Design and Implementation of a Secure URL Compression and Link Management Framework

Dr C V Madhusudhan Reddy[1], G K V Narasimha Reddy[2], Athuluri Naga Sai Harshitha[3], Bugude Gowri[4], Rastapur Sushma[5], Nallabothula Nikhitha[6]

[1,2,3,4,5,6] *Dept. Of Computer Science and Engineering (Artificial Intelligence), St. Johns College of Engineering and Technology, Yemmiganur, 518301, India.*

***Abstract*—The recent growth regarding web-based communication or sharing of digital content results in a significant challenge to usability, security, and effective management concerning the usage of long and complex URLs. URL compression services aim at converting lengthy web addresses into shorter forms; however, most of the existing solutions lack proper validation, access control, and analytical capabilities. This document describes the design and implementation of the framework for secure URL compression and link management, which enhances reliability, control, and usability.**

The approach that has been proposed guarantees that the availability and security of URLs are tested before shortening. Additionally, the approach caters to various forms of creating URLs, including URLs that are public, private, and custom. Expiry has also been included in this system, helping a user track their URLs that they have shortened, thus restricting their usage. Creating a QR code has also been included. Through a QR code, a user will be enabled to access their shortened URLs even if their device does not connect with a network. Based on the observations, it has been established that there are improvements in terms of security, effectiveness, and manageability.

***Index Terms*—URL Shortener, URL Compression, Django Framework, Web Security, Link Analytics, Expiry Control, URLs Authenticated, User, Dashboard, Custom, Public, Private, Account, Link, Long URL**

I. INTRODUCTION

The current internet community is built upon the use of hyperlinks to connect individuals to information, services, and other digital resources on the internet. As more and more robust applications are built on the Web, URLs provided to users consist more and more often of tens or even hundreds of characters due to the presence of query parameters, tracking IDs, and session IDs.

The practice of shortening URLs developed as an efficient way to minimize links while retaining accessibility. In spite of their effectiveness, traditional URL shortenings have mainly focused on redirects and tended to omit significant features such as security validation, access control, and lifecycle management. This raises a situation whereby users might unintentionally share dead links, expired pages, and even unsafe links.

On another note, many of the services being used right now have features that restrict advanced tools like analytics, customizations like aliases, and privacy-based features through subscription-based models.
Proposed system will definitely solve this issue because, unlike current systems, this system will provide a strong URL compression platform that will also validate URLs, restrict access, and provide analytical capabilities. The system will also be user-friendly, yet security and transparency will still remain a top priority.

II. METHODOLOGY

The process clearly mentions the workflow adopted while implementing URL shortening, validation, access control, and analysis in detail.

A. Acquisition of Input URL:

Essentially, the system commences by accepting a long URL through the web interface. Furthermore, the system allows the user to interact with the application as a guest user or an authenticated user. Thus, the

system grants the user as a guest basic functionality to shorten the URL. On the other hand, the authenticated user gets more advanced features. Input validation verifies if the provided URL is in appropriate format before proceeding with further processing.

B. URL Validation and Status Code Verification:
Once a URL is submitted in the system, the URL will be checked. It is checked if the URL is actually reachable or if it actually exists. If the URL is not actually reachable and is not an error response like "404 Not Found," then the URL will be rejected.
Additionally, it checks to see whether the destination site supports secure protocols. URLs that do not comply with security protocols may either be blocked or flagged. Hence, users do not land in unauthorized sites.

C. Short URL Generation Mechanism:
After this validation process, the system creates a short alias by applying a certain encoding standard in order to assure its uniqueness and eliminate any possible collisions. The formed alias has a certain optimized length in order to assure its compactness. There are options to use either system-generated or customized aliases that are easier for users to remember. In addition, the mappings from original URLs to shortcuts are stored in the database in a secure manner:

D. Access Control and Privacy Enforcement
Moreover, the system has given a password-based URL, which is termed a secured URL in order to increase security. If an individual intends to open a URL with security activated, the individual has to give a correct password before he is redirected from the URL. Similarly, access control based on roles allows only approved persons to change or disable their own URLs.

E. Expiry Management and Lifecycle Control:
There is a specified time set to terminate every shortened URL. When the time to terminate in terms of expiry elapses and goes past, it becomes inactive and redirects to a certain page.

