

Industrial Internship Report on URL SHORTENER

Prepared by
Ashlesha Shinde

Executive Summary

This report provides details of the Industrial Internship provided by upskill Campus and The IoT Academy in collaboration with Industrial Partner UniConverge Technologies Pvt Ltd (UCT).

This internship was focused on a project/problem statement provided by UCT. We had to finish the project including the report in 6 weeks' time.

My project was to convert long URLs into shorter, more manageable links. It generates a unique shortened URL and redirects users to the original URL when accessed. This enhances user experience and makes link sharing more convenient. URL shorteners offer tracking, analytics, customization, and branding options, improving user experience and facilitating effective link management.

This internship gave me a very good opportunity to get exposure to Industrial problems and design/implement solution for that. It was an overall great experience to have this internship.

TABLE OF CONTENTS

| | | |
|-----|--|----|
| 1 | Preface | 3 |
| 2 | Introduction | 4 |
| 2.1 | About UniConverge Technologies Pvt Ltd | 4 |
| 2.2 | About upskill Campus | 8 |
| 2.3 | Objective | 9 |
| 2.4 | Reference | 9 |
| 2.5 | Glossary | 10 |
| 3 | Problem Statement | 11 |
| 4 | Existing and Proposed solution | 12 |
| 5 | Proposed Design/ Model | 13 |
| 5.1 | Interfaces | 14 |
| 6 | Performance Test | 15 |
| 6.1 | Test Plan/ Test Cases | 15 |
| 6.2 | Test Procedure | 16 |
| 6.3 | Performance Outcome | 16 |
| 7 | My learnings | 17 |
| 8 | Future work scope | 18 |

1 Preface

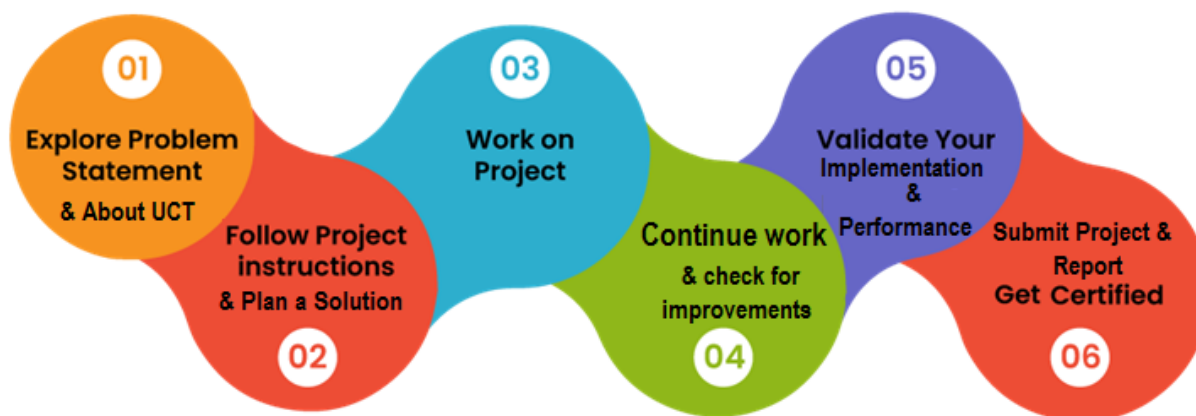
Summary of the whole 6 weeks' work.

About need of relevant Internship in career development.

Brief about Your project/problem statement.

Opportunity given by USC/UCT.

How Program was planned



Your Learnings and overall experience.

Thank to all (with names), who have helped you directly or indirectly.

Your message to your juniors and peers.

2 Introduction

2.1 About UniConverge Technologies Pvt Ltd

A company established in 2013 and working in Digital Transformation domain and providing Industrial solutions with prime focus on sustainability and RoI.

For developing its products and solutions it is leveraging various **Cutting Edge Technologies** e.g. **Internet of Things (IoT), Cyber Security, Cloud computing (AWS, Azure), Machine Learning, Communication Technologies (4G/5G/LoRaWAN), Java Full Stack, Python, Front end** etc.



