Industrial Internship Report on

URL SHORTENER

Prepared by

Riona Michael

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 URL SHORTENER. This project was about creating a web application which shortens the long URLs into a small one which is easier for copying, saving space and sharing. It uses python package such as hashlib and flask for encryption and web framework.

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

Over the 6 weeks, I chose the project title, “URL Shortener” solely because of my interest in both python and web application. At First, I learnt about packages and functions required for creating the web application and encryption, then I worked on the implementation, testing and documentation of the back-end code of the project. At last I started with the front-end development of the web after learning of HTML, CSS. And finally, the project was created after various testing.

As Python emerges as one of the most prominent languages in industry, I recognized the importance of enhancing my proficiency in it. With a keen interest in comprehending Python's role within projects, pursuing this internship presented an ideal opportunity for me to learn deeper into its workings. Through this experience, I aimed to not only expand my knowledge base but also to apply Python effectively in real-world scenarios, thereby enriching my skill set required for my career.

I extend my gratitude to the USC/UCT and team, for giving this wonderful opportunity packed with many learning resources. The prompt review of reports and meticulously structured framework of this internship played a pivotal role in propelling me forward with my project. Their support and guidance were instrumental in enhancing my skills and maximizing my potential throughout this journey.

Overall, this internship proved incredibly valuable as I learnt Python and its fundamentals through a series of quizzes. The presence of a responsive WhatsApp group provided invaluable support, swiftly addressing any doubts or queries I encountered. This journey was truly

memorable, leaving me with a profound sense of fulfillment and gratitude for the experiences gained.

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)

