International Journal of Scientific Research and Engineering Development-– Volume X Issue X, Year

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Password Strength Checker and Manager to Enhance Digital Security

Kairun Naseeba*, Dr. Harold Robinson Y**

*(Student, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
kairunnaseeba.ug.21.cs@francisxavier.ac.in)

** (Assistant Professor, Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli
haroldrobinson@francisxavier.ac.in)

----------**********----------

Abstract:

The growing complexity of digital ecosystems has made secure authentication more critical than ever, yet weak or easily guessable passwords continue to be a leading cause of data breaches. To combat this vulnerability, this project introduces a comprehensive Password Strength Checker platform focused on enhancing user security through entropy-based evaluation and intelligent feedback mechanisms. Utilizing a clean and responsive frontend built with HTML, CSS, and JavaScript, the platform delivers real-time analysis of password inputs, calculating entropy—a statistical measure of randomness and resistance to brute-force attacks. Complementary checks for character diversity, length, repetition patterns, and common dictionary words enhance the scoring model, ensuring holistic strength assessments. The backend, developed with Node.js and Express, manages deeper analysis and user interactions. It interfaces with the Google Cloud ecosystem, integrating Google Directory for user identity management and Google Sheets as a lightweight, cloud-hosted database for storing anonymized password metrics, system logs, and trend data. A RESTful API ensures efficient communication between the client and server, enabling instant updates and feedback without compromising performance or user experience. Administrators can access consolidated insights through dynamically updated Google Sheets dashboards, revealing key trends such as average password strength, most common weaknesses, and improvement over time. These analytics support informed decision-making for security training and policy updates. By combining entropy-driven password evaluation with the flexibility of Google's cloud tools and seamless user experience, this platform empowers individuals and organizations to build stronger defenses against credential-based threats—encouraging proactive, secure behavior in today's increasingly digital world.

Keywords — **Password Security, Entropy Analysis, Password Strength, Cryptographic Algorithms, Data Protection, Security Metrics, Hashing Techniques.**

----------**********----------

I. INTRODUCTION

As digital systems become increasingly integrated into personal, professional, and governmental domains, the security of user credentials remains a foundational concern. Despite advancements in encryption and authentication technologies, password-related vulnerabilities continue to be a leading cause of data breaches and unauthorized access. A 2023 report by Martin and Singh [1] highlights that weak or reused passwords contribute to over 60% of account compromises, underscoring the urgent need for user-friendly tools that promote stronger password practices.

To address this pervasive issue, this project proposes a "Password Strength Checker with Entropy-Based Evaluation", a web-based platform designed to guide users in creating secure and resilient passwords. Drawing on research by Alavi and Chen [2], which emphasizes the effectiveness of entropy as a metric for password unpredictability, the system provides real-time feedback on password strength using entropy calculations alongside traditional criteria such as length, character diversity, and repetition patterns. By quantifying resistance to brute-force and dictionary attacks, this checker helps users better understand the relative security of their passwords.

The platform employs a clean and responsive front-end interface developed with HTML, CSS, and JavaScript, delivering an intuitive user experience that visually represents strength scores and provides actionable recommendations. The backend infrastructure, built with Node.js and Express, performs validation and advanced strength assessments while maintaining secure interactions. User management is integrated with Google Directory, allowing seamless and secure identity handling, especially in organizational environments. Additionally, Google Sheets serves as the platform's lightweight cloud database, storing anonymized logs and usage statistics for auditing, analysis, and improvement purposes.

Building on the work of Nakamura and Foster [3], who advocate for transparency and educational feedback in security tools, the platform not only checks password strength but also explains the rationale behind its evaluations. This fosters user learning and engagement, promoting better long-term password hygiene. Administrators and system designers benefit from aggregated data stored in Google Sheets, which enables insight into system-wide password trends and common weaknesses.

By integrating entropy-driven evaluation, cloud-based identity management, and accessible UX design, this project seeks to bridge the gap between security awareness and user action. Its goal is to create a scalable, data-informed solution that improves password quality at both the individual and organizational levels. The following sections outline the platform's architecture, entropy computation model, user interface design, and the performance metrics used to evaluate its effectiveness in enhancing credential security.

