

Industrial Internship Report on

"URL Shortener"

Prepared by

[Soham 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 on developing a Python Project for shortening Long URL's into shorter ones to easily to remember or share to anyone. The project seamlessly generates short URL's by converting Long URL's into short ones providing URL management and reduction.

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	5
2.1	About UniConverge Technologies Pvt Ltd	5
2.2	About upskill Campus	10
2.3	Objective	12
2.4	Reference	12
2.5	Glossary	12
3	Problem Statement	13
4	Existing and Proposed solution	14
5	Proposed Design/ Model	17
5.1	High Level Diagram (if applicable)	18
5.2	Low Level Diagram (if applicable)	20
5.3	Interfaces (if applicable)	21
6	Performance Test	22
6.1	Test Plan/ Test Cases	24
6.2	Test Procedure	24
6.3	Performance Outcome	25
7	My learnings	26
8	Future work scope	27

1 Preface

- **Summary of the whole 6 weeks' work:**

Throughout my internship journey at UniConverge Technologies, I went through planned and controlled roadmap for learning and developing my skills. The program begins with an introduction to upskilling and familiarization with the company's core values and mission. In the later weeks I had gone through a professional roadmap of learning and developing my Python skills ranging from basic concepts to gaining knowledge about large and vast libraries of Python used for development. I gained practical experience, emphasizing the application of acquired knowledge in real-world scenarios, thus preparing them for future career endeavors.

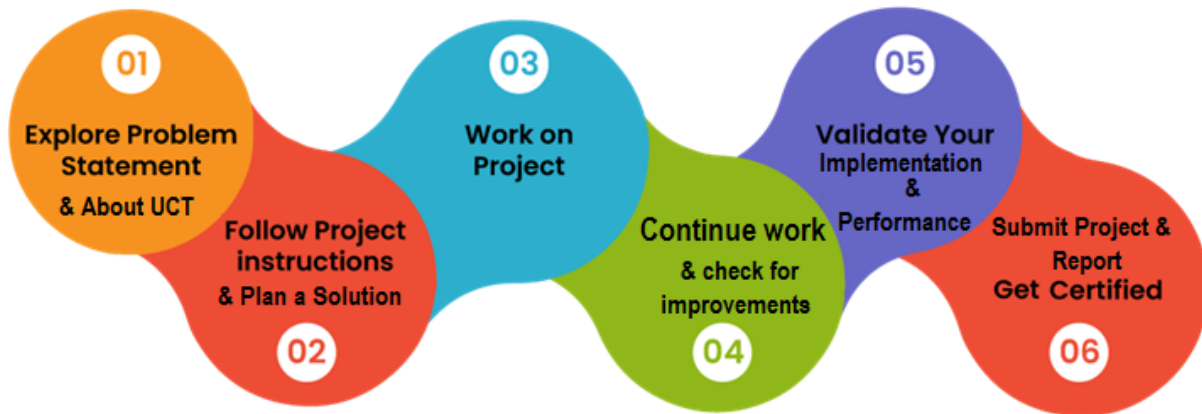
- **About need of relevant Internship in career development:**

Internships are vital for providing practical experience and give a valuable insight and working culture of industry. Internships offer an opportunity to apply theoretical knowledge gained in academic settings to real-world scenarios, thus bridging the gap between education and employment. Internships enable individuals to establish connections with professionals in their field and potentially secure future employment opportunities. A relevant internship can significantly enhance one's credibility, and employability, positioning them for success in their career journey.

- **Brief about Your project/problem statement:**

My Project was on URL Shortener using Python in which I used different Python Libraries and Python Frameworks to achieve my project goal. The main goal of my project is to Shorten the Long URL'S into Short ones for easy to maintain and sharing of URL'S.

- **How Program was planned**



- **Thank to all who have helped you directly or indirectly:**

I extend my sincere gratitude to all who have played a role, directly or indirectly, in my educational journey. I offer special recognition to the IoT Academy for their insightful videos, which significantly enriched my understanding of complex subjects. Additionally, I am deeply appreciative of UniConverge Technologies for providing me with the opportunity to intern with them. Their mentorship, coupled with an in-depth exploration of their organizational culture and expertise, has been instrumental in my professional development.

- **Your message to your juniors and peers:**

The internship gave me a valuable experience on testing and developing my Python Programming Skills. To my juniors and peers, I advise embracing every learning opportunity with zeal and diligence. Remember, each experience contributes to your growth and proficiency. Stay committed to continuous improvement and the pursuit of excellence in your chosen field. And also stay creative.

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/L0RaWAN), Java Full Stack, Python, Front end** etc.



i. UCT IoT Platform ()

