1. Developing a functional web application with email verification and JWT authentication.
2. Testing the registration and verification process using secure platforms such as Mailtrap.
3. Providing an interface that helps users understand the importance of their actions in security.
4. Developing an administrative panel for role and access control.
5. Assessing the impact of the system on security and user awareness.

Literature Review

Password Security and User Behavior

Password security remains a central element of digital protection, yet user behavior continues to weaken authentication processes. Multiple studies indicate that users consistently select weak passwords or reuse the same credentials across several accounts, exposing systems to credential stuffing, dictionary attacks, and brute force attempts (Bang et al., 2012; Das et al., 2014). This pattern reveals that the

primary vulnerability often lies not in technology, but in human factors. Research also shows that users frequently underestimate the importance of password strength, assuming they are not significant targets — a perception that significantly reduces security effectiveness.

Psychological and Cognitive Factors

Beyond technical considerations, password management is heavily influenced by psychological and cognitive limitations. Hardman et al. (2022) highlight that users experience "security fatigue" when confronted with frequent or complex password requirements, leading to non-compliance with security recommendations. Memory constraints also drive users toward insecure behaviors, such as reusing patterns, relying on personal information, or storing credentials in unsafe places. In many cases, excessive security requirements result in decreased usability, reinforcing the need to balance cognitive load with secure design principles.

Technological Security Mechanisms

As a response to the inherent weaknesses of password-based systems, more advanced mechanisms have been developed. Multi-factor authentication (MFA) and two-factor authentication (2FA) offer additional layers of protection by requiring independent verification methods, thus significantly reducing the likelihood of unauthorized access (Ometov et al., 2018). Additionally, JSON Web Token (JWT) has emerged as a widely adopted approach for managing stateless authentication in modern web applications (Bonneau et al., 2012). By embedding claims in digitally signed tokens, JWT enhances both security and scalability while reducing server-side complexity.

Educational Interventions and Institutional Policies

Recent literature emphasizes that technology alone cannot resolve authentication vulnerabilities without proper user education. Reeder et al. (2017) demonstrate that interactive, behavior-driven educational methods are more effective than traditional awareness campaigns in improving credential management practices. Institutions now adopt policy frameworks that reflect this reality by promoting longer passphrases, banning commonly used passwords, and reducing reliance on forced periodic password changes (Nieles et al., 2017). These strategies underline the importance of combining secure authentication mechanisms with user-centric approaches that encourage sustainable and secure behavior.

Methodology

Methodological Approach

This study follows an experimental approach based on the development of a functional web application, which serves as a demonstration environment to test the hypotheses raised on the impact of email verification and JSON Web Token (JWT) authentication on strengthening security and user education. The methodology is based on three main pillars: (i) designing the system architecture, (ii) technical and functional implementation of the main components, and (iii) testing to assess security, scalability, and impact on user behavior.

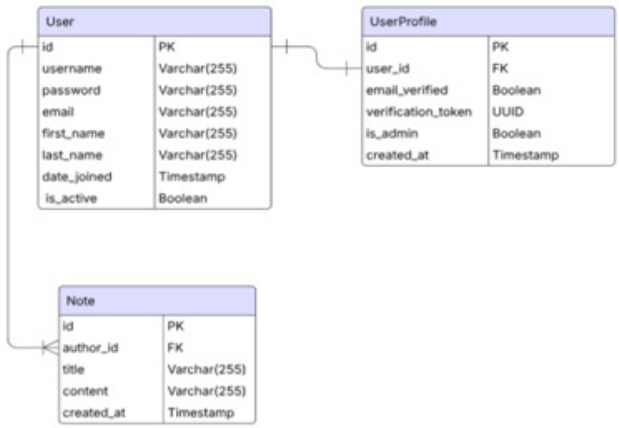
The goal is not only to create a technically viable application, but also to evaluate how technology and interface design contribute to user awareness of secure credential management practices.

System Architecture

The system is built on a client-server architecture, where the frontend (React with Vite) and the backend (Django REST Framework) communicate through a secure REST API. Data is stored in a relational SQL database, which ensures the integrity and organization of information.

The architecture is designed to provide modularity, clear separation of responsibilities and high scalability. The front end provides an interactive interface for users, while the backend contains business logic and security mechanisms.

