ANURL : An Innovative Management Scheme for Web Uniform Resource Locators

Ashish Karn[1] , Suyash Thakur[2], Pankaj Badoni[2,*]

[1]MultiPhase Flows Laboratory, Department of Mechanical Engineering, School of Engineering, University of Petroleum and Energy Studies, Energy Acres, Bidholi, Dehradun, Uttarakhand, India.

[2] School of Computer Science, University of Petroleum and Energy Studies, Energy Acres, Bidholi, Dehradun, Uttarakhand, India.
Email: pbadoni@ddn.upes.ac.in

Abstract. The developments in the Internet, social media platforms, and cloud technologies have necessitated the need for sharing online contents of various kinds. Despite a slew of popular shorteners for uniform resource locators (URL) that are available such as bitly and tinyurl, we delineate the need for a more comprehensive URL management scheme. The current paper reports the development of an innovative and comprehensive management scheme for web uniform resource locators (URL) named ANURL. The paper presents a detailed account of its features and its several use cases, which makes it stand out over the conventional ones. A brief description of the ANURL architecture and its working is provided. Finally, some preliminary data on the tool evaluation and user testing is presented to substantiate its growing user base with geography and time.

Keywords: ANURL, Uniform Resource Locators, URL, short URL, bitly, tinyurl.

1. Introduction

In recent times, the developments in Internet and Cloud technologies and the burgeoning growth of social media platforms has facilitated an easy sharing of online content such as images, videos, e-books, newsfeed, product specifications, shopping lists, commercial websites, educational quizzes, and assignments and personal and professional profiles of individuals. Usually, such online content is shared via social networks, news media, blog posts, online communities and postings, instant messaging, and other web services by individuals and organizations [1]. In all such interactions, a *Uniform Resource Locator,* or URL, is or paramount importance, and it signifies a specific location of a resource on the web. Put simply, a URL is equivalent to a web address and it has two important components: the first part typically being a *protocol identifier* that signifies the format of data transmission being used and the second one is the *resource name* that is specific to the IP address of the resource. Further, the resource name may consist of a domain name and file path. For instance, a typical web link that hosts some digital storytelling resources on the webpage of an innovative educator in thermal-fluid sciences looks like: https://www.drkarnteaching.com/digital-storytelling. In this particular case, 'https:' is the protocol, and the rest of the portion is the resource name, with 'drkarnteaching.com' being the domain name and 'digital-storytelling' being the file path. Similarly, Internet is replete with a variety of content and web pages, each of which needs its own unique URL. In order to generate such URLs in a manageable manner, some organization and structured approach is required. Usually, this is achieved by applying slashes, dates, keywords, names of individuals or random strings, etc. [2]. However, this also poses a constraint. With the introduction of these additional strings and characters, the URL

becomes longer, difficult to memorize, reproduce while typing, or distribute to others through non-digital modes, and copying and pasting it is probably the only reliable means to work with it.

To overcome these constraints, URL shortening services have been used widely for sharing and publishing online content by providing a short equivalent URL that is redirected to the corresponding long URL by the service provider through an "HTTP 301 Moved Permanently" response [3]. Initially in 2001, when the URL shorteners first surfaced, the underlying concept behind them was to prevent the breaking of long URLs while copying text and to prevent email clients from rendering the URLs unclickable by the insertion of line breaks between them [1]. Since then, its acceptance has been slow before these became prevalent in online social networks. Now, with the proliferation of social media platforms as well as its accessibility through mobile devices, URL shorteners have almost become a requirement, partly owing to character limitations in some social media such as Twitter, etc. There are many available URL shortening services available for public use, tinyurl.com and bit.ly, being the most popular. However, there are many open-source packages and web modules that can be alternatively used in a web application within a domain, such as YOURLS, Polr, Shlink etc. In addition, others are available to Python-based web application such as djanurl and microurl [1].

