[Your College Logo]

Industrial Internship Report on

”Python URL Shortner”

Prepared by

[Zaid Pathan]

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 Shorten URLs with Python! This project will show you how to build a basic URL shortener in Python. Those long links you hate sharing? We'll make them mini! Here's the gist: Python Does the Magic: We'll use Python tools to transform lengthy URLs into shorter ones. Short and Sweet: No more character limits on Twitter or other platforms. Share those links with ease! Get Started Coding: We'll focus on the core functionality: taking a long URL, making it short, and giving you the result. 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

This 6-week Python project gets you ready for an internship by building a URL shortener.

Weeks 1-3: Code the Core

- Learn Python to shorten those long URLs.
- Gain hands-on experience, understand URL shortening concepts.

Weeks 4-6: Take it Further (Optional)

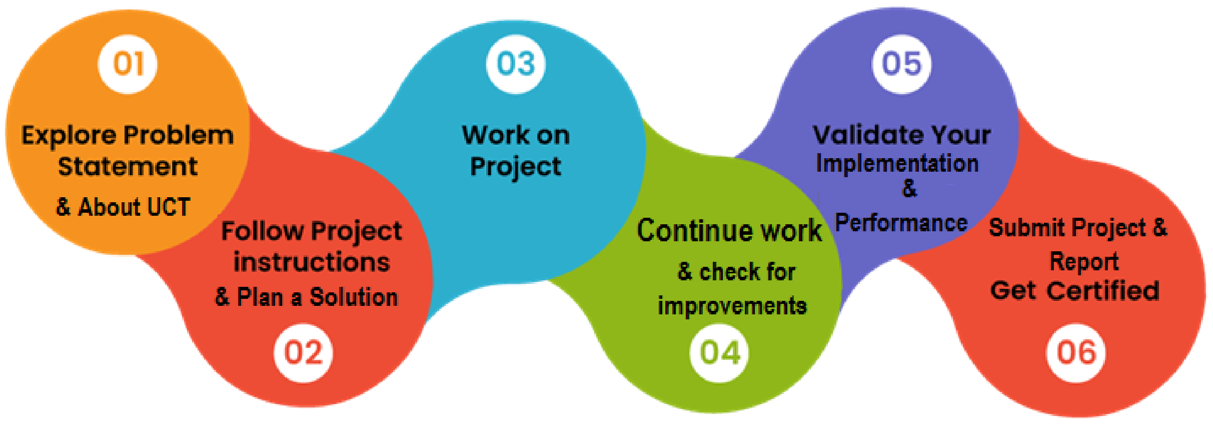
- Add features like custom URLs or click tracking.
- Deploy your shortener online (learn web dev concepts).
- Refine your code for readability and user experience.

Why This Project is Great for Internships:

- Showcases your Python skills and problem-solving abilities.
- Highlights project management skills through building and deployment.

This project is your stepping stone to an internship.

Program was planned like this :



Learnings and overall experience during internship :

Learning Python:

- Grasping core Python concepts for manipulating strings and data.
- Understanding how to use libraries for specific tasks (like URL handling).
- Sharpening my problem-solving skills by coding solutions.

Project Management:

- Breaking down the project into smaller, achievable milestones.
- Testing and debugging code to ensure functionality.
- Even in a small project, there's value in planning and execution.

Overall Experience:

- It was a confidence booster to see my code come to life and actually shorten URLs!
- This project is a great portfolio piece to showcase my Python skills.

Message to Juniors and Peers:

Don't wait for a formal internship to start learning. Take on projects like this URL shortener to:

- **Build your skills:** Hands-on experience is invaluable.
- **Boost your resume:** Showcase your initiative and technical abilities.
- **Explore your interests:** Discover if web development is your thing!