FIGURE 1. ERD diagram of the database



The ERD diagram reflects two main entities: User and Note. Each note is linked to a user through a foreign key relationship, guaranteeing referential integrity and enabling access control according to ownership.

Technologies used

- React & Vite (Frontend): chosen to build a dynamic SPA (Single Page Application), providing fluid user experience.
- Django REST Framework (Backend): robust framework in Python, with built-in support for authentication, ORM and RESTful API.
- SQL Database: SQLite during development, with the possibility of migrating to PostgreSQL in production environments.
- JWT (JSON Web Token): used for stateless authentication, providing session management without loading the server.
- Mailtrap (SMTP Sandbox): for safe testing of verification emails without risking their delivery to real addresses.
- Django-CORS-Headers: to enable secure communication between frontend and backend in separate development environments.

Implementing the main components

User registration and email verification
User registration involves a two-step process: account creation and email address verification.

- On the frontend, the user fills out the registration form with username, email, and password. Validations are performed on both the frontend and backend.
- On the back end, the user is stored in the database with an inactive status until they confirm their email.
- A verification token is generated, which is included in a custom link and sent via Mailtrap.
- Clicking the link redirects the user to a special page in React, which confirms activation via the Django API.

FIGURE 2. User registration interface

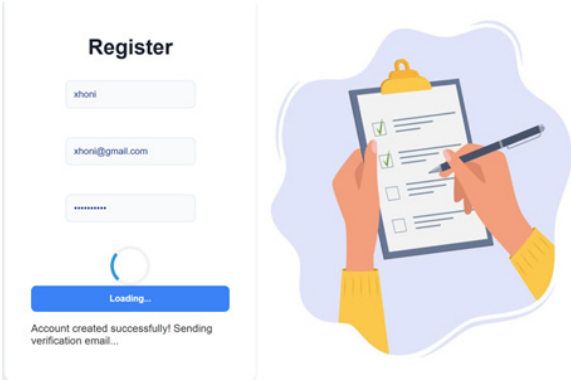


FIGURE 3. Email generated by Mailtrap for verification

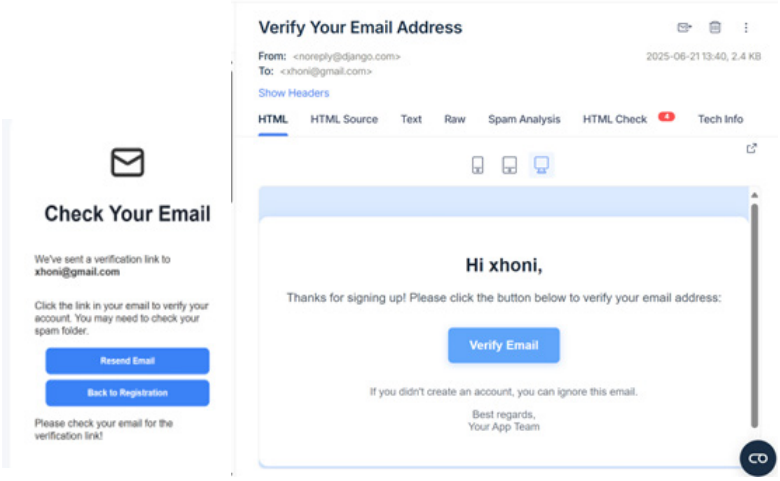
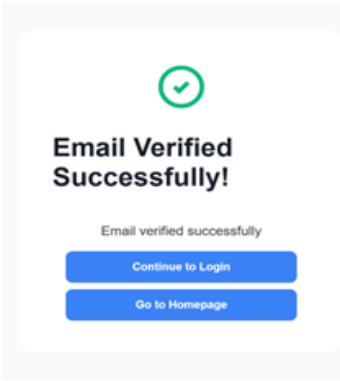