However, many of these common services are limited in the features that they provide to individuals without a premium license. For instance, tinyurl.com provides individuals to create short URL, but does not provide an option to edit the landing URL, or an option for a particular user to verify the entire list of URLs he/she has created. Similarly, many URL shorteners don't provide analytics information for all the short URLs created. To the authors' best knowledge, there is no URL shortener that provides an option for the expiry of a short URL at a chosen date/timestamp. This may be useful particularly if one needs to create a short URL for sharing over a short duration of time. But, in the current scenario, a keyword selected by a user can't be later used by any other user and this may limit the choice of keywords for many. Similarly, most of these URL shorteners don't permit the keyword to be a collection of strings and slashes which can be used to present a feel of the file directory path on a particular website and facilitate easy recollection. Another important concern stems from security issues. Short URLs have often been used recently by spammers and for malicious content phishing and malware attacks since hiding their original content URLs, they are often used for sharing malicious content such as spam or phishing [4-6]. In view of these points enumerated above, there is a clear need for the design and development of a novel URL shortener scheme that can tackle these challenges with ease and provide a simpler, more secure and free alternative, not just for businesses but also for individuals. The current paper thus reports the development of an indigenous and innovative URL shortening scheme in the Indian academic setting. The new scheme for uniform resource locators, named ANURL has been designed, developed and later deployed.

2. ANURL: A description of the innovation.

Semantically speaking, ANURL is a compound word formed by the collocation of the Sanskrit root '*anu'* meaning small, and URL, which is an acronym for Uniform Resource Locators. So, ANURL does refer to a short Uniform Resource Locator on the web, and can be accessed over the web at https://www.anurl.in/. Apart from the instructions for use and contact information, ANURL sidebar presents choices for login and signup. Further, the login presents choices for login with Google account or a 'Guest Login' that does not require any information to be fed. However, considering the security of the created short links, the validity of the short links created using the guest login is limited to 48 hours. The homepage lists different features of ANURL, which can also be described as follows.

After the login, the user lands up at a place where three pieces of information have to be entered: the long URL, a date/timestamp and a short string or a keyword which is referred to as 'hash' by Bitly or more generically, backhalf. The date/timestamp is typically an instant of time

attaining which, the ANURL assignment being created by the user deems to be invalid. This may be useful, in cases where a user may want the short link to be active only for the time duration that he/she wants it to be and wishes to make it defunct later. A possible reason behind this may be the time-specific nature of the link being shortened, or simply to ensure that keywords or strings are freed up for others to choose from. This ensures that unlike Bitly, the same backhalf could be assigned to different individuals at different instances in time, if it is available later. However, this choice is optional only. If a user does not provide an expiration timestamp, his short link will always remain active and will continue to diect him/her to the chosen URL and no one else can be granted the same backhalf. Unlike tinyurl, ANURL does not generate a randomized backhalf. However, ANURL does provide a very unique feature of editing the short links and its connection to the parent web link. So, a user can redirect the same short link to different locations at a later date. Conversely, a user can change the backhalf to refer to the same long URL. A user also has the option to delete his short URL and its assignment. The user can also track the number of clicks on his short link, and can copy the link or directly click it as well, to check its functionality. Further, as Figure 1 shows, one can also search a particular backhalf in the list of assignments one may have made in the past. This facilitates easy management of all short links in one place: search, check, edit, relink or disable.