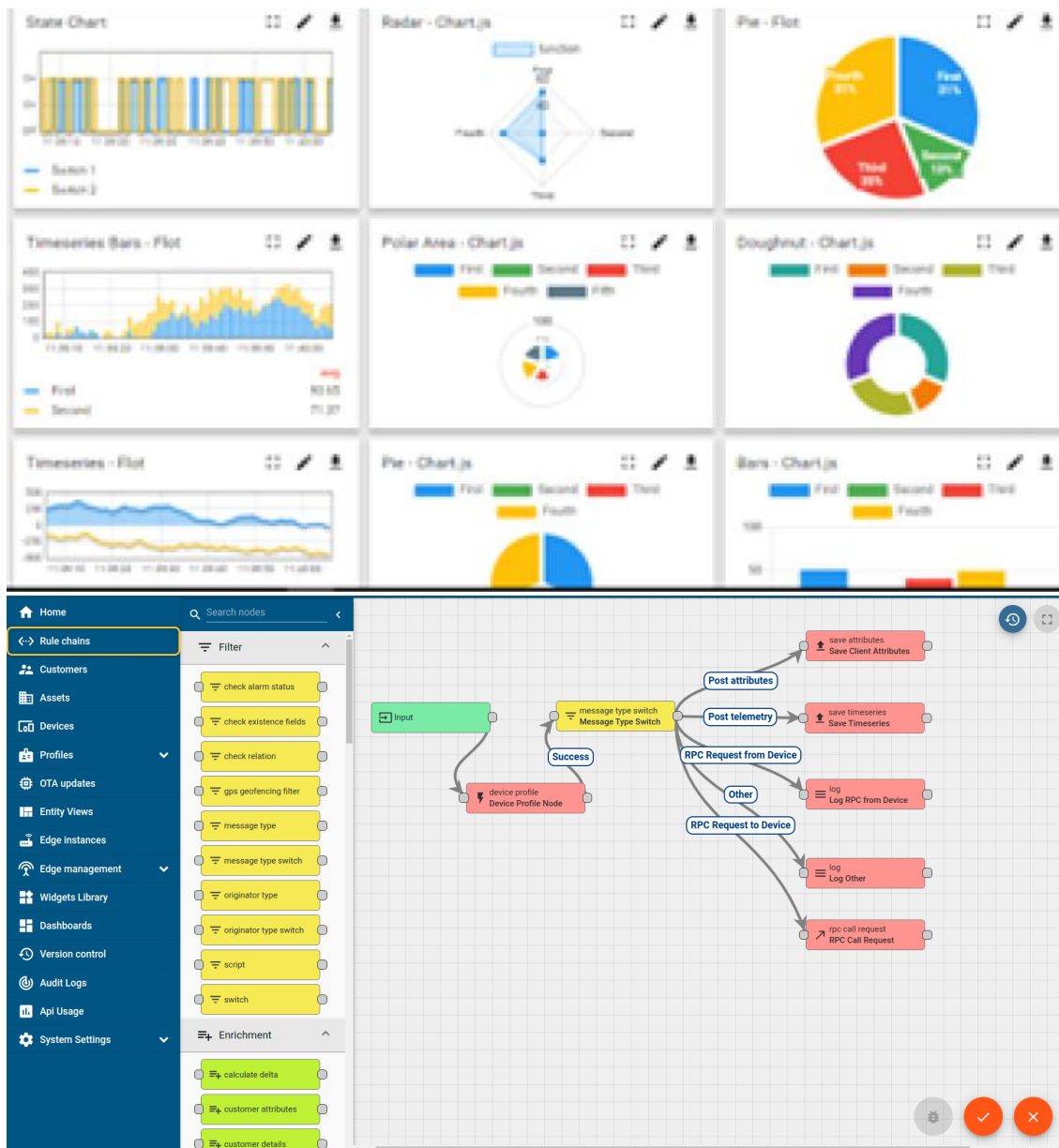
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable “insight” for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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

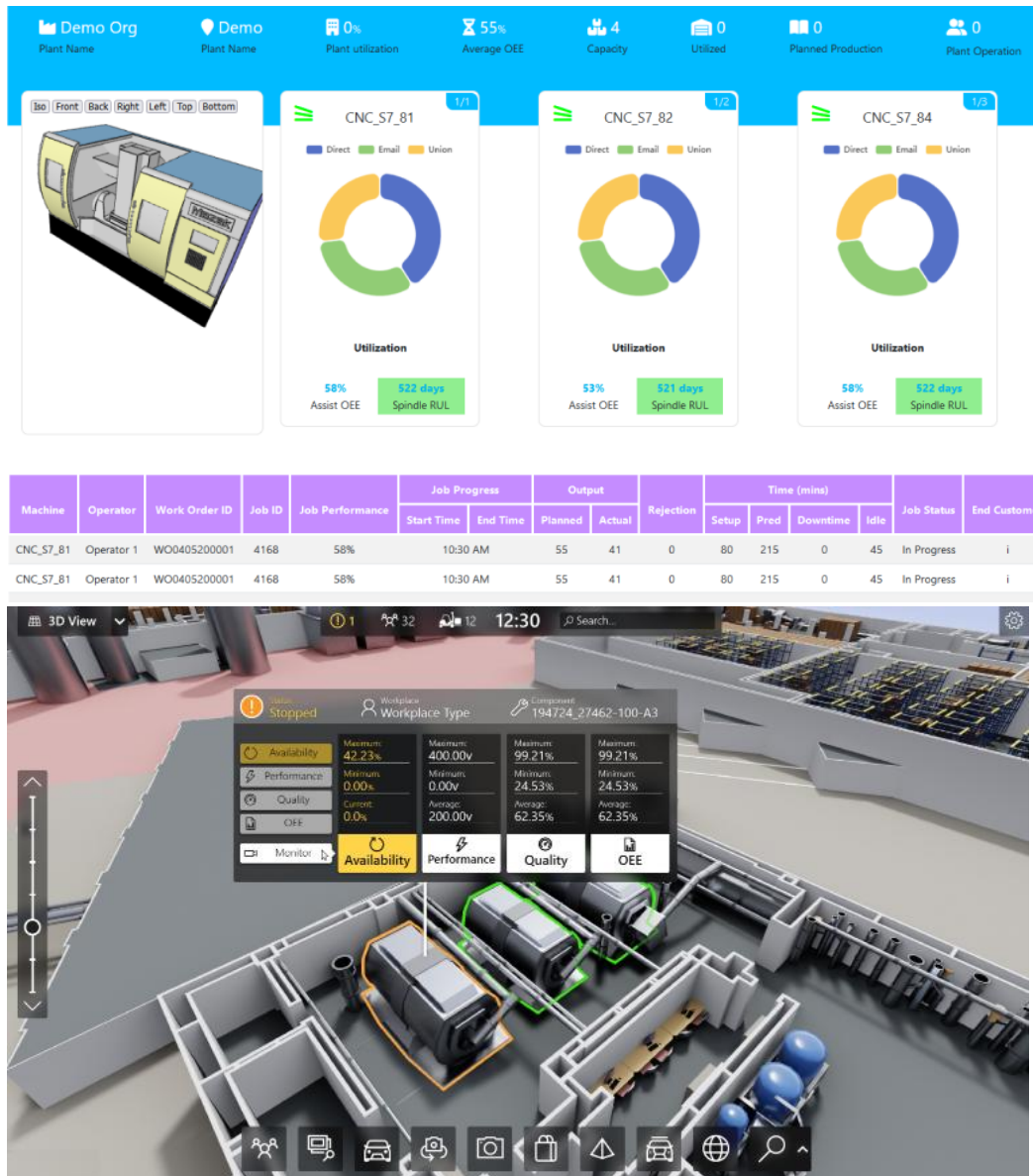
ii. Smart Factory Platform ()

