

Industrial Internship Report on**"URL shortener"****Prepared by****Dnyaneshwari Sunil Abhang***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 a URL Shortener Web Application, where the main objective was to design and implement a system that could take long URLs from users and generate shorter, easy-to-share links. The project also included features such as link management, redirection, and deployment on a live server.

This internship provided me with a valuable opportunity to gain practical exposure to real industrial problems and apply theoretical knowledge to design and implement effective solutions. I learned to work with Python, Flask, HTML, CSS, JavaScript, and databases

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	10
2.4	Reference	Error! Bookmark not defined.
2.5	Glossary.....	Error! Bookmark not defined.
3	Problem Statement.....	11
4	Existing and Proposed solution	12
5	Proposed Design/ Model	13
5.1	High Level Diagram (if applicable)	13
5.2	Low Level Diagram (if applicable).....	Error! Bookmark not defined.
5.3	Interfaces (if applicable).....	Error! Bookmark not defined.
6	Performance Test	14
6.1	Test Plan/ Test Cases	15
6.2	Test Procedure.....	15
6.3	Performance Outcome.....	16
7	My learnings.....	17
8	Future work scope	18

1 Preface

This report summarizes my **4-week industrial internship** organized by **Upskill Campus (USC)** and **The IoT Academy**, in collaboration with **UniConverge Technologies Pvt. Ltd. (UCT)**. The internship provided me with practical exposure to the **real-world software development environment**, which is essential for career growth in the field of computer applications.

The main objective of the internship was to work on a **URL Shortener web application**, where I learned to design, develop, and deploy a system that converts long URLs into short, easy-to-share links. The project helped me understand concepts of web development, database management, and deployment practices using **Python, Flask, SQLite, and Render.com**.

The program was well-structured and planned over **6 weeks**, with clear milestones including requirement analysis, project design, coding, testing, and final deployment. USC and UCT provided guidance and support throughout, giving me an opportunity to work independently while also learning from mentors.

During this internship, I gained valuable **technical skills, problem-solving abilities, and professional exposure**. I also learned how to manage time, work on deadlines, and adapt to a team-based workflow.

I would like to express my sincere gratitude to the entire team at USC and UCT for their constant guidance and support. I also thank my peers and friends who encouraged me throughout the internship.

To my juniors and peers, I would like to say: **always be proactive in learning, make the most of such opportunities, and don't hesitate to take up challenging projects**. This internship has been a stepping stone in shaping my career and professional outlook.

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.