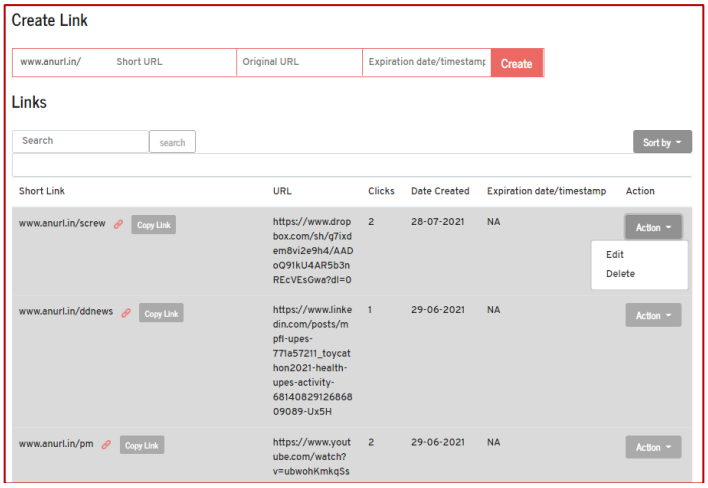

Fig. 1. A Snapshot of the ANURL interface upon logging in, showing some sample created ANURLs, alongwith the details.

3. ANURL architecture and working.

The entire ANURL coding is done in Javascript, and the technology stack used to create the application is described as follows. For the server side programming, nodeJS was employed with Express framework. As far as database is concerned, mongoDB, which is a nosql database was utilized. For frontend, Ejs, a templating language was used. The frontend was server side rendered and sent to the client side as payload to the API requests. The technical stack of ANURL is shown in Figure 2 and can be understood as follows. A user interacts with the front end, which then sends a HTTP request to the Express server with all the necessary payload, followed by the Express server making a query to the database server via Mongoose. Database holds all the data for the application.

Next, we diagrammatically present the four processes of user signup, user login, creation of a short URL and the process of navigating original URL. In the user signup process, a user

enters email and password and the password is hashed at the server with a secret value and finally the hashed value along with other user details is stored in the database. The database returns a user JSON object to the server, which then shows up a success message for the user.

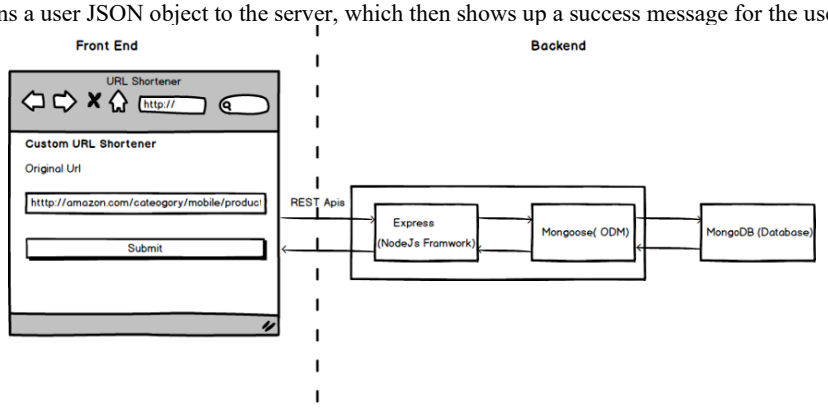

Fig. 2. Technical stack of ANURL, clearly showing the front-end and the back-end

As shown in Figure 3, while authentication (i.e. login), the user enters email and password, and server checks if the user with a given email exists in the database. Then, the server hashes the password entered and checks if the hash value matches with the password in the database. The database communicates to the server with a user JSON object, and upon successful login, an authentication token is generated.

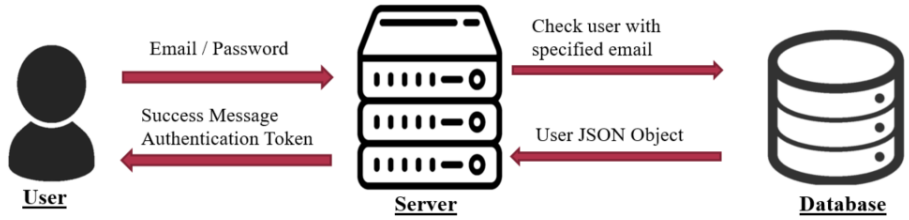

Fig. 3. The user login procedure.