FIGURE 4. Email verification confirmation message



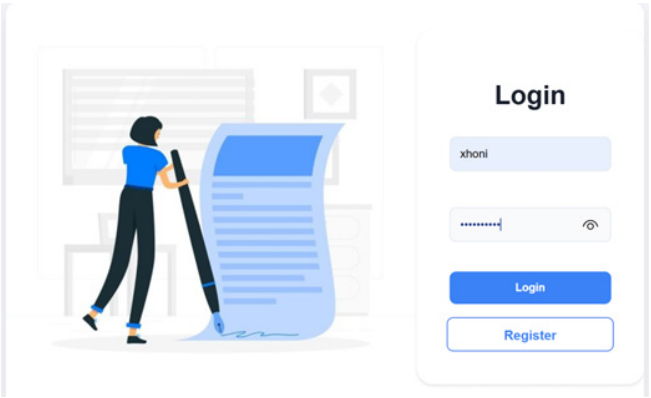
This mechanism ensures that each account is associated with a valid address, preventing the mass creation of fake accounts and increasing user awareness of the importance of this process.

Authentication and session management

After account verification, the user can log in to the system using email and password. The backend generates a JWT, which is signed with a secret key and returned to the client. The token is stored in localStorage and attached to each API request via the Authorization header.

JWTs are only valid for a limited time (15 minutes), limiting the risk in case of compromise. For longer sessions, the system provides for the use of refresh tokens.

FIGURE 5. User login interface



This mechanism eliminates the need to store sessions on the server and provides a more secure and scalable model.

User dashboard and note management

After authentication, the user is directed to a personal dashboard, where he can create, view and delete his notes. All operations are protected by JWT authentication, and each user has access only to their own records.

FIGURE 6. User Dashboard

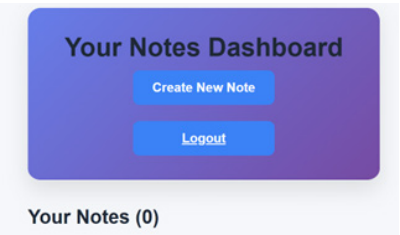


FIGURE 7. Creating a new record

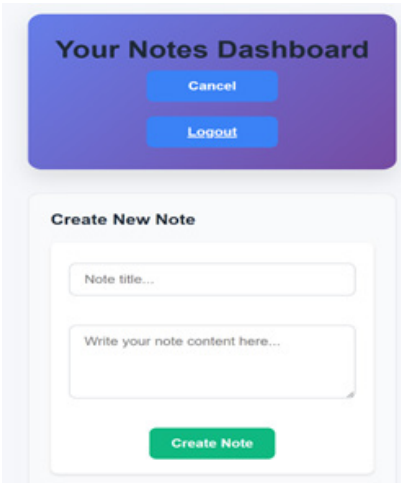
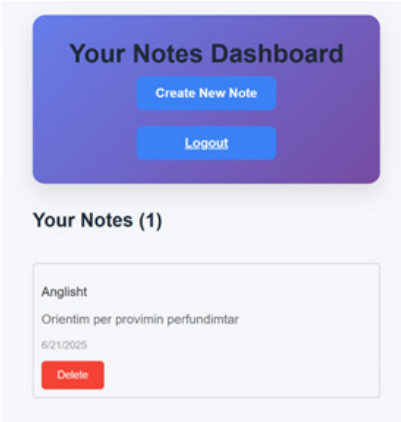


FIGURE 8. The process of deleting a record



This functionality demonstrates the practical application of access control based on data ownership and educates users on the importance of privacy.

Admin Panel

In addition to regular users, the system also provides the administrator role. Through the Django admin panel, admins can:

- Activate or deactivate accounts.
- Promote users to new roles.
- Delete accounts and all associated records.