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

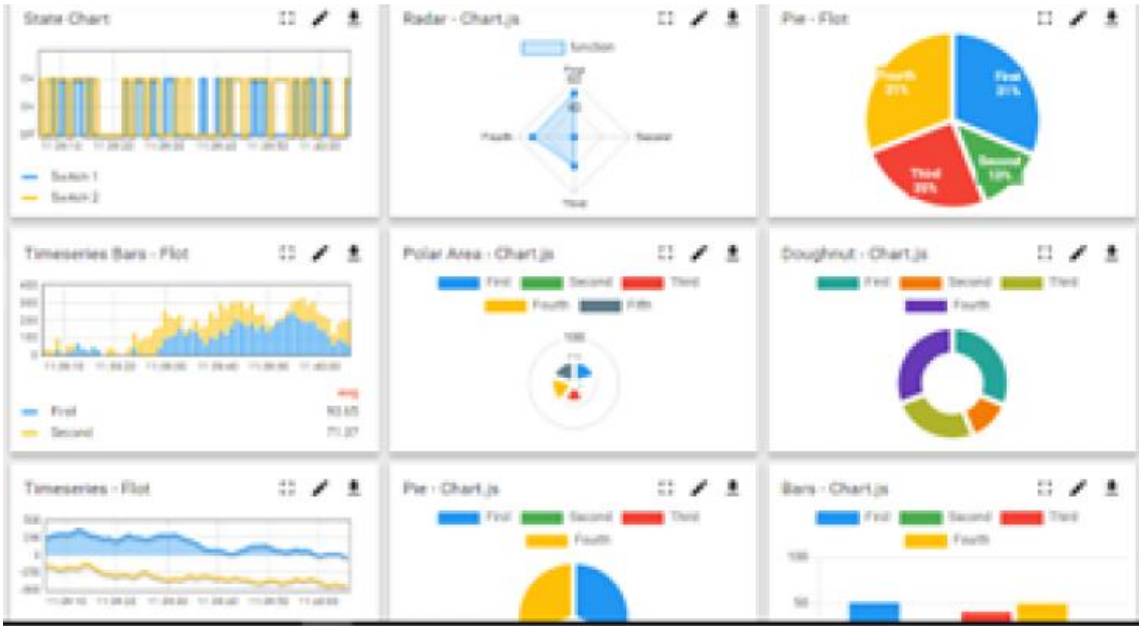
• Build Your own dashboard

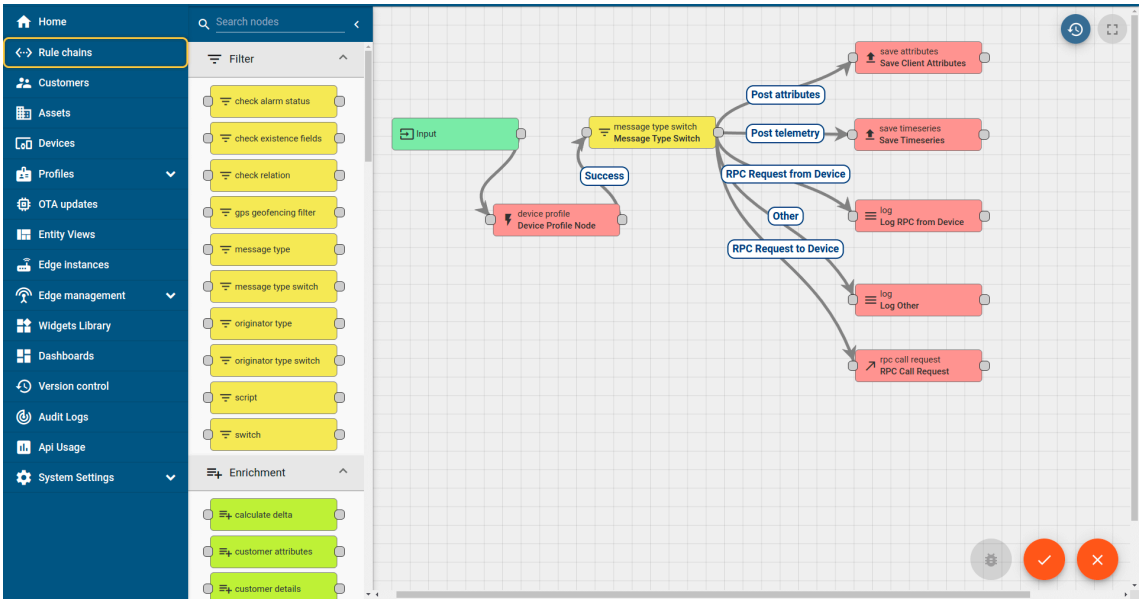
• Analytics and Reporting

• Alert and Notification

• Integration with third party application(Power BI, SAP, ERP)

• Rule Engine





ii. Smart Factory Platform (FACTORY WATCH)

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 unleased 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.

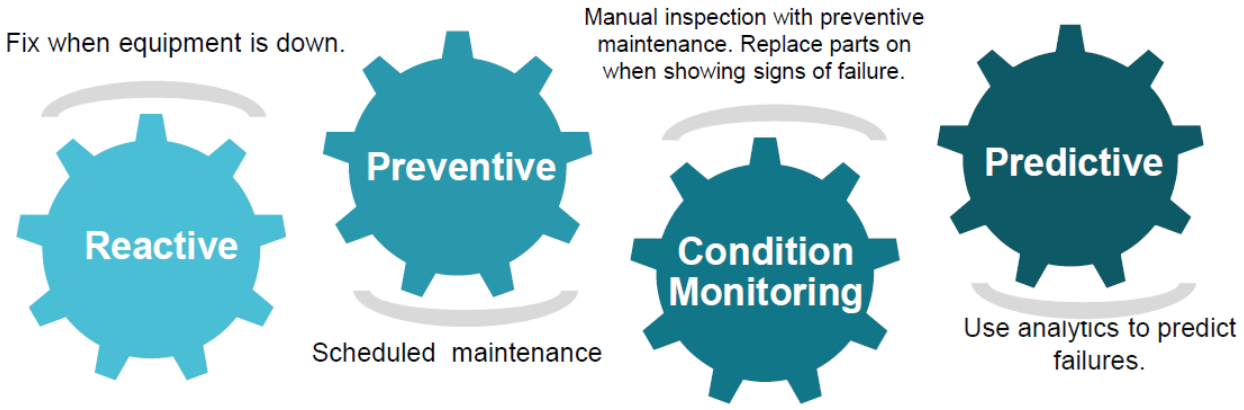
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i

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

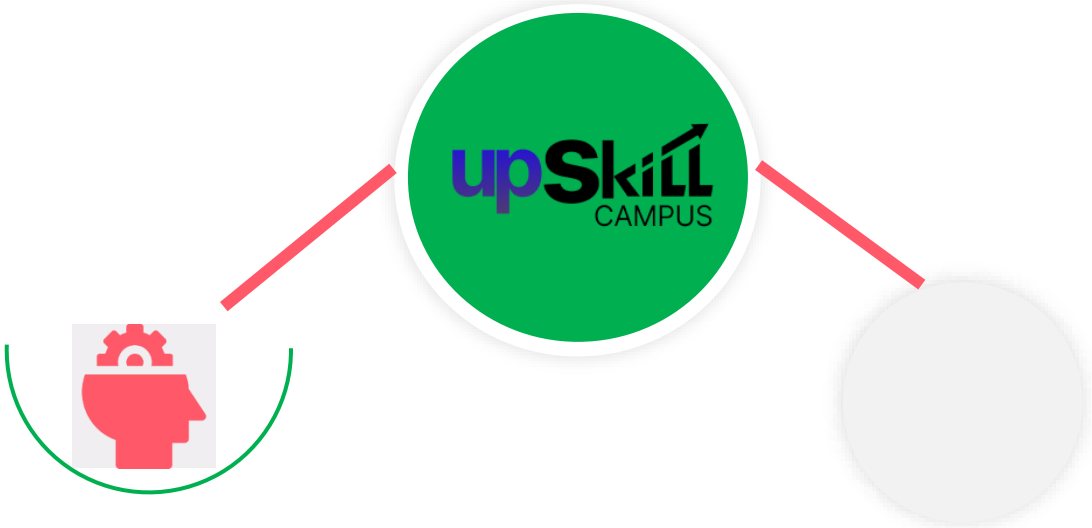
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