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 unleash 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 want 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.



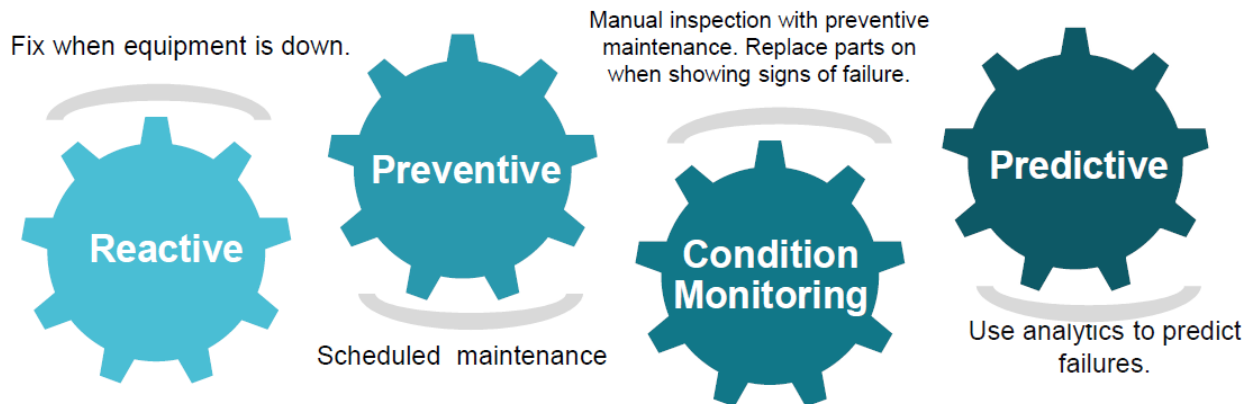


iii. based Solution

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

iv. Predictive Maintenance

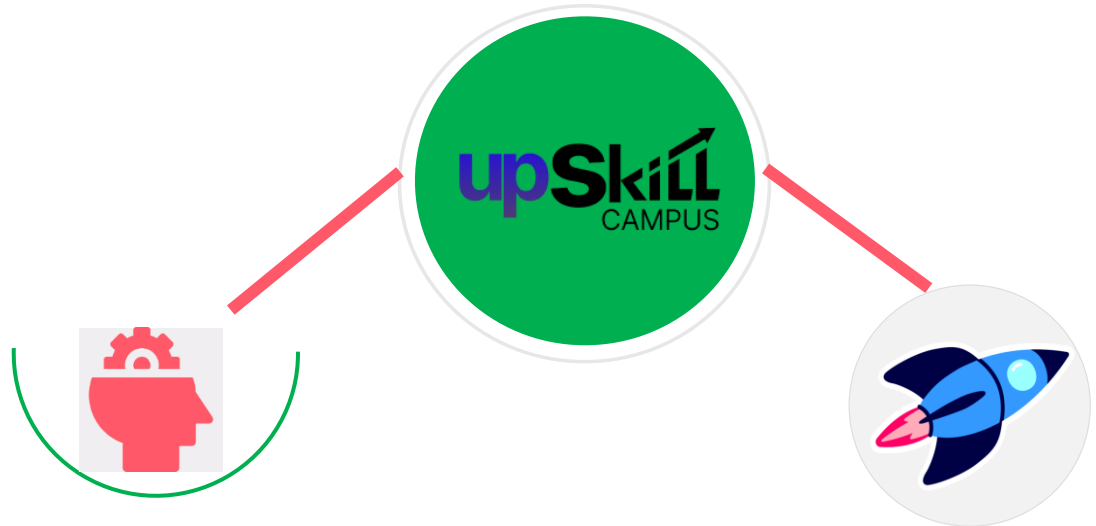
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

