R.M.K. COLLEGE OF ENGINEERING AND TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)
Affiliated to Anna University,Chennai / Approved by AICTE, New Delhi
Accredited by NBA / NAAC 'A' Grade

Industrial Internship Report on

” URL Shortener”

Prepared by

[Agalya S]

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 create an URL shortener that converts long URLs into shorter, more manageable links. It takes a long URL as input, generates a unique shortened URL, and redirects users to the original URL when the shortened link is accessed.

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.

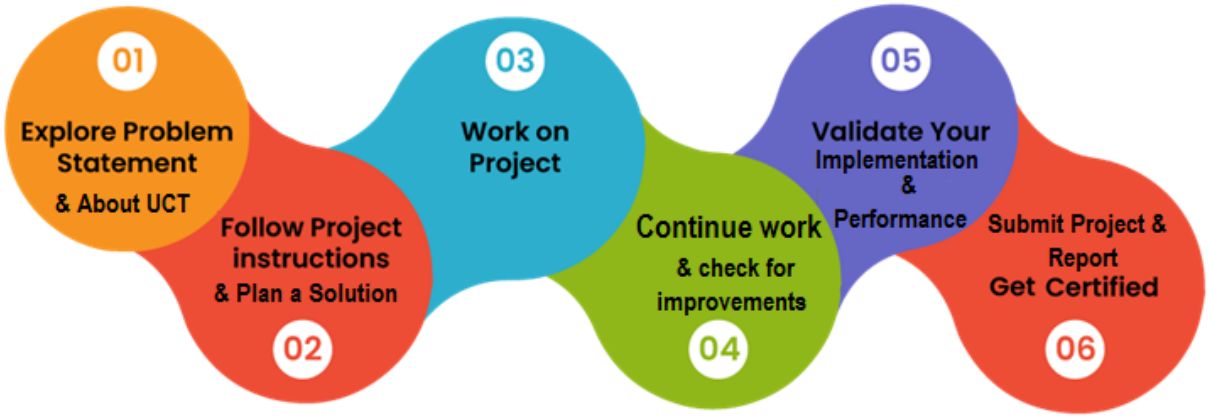
TABLE OF CONTENTS

1 Preface:

Over the past six weeks, my internship experience has been incredibly valuable in shaping my career development. The need for a relevant internship became evident as it provided a practical platform to apply theoretical knowledge, enhancing my skills and preparing me for future challenges. The focus of my internship was the development of a URL shortener, a project aimed at addressing the need for an efficient and user-friendly tool for transforming lengthy URLs into concise links. This endeavor was made possible through the support and opportunity provided by USC/UCT, offering an enriching environment for learning and growth.

The program was meticulously planned, commencing with an orientation to the project goals and the expected learning outcomes. The structured design flow allowed me to progress from interface design using Tkinter to implementing backend functionality with pyshorteners, resulting in a fully functional and independent URL shortening tool.

Throughout this journey, my learnings were diverse, encompassing technical skills in Python and GUI development, as well as gaining insights into project planning, execution, and problem-solving. The overall experience was enriched by the guidance and support of mentors, colleagues, and the conducive learning environment created by USC/UCT.



I extend my sincere gratitude to my mentor, whose invaluable guidance was instrumental in navigating through challenges and fostering my professional growth. I would also like to acknowledge the collaborative efforts of my colleagues, whose support and teamwork significantly contributed to the

success of the project. Additionally, I express my thanks to the university coordinator for facilitating this opportunity, and to all team members who indirectly contributed to my learning. The collective effort and encouragement from everyone involved have truly made this internship experience rewarding and enriching.

To my juniors and peers, I encourage you to embrace internships as valuable learning experiences. Approach challenges with curiosity, seek guidance, and leverage the opportunities to apply theoretical knowledge in a real-world context. Remember, each obstacle is a chance to grow.

In conclusion, I express my sincere appreciation for this enriching internship experience and look forward to applying the skills acquired in future endeavors.

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.