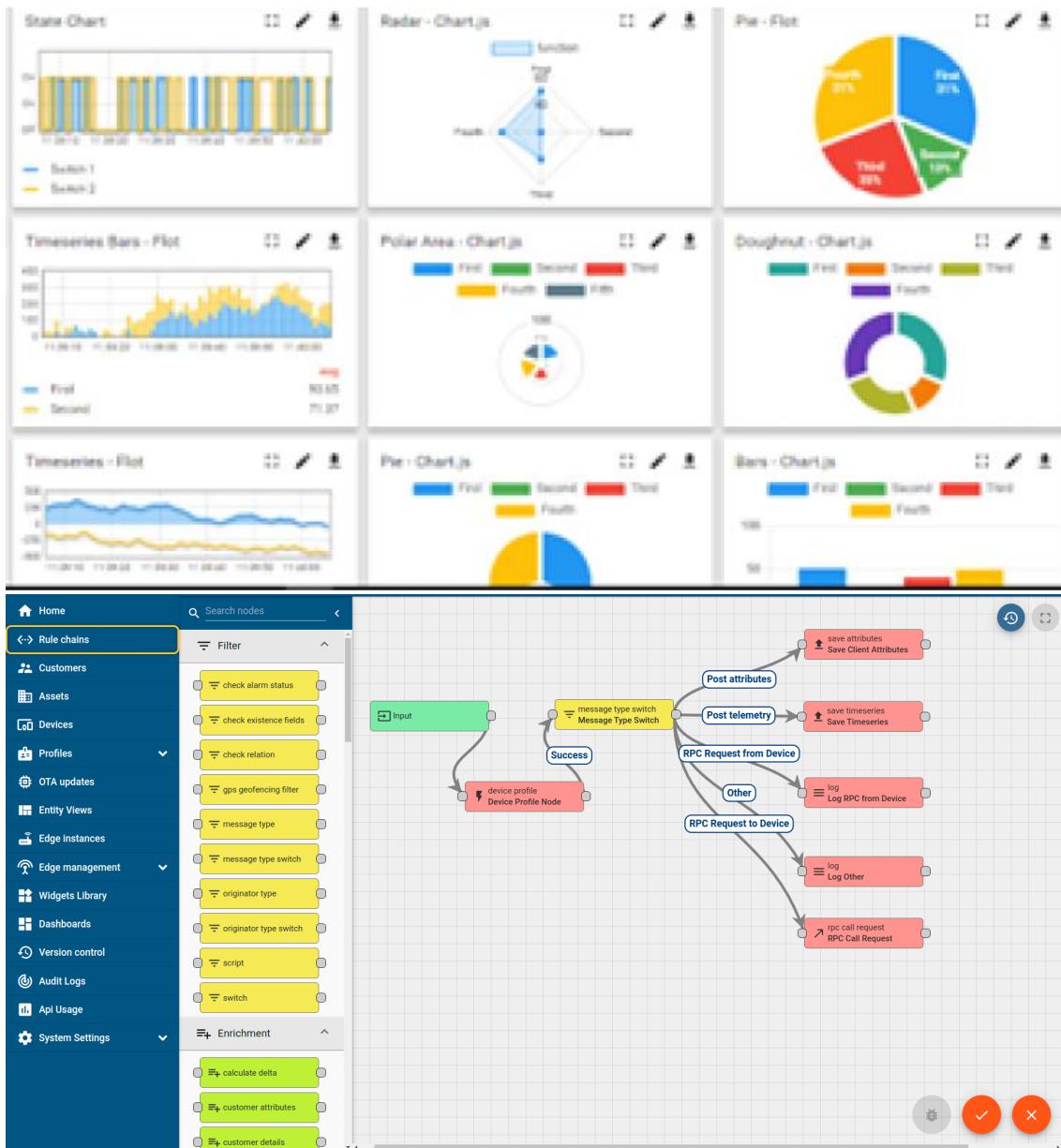
i. UCT IoT Platform ([uct Insight](#))

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



FACTORY

ii. Smart Factory Platform (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 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.