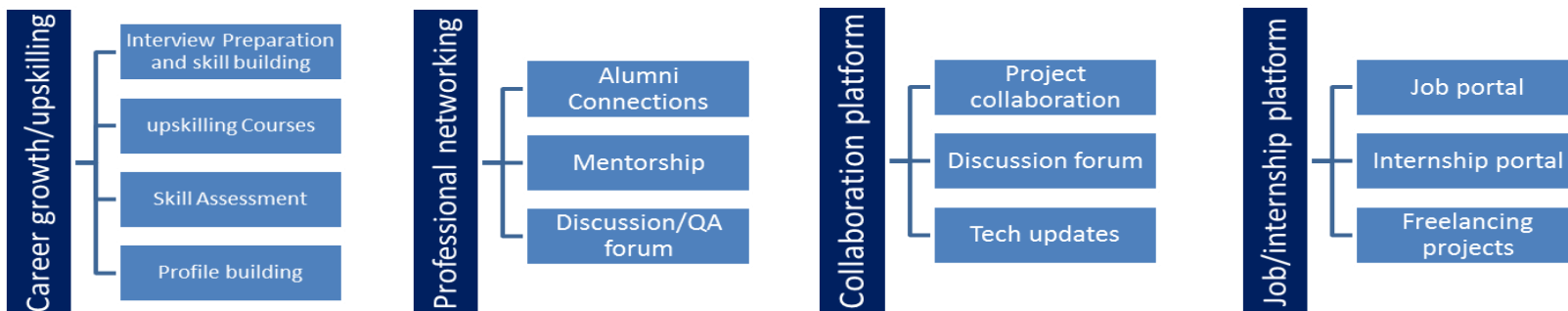
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



Seeing need of upskilling in self paced manner along-with additional support services e.g. Internship, projects, interaction with Industry experts, Career growth Services

upSkill Campus aiming to upskill 1 million learners in next 5 year

<https://www.upskillcampus.com/>



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] Python E-Learning Book

[2] IOT Academy Videos

[3] Internet

2.6 Glossary

Terms	Acronym
HTTPS	Hyper Text Transfer Text Protocol Security
CDN	Content Delivery Network
URL	Uniform Resource Locator

3 Problem Statement

The goal of URL Shortener is to Shorten Long URL'S into smaller and manageable ones for easy to remember and sharing purpose. A URL shortener is a tool that takes a long URL and produces a shorter alternative that redirects to the original URL. A URL shortener is to save space and make URLs more manageable, especially for sharing on platforms with character limits like Twitter.

The process of shortening a URL typically involves generating a unique identifier for the original URL, which is then appended to a shortened domain name, resulting in a shorter, custom URL. When users click on the shortened URL, they are redirected to the original, longer URL.

Ensuring the uniqueness of every shortened URL alias is a crucial component of the task. Due to the possibility of confusion or unintentional redirection caused by duplicate aliases, the usability and dependability of the service may be compromised. Therefore, it uses a complex hex algorithm that generates a random hex value suffixed by an "short." Text to generate the Shortened URL.

Another important factor in the "URL-Shortener" is security. Systems need to be protected from potential abuse, including shortening URLs for nefarious intent. To safeguard the integrity and dependability of the service, strong internal authentication procedures are crucial, as are strong data authentication and security best practices.

URL Shortener provides many functionalities like Space-saving, Shortened URLs take up less space, making them ideal for sharing in contexts where character limits apply, such as tweets or text messages. Improved readability, Long URLs can be cumbersome and difficult to read, especially when included in printed materials or spoken aloud. Shortened URLs are more concise and easier to communicate. Tracking and analytics, URL shortener service offer tracking and analytics features, allowing users to monitor the number of clicks, geographic location of users, and other metrics associated with the shortened URLs.

In conclusion URL Shortener enables users to create short, easy to remember and manage URL'S that meets the requirements of users. This project offers superior reliability for shortening Long URL'S into shorter ones for easy management.

4 Existing and Proposed solution

Existing Solutions to shorten the URL are easily available on internet such as:

1. Bitly:

- Custom branded short URLs
- Tracking and analytics: Provides detailed statistics on link performance, including click-through rates, geographic location of users, and referral sources
- Link management: Allows users to organize, categorize, and edit their shortened links.
- API access for integration with other applications.

2. is.gd:

- Simple and fast URL shortening service with no registration required.
- QR code generation: Allows users to generate QR codes for shortened URLs.
- Real-time statistics: Provides real-time analytics on link clicks and traffic sources.
- Bookmarklets for easy URL shortening directly from the browser.

3. Rebrandly:

- Custom branded short URLs with users' own domain names.
- Link retargeting: Enables users to add retargeting pixels to their shortened links for remarketing purposes.
- UTM tagging: Allows users to add UTM parameters to track link performance in Google Analytics.
- API access for integration with other platforms and workflows.

4. TinyURL:

- Simple and straightforward interface for quickly shortening URLs.
- Custom alias: Users can create custom short URLs to personalize their links.
- Preview feature: Allows users to preview the destination of a shortened URL before clicking on it.
- Toolbar button for easy URL shortening directly from the browser.

URL shortening services impose various restrictions or limitations on their usage, depending on the specific service provider and their terms of service. Some common restrictions that users encounter while using popular URL Shortener services are:

Limited usage: Many URL shortening services offer free plans with usage limits, such as a maximum number of shortened URLs per month or a maximum number of clicks per link. Users may need to upgrade to a paid plan to access higher usage limits.

Link expiration: Some URL shorteners impose expiration dates on shortened URLs, after which the links become inactive and no longer redirect to the original destination. Users may need to refresh or renew their links periodically to prevent expiration.

Domain restrictions: Certain URL shortening services may restrict the use of custom branded short URLs to users who have subscribed to premium plans or purchased a custom domain. Free users may be limited to using the service's default domain.

Link content restrictions: URL shortening services may have policies prohibiting the use of their service to shorten URLs containing certain types of content, such as illegal or adult content. Violating these policies may result in suspension or termination of the user's account.

Rate limiting: To prevent abuse or spamming, URL shortening services may impose rate limits on API requests or shorten URL requests from a single IP address. Users who exceed these limits may be temporarily blocked or throttled.

API access restrictions: URL shortening services that offer APIs for integration with other applications may impose usage restrictions or require API keys for access. Users may need to adhere to rate limits or usage quotas when making API requests.