FIGURE 9. Admin Panel

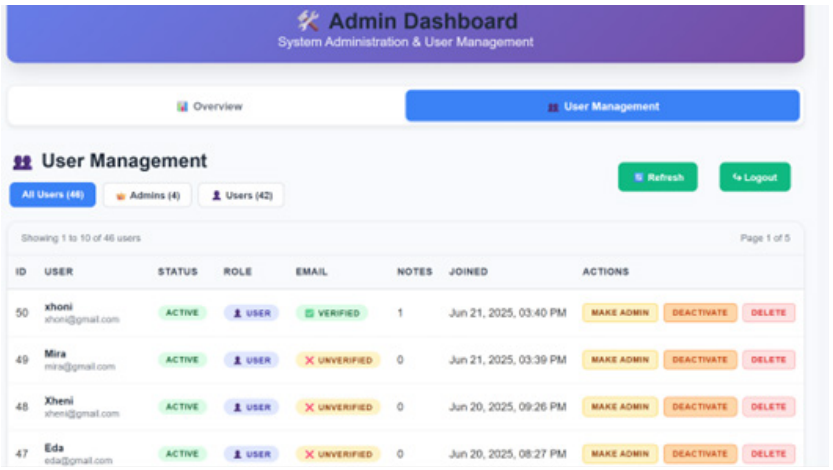
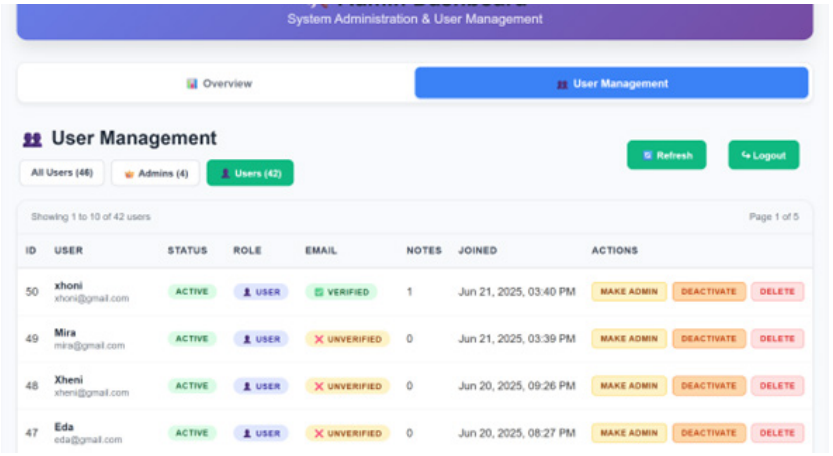


FIGURE 10. Managing users and roles



This layer of control is essential for system maintenance and the overall security of the platform.

Integration and communication security

Since the front end and backend ran on different hosts, the CORS mechanism was used to allow secure communication. This approach guarantees that only trusted origins can access the API, limiting the risk of abuse by unauthorized applications.

Security and Functionality Testing

Functional Testing

The cases of registration, email verification, login, creating and deleting notes, as well as user management from the admin panel were tested. In all cases, the system functioned according to specifications, respecting user roles and privileges.

Security Testing

Tests were conducted to verify protection against common attacks:

- SQL Injection: prevented by using Django's ORM.
- XSS (Cross-Site Scripting): limited by React's policy of not executing injected HTML.

FIGURE 11. SQL Injection Testing

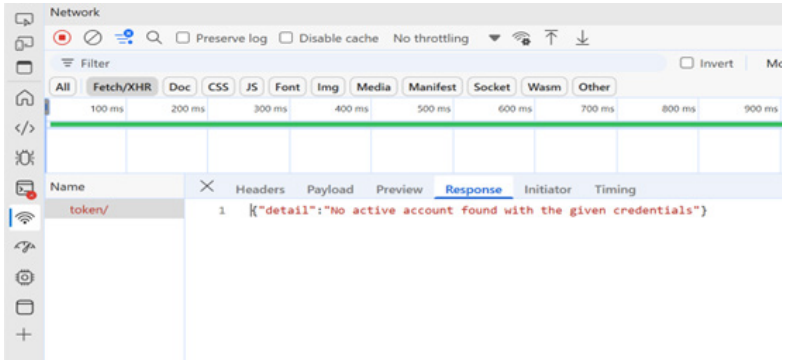


FIGURE 12. XSS Testing (Part 1)