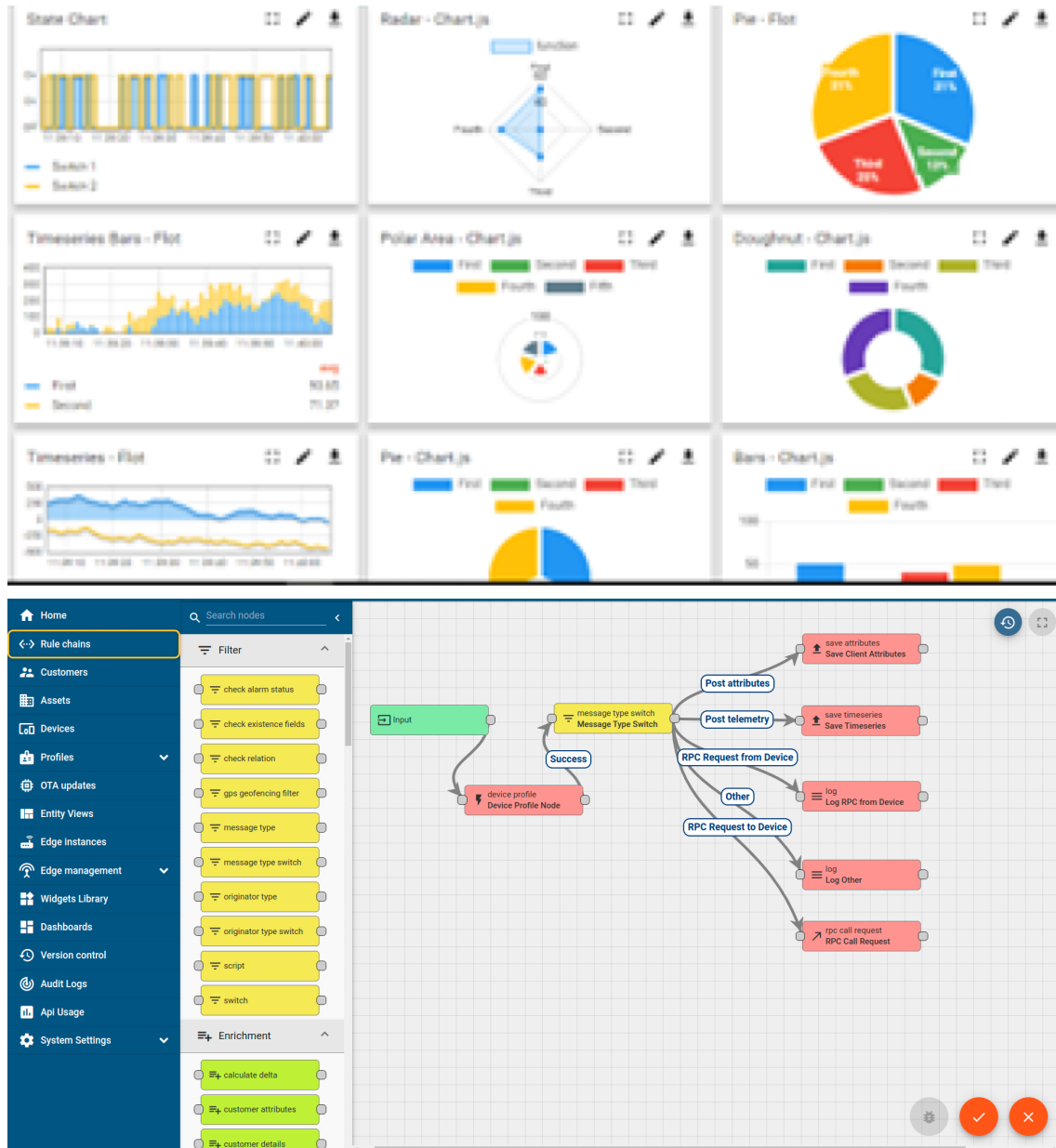
i. UCT IoT Platform ()

UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- It enables device connectivity via industry standard IoT protocols - MQTT, CoAP, HTTP, Modbus TCP, OPC UA
- It supports both cloud and on-premises deployments.

It has features to

- Build Your own dashboard
- Analytics and Reporting
- Alert and Notification
- Integration with third party application(Power BI, SAP, ERP)
- Rule Engine



FACTORY WATCH

ii. Smart Factory Platform ()

Factory watch is a platform for smart factory needs.

It provides Users/ Factory

- with a scalable solution for their Production and asset monitoring
- OEE and predictive maintenance solution scaling up to digital twin for your assets.
- to unleashed the true potential of the data that their machines are generating and helps to identify the KPIs and also improve them.
- A modular architecture that allows users to choose the service that they what to start and then can scale to more complex solutions as per their demands.