Our proposed solution for the URL-Shortener project is designed to mitigate these limitations and offer a comprehensive, user-friendly, and secure URL shortening service. Key features and enhancements of our proposed solution are as follows:

- **Scalability and Performance:** Our solution will be constructed upon a scalable and highperformance infrastructure, guaranteeing swift response times and dependable service even amidst substantial demand. We will leverage cutting-edge cloud technologies and optimization techniques to efficiently manage a significant volume of URL conversions and user traffic.

- **Enhanced Security:** Security is paramount in our solution. We will employ rigorous authentication mechanisms, robust data encryption, and proactive monitoring to safeguard against potential security threats, including phishing attacks, unauthorized access, and data breaches.

In conclusion our solution to problem statement for URL Shortener gives a reliable and dependable URL shortening service that meets all the user requirements for scalability, performance and security. By filling in the patches which are already provided by existing solutions to make better from them. Project goal is to empower users to efficiently manage and distribute their links online with confidence and ease.

4.1 Code submission (Github link):

[upskillcampus/Url-shortener\(github.com\)](https://github.com/upskillcampus/Url-shortener)

4.2 Report submission (Github link) :

5 Proposed Design/ Model

1. User Interface(UI) -

The user interface design of a URL shortener provides users with a seamless and intuitive experience when shortening URLs and managing their shortened links and contains following elements:

- **Simplicity:** It Keeps the user interface clean and uncluttered, with minimal distractions. The primary focus should be on the core functionality of shortening URLs and managing shortened links.
- **Input Field:** Provides a prominent input field where users can enter the long URL they want to shorten. Include placeholder text or a label to indicate the purpose of the input field.
- **Shorten Button:** Includes a button or action to initiate the URL shortening process. Make sure it's visually distinct and easily clickable.
- **Shortened URL Display:** Displays the shortened URL prominently so that users can easily copy it to their clipboard. Consider using a copy button or providing a one-click copy-to-clipboard functionality for convenience.
- **Link Management:** URL shortener offers features for managing shortened links, such as organizing links into categories or folders, provide an intuitive interface for users to browse, search, and manage their links effectively. Include options for editing or deleting links as needed.
- **Responsive Design:** Ensures that the user interface is responsive and works well across different devices and screen sizes, including desktops, tablets, and smartphones. Design elements should adapt dynamically to provide an optimal viewing and interaction experience.

2. URL Shortening Function –

A URL shortening function using hex values involves generating a unique hexadecimal string based on the original URL. It includes Hashing the URL meaning the function first hashes the original URL using a cryptographic hash function. This produces a fixed-length hash value, typically represented in hexadecimal format. Extracting a portion of the hash meaning the function selects a portion of the hash value to use as the short identifier for the URL. This portion is usually chosen to be a certain number of characters from the beginning or end of the hash value, depending on the desired length of the short URL. Encoding the hex value meaning the selected portion of the hash value is then encoded into a hexadecimal string, which consists of characters from the set. This hexadecimal string serves as the short identifier for the original URL. Appending to the short URL domain finally, the hexadecimal string is appended to the URL shortening service's domain name, creating a shortened URL that redirects to the original long URL.