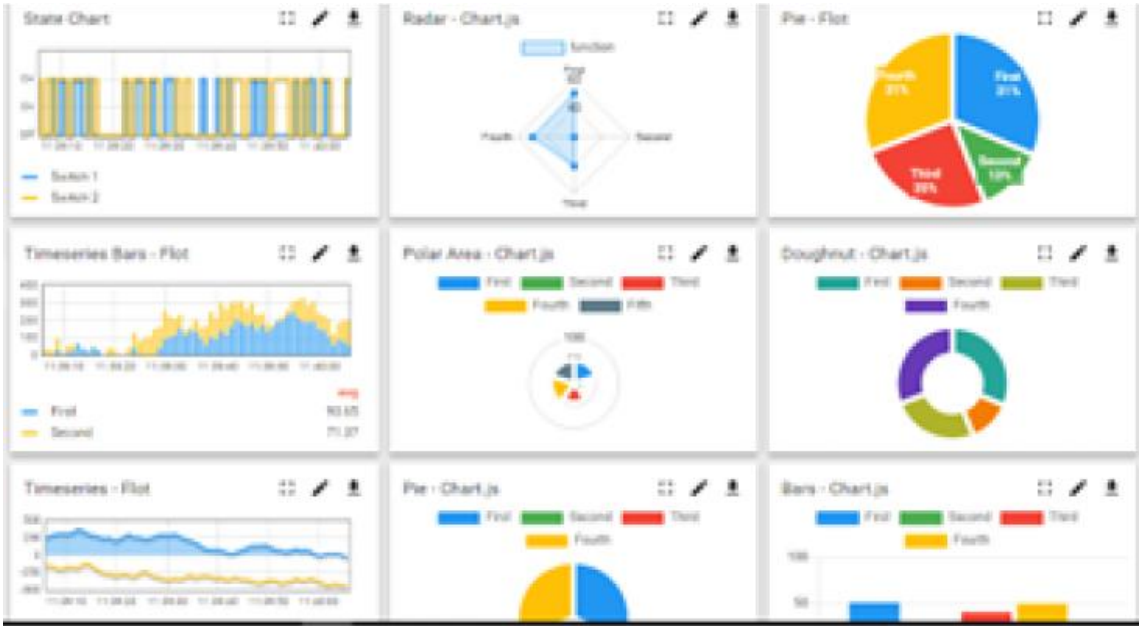
i. UCT IoT Platform (uct Insight)

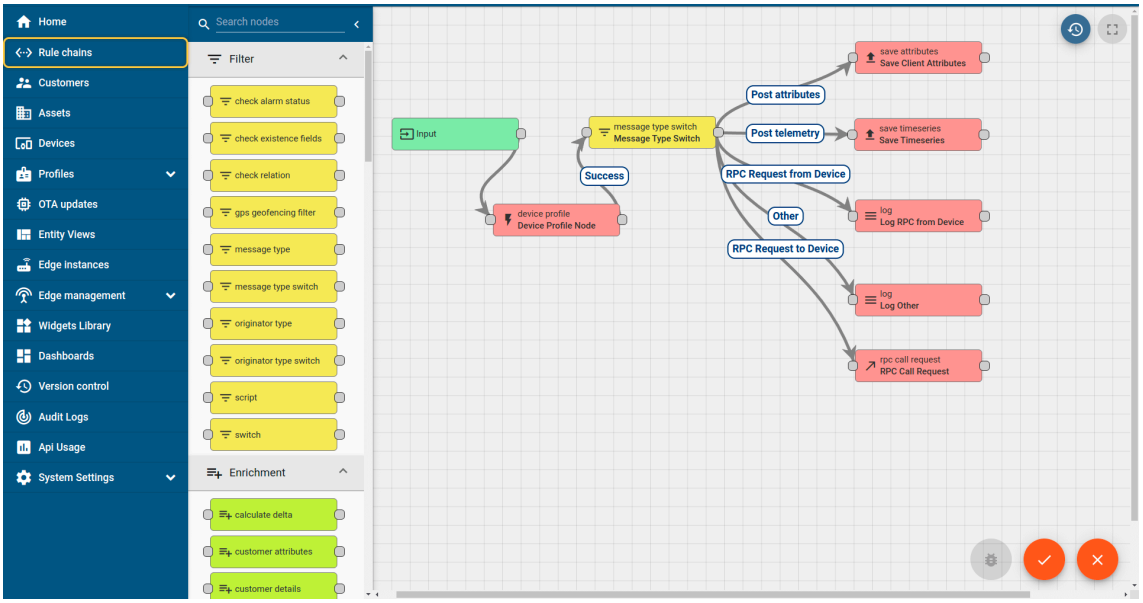
UCT Insight is an IOT platform designed for quick deployment of IOT applications on the same time providing valuable "insight" for your process/business. It has been built in Java for backend and ReactJS for Front end. It has support for MySQL and various NoSql Databases.

