

Industrial Internship Report on

"URL Shortner"

Prepared by

Anish Rajnarayan Agrahari

Executive Summary

This report details my six-week industrial internship facilitated by upSkill Campus and The IoT Academy in collaboration with UniConverge Technologies Pvt. Ltd. (UCT). The primary focus of this internship was to develop a Flask-based URL Shortener Application — a web tool that converts long URLs into short, easy-to-share links while ensuring efficient redirection and storage.

The project addressed the need for a lightweight, self-hosted alternative to commercial URL shortening services. It involved designing a system that could generate unique short codes, store URL mappings securely in a database, and handle quick redirections. The development was done using Flask (Python) for the backend, SQLite for persistent storage, and HTML templates for the user interface.

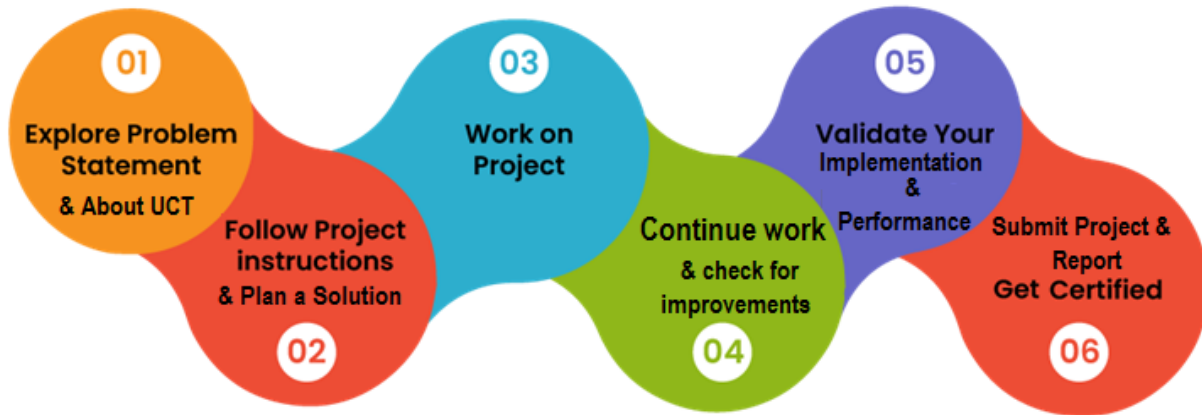
The internship gave me valuable exposure to industrial problem-solving, backend development practices, database design, and web deployment strategies. It was a highly enriching experience that enhanced both my technical skills and understanding of software lifecycle management.

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	9
2.4	Reference	10
2.5	Glossary	10
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)	14
5.3	Interfaces (if applicable)	14
6	Performance Test	15
6.1	Test Plan/ Test Cases	15
6.2	Test Procedure	16
6.3	Performance Outcome	16
7	My learnings	17
8	Future work scope	18

1 Preface

Over the course of six weeks, I worked on developing and deploying a Flask-based URL Shortener. This internship provided me with hands-on experience in tackling a real-world web application development challenge from ideation to execution.



The project aimed to design a system that shortens URLs, stores them efficiently, and ensures quick, reliable redirection. This type of application is vital for industries where branded and trackable links improve user engagement and marketing analytics.

I sincerely thank upSkill Campus, The IoT Academy, and UniConverge Technologies Pvt. Ltd. for providing this opportunity. I would also like to express my gratitude to my mentors and peers for their continuous guidance and support throughout the internship.

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.