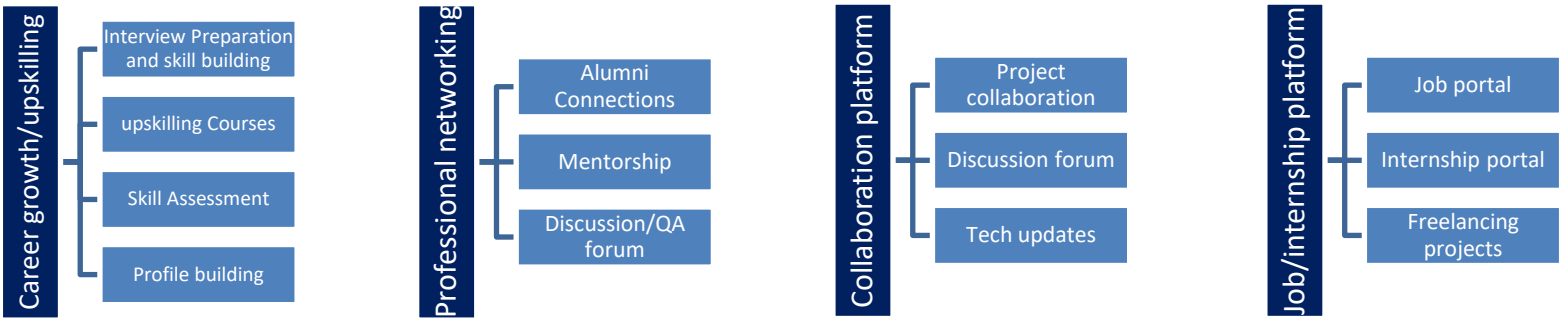
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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] URL shortening. [Online]. Available: http://en.wikipedia.org/wiki/URL_shortening

[2] Security Threats of URL Shortening: A User's Perspective, Nhien-An Le-Khac and M. Tahar Kechadi (2015)

[3] Website Link Shortener, Aeesha Sdeek Shaheen, Najla Matti Isaac (2022)

2.6 Glossary

Terms	Acronym
URL	UNIFORM RESOURCE LOCATOR
HTML	HYPER TEXT MARKUP LANGUAGE
CSS	CASCADE STYLING SHEET
HTTP	HYPER TEXT TRANSFER PROTOCOL

3 Problem Statement

DESCRIPTION: The URL shortener is a Python project that shortens lengthy URLs into easier-to-handle links. When a lengthy URL is entered, it creates a unique shortened URL and, upon access, sends users back to the original URL.

SCOPE: Project scope includes creating a database to hold the mapping between the original and shorter URLs, creating functions to create unique shortened URLs and manage redirection, and designing a user interface to input long URLs and display the shortened links.

4 Existing and Proposed solution

Existing URL shortener solutions typically offer basic functionality for generating shortened links, redirecting users, and tracking link usage. However Using external URL shortening services may introduce dependencies and potential risks, such as changes to terms of service or service discontinuation and Certain URL shortening services may have limitations on the number of links that can be shortened or the volume of traffic they can handle.

The proposed solution for the URL shortener project aims reduce dependencies on third-party services and provide greater control over uptime and reliability and accommodate a large volume of shortened links and high traffic loads.

The proposed solution aims to add value by offering a user-friendly, feature-rich URL shortening service that prioritizes customization, reliability, and security.

4.1 Code submission (Github link)

https://github.com/Riona-Michael/upskillcampus/tree/main

4.2 Report submission (Github link) :

https://github.com/Riona-Michael/upskillcampus/blob/main/URLSHORTENER_RIONA_MICHAEL_USCUTC.pdf

5 Proposed Design/ Model

5.1 Begin:

Establish the goals and specifications for the URL shortener project at the outset of the project.