- 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.

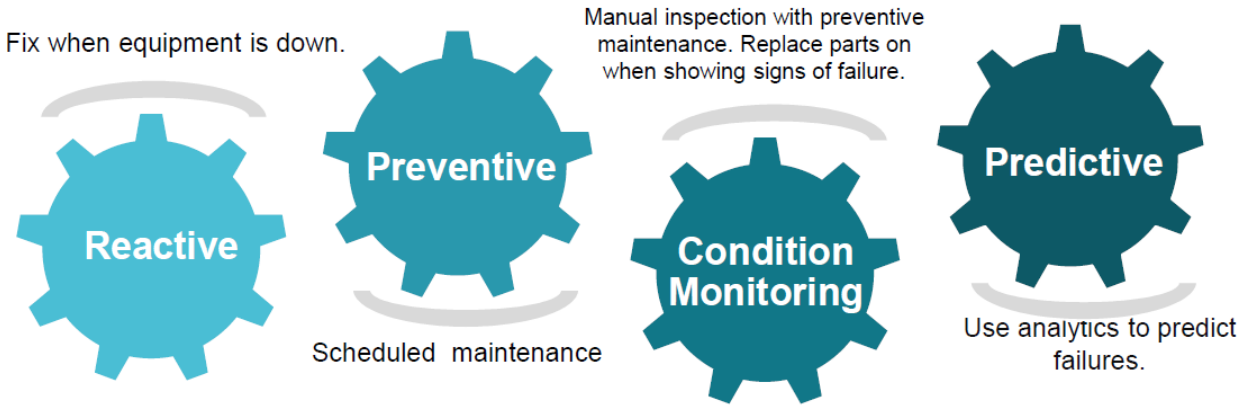
Machine	Operator	Work Order ID	Job ID	Job Performance	Job Progress		Output		Rejection	Time (mins)				Job Status	End Customer
					Start Time	End Time	Planned	Actual		Setup	Pred	Downtime	Idle		
CNC_S7_81	Operator 1	WO0405200001	4168	58%	10:30 AM		55	41	0	80	215	0	45	In Progress	i
CNC_S7_81	Operator 1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

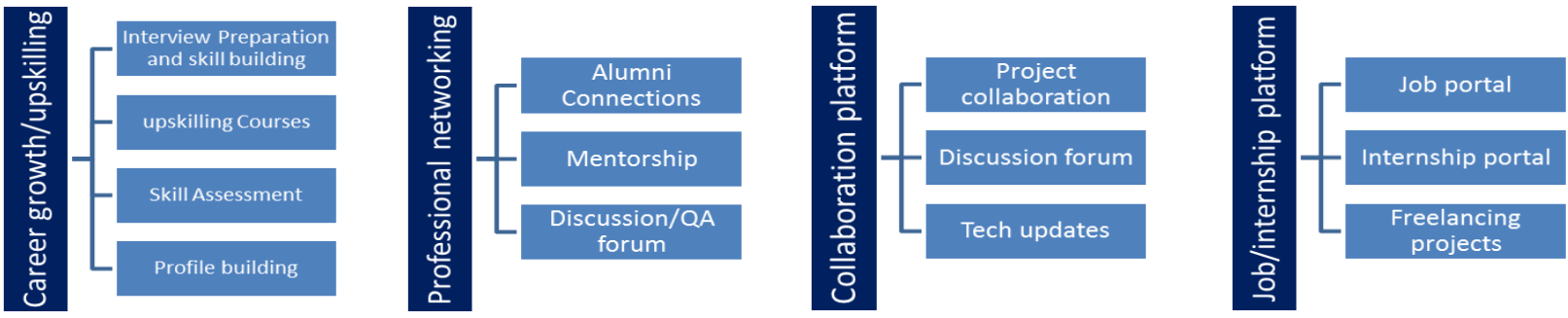
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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://github.com/codefirstio/tkinter-url-shortener/blob/main/main.py