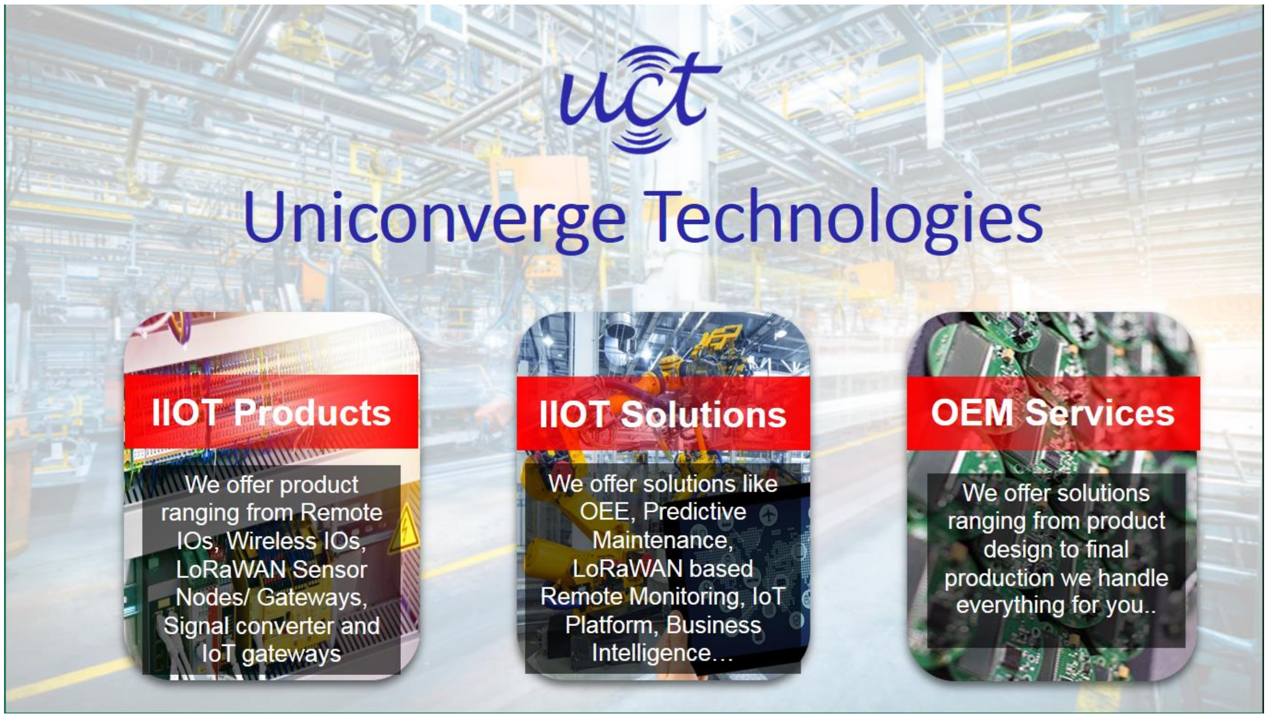
This project is just a starting point. You can extend it with features like custom URLs or analytics. There's always room to learn and grow, so keep coding and creating!

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.



Uniconverge Technologies

IIOT Products

We offer product ranging from Remote IOs, Wireless IOs, LoRaWAN Sensor Nodes/ Gateways, Signal converter and IoT gateways

IIOT Solutions

We offer solutions like OEE, Predictive Maintenance, LoRaWAN based Remote Monitoring, IoT Platform, Business Intelligence…