II. OBJECTIVE

The growing threat of cyber-attacks and data breaches has underscored the critical importance of password security in the digital age.

This research proposes the development of an Interactive Password Strength Checker with Entropy Calculation, a web-based tool designed to guide users in crafting strong, secure passwords through real-time analysis and visual feedback. The system blends usability with security science by incorporating entropy as a key metric—providing a mathematically grounded measure of password unpredictability—alongside conventional checks like character length, variety, and pattern repetition.

The primary objective of this study is to develop a lightweight, accessible platform that educates users and fosters stronger password practices through dynamic feedback, seamless integration, and intelligent analytics. The system is organized around the following key objectives:

1. Entropy-Based Password Strength Evaluation:
Design an algorithm that calculates password entropy and uses it as a core component of the strength assessment. Entropy, measured in bits, reflects the level of randomness and resistance to guessing attacks. The system will also incorporate supplementary criteria such as character diversity (uppercase, lowercase, numeric, special symbols), repetition detection, and dictionary word presence to provide a well-rounded strength score and actionable recommendations.

2. Real-Time Feedback and Educational Guidance:

Implement a responsive and interactive front-end using HTML, CSS, and JavaScript, capable of offering instantaneous visual and textual feedback as users type their passwords. This includes dynamic progress indicators, strength meters, and contextual tips to help users understand the rationale behind the score and learn how to improve it. The goal is to foster better long-term password hygiene through immediate education.

3. Node.js & Express Backend for Secure Processing and Validation:

Develop a backend system using Node.js and Express that handles entropy calculations, pattern checks, and secure server-side validations. The backend will also cross-reference inputs with anonymized breach databases to flag commonly compromised passwords, improving user awareness and minimizing reuse of high-risk credentials.

4. Google Directory Integration for User Management:

Incorporate Google Workspace Directory to manage user identities, enabling secure login, usage tracking, and policy enforcement in enterprise or educational contexts. This integration supports scalability and allows for centralized administration in multi-user environments.

5. Google Sheets as a Lightweight Analytics and Storage Backend:

Utilize Google Sheets as a flexible, cloud-based database for storing anonymized usage statistics, entropy score distributions, and system logs. This enables easy visualization of trends over time and supports administrative review and auditing without requiring a complex database infrastructure.

By addressing password insecurity through a combination of user-centric design, entropy-based evaluation, and seamless integration with cloud tools, this project bridges the gap between password awareness and actionable security behavior. It empowers users to take control of their digital security while offering administrators insights into system-wide trends and vulnerabilities—creating a more resilient digital environment for individuals and organizations alike.

III. MODULE AND ALOGRITHM

The Interactive Password Strength Checker with Entropy Calculation is designed to enhance digital security by providing real-time password analysis, visual strength feedback, and entropy-based assessments. This system transcends traditional password validators by incorporating deeper mathematical evaluations, security policy integration, and interactive guidance. Built using HTML, CSS, JavaScript, with a Node.js and Express.js backend and integrated with Google Console Directory and Google Sheets, the platform offers a modular, scalable, and security-focused environment tailored for both individuals and enterprise users.

A. Modules:

1. User Input Module:

The User Input Module is the initial point of interaction for users, providing a secure and intuitive web form for password entry. Designed for both usability and protection, the module validates inputs in real time to prevent empty or malformed entries. When connected to Google Console Directory, it can enforce pre-authentication before password evaluation begins, ensuring organizational control and access management. Optional fields like username or email enable enhanced audit tracking and policy application across enterprise-level accounts.

2. Password Validation Module:

This module filters out weak passwords through a foundational layer of validation, enforcing essential security requirements before proceeding to deeper evaluation. It ensures passwords meet a minimum length—typically eight characters—and checks them against a database of common and previously

breached credentials. With support for integration into Google Cloud's security rules and third-party APIs, the module can dynamically update its list of invalid passwords. For enterprise deployments, custom rules and restrictions—such as disallowing specific patterns or enforcing minimum entropy thresholds—can be configured through Google Console Directory policies.