Its unique SaaS model helps users to save time, cost and money.



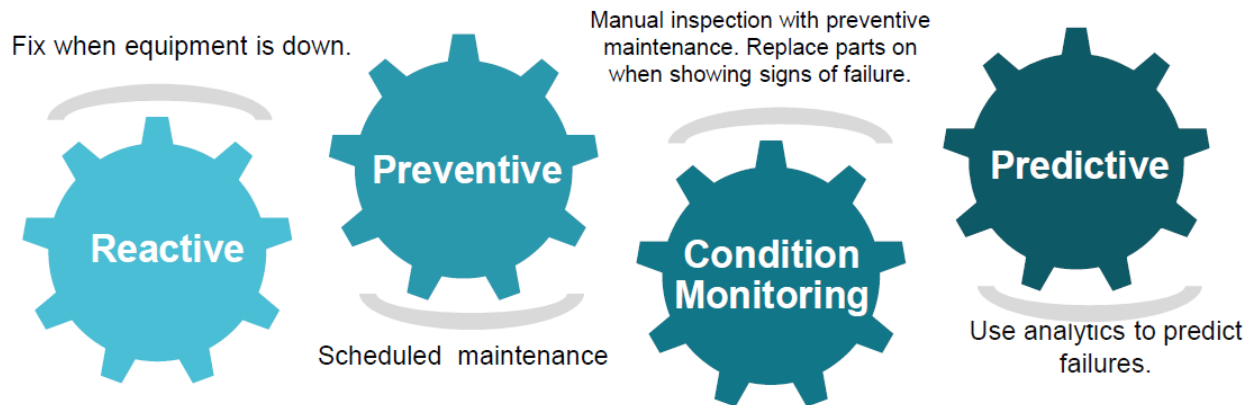


iii. LoRaWAN based Solution

UCT is one of the early adopters of LoRAWAN teschnology and providing solution in Agritech, Smart cities, Industrial Monitoring, Smart Street Light, Smart Water/ Gas/ Electricity metering solutions etc.

iv. Predictive Maintenance

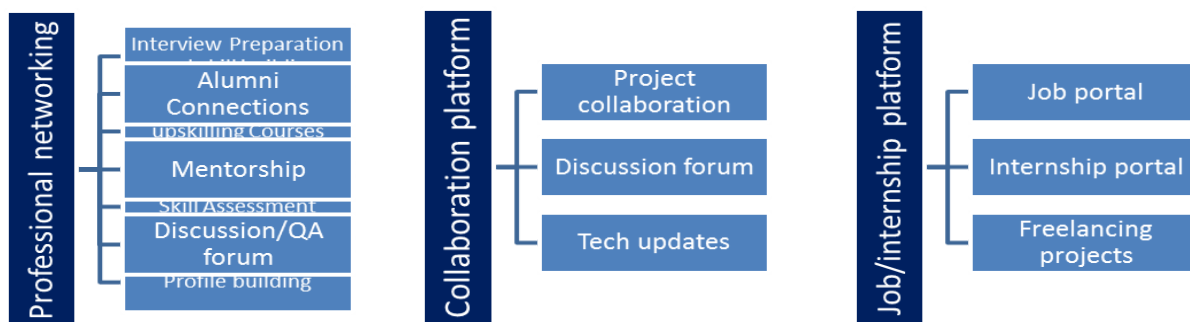
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



2.2 About upskill Campus (USC)

upskill Campus along with The IoT Academy and in association with Uniconverge technologies has facilitated the smooth execution of the complete internship process.

USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



2.3 The IoT Academy

The IoT academy is EdTech Division of UCT that is running long executive certification programs in collaboration with EICT Academy, IITK, IITR and IITG in multiple domains.

2.4 Objectives of this Internship program

The objective for this internship program was to

- ☛ get practical experience of working in the industry.
- ☛ to solve real world problems.
- ☛ to have improved job prospects.
- ☛ to have Improved understanding of our field and its applications.
- ☛ to have Personal growth like better communication and problem solving.