OEM Services

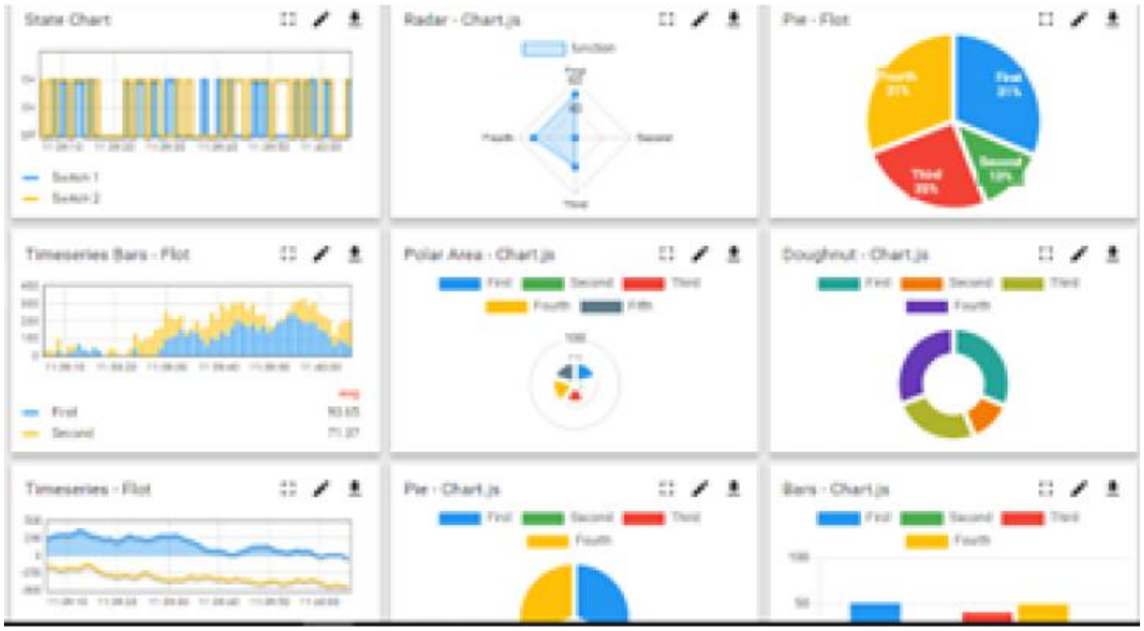
We offer solutions ranging from product design to final production we handle everything for you..

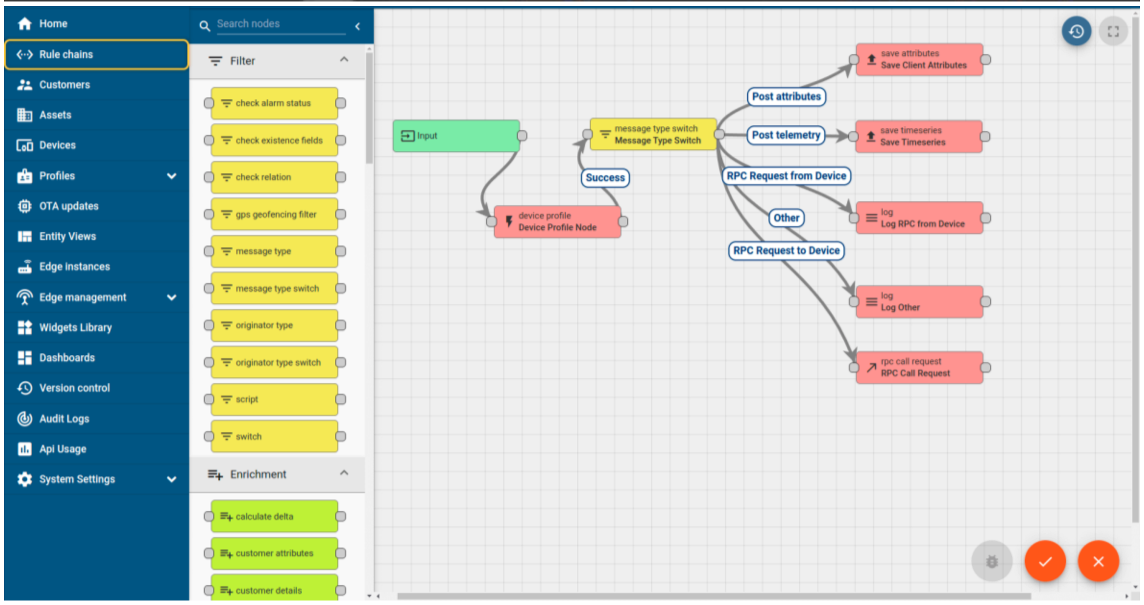
i. UCT IoT Platform (uct Insight)

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


State Chart
Switch 1
Switch 2
Radar - Chart.js
Pie - Flot
Fourth 31%
First 31%
Third 26%
Second 13%
Timeseries Bars - Flot
First
Second
Polar Area - Chart.js
First
Second
Third
Fourth
Fifth
Doughnut - Chart.js
First
Second
Third
Fourth
Timeseries - Flot
Pie - Chart.js
First
Second
Third
Fourth
Bars - Chart.js
First
Second
Third
Fourth