3. Strength Evaluation & Entropy Module:

The Strength Evaluation & Entropy Module is the analytical core of the system, using a combination of structural rules and entropy theory to evaluate password robustness. The algorithm considers multiple factors including password length, character type diversity (uppercase, lowercase, digits, symbols), and recognizability of patterns such as "1234" or repeated characters. It applies Shannon entropy to calculate the statistical randomness of the password, yielding a quantifiable strength score. These evaluations are deterministic to ensure fairness and reproducibility, and all computations are performed without storing the password, ensuring complete user privacy and compliance with regulations like GDPR.

4. Feedback Generation Module:

This module is responsible for translating technical evaluations into meaningful, user-friendly insights. Once a password is analyzed, it returns a strength rating—categorized as Weak, Medium, or Strong—alongside specific, rule-based suggestions for improvement. Users receive tailored tips, such as encouraging the addition of special characters or the avoidance of dictionary words, while compromised credentials are clearly flagged when identified. The language used in the feedback is intentionally structured and transparent, building user trust and making the improvement process educational and actionable.

5. Visual Indicator Module:

To enhance user experience, the Visual Indicator Module provides a dynamic, color-coded strength bar that updates in real time as the user types. Weak passwords trigger a red status, moderate ones shift to yellow, and strong passwords earn a green rating. These visuals are paired with icons and tooltips to improve accessibility and help users intuitively understand their progress. The real-time nature of the indicator reinforces good password habits and encourages users to iterate toward stronger, more secure choices with immediate feedback.

6. User Feedback & Education Module:

More than just a validator, this module is designed to educate users about secure password practices. After submission, users receive a detailed breakdown that includes their entropy score, vulnerabilities detected, and actionable suggestions for improving their password security. When integrated with Google Directory, enterprise rules may require users to update weak passwords before proceeding. The module also advocates for best practices such as using passphrases, avoiding reused credentials, and enabling multi-factor authentication (MFA), positioning it as both a tool and a tutor in digital hygiene.

7. Backend Security Module (Node.js & Express.js):

The backend infrastructure is built with security and scalability at its core. It leverages JWT-based authentication for secure session handling and API protection. Middleware tools such as `express-validator` and `sanitize-html` help ensure incoming data is clean and safe from injection attacks. Brute-force protection is implemented using `express-rate-limit`, and OAuth2 integration enables secure logins via platforms like Google and Facebook. Abnormal usage patterns and failed login attempts are logged for administrative review, helping detect potential threats in real time.

8. Frontend Security Module (JavaScript):

On the client side, this module ensures sensitive user input is managed with the utmost care. Passwords are processed only in memory and are never stored or logged. Controlled DOM elements manage input behavior securely, while JWT tokens

support stateless session tracking. All communications are conducted over HTTPS with TLS encryption, ensuring that data remains secure in transit and that users interact with the platform in a fully protected environment.

9. Database & Logging Module (Google Sheets + Google Cloud):

Designed for lightweight but insightful data tracking, this module uses Google Sheets and Google Cloud services to log password evaluation metadata—such as entropy scores, timestamps, and anonymized user identifiers. While never storing the actual passwords, it provides visibility into user trends, common vulnerabilities, and compliance metrics. It also integrates with breach detection APIs like HaveIBeenPwned to enrich analysis and flag at-risk credentials. These logs serve as the foundation for audit trails, dashboards, and security oversight across the organization.

10. Testing and Evaluation Module:

To ensure the system remains effective and resilient, this module simulates high-volume password entry to test system performance and stability. It also supports A/B testing to measure the impact of different types of feedback—visual versus textual—on user behavior and learning outcomes. Over time, it collects insights into how users respond to feedback, helping refine educational prompts and security messaging. Additionally, it ensures adherence to OWASP guidelines and best practices for password handling and user data protection.

B. Algorithm:

The core algorithms of the Interactive Password Strength Checker are designed to deliver real-time password analysis through entropy-based evaluation, actionable feedback, policy compliance, and secure logging. These processes not only empower users to create stronger, more secure passwords, but also provide system administrators with critical visibility into password hygiene and compliance across the organization.

1. Entropy-Based Strength Evaluation Algorithm:

At the heart of the password strength analysis is the entropy-based evaluation algorithm. This algorithm quantifies the unpredictability of a password using the formula **Entropy = $\log_2$(R^L)**, where `R` is the pool of unique character types (such as lowercase letters, digits, or symbols), and `L` represents the password's length. Based on this entropy score, passwords are categorized into three strength levels: weak (less than 40 bits), moderate (40–60 bits), and strong (greater than 60 bits). The algorithm considers character diversity, penalizes common patterns (e.g., "1234" or "password"), and ensures that no password content is ever stored entropy is computed on the fly. Additionally, organizations integrating with Google Console Directory can dynamically adjust scoring thresholds to enforce internal security policies.