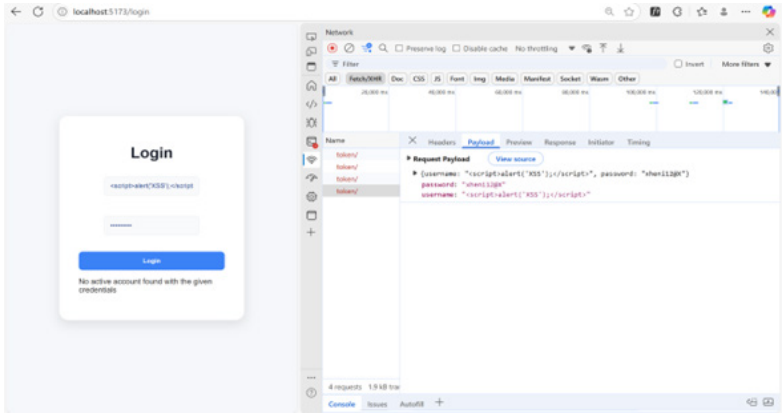
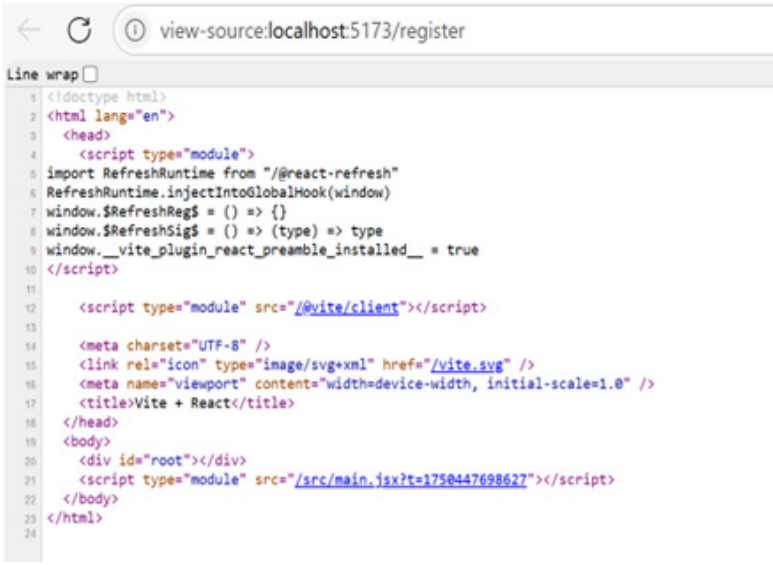


FIGURE 13. XSS Testing (Part 2)



Reflection on Methodology

The approach used demonstrated that combining modern technologies with secure development practices provides a robust model for web applications. The implementation of email verification and JWT authentication not only increased the level of security but also contributed to user education by making them aware of authentication processes and their role in maintaining security.

Results and Discussion

Main Results

The registration and email verification process worked as expected, ensuring that no user was able to log in without first confirming their email address. This mechanism proved its value as an indispensable tool to prevent the creation of fake accounts, making the system more stable and reliable. Furthermore, users were faced with a clear experience that guided them on the importance of this process, transforming it into an educational tool in addition to its technical function.

JSON Web Token (JWT) authentication also showed high effectiveness in managing user sessions. Each user, after successful identification, was provided with a token that had a limited time validity, thus reducing the risk of abuse in case of compromise. Testing confirmed that the system automatically rejected any

unauthorized request, proving that access control was implemented correctly and securely. The user dashboard was created as an isolated environment where each person had access only to their personal records. This organization ensured that data ownership rights were fully respected, guaranteeing information privacy and making the interface clear and functional for individual content management.

At the same time, the admin panel offered advanced functionality for user management. Administrators had the ability to activate or deactivate accounts, delete specific users, or promote them to new roles. Any intervention made in this panel was immediately reflected in the system's behavior, confirming full and centralized control over the platform's access and operation.

Finally, security testing confirmed that the application was protected from the most common attacks, such as SQL Injection and Cross-Site Scripting (XSS). Thanks to the use of ORM in Django, the injection of malicious commands into the database became impossible, while React's policy for handling injected HTML effectively limited the possibility of executing malicious scripts. These results showed that the developed application complies with basic security standards and provides a reliable level of protection for user data.

Discussion of the results

The test results confirm that the combination of email verification and JWT provides a level of security comparable to industry's best practices. In addition to technical assurance, the system also contributed to user awareness, as the verification process made them more aware of the importance of authentication and identity control.

Educational value:
Users were presented with clear messages during registration and login, where the system explained why email verification was required and why login could not be completed without this step. This created an educational interaction that goes beyond technical functionality.

Accessibility limitations:
However, the current system does not include multi-factor authentication (MFA), which is considered one of the highest security standards. Also, the use of Mailtrap is limited to test environments; in a real environment, integration with an external email service and verification of message delivery would be necessary.