Home
Rule chains
Customers
Assets
Devices
Profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
Search nodes
Filter
check alarm status
check existence fields
check relation
gps geofencing filter
message type
message type switch
originator type
originator type switch
script
switch
Enrichment
calculate delta
customer attributes
customer details
Input
device profile
Device Profile Node
Success
message type switch
Message Type Switch
Post attributes
Post telemetry
RPC Request from Device
Other
RPC Request to Device
save attributes
Save Client Attributes
save timeseries
Save Timeseries
log
Log RPC from Device
log
Log Other
rpc call request
RPC Call Request

ii. Smart Factory Platform (FACTORY WATCH)

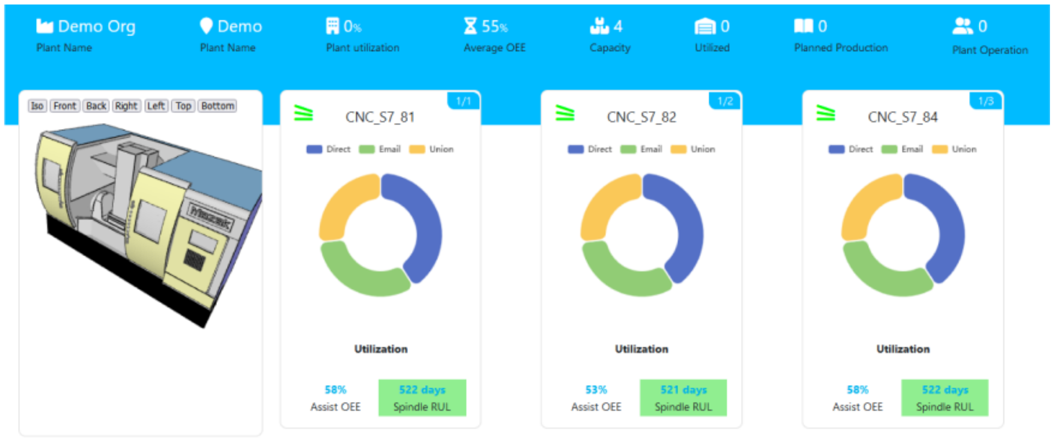
Factory watch is a platform for smart factory needs.

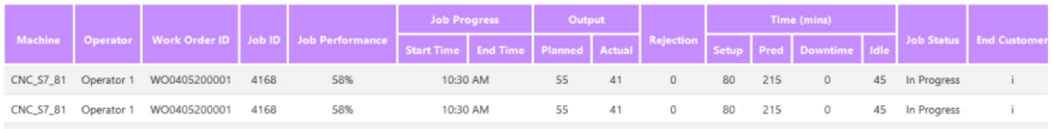
It provides Users/ Factory

- 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.

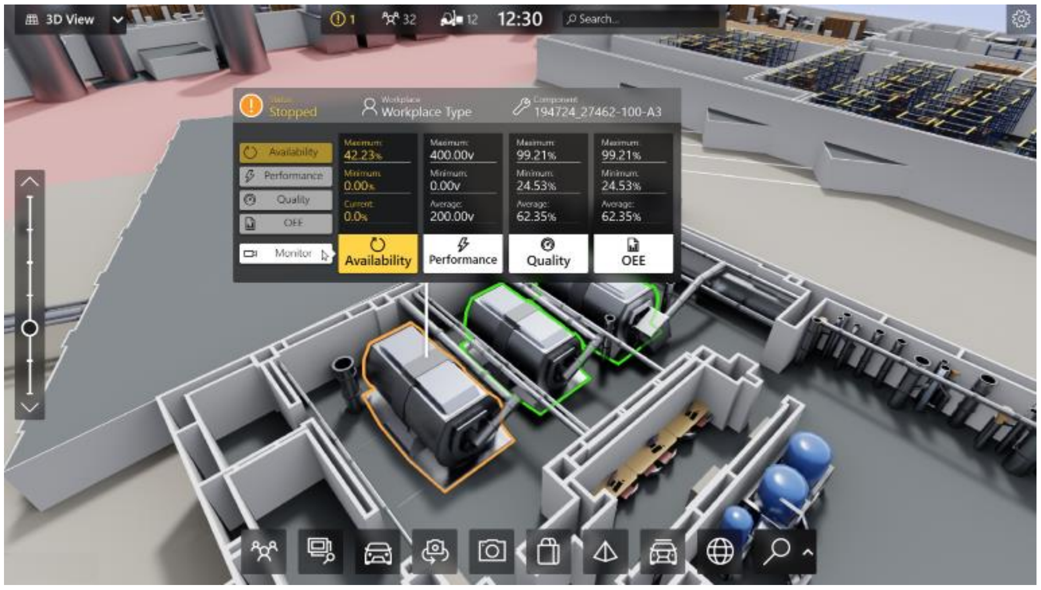
Its unique SaaS model helps users to save time, cost and money.