2. Real-Time Validation and Feedback Algorithm:

This algorithm operates as the user types, delivering immediate, personalized feedback to help them improve their password strength. It performs real-time validation against basic complexity rules and checks against static or dynamically fetched lists of compromised passwords. Suggestions are presented contextually to encourage the use of diverse characters, avoid dictionary-based or sequential patterns, and increase overall length. A strength bar, dynamically synced with backend entropy evaluations, provides visual feedback to enhance the user experience. The underlying rule-based heuristics ensure that this guidance remains consistent and reliable across sessions and platforms.

3. Google Directory Policy Enforcement Algorithm:

When integrated with Google Console Directory, this algorithm enforces enterprise-specific password policies with precision. It authenticates users via OAuth2 and JWT before initiating password analysis. Once authenticated, it ensures compliance with organizational mandates such as enforcing minimum entropy scores, restricting password reuse, and requiring password updates after defined intervals.

All user activity related to password changes is logged securely in accordance with Google Workspace security protocols, without storing any raw password data. This system can also initiate compliance workflows—such as prompting for mandatory password resets—based on evaluation outcomes.

4. Secure Transaction & Logging Algorithm (Node.js + Google Sheets):

To ensure secure handling of password evaluation metadata, this algorithm facilitates anonymized logging to platforms like Google Sheets or Google Cloud Storage. It records entropy scores, feedback, and usage patterns while omitting any actual password strings. All data transfers are protected by token-based authentication, enabling real-time dashboards that highlight potential vulnerabilities and track password hygiene at scale.

5. Frontend & Backend Security Enforcement Algorithm:

This comprehensive security algorithm spans both the frontend and backend layers. On the frontend, JavaScript-based controls ensure that user inputs are clean, conform to formatting rules, and are processed exclusively in-memory without being stored or transmitted in raw form. On the backend, the system leverages Node.js and Express with robust middleware such as `express-validator` for API request validation, `sanitize-html` for input sanitization, and `express-rate-limit` to guard against brute-force attempts. OAuth2 and JWT protocols govern secure API access, ensuring that only authorized users can interact with evaluation and logging functions.

These algorithms form a cohesive framework that balances robust password security with usability and compliance. By combining entropy theory, intelligent heuristics, secure infrastructure, and enterprise integration, the Interactive Password Strength Checker serves as a proactive defense mechanism against credential-based threatsempowering users to make smarter security choices while equipping organizations with the tools to monitor and enforce best practices.

IV. METHODLOGY

The methodology behind the Interactive Password Strength Checker with Entropy Calculation focuses on secure data handling, real-time evaluation, dynamic feedback generation, policy enforcement, and audit-friendly visualization. The system is designed to guide users toward better password practices while offering enterprise-level customization through integrations with Google Console Directory and cloud-based logging via Google Sheets.

1. Data Acquisition and Processing:

Password evaluation begins at the client-side interface, developed using HTML, CSS, and JavaScript, where user input is captured in real time. To maintain data integrity and security, inputs are sanitized on the frontend and revalidated on the backend using middleware like `express-validator` and `sanitize-html` in the Node.js and Express.js stack. Importantly, no passwords are stored at any point in the process. Instead, anonymized metadata—such as entropy scores, timestamp, character diversity, and overall strength rating—is securely logged into Google Sheets. This setup supports both individual assessments and broader organizational analytics. When integrated with Google Console Directory, additional authentication metadata (excluding passwords) is captured to support enterprise security enforcement and auditing without compromising user privacy.

2. Password Feature Engineering:

To assess password strength, the backend converts each password input into a rich set of features that reflect structural and statistical complexity. These include counts of character types (e.g., lowercase, uppercase, digits, special symbols), pattern detection

for sequences like "abc" or "qwerty", and entropy calculations based on Shannon's formula `Entropy = $\log_2$(R^L)`, where R represents the size of the character set and L is the password length. Additionally, passwords are checked against a list of commonly used or previously breached credentials. These engineered features form the basis for strength classification, policy compliance checks, and real-time visual feedback. For enterprise administrators, they also serve as input for refining organizational security policies and adjusting threshold rules based on actual usage data.