Figure 4 illustrates the procedure of the creation of a short URL. Typically, a user enters all the required fields to create a short URL, whereafter the front-end sends the HTTP request to the server with the required information as payload in the request body along with the authentication token. Then, the server middleware verifies the token and the server creates a new entry in the URL collection of the database which contains reference to the user collection.

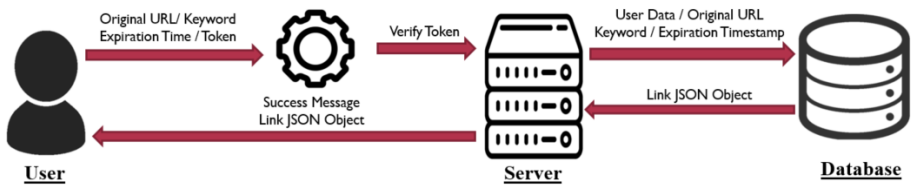

Fig. 4. The process of creating a short URL.

Next, Figure 5 demonstrates the set of steps involved in navigating the original URL. When opening a short URL, the keyword is extracted from the URL as params. In the case of a directory-like route (for instance, "anurl.in/keyword1/keyword2") all the keywords are extracted and concatenated to create a single searchable string. Then, the server sends a query to the database to extract the requested URL mapped to the given keyword and redirects to the original URL.

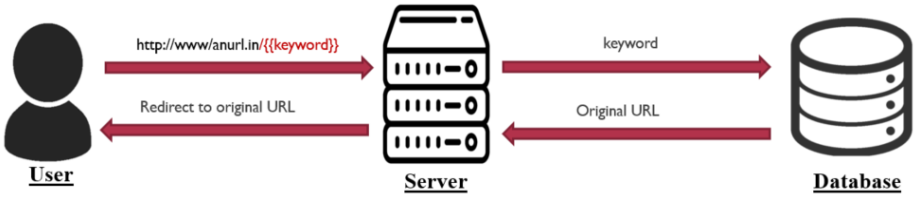

Fig. 5. The process of navigating a short URL.

Figure 6 presents a schema of the database in the current study. Although nosql database does not have a strict schema but a visual representation helps to understand the type of data stored in the collection. There are broadly two collections: User and the Link. Every unique user has an entry in the User collection and every link created has an entry in the Link collection. To map Link collection to User collection there exists a field in the Link collection called author which is of type ObjectId and has reference to User collection. The author field contains the Id of the user that has created the link (similar to a foreign key in SQL database). The TTL field in the Link collection is used to track the expiration time of the links.

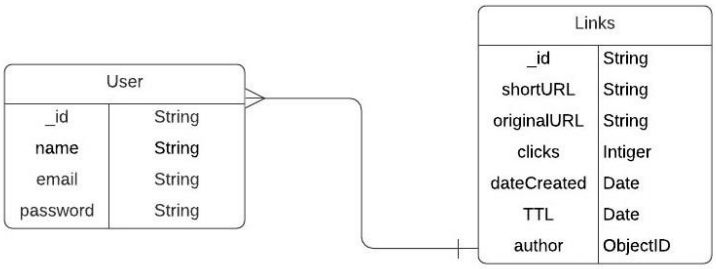

Fig. 6. The schema of the database.

4. ANURL Use cases

ANURL provides individuals with a way to shorten links to get a manageable and readable URL to share content [3, 5-6]. These links are not just personalized but easy to recollect and distribute to others at any opportune moment. An obvious use-case is that of microbloggers, who use short URLs to share content in their microblogs that have a specified character length limit, say, a limit of 140 characters as in Twitter [3]. In addition, users who want to get rid of semantics from an original URL may use ANURL for content sharing. In general, users can choose to create a customized backhalf that are easy to recollect, as long as the keyword (or backhalf) is not already taken. For instance, "anurl.in/rajcv" is a customized backhalf for a resume of a person Raj. An instructor, when asked for his calendar may direct his students to "anurl.in/calendar", or when asked orally for a meeting link, may simply text "anurl.in/zoom".