2.5 Reference

- [1] Babu, B. Suresh, and Sriharsha Thota Ravisankar. "A URL Shortening Service by Using Flask Framework Based on Base-62 Algorithm" (2022).
- [2] Le Page, Sophie, et al. "Using url shorteners to compare phishing and malware attacks." 2018 APWG Symposium on Electronic Crime Research (eCrime). IEEE, 2018.
- [3] Prasetyadi, Gottfried, Tri Hantoro Utomo, and Achmad Benny Mutiara. "Singkat: a keyword-based URL shortener and click tracker package for Django web application." International Journal of Advanced Computer Science and Applications 9.9 (2018).
- [4] Nikiforakis, Nick, et al. "Stranger danger: exploring the ecosystem of ad-based url shortening services." Proceedings of the 23rd international conference on World wide web. 2014.
- [5] Neumann, Alexander, Johannes Barnickel, and Ulrike Meyer. "Security and privacy implications of url shortening services." Proceedings of the Workshop on Web 2.0 Security and Privacy. 2010.

2.6 Glossary

| Terms | Acronym |
|-------|-----------------------------------|
| URL | Uniform Resource Locator |
| SMS | Short Message Service |
| AR | Augmented Reality |
| API | Application Programming Interface |
| | |

3 Problem Statement

The problem is to design and implement a URL shortener service. URL shorteners are tools used to convert long URLs into shorter and more manageable links that redirect to the original, longer URL when clicked. These shortened URLs are particularly useful in situations where character limits exist, such as in social media posts, and they also make links more visually appealing. Given a long URL, the service should generate a shorter and unique alias or key for it. This shortened URL should ideally be much shorter than the original one. When a user accesses the shortened URL, they should be redirected to the original long URL associated with that alias. Overall, the goal of the URL shortener is to simplify and optimize the process of sharing and accessing long URLs, making them more user-friendly and convenient.

4 Existing and Proposed solution

Existing system:

The existing system of URL shortener is a web-based service that takes long URLs and converts them into shorter, more manageable links. Users can paste their lengthy URLs into the URL shortener's input field, and the service generates a condensed version of the link. These shortened URLs redirect users to the original long URL when clicked. Many URL shorteners offer additional features such as link tracking, analytics, and customization options. The service is commonly used for sharing links on social media platforms, email, and messaging services to make links more visually appealing and easier to share.

Proposed system:

The proposed system of URL shortener aims to offer a user-friendly web-based platform that efficiently converts long URLs into concise links. Users can simply paste their lengthy URLs into the provided input field, and the system will generate a shortened link. The system will prioritize speed, ensuring minimal redirect time for users clicking on the shortened links. It will also include comprehensive analytics, providing users with real-time data on link performance, such as click-through rates, geographical distribution, and device types.

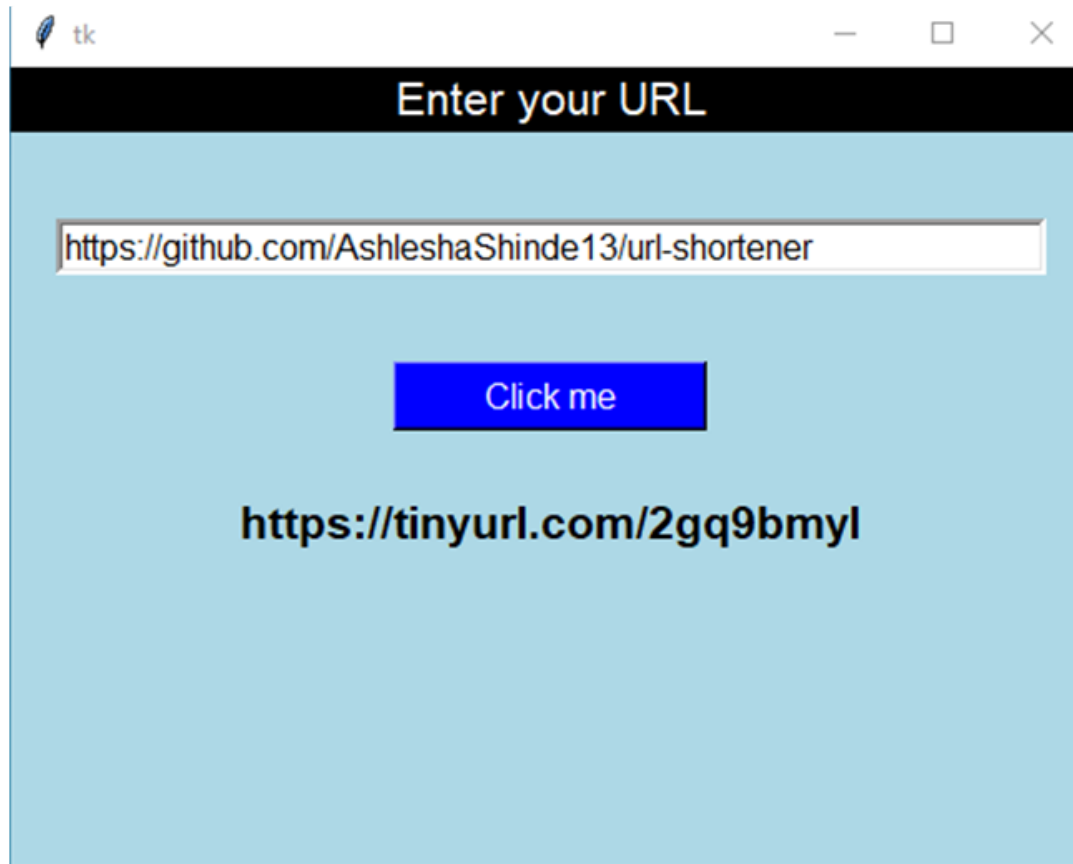
4.1 Code submission (Github link)