3. Real-Time Evaluation and Policy Enforcement:

Real-time responsiveness is a key feature of the system. As users type, transmitting password data securely for continuous evaluation. The backend immediately returns a comprehensive strength profile, including entropy score, classification (Weak, Medium, Strong), and customized suggestions for improvement. For enterprise environments leveraging Google Console Directory, organizational policies are enforced dynamically such as minimum entropy requirements or disallowed character patterns. Non-compliant submissions trigger contextual feedback and block form completion until password criteria are satisfied. Enterprises can also incorporate historical password logging or expiration tracking into their Google Workspace environment to centralize credential policy enforcement.

4. Notification and Feedback System:

User interaction is enhanced through a multi-layered feedback mechanism that informs and educates in real time. As users construct their passwords, the system delivers live, dynamic feedback directly on-screen, offering immediate suggestions and highlighting detected weaknesses. A color-coded strength bar visualizes password quality intuitively ranging from red (weak) to green (strong). Optional email notifications can be configured to summarize password strength and deliver improvement tips to users, particularly useful in enterprise setups. Additionally, admin-facing alerts are triggered when users frequently fail to meet security standards or entropy thresholds, enabling IT teams to intervene and support users in strengthening their password practices. All events are logged anonymously to Google Sheets for long-term analysis and policy optimization.

V. EXISTING SYSTEM

1. Basic Password Validation in Legacy Systems:

Traditional web systems use simplistic password checks, focusing mainly on minimum length or character type requirements (e.g., "must include one number and one special character"). These validations are typically enforced only on the frontend using basic JavaScript logic. Such static rules fail to accurately reflect true password strength or resist sophisticated brute-force and dictionary attacks.

2. No Entropy-Based Strength Analysis:

Most existing password checkers do not consider entropy, which is the mathematical measure of unpredictability. Without entropy analysis, a password like "Password123!" may pass validation but remains highly vulnerable. This lack of depth results in false positives, where users assume their passwords are secure even when they are not. As a result, security remains superficial.

3. Absence of Real-Time Feedback and Guidance:

Conventional password input forms offer little to no live feedback. If feedback exists, it is typically limited to "Weak/Medium/Strong" labels without explanation. This absence of real-time, educational feedback limits user understanding and improvement, leaving them uninformed about best practices like increasing character diversity or avoiding common patterns.

4. No Centralized Logging or Administrative Oversight:

Legacy systems rarely log password strength metrics or store analytics related to user behavior.

There is no central dashboard for IT administrators to view strength trends, detect weak usage patterns, or adjust policies proactively. Without integration into tools like Google Sheets or Google Workspace Directory, administrators are unable to enforce enterprise-wide standards or track compliance.

5. Lack of Directory Integration for Enterprise Policies:

Most password systems operate in isolation and are not linked with Google Console Directory or other identity management platforms. This disconnect limits policy enforcement, such as requiring stronger entropy thresholds for privileged accounts, or syncing password change behavior across platforms.

6. Insecure Input Handling and Minimal Backend Validation:

In many existing applications, password validation occurs entirely on the frontend without secure backend verification. Inputs may be susceptible to manipulation, bypassing weak checks using browser developer tools. Additionally, some systems inadvertently log plain-text passwords during debugging or form submissions—creating significant security vulnerabilities.

7. No Visual Data Representation or Trend Analysis:

Current systems provide no visualization of password health across users or time. Admins and users alike lack access to meaningful dashboards showing average entropy scores, common password patterns, or compliance statistics. This invisibility results in poor decision-making, no actionable insights, and missed opportunities to improve password hygiene at scale.

8. Challenges in Existing Systems:

Existing password validation mechanisms are outdated, shallow, and reactive. They rely on rigid rules rather than adaptive logic and provide no intelligent analysis or contextual feedback. Without entropy calculations, real-time interactivity, or integration with tools like Google Sheets and Google Console Directory, these systems fail to meet the demands of modern digital security—especially for organizations with large user bases or regulatory responsibilities.