[Your College Logo]





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

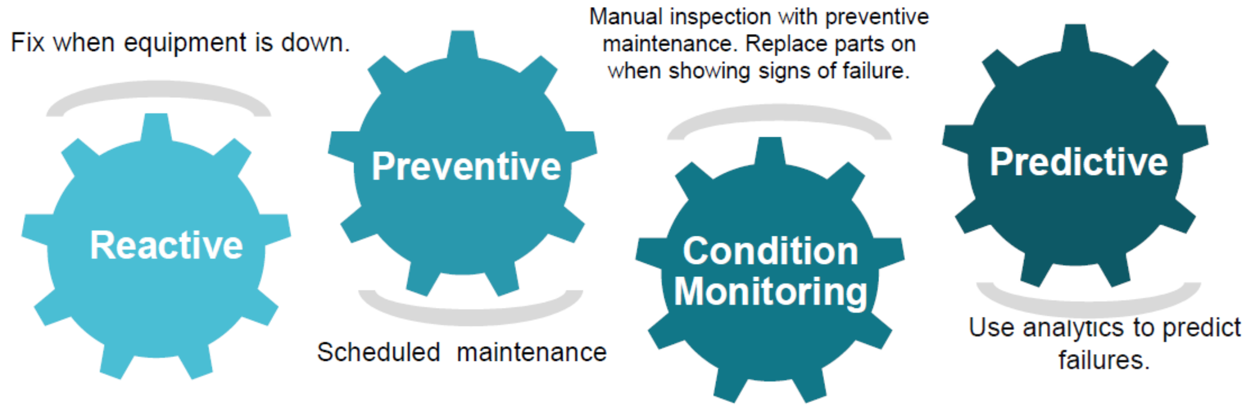


iii. LoRaWAN based Solution

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.

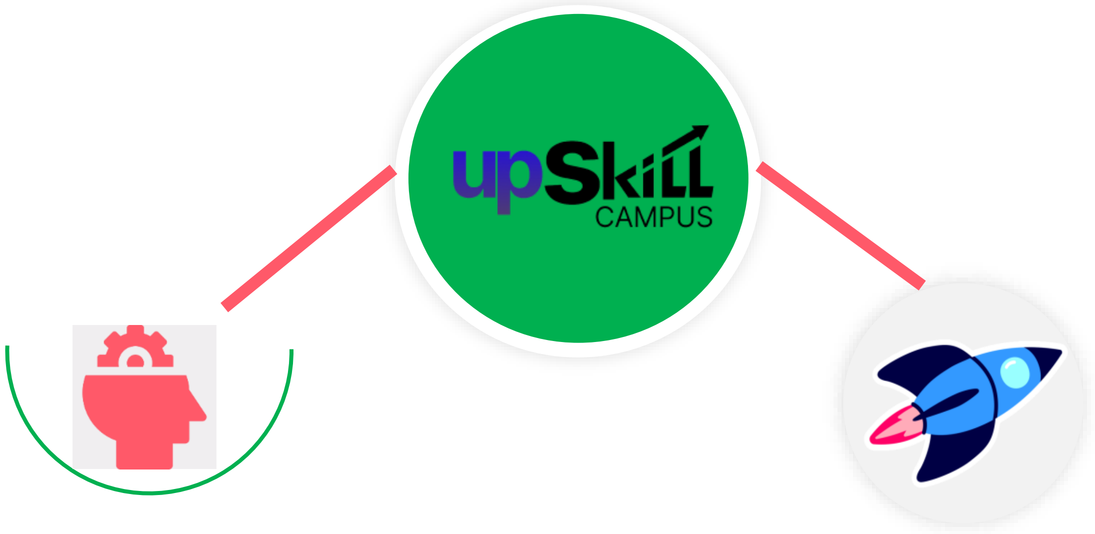

Fix when equipment is down.
Reactive
Preventive
Scheduled maintenance
Manual inspection with preventive maintenance. Replace parts on when showing signs of failure.
Condition Monitoring
Predictive
Use analytics to predict failures.