F. Figure:

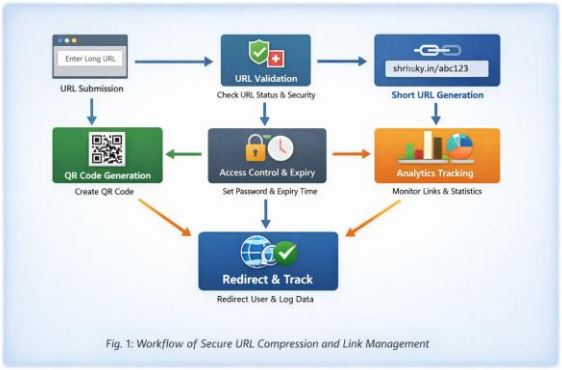

Fig. 1: illustrates the complete process flow from URL submission, validation, shortening, access control, analytics tracking, and final redirection

III. SYSTEM ARCHITECTURE

The system architecture is based on a layered system to ensure scalability and modularity.

A. Input Layer: The input layer deals with the management of user inputs like the submission of URLs, authentication of the user, and the handling of the user's requests while maintaining the security of the user's interaction with the system.

B. Processing Layer: The functionalities carried out in this stage involve URL validation, alias generation, password verification, expiry analysis, and QR code creation. It serves as the core logic engine of this system.

C. Database Layer: At the database level, the data includes user credentials like username and password, the original URL, the shortened alias, the permissions specific to the URL, the timestamp until which the URL is valid, and the analytics data.

D. Redirection and Analytics Layer: These layers manage access requests, implement access rules, control redirections, as well as manage analytics such as the clicks and the timing of the accesses.

IV. IMPLEMENTATIONS STACK

The proposed system will be developed using only software tools and frameworks. This will make the system lightweight and easier to implement.

- *Python:* This is used for backend logic and server-side operations.
- *Django Framework:* Involves URL configuration, authentication, ORM, as well as security features.
- *HTML, CSS, JavaScript:* It allows for the development of responsive user interfaces.
- *SQLite Database:* Utilized to store and manage URL mappings as well as user information.
- *Environment QR Code Libraries:* Generate Scannable QR Codes Dynamically stereo stable.

Such a stack assures portability, quick development, as well as inexpensive deployment.

V. OPERATIONAL ADVANTAGES

The system presented has a number of advantages:

- Prevents broken and invalid URL redirections.
- It improves security through validation and password protection.
- Allows custom aliasing for easier memory access.
- Offers expiry-based lifecycle control.
- Offers detailed analytics for link usage monitoring.
- Enables QR-based access for Mobile and Offline scenarios.
- Allows multi-user environment with role-based control.

VI. RESULTS AND DISCUSSION

The process of proposed safe compression of URLs and management link framework implementation with respect to providing security is done. The proposed system is implemented with respect to two types of users. The two types of users are considered as guest users and authenticated users. This will allow users with simple needs and complex needs to use this proposed system of ours.

With regard to guest users, this system has been successful in providing those users a facility of easily creating shortened URLs without requiring registration in this system, as those URLs would be available in an instant manner, creating a period of 24 hours.

A well-designed dashboard interface can give users who are logged in a chance to have various capabilities. Some of the features include: the ability to make a given link public or private or generate a custom URL. The creation of a protected link with the addition of a password made the feature more useful not only because of the security involved but most importantly due to the simplicity of the command. The addition of the feature to generate a QR code enabled the creation of seamless redirects via the mobile gadget. Lastly, there is a module that can be added for analytics. Using the analytics feature can give a chance to view the statistics of the usage of the given link created. The results indicate the need for a system that can make a URL-shortening feature much better.

A. Guest Users:

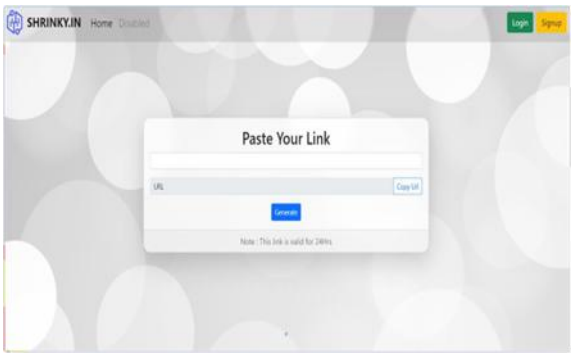

Fig. 1: It depicts the visibility for guest user, the user can short the URL by pasting the link in the specified field and should click on the generate button, if the link is valid then the short link is generated in the field, user can copy the link by clicking on copy URL button, the link will be copied to the clipboard. This link is now ready to use and is valid for 24hrs.