https://github.com/AshleshaShinde13/upskill_campus

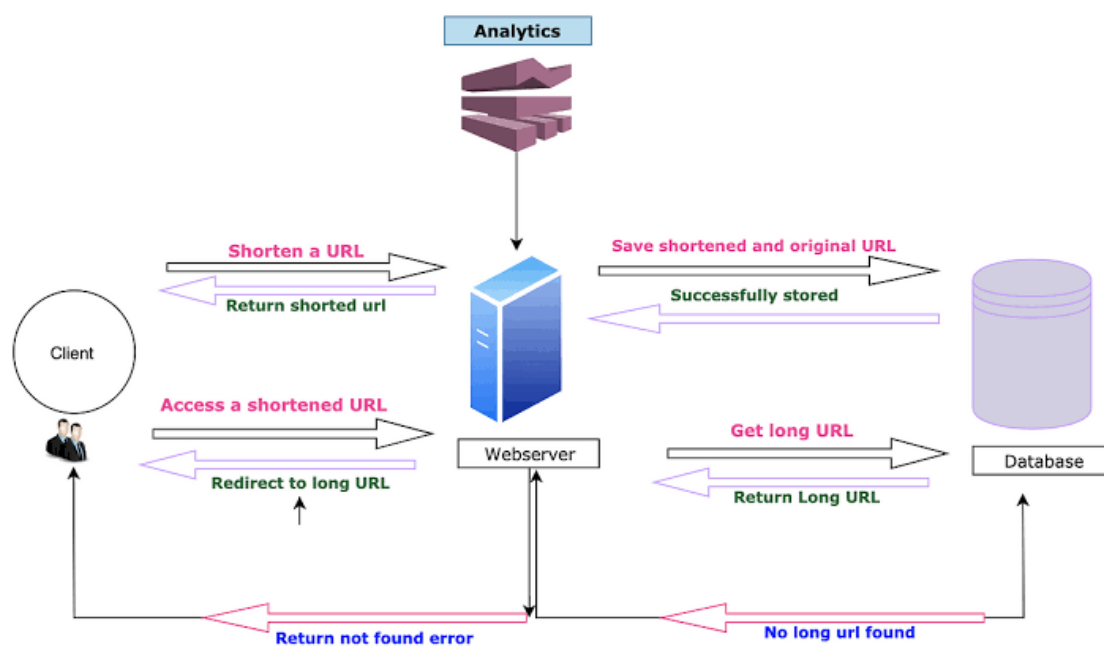
4.2 Report submission (Github link) : first make placeholder, copy the link.

https://github.com/AshleshaShinde13/upskill_campus

5 Proposed Design/ Model



5.1 Interfaces



6 Performance Test

URL shorteners have become an integral part of real industries rather than just academic projects due to their practicality and value in addressing real-world challenges. In industries like digital marketing, social media management, and online advertising, where concise and visually appealing links are crucial, URL shorteners offer a simple and effective solution. They enable businesses to share content easily on platforms with character limits, track link performance through analytics, enhance user convenience, and reinforce branding. With their widespread adoption in affiliate marketing, mobile apps, SMS communications, and offline marketing, URL shorteners have evolved into indispensable tools that streamline link management and significantly improve user experience in the digital era.