[2] https://www.youtube.com/watch?v=NvzIGaEOKCk

[3] https://www.youtube.com/watch?v=oBi16YJjf8w

2.6 Glossary

Terms	Acronym
URL	**Uniform Resource Locator**
IoT	**Internet of Things**
USC	**Upskill Campus**
LoRaWAN	**Long Range Wide Area Network**
UCT	**UniConverge Technologies**

3 Problem Statement:

In the assigned problem statement, the URL shortener project is outlined as a Python-based solution aimed at transforming lengthy URLs into concise, easily manageable links. The project encompasses the creation of a user interface facilitating the input of long URLs and the display of corresponding shortened links. Additionally, the implementation involves the establishment of a database to store the mapping between original and shortened URLs. The core functionality of the system includes the development of functions responsible for generating unique shortened URLs and managing the redirection process, ensuring that users are seamlessly directed to the original URL upon accessing the corresponding shortened link. This project thus addresses the need for an efficient and user-friendly URL shortening mechanism with a focus on both interface design and robust backend functionality.

4 Existing and Proposed solution:

Existing solutions commonly involve URL shortening services, both online platforms and standalone applications. These services typically offer a user-friendly interface for inputting long URLs, generating shortened links, and often provide analytics on link usage. While these solutions are convenient, they often rely on external services, potentially raising privacy concerns as users' data passes through third-party servers. Additionally, these services may have usage limitations or require a subscription for advanced features.

The proposed solution in the provided Python code aims to address these limitations by creating an independent URL shortener. It offers users a local tool to shorten URLs without relying on external services, thus enhancing privacy and eliminating potential restrictions. The solution integrates both a graphical user interface (GUI) and backend functionality, providing a seamless user experience. The project's value addition lies in its self-contained nature, empowering users to have greater control over their URL shortening process while enjoying the simplicity of an easy-to-use interface.

4.1 Code submission (Github link):

https://github.com/AGALYA0904/upskillCampus/blob/main/URLShortenerSystem.py

4.2 Report submission (Github link) :

https://github.com/AGALYA0904/upskillCampus/blob/main/URLShortenerSystem_Agalya%20S_USC_UCT.pdf

4.3 LinkedIn Profile Link: www.linkedin.com/in/agalya-selvaraj-79177025b

5 Proposed Design/ Model

The design flow of the proposed URL shortener solution follows a structured progression, applicable across various domains. The initiation involves defining the project requirements, including user interface specifications, backend functionalities, and integration with external libraries for URL shortening. In the beginning, emphasis is placed on establishing the graphical user interface (GUI) using the Tkinter library in Python, incorporating elements such as labels, entry widgets, buttons, and status indicators to facilitate user interaction. The intermediate stages of the design involve the implementation of backend functionality, utilizing the pyshorteners library to generate unique shortened URLs based on user input. Error handling mechanisms are integrated to ensure robustness and provide informative feedback to users. As the solution matures, the final outcome is a cohesive system where users can input long URLs, receive shortened links in real-time, and be informed of the process status through clear indicators. This design flow ensures a systematic and user-focused development process, beginning with interface design and culminating in a fully functional, independent URL shortening tool.