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.

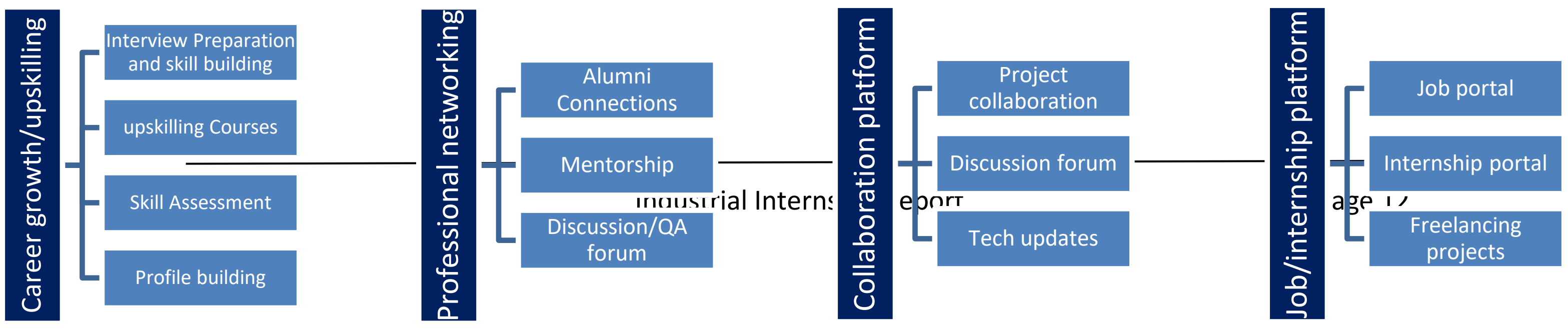
[Your College Logo]



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] https://kinsta.com/docs/application-hosting/app-quick-start/python-examples/

[2] https://www.geeksforgeeks.org/python-how-to-shorten-long-urls-using-bitly-api/

[3] https://www.amazon.com/Reema-Thareja/e/B00357V8ME

2.6 Glossary

Terms	Acronym
URL	Uniform Resource Locator - The web address you type into your browser.
Shortened URL	A compressed version of the original URL, typically used for sharing.
API	Application Programming Interface - A set of commands that allows programs to communicate with each other. A URL shortener service might have an API for developers to integrate shortening functionality.
Click Tracking	Recording data on how many times a shortened URL is clicked.
Base62 Encoding	A way to represent numbers using 62 characters (a-z, A-Z, 0-9). This is often used to create shortened URLs because it allows for shorter representations compared to base-10 (decimal).

3 Problem Statement: Building a Python URL Shortener

The Challenge:

Traditionally, web addresses (URLs) can be quite lengthy. This can be cumbersome when sharing them in emails, text messages, or on social media platforms with character limits.

Our Solution:

Develop a Python application that acts as a URL shortener. This program will take a long URL as input and generate a shorter, more manageable version.

Explanation:

Here's a breakdown of the functionalities involved:

- **Input:** The program will prompt the user to enter the long URL they want to shorten.
- **Processing:** Python code will handle the URL shortening logic. There are two main approaches:
  - **Hashing:** The URL can be converted into a unique string of characters using a hashing algorithm. This string can then be used as the shortened URL.
  - **Database Storage:** The long URL can be stored in a database, and a unique identifier (like a random code) is assigned. This identifier becomes the shortened URL. When someone clicks the shortened URL, the program redirects them to the original long URL stored in the database.
- **Output:** The program will present the user with the shortened URL. Additionally, it could offer options to:
  - Copy the shortened URL to the clipboard.
  - View the shortened URL in a web browser (if using a database approach).