6.1 Test Plan/ Test Cases

1. Test Environment Setup:

The libraries used in the code:

- 1) pyshorteners: This library provides an easy way to shorten URLs using various URL shortening services. In this code, it is used to shorten the user-provided URL using TinyURL.
- 2) tkinter: This library is used to create the graphical user interface (GUI) for the URL shortener application. It is a standard Python GUI library and comes bundled with Python.

2. Unit Testing:

- 1) Test for Valid URL Shortening:
 - Input: Valid long URL (e.g., "https://www.example.com")
 - Expected Output: A shortened URL is generated, and it is not the same as the input URL.
- 2) Test for Shortened URL Redirection:
 - Input: A shortened URL generated from the previous test case.
 - Expected Output: The application will redirect to the original long URL.

6.2 Test Procedure

1) Test GUI Launch:

- Open the Python script containing the URL shortener code and run it.
- Verify that the GUI window opens successfully without any errors.

2) Test GUI Elements:

- Check that the GUI contains the following elements:
 - Label with the text "Enter your URL."
 - An entry box for the user to input the long URL.
 - A button labeled "Click me" to trigger the URL shortening process.
 - Another entry box to display the shortened URL.

3) Test URL Shortening:

- Enter a valid long URL (e.g., "<https://www.example.com>") in the input entry box.
- Click the "Click me" button.
- Verify that the shortened URL appears in the second entry box.
- Manually check that the shortened URL is valid and correctly redirects to the original long URL.

6.3 Performance Outcome

The performance outcome of this code primarily depends on two factors:

- 1) URL Shortening API Performance: The pyshorteners library is used to interact with an external URL shortening service (in this case, TinyURL). The time taken to shorten a URL will depend on the performance and response time of the external service. Since the code uses the TinyURL API, the performance will be affected by the speed and availability of TinyURL's servers.
- 2) GUI Responsiveness: As the code runs in the main thread and uses the `mainloop()` function, the GUI will be responsive to user interactions. However, if the URL shortening process takes a significant amount of time due to slow API response or network latency, the GUI might become unresponsive during that time.

7 My learnings

URL shortener Python project aims to create a program that can shorten long URLs into concise and user-friendly links. The project involves implementing the logic for generating shortened URLs, redirecting users to the original URLs, and storing the mappings between the shortened and original URLs. Throughout the project, it's essential to handle error cases, ensure proper URL validation, and consider security aspects to prevent abuse or misuse of the URL shortener. Overall, a URL shortener Python project focuses on implementing the logic and functionality necessary to shorten URLs, redirect users, and manage the mappings between the shortened and original URLs in a reliable and efficient manner.

8 Future work scope

The future scope of URL shorteners is likely to involve several advancements and innovations to meet evolving user needs and technological trends. Here are some potential future developments for URL shorteners:

1. **Enhanced Link Analytics:** URL shorteners may offer even more sophisticated and detailed analytics, providing valuable insights into user behavior, engagement patterns, and conversion rates. AI-powered analytics could offer actionable recommendations for optimizing link performance.
2. **Deep Linking:** URL shorteners could support deep linking, allowing users to create shortened links that direct users to specific content within mobile apps, enhancing the user experience and app engagement.
3. **Augmented Reality (AR) Integration:** With the rise of AR technology, URL shorteners might adapt to create shortened links that lead to AR experiences, bridging the physical and digital worlds.
4. **Voice Assistant Compatibility:** As voice search and virtual assistants become more prevalent, URL shorteners could integrate with voice platforms, enabling users to create and share shortened links using voice commands.
5. **Blockchain-based URL Shorteners:** Blockchain technology could provide enhanced security and transparency for URL shorteners, ensuring the integrity and authenticity of shortened links.
6. **Improved Link Management:** Future URL shorteners may offer more comprehensive link management features, allowing users to organize, categorize, and track large volumes of shortened links efficiently.
7. **Advanced Security Features:** With growing cybersecurity concerns, URL shorteners might implement advanced security measures, including automatic link scanning for malicious content and potential threats.
8. **URL Shortener APIs:** Expanding API capabilities could lead to a broader range of integrations with various applications and services, enhancing the versatility and usability of URL shorteners.
9. **Link Monetization:** URL shorteners could explore options for link monetization, allowing users to earn revenue based on the click-through rates of their shared links.
10. **Privacy and Data Protection:** Future URL shorteners may prioritize user privacy, ensuring that data collected from link analytics is handled responsibly and securely.