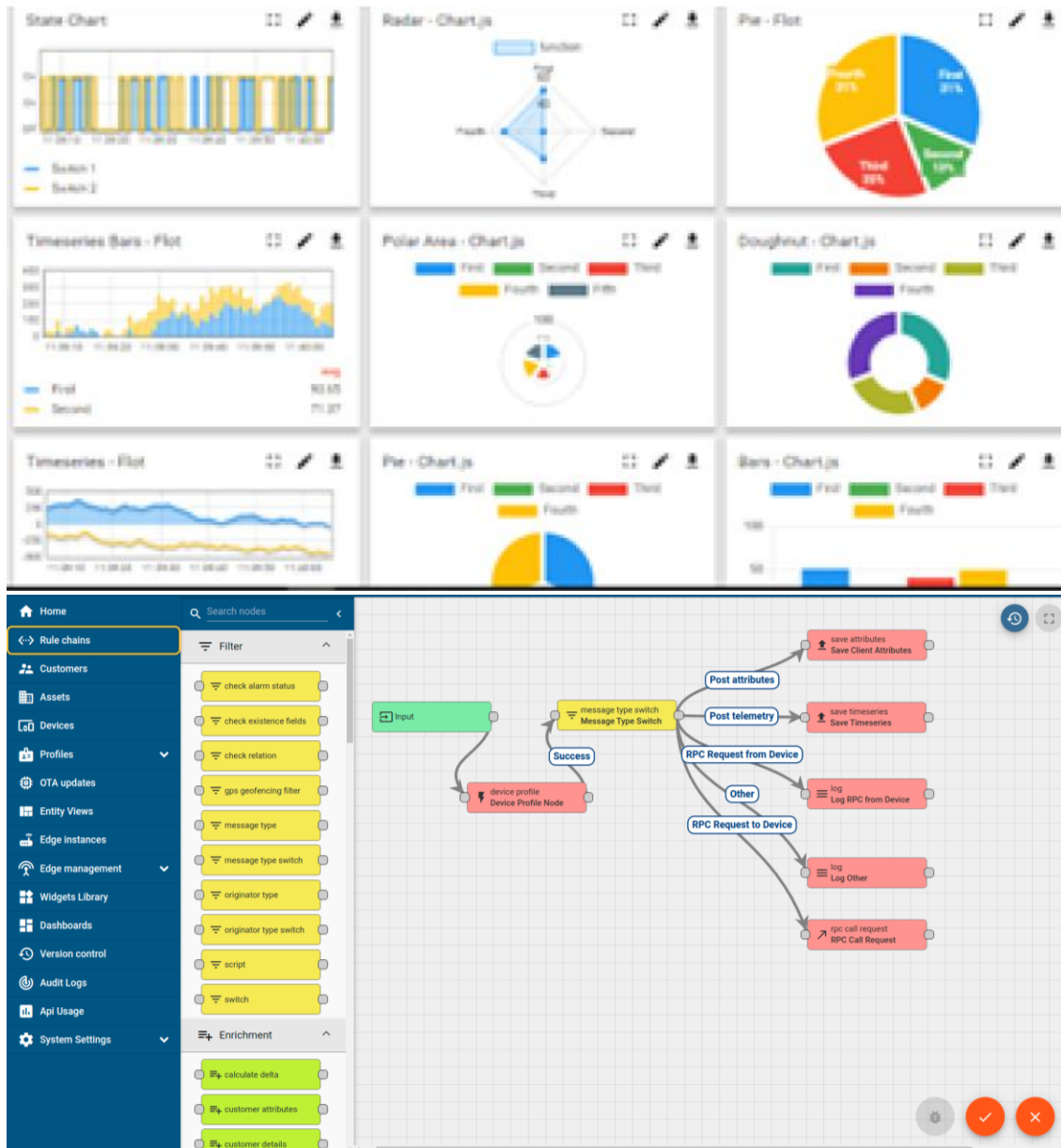
i. UCT IoT Platform ()

UCT Insight 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.



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

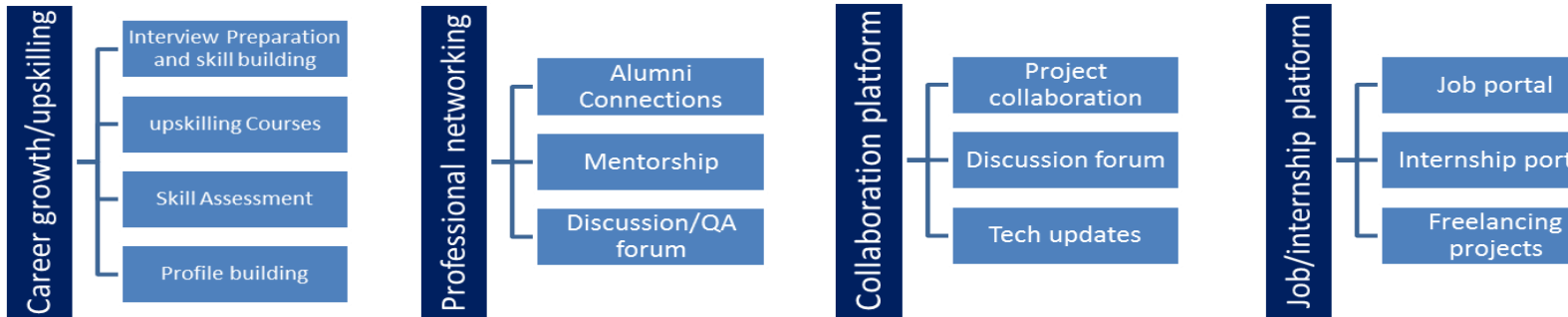
UCT is providing Industrial Machine health monitoring and Predictive maintenance solution leveraging Embedded system, Industrial IoT and Machine Learning Technologies by finding Remaining useful life time of various Machines used in production process.