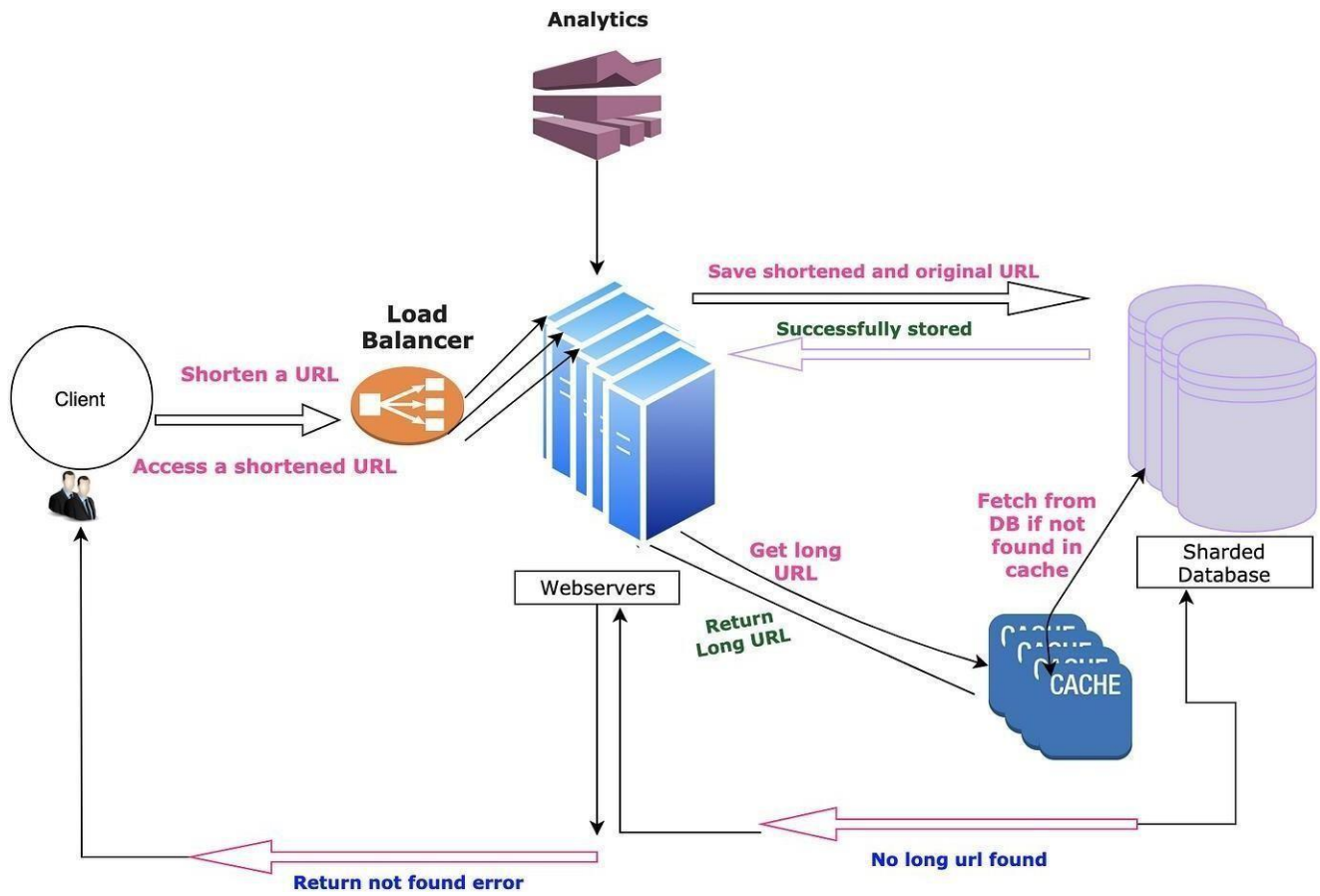
3. Security Measures: -

- Security stands as a paramount consideration in our solution's design.
- We will implement robust authentication mechanisms to ensure secure access to the service, incorporating advanced features such as multi-factor authentication and token-based authentication.
- Additionally, encryption techniques will be employed to safeguard sensitive data, such as user credentials and analytics information, during transit and while at rest.

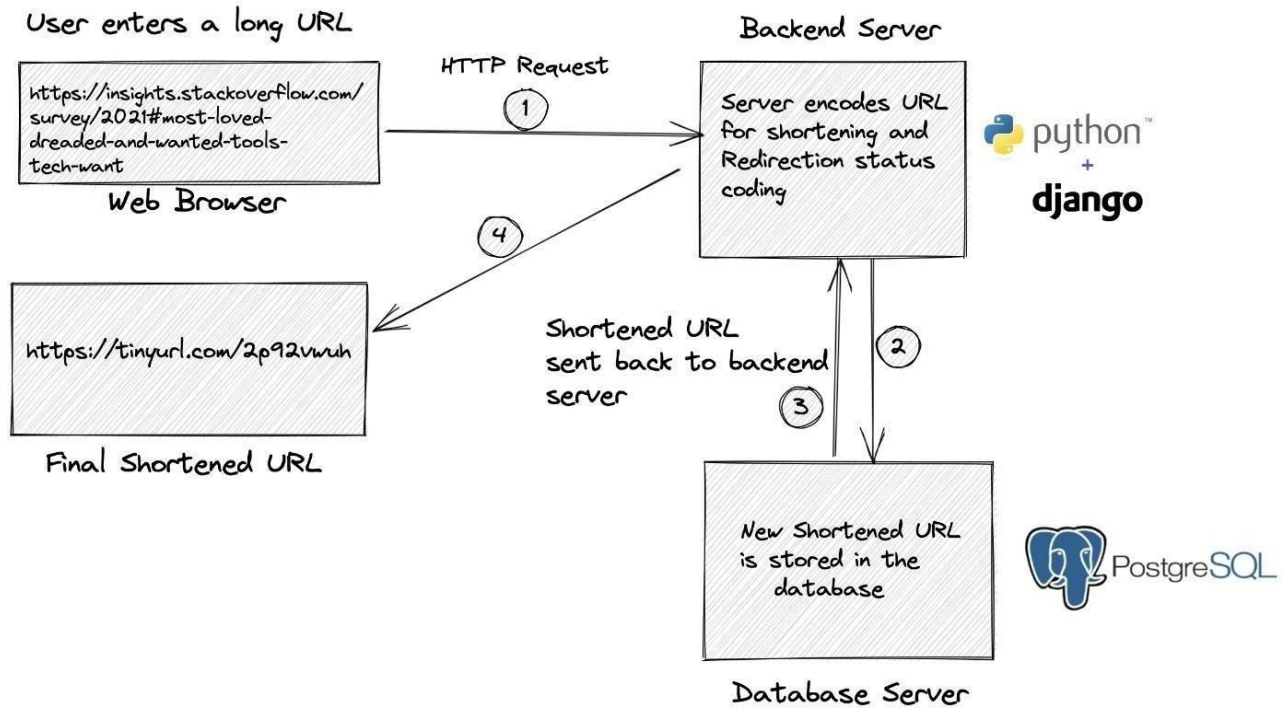
4. Scalability and Performance:

- The solution will be architected to exhibit exceptional scalability and performance, capable of seamlessly accommodating substantial volumes of URL conversions and user traffic.
- We will leverage cutting-edge cloud technologies and horizontal scaling methodologies to dynamically allocate resources in response to demand fluctuations, ensuring consistently optimal performance.
- By adhering to this meticulous design approach, our proposed URL shortening solution will furnish users with a comprehensive, user-friendly, and secure platform for shortening URLs, managing links, and effectively tracking analytics data.