There is another interesting feature as far as customization of short URLs is concerned, what we call as “URL directories”, that may make dealing with the URLs much simpler, particularly for individuals and small organizations owing to the absence of ready availability of such tools. Apart from characters a-z, A-Z, and digits 0-9, ANURL allows insertion of ‘/’ symbols in the short links, allowing to form strings that appear as directories and filepath. For instance, an individual by the name of ‘Hari’ can store his pictures in this way – “anurl.in/hari/codes/year1” (alluding to Hari’s codes from year 1), “anurl.in/hari/research/year2” (directs to a cloud folder that houses all research done by him in the second year) etc. This can provide great ease of tracking in organizations such as the list of research publications in different departments of a university: “anurl.in/mech/publications” (Publications of Mechanical Engineering department), “anurl.in/physics/publications” (a reference to publications of physics department). Or, for that matter, in storing the repositories of a conference such as “anurl.in/icissc/2021/day1/ppts” may be used to store all the presentations from day-1 of the conference ICISSC-2021. Similarly, “anurl.in/2020/day3/papers” may be used to refer to all the papers related to the research presented at third day of ICISSC-2020.

The URL editing feature of ANURL can serve a number of useful applications in scenarios when the same link can be used for a variety of chronological operations. For instance, in the submission, online review, and result declaration for a proposed research call, the same link “anurl.in/project21” may point one to the webpage for submission of the proposal before the submission deadline, to a link for a Zoom meeting for the online evaluation, and later to the document showing the final results. This feature also ensures that, if required, the links can be shared with others even before the final link of content on the web is known (for instance, a Youtube video). This can prove to be very useful in situations of strict deadlines where content has to be submitted. In such scenarios when the evaluation of the submitted content is clearly a later event, ANURL provides a sound buffer time. Similarly, ANURL can provide a unique link for all meetings of a department, even if the meeting can be scheduled by different people at different times, provided the backend correction is made.

The analytics feature of ANURL can help one to check the number of hits on a short URL and may reveal the accessibility and extent of use of particular online content. This can be profitably used by an instructor, for instance, in gauging the student's response to a particular theme or aspect of the course. Similarly, the ANURL allows one to set an expiration date/time stamp for every URL one creates and deactivate a URL when required. This may again prove very useful in educational settings for instance, to auto-set deadlines for a task. By setting up an expiration timestamp, an instructor can automatically ensure the closure of the submission of assignments since the short links may not remain functional later.

However, an objection may be raised regarding the security concerns in the usage of short links such as ANURL, since short URLs are convenient tools in the hands of spammers and attackers who hide original URLs in this way [5]. However, multiple steps have been taken to ensure the security of ANURL short links. First, the Google sign-in makes sure that user authentication is done, and minimizes the chances of misuse to some extent. However, in order to minimize the misuse by guest login, the validity of the short URLs is limited to a maximum of 48 hours. In addition, a rate limit has been applied to the API to protect it from spam, with the current limit being 100 requests per five minutes. This ensures that if any short link receives more than a hundred hits in five minutes, the short URL is automatically blocked at that particular instant. These features greatly curtail the scope of misuse of ANURL by spammers.

5. ANURL deployment and testing: survey and statistical analysis

Finally, we ran behavioral testing to ensure excellent user experience, seek feedback on the novelty, working, and usefulness of the tool, and also collect inputs that could be incorporated into it. The developed tool underwent rigorous testing and evaluation by computer science students and experts through a Google Form, and sixty responses were received and analyzed. The survey queries consisted of both qualitative and quantitative responses, and the received

responses are then examined through a MATLAB script. The response of the users towards three aspects of ANURL: its conceptual clarity on the website, the usefulness of the ANURL scheme, and its ease of use, was collected. Figure 7a shows the user responses towards these in stacked bar format. In all three cases, about 40-60% of users thought that ANURL was a 'great' tool, whereas approximately 30-50% of users in all the cases claimed it to be a 'good' tool. Overall, around 90% of the users attest to the utility of ANURL with the top two responses. Upon digging deeper as to what fascinated the users most about ANURL (as per their understanding), the users had varying opinions. Figure 7b illustrates that 63% of the users liked the 'custom short URL' feature of the tool most (with its edit feature), 17% vouched for automatic deletion using a chosen timestamp and the remaining 20% approved the analytics feature.