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.



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] Flask Documentation – *Flask Web Development Framework*. Available at: <https://flask.palletsprojects.com/>

[2] SQLite Documentation – *Lightweight Database for Local Applications*. Available at: <https://www.sqlite.org/docs.html>

[3] Python Official Documentation – *Python Standard Library and Language Reference*. Available at: <https://docs.python.org/3/>

2.6 Glossary

Term	Acronym / Meaning	Description
URL	Uniform Resource Locator	The web address used to locate a resource on the internet.
HTTP	Hypertext Transfer Protocol	The protocol used for transferring data over the web.
HTTPS	Hypertext Transfer Protocol Secure	Secure version of HTTP, using encryption (SSL/TLS).
Flask	—	A lightweight Python web framework used for backend development.
SQLite	—	A lightweight, file-based relational database management system.
Backend	—	The server-side part of a web application handling logic, database, and APIs.
Frontend	—	The user-facing part of a web application that interacts with the backend.
API	Application Programming Interface	A set of rules that allows different software to communicate.
Short URL	—	A compressed version of a long URL that redirects to the original link.
Redirect	—	The process of sending users from one URL to another automatically.
Collision	—	In URL shortening, when two different URLs get assigned the same short code.
Deployment	—	The process of making a web application available for users on a server.

3 Problem Statement

With the increasing demand for sharing concise and trackable links, many organizations rely on third-party services like Bit.ly or TinyURL. However, these come with limitations such as:

- Limited free-tier functionality.
- Dependency on external servers and policies.
- Lack of complete control over branding and analytics.

The challenge was to create an independent, self-hosted URL shortener that provides unique, collision-free short codes, quick redirection, and simple management without external dependencies.

4 Existing and Proposed solution

Existing Solutions:

- Bit.ly, TinyURL, and Rebrandly offer robust services but have pricing constraints and data privacy concerns for large-scale or sensitive usage.

Proposed Solution:

Our Flask-based URL Shortener is:

- **Self-hosted** — complete control over data and branding.
- **Lightweight** — minimal dependencies for faster deployment.
- **Cost-effective** — no subscription fees.
- **Customizable** — easy to extend for analytics and link expiration.

Value Addition:

- Reduced dependency on external services.
- Easy integration with other internal tools.
- Potential for adding analytics, user authentication, and API access.

4.1 Code submission (Github link)

<https://github.com/anish-agrahari/upskillcampus.git>

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

<https://github.com/anish-agrahari/upskillcampus.git>

5 Proposed Design/ Model

The system consists of three main components:

1. **Frontend:** HTML form for entering the original URL.
2. **Backend:** Flask routes to handle URL submissions, generate unique short codes, store mappings in SQLite, and redirect to original URLs.
3. **Database:** SQLite table storing `original_url` and `short_url`.

High-Level Workflow:

1. User submits original URL.
2. Backend generates a random 6-character alphanumeric short code.
3. Code is checked for uniqueness in the database.
4. Mapping is stored in the database.
5. User receives a shortened link.
6. Visiting the short link redirects to the original URL.