Comparison with literature:
The literature suggests that user behavior remains the weakest link in the security chain (Bang et al., 2012; Hardman et al., 2022). The results of this study support

this finding but also demonstrate that combining technological practices with interface design can have a significant impact on improving their behavior.

Practical implications:
This project provides a simple and applicable model for web applications that want to integrate basic security mechanisms, focusing on both the technical aspect and user education. The system can serve as a practical environment for cybersecurity training, as well as a basis for further developments in larger applications.

Conclusions and Future Work

This study aimed to explore the impact of email verification and JSON Web Token (JWT) authentication mechanisms in strengthening the security of web applications and in increasing user awareness of proper credential management practices. The implementation of a functional application served as a practical test that demonstrated that these mechanisms do not only provide technical protection but can also be used as educational tools for users.

The results clearly showed that the email verification process limits the creation of fake accounts and strengthens identity control, while the use of JWT provides a secure and scalable authentication model. The personalized user dashboard and the administrator panel illustrated how access control can be implemented at different levels, respecting the principles of privacy and centralized management. Security testing demonstrated that the developed system was protected against a range of common attacks, such as SQL Injection and Cross-Site Scripting (XSS), ensuring data integrity.

Another important finding was the educational dimension of the system. Users were presented with a clear and structured registration and login process, which not only ensured controlled access, but also made them more aware of their role in protecting their personal data. This element reflects the findings of the literature that emphasize that the user often remains the weakest link in the security chain, but a good interface design and clear explanatory messages can directly influence the change of their behavior.

Although the system worked as expected, its limitations should not be overlooked. Currently, the application does not include multi-factor authentication (MFA), which is a standard practice in modern applications with a high level of security. Also, the use of Mailtrap is limited to test environments only; for real applications, integration with a professional email service and large-scale delivery management would be required.

In future work, this system could be extended with additional modules such as password recovery, multifactor authentication integration, and the development

of real-time security monitoring mechanisms. Another valuable direction is the integration of user behavior analysis, using simple artificial intelligence algorithms to identify abnormal behavior and prevent unauthorized access.

In conclusion, the paper contributes by providing a practical and simple model for building more secure web applications, which combine technological mechanisms with an educational approach to users. The results show that even with simple and accessible tools, a balance between technical protection and awareness raising can be achieved, making these solutions suitable for adoption in a wide range of web applications in practice.

References

Bang, J. M., Karlsson, F., & Tehler, H. (2012).User compliance and password security: Investigating a user-centric framework.*Information & Computer Security, 20(4), 332–348.*

Bonneau, J., Herley, C., Van Oorschot, P. C., & Stajano, F. (2012). The quest to replace passwords: A framework for comparative evaluation of authentication schemes. *IEEE Symposium on Security and Privacy*, 553–567.

Dhanjani, N. (2015). *Abusing the Internet of Things: Blackouts, freakouts, and stakeouts.* O'Reilly Media.

Hardman, D., Zois, D. S., & Tzovaras, D. (2022). Usable security: Balancing user behavior and authentication security. *Computers & Security, 113*, 102546.

Krombholz, K., Busse, K., Pfeffer, K., Smith, M., & Grechenig, T. (2017)."If HTTPS were secure, I wouldn't need 2FA"—End user and administrator mental models of HTTPS.*IEEE Symposium on Security and Privacy*, 246–262.

Nash, J., & Biddle, R. (2020).Passwords at the Crossroads: User choices and system design. *International Journal of Human-Computer Studies, 137*, 102383.

Nieles, M., Dempsey, K., & Pillitteri, V. Y. (2017).An introduction to information security (NIST SP 800-12 Rev.1).National Institute of Standards and Technology.

Ometov, A., Bezzateev, S., Mäkitalo, N., et al. (2018). Multi-factor authentication: A survey. *Cryptography, 2(1), 1.*

Pfleeger, C. P., Pfleeger, S. L., & Margulies, J. (2015).*Security in computing* (5th ed.). Pearson.

Schneier, B. (2015). *Secrets and lies: Digital security in a networked world* (Updated ed.). Wiley.

Ur, B., Shay, R., Komanduri, S., et al. (2015).Can users behave securely without suffering a usability penalty? *CHI Conference on Human Factors in Computing Systems*, 157–166.