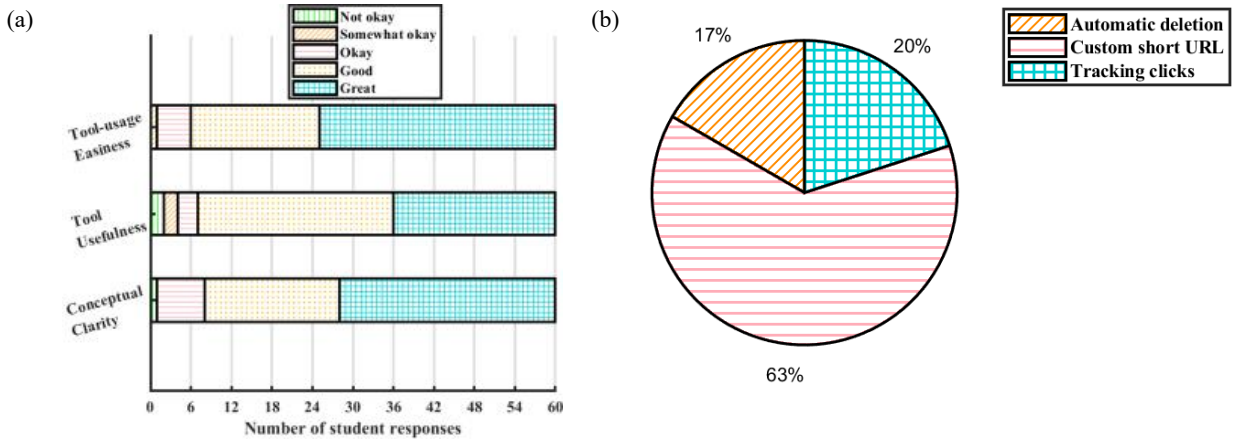

Fig. 7. : (a) General acceptance and comments on ANURL, and (b) the best feature of ANURL as per the survey results

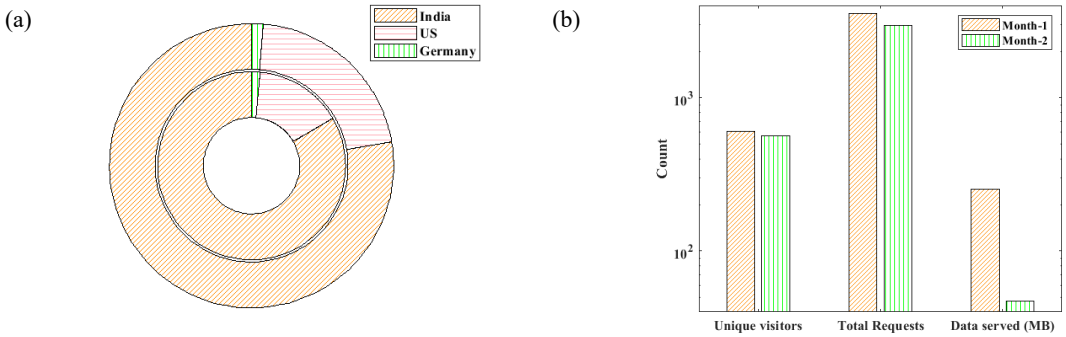

Figure 8: a) Geographical (The inner bar denotes Month-1) and, b) variation in the usage of ANURL.