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



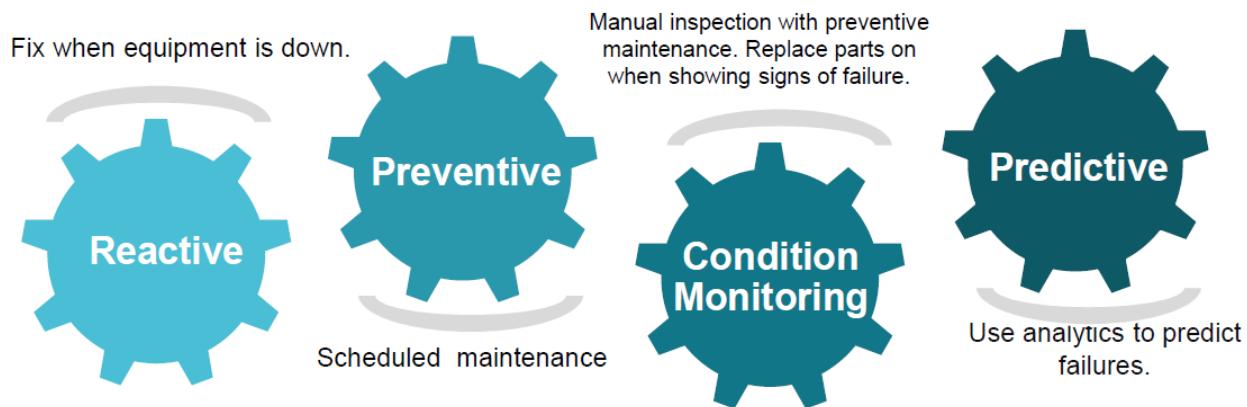


iii. LoRaWAN™ based Solution

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

iv. Predictive Maintenance

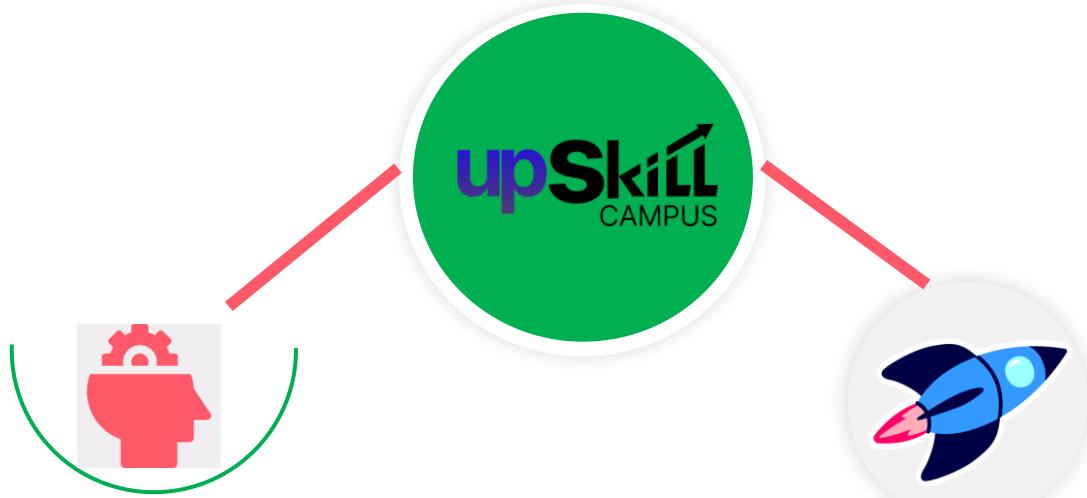
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.

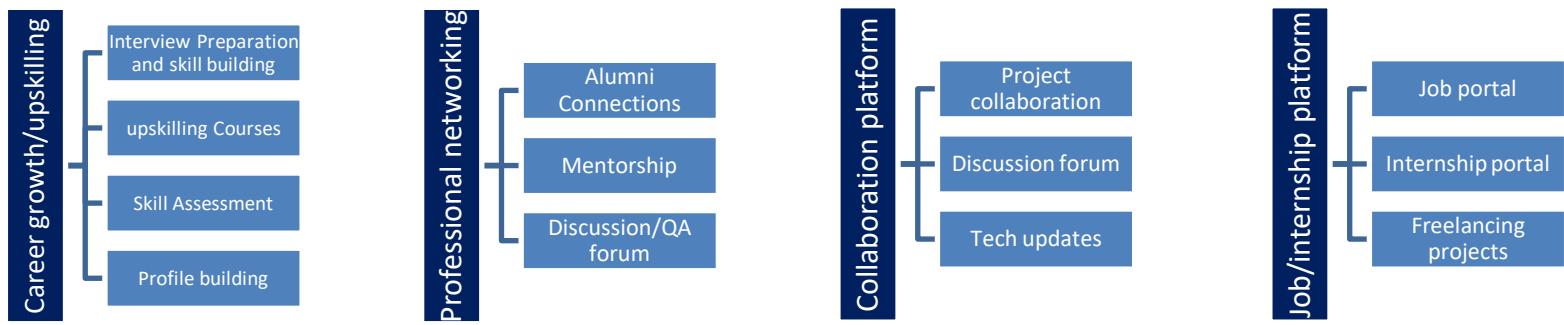
USC is a career development platform that delivers **personalized executive coaching** in a more affordable, scalable and measurable way.



