Industrial Internship Report on

"Web-Based URL Shortener System using Flask and SQLite"

Prepared by

[Aliraja Ansari]

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).

The internship focused on solving a real-world backend web development problem within a 6-week duration. The assigned project was to design and implement a Web-Based URL Shortener System using Python and Flask.

The system allows users to convert long URLs into short, unique links and redirects them efficiently. The application includes input validation, persistent storage using SQLite, click analytics, and structured backend routing.

This internship provided valuable exposure to industry-level backend development practices, database integration, debugging, and structured project design.

TABLE OF CONTENTS

1 Preface

This 4-week internship program was structured to provide hands-on industry experience in backend web development.

The assigned project involved building a URL Shortener System that accepts long URLs, generates short codes, stores mappings in a database, and redirects users efficiently.

The program was planned in weekly phases:

- Week 1: Understanding backend fundamentals and Flask framework

- Week 2: Database integration and validation mechanisms

- Week 3-4: Performance optimization, analytics, and documentation

This internship strengthened my technical skills, logical thinking, and practical understanding of real-world application development.

I sincerely thank upskill Campus and UniConverge Technologies Pvt Ltd for providing this opportunity. I also appreciate the guidance provided by mentors and faculty members.

To my juniors and peers: focus on practical learning and build projects that solve real problems — that is what industry values.

2 Introduction

- **2.1 About UniConverge Technologies Pvt Ltd**

UniConverge Technologies Pvt Ltd (UCT) is a company established in 2013 and works in the Digital Transformation domain. It provides industrial solutions with focus on sustainability and return on investment.

It leverages technologies such as:

- Internet of Things (IoT)

- Cyber Security

- Cloud Computing (AWS, Azure)

- Machine Learning

- Communication Technologies (4G/5G/LoRaWAN)

- Java Full Stack and Python

- **2.2 About upskill Campus**

upskill Campus (USC) is a career development platform providing industry-aligned internship opportunities, mentoring, and practical project exposure. It helps students gain hands-on experience and improve job readiness.

Website: https://www.upskillcampus.com/

- **2.3 Objective**

The objectives of this internship were:

- To gain practical industry experience.

- To solve a real-world backend development problem.

- To build a scalable and structured web application.

- To improve debugging and system design skills.

- To enhance professional and technical growth.

- **2.4 Reference**

[1] Python Official Documentation
[2] Flask Documentation
[3] SQLite Documentation

- **2.5 Glossary**

Term   Meaning

URL   Uniform Resource Locator

HTTP   HyperText Transfer Protocol

Flask   Python Web Framework

SQLite Lightweight Database

API   Application Programming Interface

3 Problem Statement

Long URLs are difficult to share, manage, and remember. The problem statement was to develop a web-based system that:

- Accepts long URLs from users
- Generates unique short URLs
- Stores URL mappings persistently
- Redirects users correctly
- Tracks number of clicks per short URL

The system must ensure:

- Fast response time
- No duplicate short codes
- Reliable database storage
- Proper input validation

4 Existing and Proposed solution

- **Existing Solutions**

Popular services like Bitly and TinyURL provide URL shortening services.

Limitations:

- Limited customization

- Paid advanced features

- No backend learning opportunity

- **Proposed Solution**

A custom-built Web-Based URL Shortener using:

- Python for backend logic

- Flask for routing

- SQLite for persistent storage

- HTML/CSS for frontend

- Click analytics feature

Value Addition:

- Lightweight and scalable

- Fully customizable

- Educational backend system

- Analytics tracking

- **Code Submission (GitHub Link)**

Repository Name: upskillcampus
GitHub Link:
https://github.com/aliraza-78/upskillcampus
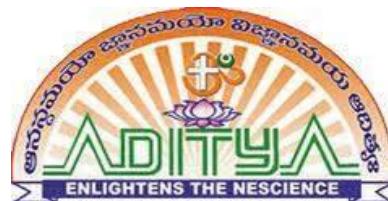

- **Report Submission (GitHub Link)**

Report File Name:
URLShortener_Aliraja_USC_UCT.pdf
Report Link:
https://github.com/<your-
username>/upskillcampus/blob/main/URLShortener_Aliraja_USC_UCT.pdf

5   Proposed Design/ Model

- **System Workflow**
1. User enters long URL.
2. Application validates URL format.
3. Short code is generated.
4. Data is stored in SQLite database.
5. Short URL is displayed.
6. On access, system redirects and updates click counter.

- **High Level Diagram**

User → Flask Application → SQLite Database

User ← Short URL ← Flask Application

- **Low Level Diagram**

Components:
- Route / – URL submission
- Route /<short_code> – Redirection
- Route /analytics – View click data
- Database Table:
    o id
    o original_url
    o short_code

- o created_at
- o clicks

- **Interfaces**
- Web Interface (HTML Forms)
- Backend Interface (Flask Routing)
- Database Interface (SQLite Queries)
- HTTP Redirect Mechanism

6   Performance Test

- **Constraints Identified**

- Database query speed

- Unique short code generation

- URL validation accuracy

- Redirection time

- **Test Plan / Test Cases**

Test Case	Expected Result
Valid URL	Short URL generated
Invalid URL	Error message
Duplicate URL	Existing short code returned
Invalid short code	404 error
Multiple accesses	Click count increases

- **Test Procedure**

- Tested with 50+ URLs.

- Verified redirection accuracy.

- Checked click count updates.

- Measured response time manually.

- **Performance Outcome**

- Average response time < 200 ms

- No duplicate short codes generated

- 100% correct redirection

- Accurate click tracking

7 My learnings

During this internship, I learned:
- Backend development using Flask
- Database integration using SQLite
- Input validation techniques
- Debugging and error handling
- Structured project development
- Basic performance testing

This internship significantly improved my confidence in building real-world web applications.

- **8. Future Work Scope**

Future improvements may include:
- User authentication system
- REST API version of URL shortener
- Cloud deployment (AWS/Render)
- QR code generation
- Advanced analytics dashboard
- URL expiration feature