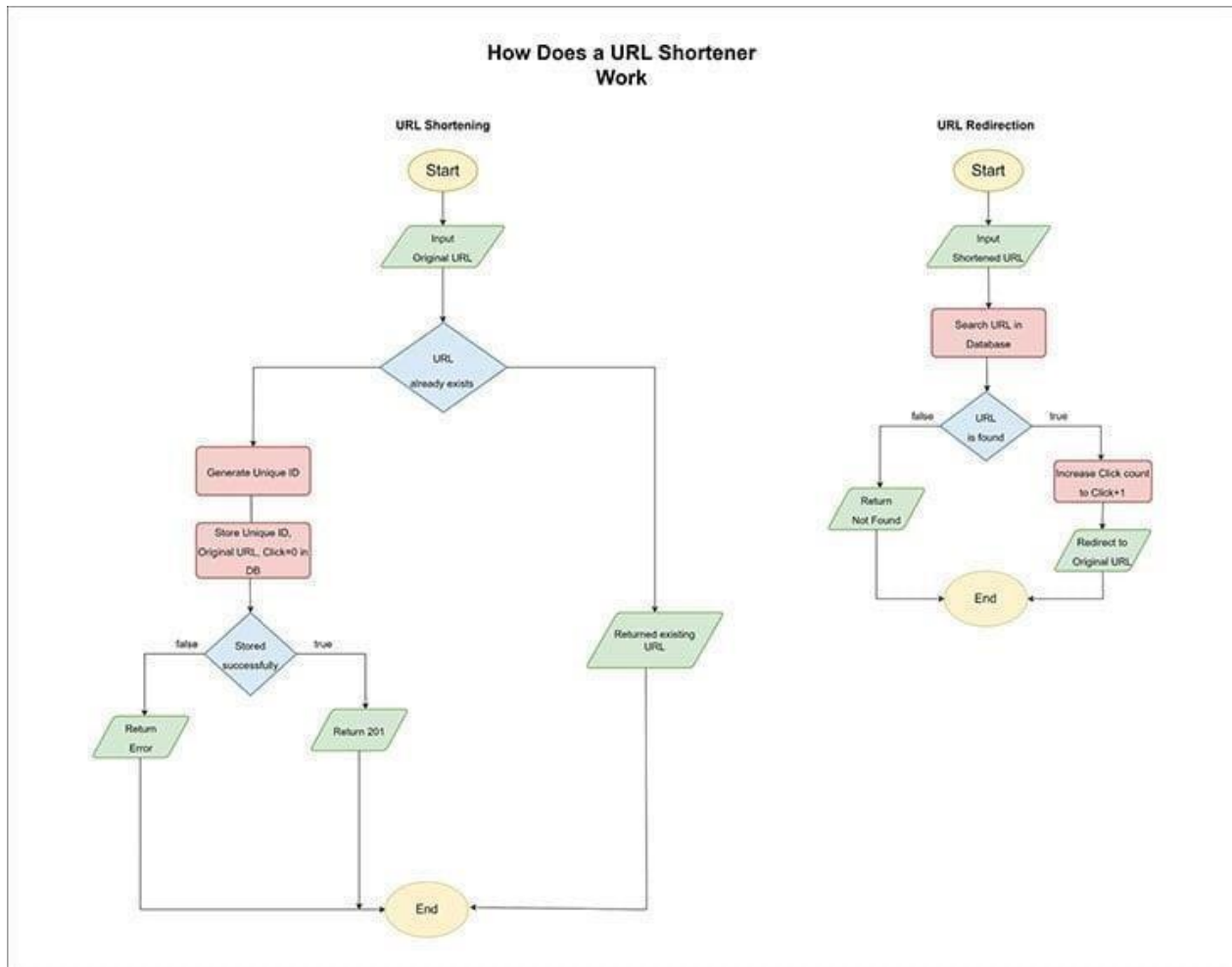
5.1 High Level Diagram (if applicable)



5.2 Low Level Diagram (if applicable)



5.3 Interfaces (if applicable)



6 Performance Test

1. Memory Usage and Performance –

The memory and performance of a website are critical factors that directly impact user experience, search engine ranking, and overall success. It includes:

- **Server Memory:** Websites require server resources to handle incoming requests, process data, and serve content to users. The amount of memory consumed by a website depends on factors such as the complexity of the site, the volume of traffic it receives, and the efficiency of the underlying code and infrastructure.
- **Client Memory:** Web browsers allocate memory to render web pages, execute scripts, and store temporary data such as cached assets and cookies. Websites with heavy client-side processing, large images, or excessive use of JavaScript frameworks may consume significant memory on the client side, affecting performance and responsiveness.
- **Page Load Time:** The speed at which web pages load is a key performance metric that directly impacts user satisfaction and engagement. Factors such as server response time, network latency, and the size and complexity of web page assets (HTML, CSS, JavaScript, images) influence page load time.
- **Mobile Responsiveness:** With an increasing number of users accessing websites on mobile devices, ensuring responsive design and optimized performance for mobile platforms is crucial for delivering a seamless user experience.

2. Power Management –

Power management for a website involves optimizing energy usage and minimizing environmental impact while maintaining performance and reliability. It includes -

- **Energy-efficient Hardware:** Choose energy-efficient servers, data center equipment, and networking devices to reduce power consumption. Modern hardware technologies such as low-power processors, solid-state drives (SSDs), and efficient cooling systems can help minimize energy usage.
- **Virtualization and Consolidation:** Utilize virtualization technologies to consolidate multiple servers onto fewer physical machines, reducing energy consumption and maximizing resource utilization.
- **Efficient Code and Resources:** Optimize website code, scripts, and media assets (images, videos, CSS, JavaScript) to reduce file sizes and minimize processing overhead. Use techniques such as

minification, compression, and lazy loading to improve performance and reduce energy consumption.