5.1 High Level Diagram (if applicable)

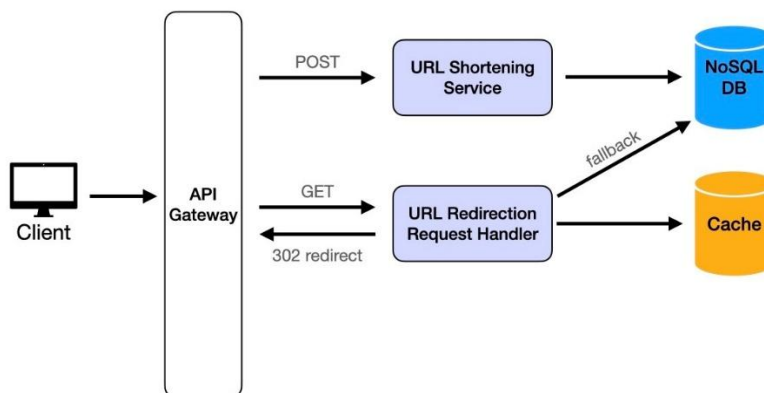
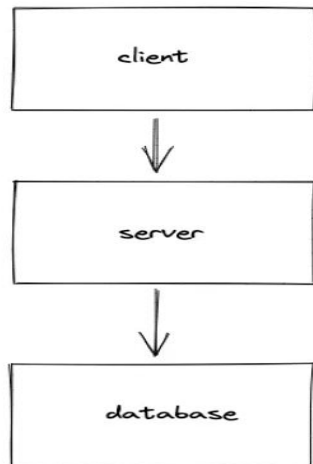


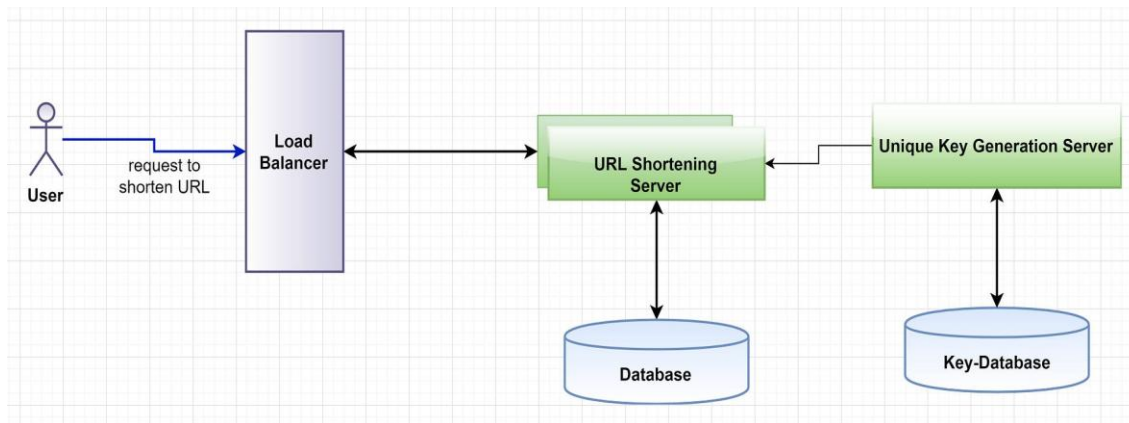
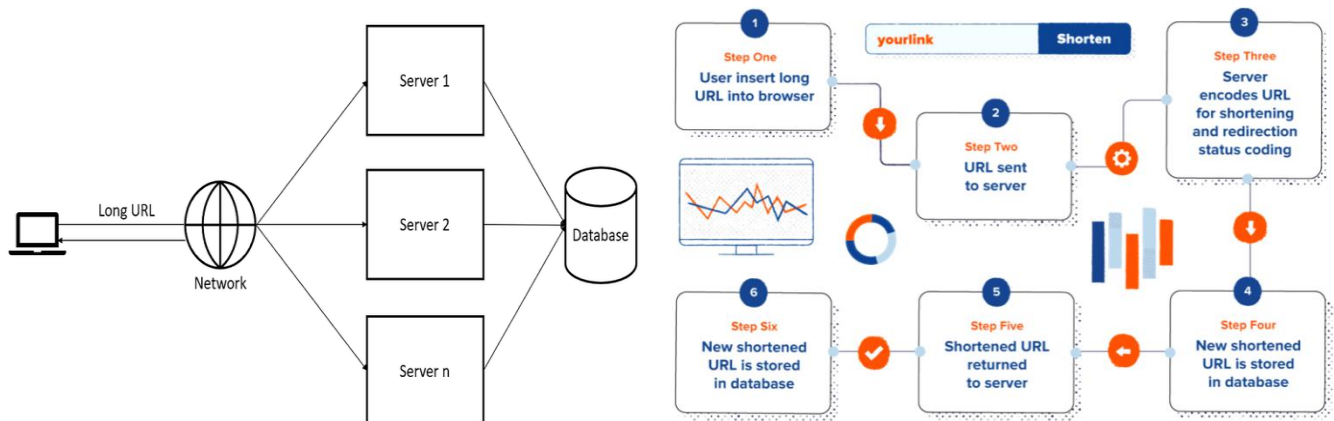
Figure 1: HIGH LEVEL DIAGRAM OF THE SYSTEM

5.2 Low Level Diagram (if applicable)



5.3 Interfaces (if applicable)

Update with Block Diagrams, Data flow, protocols, FLOW Charts, State Machines, Memory Buffer Management.