VI. PROPOSED SYSTEM

The Interactive Password Strength Checker with Entropy Calculation is designed to enhance user security awareness by providing real-time password evaluation, strength visualization, and entropy-based analysis. Unlike traditional password validators that rely solely on static rules, this system applies dynamic entropy metrics, structured feedback modules, and visual indicators to promote the creation of stronger credentials. Built with HTML, CSS, JavaScript, and powered by a Node.js and Express.js backend, it integrates with Google Console Directory and logs evaluation data via Google Sheets to support secure enterprise usage and monitoring.

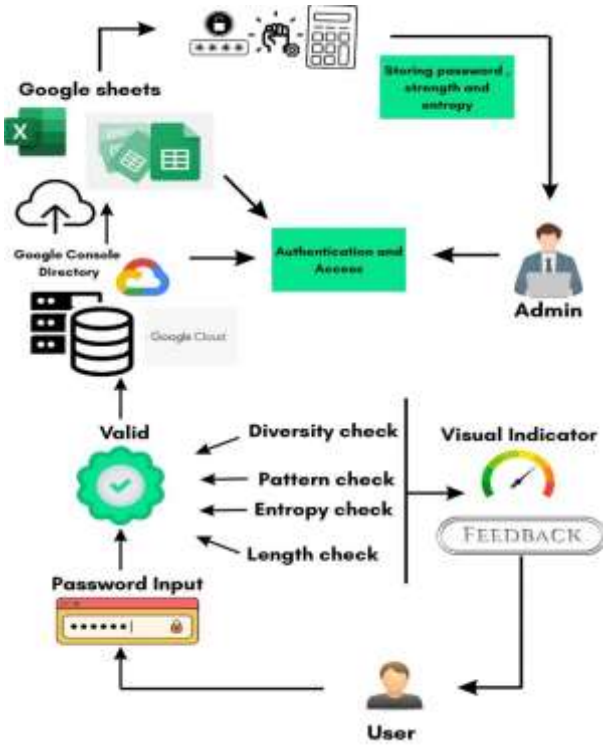

Fig. 1: Architecture diagram of Password Strength Checker

1. Real-Time Password Input and Sanitization:

The system starts with a user-facing input module where users enter their password into a web form. Input is validated using JavaScript to ensure it's not empty or malformed. To prevent injection attacks, all fields are sanitized using `sanitize-html`, and backend validation via `express-validator` ensures clean request bodies. When integrated with Google Console Directory, the system authenticates users before allowing password evaluation, ensuring that organizational policy enforcement is context-aware and secure.

2. Password Validation Engine:

The backend validation engine checks that the submitted password adheres to essential criteria:

- Minimum length (default: 8 characters).
- Not present in a commonly used or breached password list (cross-referenced via external breach databases).
- Conforms to enterprise policies if the user is authenticated via Google Workspace Directory.
- Ensures that users are not submitting easily guessable passwords like "admin123" or "password1."

This logic is executed on the server side using Express.js and is designed to be extendable based on organization or application-specific rules.

3. Strength Evaluation and Entropy Calculation Module:

After passing initial validation, the password enters the evaluation module, where strength is determined based on:

- Length Score – Longer passwords yield higher base entropy.
- Character Diversity – Use of uppercase, lowercase, numbers, and special characters.
- Pattern Detection – Identifies repetition, sequences (e.g., "abcd"), and keyboard patterns (e.g., "qwerty").
- Entropy Estimation – Uses Shannon entropy calculation to compute bits of entropy, reflecting the unpredictability of the password.

Entropy results are mapped to qualitative ratings:

- < 40 bits – Weak
- 40–60 bits – Medium
- > 60 bits– Strong.

4. Dynamic Feedback and Recommendation System:

The system generates live feedback in real-time as users type their password:

- Displays categorized strength level (Weak, Medium, Strong).
- Provides tips such as "Add a symbol," "Avoid predictable sequences," or "Increase length."
- Warns users about reused, leaked, or compromised passwords.

Feedback is standardized to ensure consistency, making the experience both educational and corrective.

