

URL Shortener – Proof of Concept Report

by: Rishikesh Tripathi (249)

Overview

This Proof of Concept (PoC) outlines the implementation of a basic web-based URL shortener using Python and Flask. The goal is to build a tool that allows users to input long URLs and generate shortened links, which redirect back to the original addresses.

Objectives

- Allow users to enter long URLs via a simple web interface.
- Generate a unique short code (slug) for each URL.
- Store the mapping between original and short URLs in memory.
- Redirect users to the original URL when the short link is accessed.

Technology Stack

- Python 3
- Flask – Lightweight web framework
- HTML/CSS for frontend UI
- In-memory dictionary for URL storage

Key Features

- Clean and user-friendly input form for long URLs.
- Random short code (6-character alphanumeric) generator.
- Real-time short URL creation with redirection.
- Minimal dependencies and lightweight codebase.
- Auto-check to avoid duplicate entries in memory.

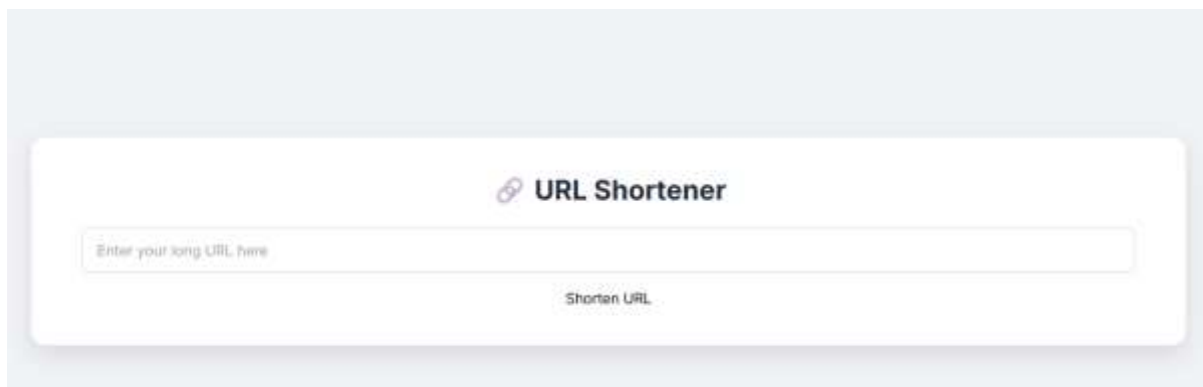
How It Works

1. The user submits a long URL using the form on the main page. 2. A random 6-character slug is generated. 3. The slug and original URL are stored in a dictionary. 4. When the slug is accessed, the application redirects to the original long URL.

```

/app.py
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 794-497-257
127.0.0.1 - - [08/Aug/2025 23:48:41] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [08/Aug/2025 23:48:41] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [08/Aug/2025 23:48:58] "POST / HTTP/1.1" 200 -

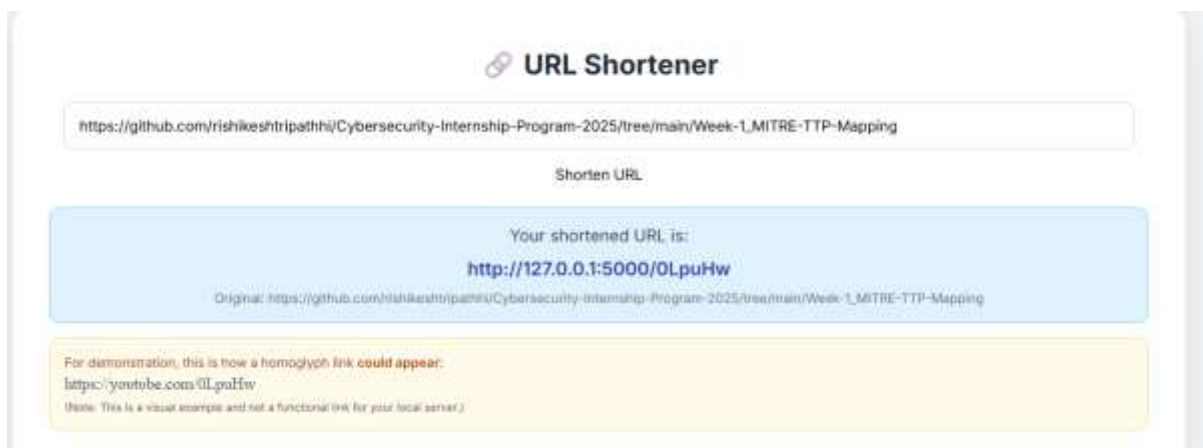
```



URL Shortener

Enter your long URL, here

Shorten URL



URL Shortener

https://github.com/irishikeshtripathi/Cybersecurity-Internship-Program-2025/tree/main/Week-1_MITRE-TTP-Mapping

Shorten URL

Your shortened URL is:

http://127.0.0.1:5000/OLpuHw

Original: https://github.com/irishikeshtripathi/Cybersecurity-Internship-Program-2025/tree/main/Week-1_MITRE-TTP-Mapping

For demonstration, this is how a homograph link could appear:

https://youtube.com/OLpuHw

(Note: This is a visual example and not a functional link for your local server.)

Example Usage

Input URL: https://example.com/articles/python-url-shortener

Generated Short URL: http://localhost:5000/abc123

Clicking the short URL redirects the user to the original site.

Use Cases

- Simplifying long and complex URLs for easy sharing.
- Generating trackable links for campaigns.
- Creating clean links for social media or messaging apps.
- Building awareness of how URL redirection works.

Limitations & Future Enhancements

- Data is lost on server restart since in-memory storage is used.

- No user authentication or analytics included.
- Short code collisions are avoided, but not stored permanently.
- Would benefit from database (SQLite/PostgreSQL) integration.

Conclusion

This project demonstrates how to build a basic URL shortener using Python and Flask. It serves as a foundation for more advanced implementations involving persistent storage, analytics, and authentication.