Apart from the quantitative feedback, users also provided much qualitative feedback as well. Most of the users responded in affirmative lauding this new initiative, with pithy comments such as 'Good idea', 'Great work', 'Really useful website for sure', 'I found this project realy cool', 'I've seen these features for the first time', 'Amazing UI, smooth and functional! Loved it.', 'Excellent functionality', etc. Needless to say, since the entire sample set for the data collection consisted of computer science fraternity, they were quite familiar with the conventional Url shorteners. Some users provided constructive feedback of integrating the feature of guest login, a page outlining the instructions for users etc. Users also requested to

provide a direct link to redirect instantly, a feature to copy links, enhanced security features, etc., and these features were eventually integrated into the web tool. Users also commended the logo and the simple user interface of the tool.

Finally, we present some data on the usage of ANURL with respect to geography and time. The data was collected during two months labeled as Month-1 (18 April 2021 – 18 May 2021) and Month-2 (18 June 2021- 18 July 2021). Figure 8a shows that during these two periods, India, US and Germany constituted the most of the users followed by the Russian federation, Belgium, UK, and Norway. In addition, during Month-3 (18 July 2021-18 August 2021), 184 users from China and 242 users from Hongkong also registered for the tool. Further, Figure 8b shows that during the two periods total requests and the number of unique visitors remain largely the same. It can then be safely concluded that the number of visitors from the US has increased steadily from month-1 to month-2. This assertion can further be validated from Figure 9 below, which shows the expanding user-base of ANURL over all the three month periods in 2021, for three leading countries: India, US and Germany.

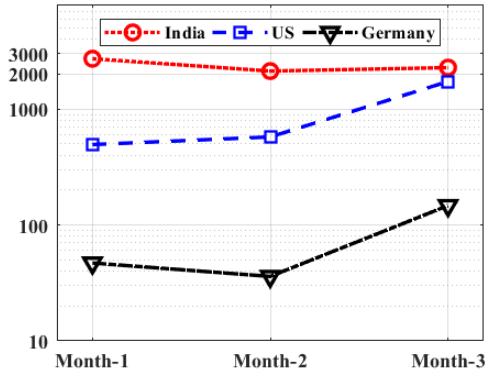

Figure 9: The increase in userbase of ANURL over three months in 2021.

It is expected that owing to its unique features, the user base of ANURL will expand in the future, particularly for individual segments. Clearly, this humble web tool emerges from an academic setting and does not target to contend with the conventional tools that are being marketed currently. Yet, the current development should be seen only as an innovative tool that was targeted to simplify things for individuals, particularly educators, who are encumbered by the conundrum of weblinks in today's digital age.

References

1. Prasetyadi, G., Hantoro, U. T., & Mutiara, A. B. (2018). Singkat: A keyword-based URL shortener and click tracker package for Django web application. *International Journal of Advanced Computer Science and Applications*, *9*(9), 118-122.
2. Azar E, Alebicto ME. Swift Data Structure and Algorithms. Birmingham:Packt Publishing. 2016: 236.
3. Antoniades, D., Polakis, I., Athanasopoulos, E., Ioannidis, S., Markatos, E.P., Karagiannis, T., 2011. we.b: The web of short urls. In: Proceedings of the 20th International World Wide Web Conference (WWW 2011).
4. Chhabra, S., Aggarwal, A., Benevenuto, F., Kumaraguru, P., 2011. Phi.sh/ocial: The phishing landscape through short urls. In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference.
5. Gupta, N., Aggarwal, A., Kumaraguru, P., 2014. bit.ly/malicious: Deep dive into short URL based e-crime detection. CoRR abs/1406.3687.
6. Nikiforakis, N., Maggi, F., Stringhini, G., Rafique, M.Z., Joosen, W., Kruegel, C., Piessens, F., Vigna, G., Zanero, S., 2014. Stranger danger: exploring the ecosystem of ad-based url shortening services. In: Proceedings of the 23rd International Conference on World Wide Web Conference (WWW 2014).