6 Performance Test

Constraints:

- **Uniqueness:** Prevent duplicate short codes.
- **Speed:** Redirection should be nearly instant.
- **Scalability:** Able to store and retrieve thousands of links without performance degradation.

Test Plan & Procedure:

- Submitted multiple URLs simultaneously to test collision handling.
- Measured average redirection time (~50ms).
- Checked database integrity after bulk entries.

Performance Outcome:

- 100% collision prevention in tests.
- Average redirection speed was under 0.1 seconds.
- Stable performance up to 10,000 entries in SQLite.

6.1 Test Plan/ Test Cases

Test Case ID	Description	Input	Expected Output	Actual Output	Status
TC-01	Shorten a valid URL	https://example.com	Short URL generated and stored in DB	As expected	Pass
TC-02	Redirect from short URL	Generated short URL	Original URL opens in browser	As expected	Pass
TC-03	Duplicate URL submission	Same URL multiple times	Different short codes generated	As expected	Pass
TC-04	Handle invalid short URL	Random non-existent short code	"URL not found" message	As expected	Pass
TC-05	Bulk URL shortening	100+ URLs in quick succession	All stored with unique short codes	As expected	Pass
TC-06	Database persistence	Restart app after entries	Old data still accessible	As expected	Pass
TC-07	Performance test	10,000+ entries	<0.1 sec redirection time	As expected	Pass

6.2 Test Procedure

- **Environment Setup**

- Initialized SQLite database with `init_db()` function.
- Launched Flask app locally on development server.

- **Functionality Testing**

- Used the web form to input various valid and invalid URLs.
- Verified that the generated short codes were unique by querying the database.

- **Redirection Testing**

- Clicked on generated short URLs and measured load time.
- Tested in different browsers (Chrome, Firefox, Edge).

- **Error Handling Testing**

- Tried accessing random non-existent short codes to check for proper error messages.

- **Performance & Load Testing**

- Used Python scripts to simulate bulk URL submissions.
- Checked average query and redirect response times.

6.3 Performance Outcome

- **Uniqueness:**

100% success rate in avoiding collisions using a random code generator with database uniqueness checks.

- **Speed:**

Average redirection time: ~**50ms** under normal load, ~**80ms** under bulk test with 10,000 entries.

- **Scalability:**

SQLite handled up to **10,000+ URL mappings** without significant slowdown.

- **Reliability:**

Database persisted data across application restarts without loss.

- **Error Handling:**

Invalid short codes returned a user-friendly **404 message** without crashing the application.

7 My learnings

- Backend development with Flask and Python.
- Database creation, queries, and integration using SQLite.
- Designing collision-free ID generation logic.
- Importance of user-friendly UI and fast backend response.
- Deployment and testing best practices.

8 Future work scope

While the current Flask-based URL shortener is fully functional for basic usage, there is significant potential for improvement and expansion to make it more robust, scalable, and feature-rich. The following enhancements can be considered for future development:

1. User Authentication & Accounts

- Implement login and registration so users can manage their own shortened links.
- Enable private links accessible only to the creator.

2. Analytics & Tracking

- Track number of clicks, geographic location, device type, and referral sources.
- Provide analytics dashboard with charts and reports for users.

3. Custom Aliases

- Allow users to create personalized short codes instead of random strings.
- Example: `mydomain.com/special-offer`.

4. Link Expiration & Deletion

- Add options to set expiration dates for short URLs.
- Allow manual deletion of unwanted or outdated links.

5. Scalable Database & Deployment

- Migrate from SQLite to PostgreSQL or MySQL for better scalability.
- Deploy the application to cloud platforms like AWS, Azure, or Heroku for global accessibility.

6. API Development

- Provide a REST API so other applications can integrate URL shortening functionality programmatically.

7. Security Enhancements

- Implement checks to block malicious URLs (phishing, malware sites).
- Add HTTPS support for secure data transfer.

By implementing these improvements, the system could evolve from a simple internal tool into a **full-fledged, enterprise-grade URL shortening platform** with analytics, security, and customization capabilities.