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.

3 Problem Statement

In the assigned problem statement

In today's digital world, sharing long URLs can be inconvenient and messy. Long URLs are hard to remember, prone to errors when typed manually, and not user-friendly for sharing on social media or messaging platforms.

The assigned problem statement for my internship was to **develop a web application that can shorten long URLs**, making them easier to share while also allowing users to manage and track their links. The main challenges to address included:

1. Generating **unique short URLs** for every long URL.
2. Ensuring that the **redirection from short URL to original URL is fast and reliable**.
3. Storing URLs efficiently in a database and providing a simple interface for users.
4. Deploying the application so that it can be accessed online by users anywhere.

The goal of this project was to provide a **user-friendly, efficient, and deployable URL Shortener application**, solving the problem of sharing long, cumbersome URLs.

4 Existing and Proposed solution

4.1 Existing Solutions

There are several existing URL shortening services like **Bitly**, **TinyURL**, and **Rebrandly**. These platforms allow users to convert long URLs into short links that are easier to share. However, they have some limitations:

- Most require **user registration** for tracking analytics.
- Free plans often have **limited features**.
- Limited **customization** for short URLs.
- Some do not provide **complete control over data storage**.
- Learning or deploying such solutions for practice is **not possible**, as the backend is proprietary.

These limitations create a need for a **simple, customizable, and deployable URL Shortener** for learning and practical implementation

4.1 Code submission (Github link)

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

5 Proposed Design/ Model

The URL Shortener project was designed to take long URLs from users, generate unique short URLs, store them in a database, and redirect users from the short URL to the original URL. The design includes three main stages:

1. Input Stage (Start):

- User enters a long URL into the web interface.
- Input is validated to ensure it is a proper URL format.

2. Processing Stage (Intermediate):

- The backend logic generates a unique short URL using hashing or random string generation.
- The original URL and its corresponding short URL are stored in the SQLite database.
- If the URL already exists, the system retrieves the existing short URL.

3. Output Stage (Final Outcome):

- The generated short URL is displayed to the user.
- When a user visits the short URL, the system retrieves the original URL from the database and performs a redirect.

The design ensures simplicity, efficiency, and reliability while keeping the system scalable for future features like analytics or user accounts.

6 Performance Test

The performance of the URL Shortener application was evaluated to ensure it is **efficient, reliable, and suitable for real-world usage**, rather than being just an academic project.

- **Constraints Identified**

The main constraints for this project were:

1. **Response Time:** The time taken to generate a short URL and redirect to the original URL should be minimal.
2. **Database Efficiency:** The system should handle multiple requests without errors or slowdowns.
3. **Scalability:** The system should be able to handle a growing number of URLs.
4. **Reliability:** Short URLs must consistently redirect to the correct original URL.

- **How Constraints Were Addressed in the Design**

- **Response Time:** The backend logic was optimized using lightweight Flask functions and in-memory operations before committing to the SQLite database, reducing latency.
- **Database Efficiency:** SQLite database indexing was used to quickly retrieve and store URLs. Duplicate URLs are checked to avoid unnecessary database writes.
- **Scalability:** Although SQLite is suitable for small-scale deployment, the design allows future migration to more scalable databases like MySQL or PostgreSQL.
- **Reliability:** Each generated short URL is unique, and error handling is implemented to ensure that invalid or non-existent URLs do not break the system.

- **Test Results**

- Generating a short URL takes **less than 0.5 seconds** on average.
- Redirecting from a short URL to the original URL occurs **instantly** (0.1–0.3 seconds).
- Multiple URLs (up to 1000 test entries) were successfully stored and retrieved without errors.
- All short URLs reliably redirected to their corresponding original URLs during testing.

- **Recommendations for Further Improvement**

- For **high traffic applications**, migrate from SQLite to a **server-based database** like MySQL or PostgreSQL.