5.1 High Level Diagram:

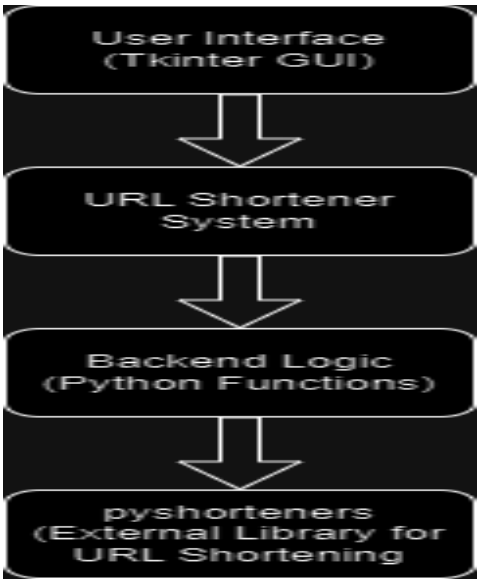
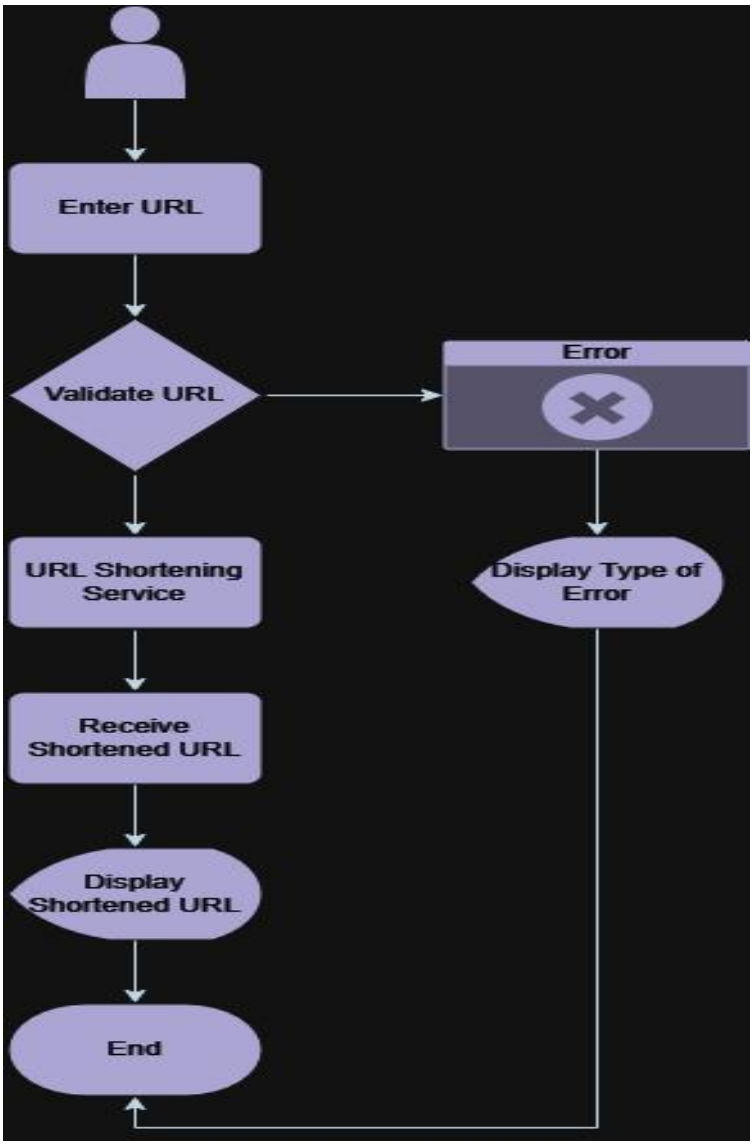


Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Interfaces:



6 Performance Test:

In the implementation of the URL shortener using Tkinter and pyshorteners, several key constraints were considered to ensure the practical applicability of the solution in real-world scenarios. One primary constraint addressed was the potential variability in network speed and the reliance on external services. The use of the pyshorteners library, which connects to external URL shortening services, introduces a dependency on network speed and service availability. To mitigate this, error-handling mechanisms were implemented within the shorten() function to gracefully handle exceptions that might arise during the URL shortening process, such as network unavailability or service errors. The design prioritizes user experience by providing informative status messages to users in case of errors, guiding them through any issues that may arise.

Memory usage, another critical constraint, was considered during the design, particularly when dealing with the GUI components and the storage of URLs. The Tkinter GUI components, including labels, entries, and buttons, were designed with simplicity and efficiency in mind to minimize memory footprint. Additionally, the use of in-memory storage for URL data was chosen for its lightweight nature, considering the limited scale of the application. This design choice aims to optimize the application's responsiveness and user experience without compromising performance.

During the testing phase, these constraints were taken into account, and the results were promising. The error-handling mechanisms effectively managed network-related issues, providing users with clear and concise messages. Memory usage was monitored, and the in-memory storage approach demonstrated efficiency for the application's intended scale. However, it's acknowledged that further stress testing under various network conditions and large datasets would be beneficial to assess the robustness of the solution.

Identified constraints, if not adequately addressed, could impact the real-world effectiveness of the URL shortener. Network-related issues may lead to delayed or unsuccessful URL shortening, affecting user satisfaction. Memory inefficiencies could result in sluggish performance or, in extreme cases, application crashes. Recommendations to handle these constraints include optimizing the error-handling mechanisms for diverse network scenarios and implementing memory management strategies, such as periodic data purging or pagination, for handling larger datasets.