4 Existing and Proposed solution

There are two main approaches to building a Python URL shortener:

Existing Solution 1: Hashing-based Shortening

- **Concept:** The long URL is converted into a unique string of characters using a hashing algorithm (e.g., MD5, SHA-256). This string serves as the shortened URL.
- **Pros:**
  - Simple to implement.
  - No database required.
- **Cons:**
  - Collision possibility: Two different URLs might generate the same shortened string. You'll need a strategy to handle this (e.g., adding a random prefix or checking for existing hashes).
  - No information about the original URL is retained. Once shortened, you can't easily retrieve the original long URL.

Existing Solution 2: Database-driven Shortening

- **Concept:** The long URL is stored in a database. A unique identifier (like a random code) is assigned and becomes the shortened URL. When someone clicks the shortened URL, the program retrieves the original long URL from the database and redirects them.
- **Pros:**
    - No collision issues.
    - Allows for additional features like custom URL creation and click tracking.
- **Cons:**
    - Requires setting up and managing a database.
    - Introduces security considerations for protecting user data in the database.

Proposed Solution:

This project can explore a combination of these approaches:

- **Start with Hashing for Basic Functionality:** Implement a basic shortener using hashing initially to grasp core concepts.
- **Optional Expansion: Integrate Database for Advanced Features:** As you gain experience, consider adding a database layer to enable features like:
    - **Custom URLs:** Allow users to define a specific shortened URL for their long URL. (e.g., "[invalid URL removed]")
    - **Click Tracking:** Track the number of times a shortened URL is clicked. This can provide insights into user behavior (useful for marketing or analytics).

4.1 Code submission (Github link) :

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

5 My learnings :

Building a Python URL shortener might seem like a simple task, but it packs a powerful learning punch. Here's how this project can fuel your Pythonic journey:

Solidifying Core Concepts:

- **String Manipulation:** You'll master string handling techniques, essential for working with URLs. Slicing, concatenation, and character encoding will become second nature.
- **Data Structures:** You'll gain experience with data structures like dictionaries (useful for storing shortened URLs and their corresponding originals - database approach).
- **Conditional Statements:** Mastering if-else statements and loops becomes crucial for handling user input, validating URLs, and implementing error handling.

Exploring Advanced Techniques:

- **Hashing Algorithms:** You'll delve into hashing algorithms (MD5, SHA-256) and understand how they can be used for URL shortening (with an understanding of collision handling).
- **Working with Libraries:** You might explore libraries like pyshorteners to interact with existing URL shortening services or leverage database libraries like sqlite3 for advanced functionality.
- **Optional: Web Development Concepts:** If you deploy your shortener online (database approach), you'll dabble in web development concepts like frameworks (Flask, Django) to handle user interaction and URL redirects.

Boosting Your Career Prospects:

- **Demonstrated Skills:** This project showcases your proficiency in Python fundamentals, problem-solving abilities, and an eagerness to learn (especially if you explore advanced features).
- **Project Management:** You'll gain experience in breaking down tasks, planning code structure, and testing functionalities - valuable project management skills for professional settings.

6 Future work scope :

While your Python URL shortener handles the core functionality, there's room for exciting future enhancements! Here are some ideas to consider:

Advanced Shortening Techniques:

- **Custom Shortened URLs:** Allow users to define their own shortened URLs for a more personalized touch. They could create shortened URLs like "https://accounts.google.com/AccountChooser?service=urlshortener"
- **Vanity URLs:** Implement a system for vanity URLs where users can pay a small fee to claim a specific, memorable shortened URL.
- **URL Expiration:** Add an option for shortened URLs to expire after a set time. This could be useful for temporary promotions or links containing sensitive information.

Enhanced Analytics & User Management:

- **Click Tracking:** Integrate click tracking to monitor how many times each shortened URL is clicked. This data can provide insights into user behavior and campaign effectiveness.
- **User Accounts:** Implement a user account system. Users could manage their shortened URLs, view click statistics, and potentially create custom short URLs (as mentioned above).

Scalability & Integration:

- **Database Optimization:** If using a database, explore optimization techniques to handle a large number of shortened URLs efficiently.
- **API Integration:** Develop an API for your URL shortener. This would allow other applications to integrate your shortening functionality.

THANK YOU !