5. Visual Strength Indicator Module:

An interactive visual progress bar reflects password strength dynamically:

- Red Bar (Very Weak) – Entropy is too low or pattern detected.
- Yellow Bar (Medium) – Meets minimum criteria but could be stronger.
- Green Bar (Very Strong) – Secure password with sufficient randomness.

This bar updates as users modify the password, offering instant visual feedback and helping guide better password formation.

6. Post-Submission Evaluation and Policy Enforcement:

Upon form submission, a security summary is generated:

- Displays the total entropy score and a final strength verdict.

- Suggests further improvements if the password is below a defined enterprise threshold.
- If Google Console Directory is integrated, enforces role-based password policies.

E.g., requiring higher entropy for admins or IT roles.

7. Backend and Frontend Security Framework:

JWT & OAuth2 for authentication.

- Rate Limiting using `express-rate-limit` to prevent brute-force attacks.
- Input validation & sanitization using `express-validator` and `sanitize-html`.

Frontend (HTML, CSS, JavaScript)

- Controlled input fields prevent keystroke logging.
- Real-time password feedback is securely handled in-browser.
- JWT tokens used for session management.

VI. OUTPUT

Password Strength Checker System Outcomes:

The Password Strength Checker with Entropy Calculation is a secure, intelligent system designed to evaluate password strength in real-time, educate users about secure password practices, and log relevant data for analytics. Integrated with a Google Sheets database and optionally authenticated via Google Console Directory, the system supports both end-user guidance and administrator insights

1. User Interface and Password Entry:

The main page features a responsive, user-friendly interface built with HTML, CSS, and JavaScript. Users are prompted to input a password, and the interface immediately begins analyzing and displaying results.

Displayed Elements:

-Password Input Field: Users enter the password here.

- Live Strength Bar: Visual feedback through a horizontal strength meter.
- Strength Label: Displays one of the categories: "Very Weak," "Weak," "Moderate," "Strong," "Very Strong."
- Character Breakdown: Shows count of lowercase, uppercase, numeric, and special characters.
- Visibility Toggle: Option to show/hide password input.

Fig. 2: Password Strength Checker Interface

2. Real-Time Entropy Calculation and Strength Categorization:

The system calculates the password entropy (in bits) using the formula:

Entropy = log2(poolSize^length)

- 0 – 28 bits → Very Weak
- 28 – 35 bits → Weak
- 36 – 59 bits → Moderate
- 60 – 127 bits → Strong
- 128+ bits → Very Strong

3. Smart Suggestions and Validation Results:

The system validates password quality and provides real-time suggestions for improvement.

Feedback Includes:

- Messages like: "Try adding a special character," or "Password is too short."
- Icons or colors for unmet rules (e.g., No uppercase).
- Final verdict summarizing overall password quality.

Password Policy Checks:

- Minimum length (e.g., 8 characters)

- At least 1 uppercase, 1 lowercase, 1 number, and 1 special character
- No repetitive or sequential characters (optional rule).

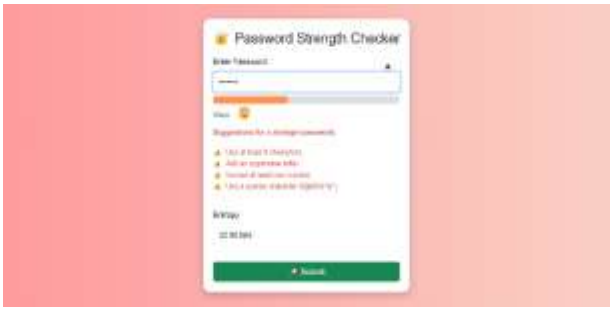

Fig. 3: Password Strength Checker Interface

4. Logging to Google Sheets Database (via Node.js Backend):

When a user submits a password for evaluation, the backend built with Node.js (Express) processes the entropy score and logs relevant metadata into a Google Sheets database using the Google Sheets API and service credentials from the Google Console Directory.

Fig. 4: Google Console Directory

VII. CONCLUSIONS

The Password Strength Checker with Entropy Calculation offers a robust, educational, and data-driven solution to improving digital security practices. By combining real-time entropy analysis with user-friendly visual feedback and integrated data logging, this system helps individuals and organizations create stronger, more secure passwords while promoting awareness of password strength metrics.