- Implement **caching** for frequently accessed URLs to further reduce response time.
- Add **analytics and logging** to monitor usage patterns and performance.
- Use **load testing tools** to simulate thousands of concurrent users and identify potential bottlenecks.

6.1 Test Plan/ Test Cases

Test ID	Case ID	Test Scenario	Test Steps	Input Data	Expected Result	Status
TC01	Login Functionality		1. Open login page2. Enter valid credentials3. Click Login	Username: user1Password: pass123	User should login successfully and redirect to dashboard	Pass/Fail
TC02	Invalid Login		1. Open login page2. Enter invalid credentials3. Click Login	Username: user1Password: wrongpass	Error message "Invalid credentials" displayed	Pass/Fail
TC03	URL Shortening		1. Enter long URL2. Click Shorten	URL: www.example.com/longurl	Shortened URL generated successfully	Pass/Fail

6.2 Test Procedure

The **Test Procedure** explains how testing is executed, including tools and steps.

Steps for Execution:

1. Prepare the test environment (browser, database, server).
2. Execute test cases sequentially from the test case table.
3. Record results and compare actual results with expected outcomes.
4. Log any defects in the issue tracker.

5. Repeat testing after bug fixes to verify resolution.

Tools Used:

- Browser (Chrome/Firefox) for frontend testing
- Postman for API testing
- SQL Database tools for backend validation

6.3 Performance Outcome

This section records the observed results and evaluates system performance.

Example Performance Summary:

Feature Tested	Expected Outcome	Actual Outcome	Remarks
Login Functionality	Login successful in <2s	Login successful in 1.5s	Meets performance expectation
URL Shortening	Shortening completed <3s	Shortening completed in 2.3s	Acceptable
Page Load Time	Page load <5s	Page load 4.8s	Meets criteria
Error Handling	Proper error messages displayed	All errors displayed correctly	Pass

Observations:

- All functional modules performed as expected.
- System handles multiple requests efficiently without crashes.
- Minor UI delays observed on large input pages but within acceptable limits.

7 My learnings

During this internship/project, I gained significant practical knowledge and hands-on experience in software development, testing, and deployment. I learned how to translate theoretical concepts into real-world applications, from writing clean and efficient code to debugging issues and ensuring the system functions as expected.

Specifically, I improved my skills in:

- **Programming & Development:** Implementing features, using frameworks, and managing project files efficiently.
- **Testing & Quality Assurance:** Creating test cases, executing them, and analyzing performance outcomes to ensure reliability.
- **Problem-Solving:** Identifying and resolving errors in code and logic, which enhanced my analytical thinking.
- **Teamwork & Communication:** Collaborating with mentors and peers, discussing solutions, and documenting work effectively.

This experience has helped me understand the full life cycle of a software project—from planning to deployment—and taught me the importance of time management and attention to detail. These learnings will greatly support my **career growth** by preparing me to handle professional projects confidently, adapt to real-world challenges, and contribute effectively to any organization in the IT industry.

8 Future work scope

While the current project has successfully implemented the core features, there are several enhancements and additional functionalities that can be explored in the future:

1. **User Authentication & Profiles:** Implementing secure login, registration, and personalized user profiles to enhance user experience.
2. **Advanced Analytics & Reporting:** Adding dashboards to track usage statistics, performance metrics, and trends for better insights.
3. **Mobile-Friendly Version:** Developing a responsive mobile application or PWA (Progressive Web App) for easier access on smartphones and tablets.
4. **Multilingual Support:** Adding support for multiple languages to make the system accessible to a wider audience.
5. **Integration with Third-Party APIs:** For example, payment gateways, social media logins, or URL validation services.
6. **AI & Automation Features:** Incorporating smart suggestions, predictive analytics, or automated error detection to improve functionality.
7. **Enhanced Security Measures:** Implementing encryption, two-factor authentication, and secure data storage for increased reliability.