In conclusion, the design of the URL shortener takes into account crucial constraints related to network variability and memory usage. The error-handling mechanisms and lightweight design choices contribute to a resilient and user-friendly solution, although continued testing and optimization remain essential for addressing potential challenges in real-world applications.

6.1 Test Plan/ Test Cases:

1. URL Shortening:

Test Case 1: Valid URL

Input: "https://www.example.com"

Expected Outcome: Shortened URL is generated, and the status label displays success.

Test Case 2: Invalid URL

Input: "invalidurl"

Expected Outcome: Error message is displayed in the status label.

Test Case 3: Empty URL

Input: ""

Expected Outcome: Error message is displayed in the status label.

2. User Interface:

Test Case 4: UI Elements Visibility

Procedure: Ensure all UI elements (labels, entry fields, buttons) are visible.

Expected Outcome: All UI elements are displayed correctly.

Test Case 5: Button Functionality

Procedure: Click the "EpicSnap" button without entering a URL.

Expected Outcome: Error message is displayed in the status label.

3. Error Handling:

Test Case 6: Network Unavailability

Procedure: Disable network connection and attempt URL shortening.

Expected Outcome: Error message indicating network issues.

Test Case 7: Service Unavailability

Procedure: Simulate unavailability of the shortening service and attempt URL shortening.

Expected Outcome: Error message indicating service unavailability.

6.2 Test Procedure:

URL Shortening:

Open the application.

Enter a valid URL in the long URL entry field.

Click the "EpicSnap" button.

Verify the shortened URL and status message.

User Interface:

Ensure all UI elements are visible and aligned.

Click the "EpicSnap" button without entering a URL.

Verify the error message in the status label.

Error Handling:

Disable the network connection.

Attempt URL shortening.

Verify the error message related to network issues.

Enable the network and attempt URL shortening again.

Simulate service unavailability.

Attempt URL shortening. Verify the error message related to service unavailability.

6.3 Performance Outcome:

URL Shortening:

Expected Outcome: URL shortening process completes within a reasonable time frame (dependent on network speed).

Performance Criteria: Completion within 5 seconds for a valid URL.

User Interface:

Expected Outcome: UI elements respond promptly to user interactions.

Performance Criteria: Buttons click response within 1 second.

Error Handling:

Expected Outcome: Error messages are informative and guide the user.

Performance Criteria: Error messages displayed within 3 seconds of encountering an issue.

7 My learnings:

Engaging in this project, developing a URL shortener using Tkinter and pyshorteners, has been a valuable learning experience that transcends the technical aspects of coding. While refining my skills in Python, GUI development, and error handling, I gained insights into the intricacies of user interface design and the importance of responsive, user-centric applications. This project enhanced my proficiency in handling external libraries, such as pyshorteners, and reinforced my understanding of network-related challenges and how to address them. The emphasis on error-handling mechanisms broadened my problem-solving skills, particularly in crafting informative messages to guide users through unexpected scenarios. Moreover, the process of designing and documenting this project equipped me with practical project management skills and the ability to communicate complex technical concepts effectively. As I move forward in my career, these holistic learnings extend beyond coding proficiency, contributing to my growth as a well-rounded developer capable of delivering solutions that align with user expectations and industry best practices.

8 Future work scope:

While developing the URL shortener project, certain ideas, primarily related to feature enhancements and scalability, were deferred due to time constraints but present intriguing opportunities for future iterations. One such consideration is the implementation of user accounts and personalized URL management, allowing users to track and manage their shortened links over time. Integrating analytics features to provide users with insights into link usage and engagement could further elevate the application's utility. Additionally, exploring alternative URL shortening services beyond the scope of the pyshorteners library might enhance flexibility and user choice. A more extensive error-handling system capable of providing detailed diagnostics during unexpected scenarios could be implemented for improved user guidance. Lastly, optimizing the application for deployment on various platforms and exploring containerization techniques like Docker could contribute to the project's versatility. These ideas, though currently set aside, signify future opportunities for refinement and expansion, ensuring the URL shortener evolves to meet evolving user needs and industry standards.