5.2 Research: Look into the newest technology, best practices, and URL shortening solutions available.

5.3 Planning: Draft a project plan with the necessary resources, tasks, and deadlines.

5.4 Stages in Between:

Describe the general architecture of the URL shortening system, taking into account the frontend, backend, database, and API components.
Database Design: Create the database schema needed to hold the original URLs together with the short codes that go with them.
Frontend Development: Create the application's user interface, including the form that users use to submit lengthy URLs and see their abbreviated connections.
Backend Development: Put into practice the backend logic that manages redirection requests, creates short codes, and saves URLs in databases.

5.5 Testing: Perform end-to-end, integration, and unit tests to High Level Diagram (if applicable)

5.6 Interfaces

The user interface, or UI:

Description: By entering lengthy URLs and receiving abbreviated links, users can communicate with the URL shortening system through this interface.

Representation: The user interaction process, from entering a large URL to obtaining the shortened link, can be shown using a flowchart. Furthermore, the arrangement and style of the user interface might be visually represented using wireframes or mockups.

Management of Memory Buffers:

Description: Effective storing and retrieval of URL records may still be crucial in a URL shortener,

even though memory management may not be as crucial as it is in memory-intensive applications.

Representation: The storage and administration of URL records in memory, including caching techniques and eviction rules to maximize efficiency, can be seen in a memory buffer management diagram.

6. Performance Test

Memory Constraints:

Identification: With a large number of URLs to be stored along with their corresponding shortened codes, memory usage can become a constraint, particularly for systems with limited memory resources.

Addressing in Design: Implementing efficient data structures such as hash tables or databases can optimize memory usage. Additionally, implementing caching mechanisms can reduce the load on memory by storing frequently accessed URLs.

Test Results: Memory usage can be monitored during stress testing to ensure efficient utilization. Testing can involve analyzing memory consumption under various load conditions to identify potential bottlenecks.

Speed and Throughput Constraints:

Identification: The speed and throughput of the URL shortener system can be constrained by factors such as server processing capabilities and network latency.

Addressing in Design: Employing asynchronous processing techniques, optimizing database queries, and implementing caching mechanisms can improve system speed and throughput. Additionally, load balancing can distribute incoming requests across multiple servers.

Test Results: Performance testing can assess the system's speed and throughput under various load conditions. Test results may include response times, throughput metrics, and scalability limits.

Accuracy Constraints:

Identification: Accuracy constraints involve ensuring that the URL shortener consistently generates correct short URLs and accurately redirects users to the original URLs.

Addressing in Design: Implementing error handling mechanisms and redundancy in URL storage can maintain accuracy. Thorough testing, including unit tests and integration tests, can validate functionality.

Test Results: Test results should demonstrate the system's ability to generate correct short URLs and accurately redirect users under various scenarios, including edge cases.

Durability Constraints:

Identification: Durability constraints pertain to the resilience of the URL shortener system against data loss or corruption.

Addressing in Design: Implementing data backup and recovery mechanisms, such as database backups and replication, can ensure durability. Additionally, fault-tolerant architectures can mitigate the impact of hardware failures.

By addressing these constraints in the design and testing phases, the URL shortener system implemented with Flask and hashlib can meet the requirements of real-world deployment in industries, ensuring performance, reliability, and scalability.

6.1 Test Plan/ Test Cases

Identify constraints such as memory usage, speed, accuracy, and durability. Develop test cases to validate URL shortening functionality, database interactions, error handling, and system scalability. Ensure comprehensive coverage of both normal and edge cases.

6.2 Test Procedure

Execute test cases against the URL shortener system, including unit tests, integration tests, and stress tests. Monitor system behaviour, response times, and resource utilization. Record and analyze test results to identify any issues or performance bottlenecks.

6.3 Performance Outcome

Evaluate system performance based on test results, including throughput, response times, memory usage, and error rates. Address any identified issues and optimize system components as needed to meet performance requirements. Provide documentation summarizing performance outcomes and any recommendations for improvements.

7. My learnings

Through the URL shortener project, I've honed my skills in Python web development, problem-solving, project management, testing, and communication. These learnings will bolster my career growth in software development and engineering.

Overall, this internship was really beneficial since it taught me the basics of Python through a series of tests. This trip was genuinely unforgettable, and I came away from it feeling incredibly fulfilled and appreciative of the experiences I had.

8. Future work scope

Future work for the URL shortener project may involve implementing features such as user authentication, custom URL aliases, link expiration, API integration, enhanced analytics, URL preview, browser extensions, and a mobile application.