- **Client-Side Optimization:** Implement client-side optimization techniques such as browser caching, resource prefetching, and asynchronous loading to minimize unnecessary network requests and conserve bandwidth.
- **Responsive Design:** Design websites with responsive layouts that adapt to different screen sizes and devices, optimizing energy usage on mobile devices and reducing the need for server-side processing and data transfer.
- **Content Delivery Networks (CDNs):** Utilize CDNs to distribute website content across multiple edge servers located closer to end users, reducing latency and minimizing energy consumption associated with long-distance data transfer.
- **Caching and Content Optimization:** Implement caching strategies to store static content (e.g., images, CSS, JavaScript) at edge locations or client browsers, reducing the need for repeated data retrieval and processing from origin servers.

3. Security –

Ensuring the security of a website is paramount to protect sensitive data, maintain user trust, and prevent unauthorized access or malicious activities. It includes

- **Authentication and Authorization:** Use strong authentication mechanisms, such as multi-factor authentication (MFA), to verify the identity of users and prevent unauthorized access to protected resources. Implement access controls and authorization policies to restrict access to sensitive areas of the website based on user roles and permissions.
- **Secure Coding Practices:** Follow secure coding practices and guidelines to minimize the risk of security vulnerabilities in website code. This includes using parameterized queries for database access, escaping output to prevent XSS attacks, and avoiding the use of deprecated or insecure functions.
- **Security Headers:** Set HTTP security headers such as Content Security Policy (CSP), X-Frame-Options, X-XSS-Protection, and X-Content-Type-Options to mitigate common web security threats and protect against cross-site scripting, clickjacking, and MIME sniffing attacks.
- **Regular Updates and Patching:** Keep software components, including web servers, content management systems (CMS), libraries, and frameworks, up to date with the latest security patches and updates to address known vulnerabilities and security issues.

6.1 Test Plan/ Test Cases

The test cases were designed to cover various aspects of the application, including short URL Processing, redirection, database processing, input validation, error handling, and performance under load. Test cases include:

1. Functional Testing:

- Verify that users can shorten a valid URL successfully.
- Ensure that the shortened URL redirects to the original long URL when accessed.
- Verify that short URLs expire or become inactive after a specified period if expiration functionality is implemented.

2. Usability Testing:

- Evaluate the user interface for ease of use, clarity, and intuitiveness.
- Test the responsiveness of the website across different devices and screen sizes.
- Verify that users can easily copy and share shortened URLs.
- Test the speed and responsiveness of the website, ensuring quick load times and minimal latency.

3. Security Testing:

- Test for proper authentication and authorization mechanisms, including login/logout functionality and access controls.
- Test for protection against CSRF (Cross-Site Request Forgery) attacks by validating and verifying request tokens.
- Verify that the website implements security headers such as Content Security Policy (CSP) to mitigate common web security threats.

6.2 Test Procedure

The Django testing framework was employed to conduct unit tests, ensuring the validation of individual components such as models, views, forms, and utility functions. - Integration tests were carried out to assess the interaction between modules and application components. - Configuration tests were undertaken to evaluate the comprehensive functionality, covering user interface operations, database transactions, and error handling procedures. - Performance tests were simulated using tools like JMeter or Locust to gauge response time, throughput, and server stability under different load scenarios.

6.3 Performance Outcome

Performance tests yielded favorable results, demonstrating the application's capability to accommodate a substantial volume of concurrent users while sustaining an acceptable response time. - Future enhancements aim to bolster application performance through optimizations including caching mechanisms, database indexing, and refined query design. - To track important performance measures and identify areas that were ready for more optimization, monitoring tools were employed.

7 My learnings

1. Programming Skill Development –

This internship gave me a deep dive into programming and developing an website using Python Programming Language, Django Framework Development, Frontend Technologies, Bootstrap, Backend Technologies and Databases. This exposé gave my programming skills an huge uplift particularly in Web development.

2. Project Management -

Completing the targeted goal with the timeline gave me right ideas to work more efficiently and effectively by giving priorities to critical task and proper workflow management and following an professional roadmap gave me an idea on project management.

3. Collaboration and Communication -

Collaboration with industry leaders like UniConverge Technologies Pvt Ltd and educational institutions such as upskill Campus and The IoT Academy has underscored the pivotal role of effective communication, teamwork, and collaborative efforts in achieving project objectives.

4. Testing and Best Practices -

Proficiency in devising test designs, crafting test cases, and executing performance evaluations ensures the robust functionality, reliability, and performance of URL shortening applications. Testing encompasses memory efficiency, accuracy, stability, power consumption, and security considerations.

5. Real-world Applications -

Translating theoretical knowledge into real-world project scenarios has provided invaluable insights into technical dilemmas, strategic solutions, and practical business implications, thereby contributing to overall professional efficacy.

8 Future work scope

1. Advanced Analytics and Reporting -

Implement advanced analytics capabilities to provide insights into link performance, user engagement, and referral sources. Develop customizable reporting features that allow users to generate and export detailed analytics reports.

2. Link Management and Organization -

Enhance link management capabilities by introducing features such as link categorization, tagging, and folder organization. Implement search functionality to allow users to easily find and manage their shortened links.

3. Customization and Branding -

Expand customization options to allow users to customize the appearance and branding of shortened URLs with custom domains, branded short URLs, and custom landing pages. Introduce white-label solutions for businesses and organizations to offer branded URL shortening services to their customers or clients.

4. Security Enhancements -

Strengthen security measures by implementing additional security controls such as IP whitelisting, rate limiting, and bot detection mechanisms. Conduct regular security audits and penetration testing to identify and address potential vulnerabilities and security risks.

5. Mobile Applications -

Develop mobile applications for iOS and Android platforms to provide users with a seamless and intuitive experience for URL shortening and management on mobile devices. Implement push notifications and offline access capabilities to enhance user engagement and convenience.