B. Authenticated Users:

1. Login page: The Login Page provide authorized users with the ability to log in by entering the authorized information in the form.

2. Dashboard: The Dashboard is the principal window that is exposed to users after they have successfully logged in. In other words, this is a hub where users have access to and can interact with various features within an app, for instance, generating short URLs.

3. Custom Section: The Custom Section allows users to customize their shortened URLs if they are logged in to their accounts. The users are given the option to use convenient and easy-to-remember URLs instead of randomly generated links.

4. QR Code: The QR Code Module allows authorized users to generate a QR code based on a shortened URL. This makes it possible to share a URL digitally by scanning a QR code.

C. Table:

Test Id	User Type	Testcase / Input	Expected Output
1.	User	Login: Valid Credential	Successful Login
		Invalid Credential	Unsuccessful Login
2.	End User	URL Validation: Valid URL	Generate Short URL
		Invalid URL	Prompt as Invalid URL
3.	Guest User	Validating URL Expiry: If URL not expired	Redirect to original URL
		If URL expired	Page 404
4.	Authorized User	Private URL (Auth with password): Valid Password	Redirect to original URL
		Invalid Password	Page 404

This table describes the testing of the application done for the login page or account creation page. It is tested with or without correct details to confirm that only authenticated users are able to access the application. It is also tested for the account creation feature to confirm whether a new user can successfully create an account with a new username or if the system will prompt the user to type in a different username if the username already exists. The table also gives an overview of how testing of the application with regards to the set requirements is conducted. It certifies the proper validation of URLs or the prevention of their expiry. This helps in preventing users from misusing the application or accessing invalid URLs. It offers validation of valid URLs for the purpose of shortening them to ease accessibility.

The table gives an overview of the testing procedure for URLs to confirm if they are private by incorporating a password for access. It also certifies whether access to the private link is denied upon typing in an improper password or if a user is redirected to the original link if they type in the proper password.

VII. CONCLUSION

It can be determined that the implementation and integration of the secure URL compression and link management framework were a complete success in addressing the various drawbacks that exist within traditional URL shorteners by incorporating validation, access control, customization, as well as tracking, within a single framework. The application allows users, both old and young, an interface that seems friendly to use, considering that the URLs provided are not only useful for easier sharing but also tracking.

This security concerns related to malicious redirection are eliminated by the URL validation mechanism as well as the password protection features. Analytical insights gained through the link performance provided by such a system reduce the complexities. Thus, the above-discussed link management system is found to be efficient in terms of providing improved security along with the scalability required to manage the link in the modern web environment.

REFERENCES

[1] Rohit Sankhala, Manan Kharbanda, Ankit Yadav, Pardeep Suthar, Parampreet Kaur, A URL Shortening Service Project in 2022.
[2] Coleman, J. and. (2013), "Putting QR codes to the test", New Library World, Vol. 114 No. 11/12, pp. 459-477. https://doi.org/10.1108/NLW-05-2013-0044 [3] Michael A. Veronin, ‘URL 404 File Not Found’ in 2009.
[3] Sandy J.J. Gould a,n , Anna L. Cox a, Duncan P. Brumby a, Sarah Wiseman, A systematic exploration of design trade-offs in link shortening services in 2016.
[4] Michael A. Veronin, ‘URL 404 File Not Found’ in 2009.

[5] Daejin Choia,1, Jinyoung Hanb, Selin Chuna, Efstratios Rapposc, Stephan Roberts, Ted Taekyoung Kwona, Bit.ly:Uncovering content publishing and sharing through URL shortening services in 2018.
[6] M. Alajmi, I. Elashry, H. S. El-Sayed and O. S. Farag Allah, "Steganography of Encrypted Messages Inside Valid QR Codes," in IEEE Access, vol. 8, pp. 27861-27873, 2020, doi: 10.1109/ACCESS.2020.2971984.
[7] Gibbs, M. (2009). URL shortening: * A list of URL shorteners worth checking out. Network World (Online).
[8] Hyung-Jin, M., & Li, Y. (2017). Secure short URL generation method that recognizes risk of target URL. Wireless Personal Communications, 93(1), 269-283.
[9] Munro, J. (2004, Dec 28). Cut URLs down to size; long URLs can be unwieldy. here's how to make them easy to use.: The independent guide to IBM personal computers. PC Magazine, 23, 84.