1. Accurate Password Strength Evaluation:

The system leverages entropy-based calculations to deliver an objective and precise measure of password strength. It analyzes character types, length, and overall complexity to generate a secure entropy value, categorizing each password from "Very Weak" to "Very Strong." This method goes beyond basic pattern checks, offering deeper insight into actual password robustness.

2. Real-Time Feedback and Smart Suggestions:

Users receive instant feedback on their password quality as they type. The strength bar, accompanied by live entropy scoring, offers a visual cue to motivate users toward stronger password creation. Smart tips and unmet rule highlights guide users to improve their passwords effectively and immediately.

3. User-Friendly and Accessible Interface::

Built with HTML, CSS, and JavaScript, the front-end interface is responsive, clean, and intuitive. Whether accessed on desktop or mobile, users can easily interact with the tool, view password evaluations, and understand the security of their input. The design ensures usability for both technical and non-technical users.

4. Data Logging and Analytics Integration:

The backend, developed with Node.js and Express, integrates with Google Sheets through the Google Console Directory API, securely storing metadata such as entropy values, password lengths, and timestamps. This logged data enables administrative users to monitor trends, assess overall system usage, and visualize strength distribution over time.

5. Enhancing Security Awareness and Best Practices:

Beyond evaluating passwords, the system serves an educational purpose—teaching users the principles of secure password creation and the

importance of entropy. By encouraging better practices, it contributes to a more security-conscious digital environment, helping mitigate the risk of password-based breaches.

6. Scalable and Adaptable for Future Enhancements:

Thanks to its modular architecture and cloud-integrated backend, the system can be expanded with features such as user login history, password reuse warnings, or machine learning-based suggestions. It provides a strong foundation for future development in the field of cyber-security tools.

ACKNOWLEDGMENT

I truly value **Dr. Harold Robinson Y** advice and mentoring, as her knowledge and assistance have been invaluable in forming this study. Her insightful observations have significantly improved the study's quality and depth.

I am also appreciative of everyone who shared their thoughts and offered helpful criticism. A particular thank you to everyone who voluntarily contributed their time and ideas, providing vital information that made this study stronger. Their input has been crucial in improving the study's conclusions and scope.

Finally, I want to express my sincere gratitude to my family, teachers, and peers for their unwavering support and encouragement. Their encouragement and support have been essential to finishing this work successfully.

REFERENCES

[1] Bonneau, J., Herley, C., van Oorschot, P. C., & Stubblefield, A.(2012). The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. https://doi.org/10.1145/2382196.2382198

[2] Garfinkel, S. L., & Miller, F.(2010). Password Strength: An Empirical Analysis. IEEE Security & Privacy, 8(5), 60-66. https://doi.org/10.1109/MSP.2010.143

[3] Zhang, L., & Zhang, Y. (2018). Measuring the effectiveness of password strength meters. Proceedings of the 2018 International Conference on Computer Science and Network Technology, 68-73. https://doi.org/10.1109/CSNT.2018.00018

[4] Miller, F., & Garfinkel, S. L. (2007). Password Strength: An Empirical Analysis. IEEE Transactions on Dependable and Secure Computing, 4(1), 1-8. https://doi.org/10.1109/TDSC.2007.1121

[5] Sotiriadis, S., & Deligkas, V.(2021). Secure Password Management Using Web Authentication Frameworks. Computers & Security, 100, 102123. https://doi.org/10.1016/j.cose.2020.102123

[6] Rivest, R. L. (2001). The MD5 Message-Digest Algorithm. RFC 1321. https://doi.org/10.17487/RFC1321

[7] Burr, W. E., Dodson, D. F., & Polk, W. T. (2005). NIST Special Publication 800-63-3: Digital Identity Guidelines. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63-3

[8] Sparrow, S. R., & Blackwell, E. J. (2019). Evaluating the performance of password strength meters in real-world applications. https://doi.org/10.3233/JCS-180643

[9] Shah, R., & Ramanathan, V. (2017). Real-time Password Strength Evaluation and Feedback. https://doi.org/10.1109/CSCloud.2017.40

[10] Barkley, S., & Byrne, S. (2016). An analysis of password strength meters: User behavior and system security. Proceedings of the 2016 ACM Conference on User Interface Software and Technology, 45–55. https://doi.org/10.1145/2984511.2984525

