Proof of Concept (PoC) — URL Shortener Tool

Tool Name: URL Shortener

Name : Soham Kadu

Intern id: 235

Description

This is a simple **Python + Flask** web-based tool that allows users to enter a long URL and receive a shortened version.
It demonstrates basic web app development, random slug generation, URL mapping storage, and HTTP redirection.

What Is This Tool About?

The tool creates unique short codes (slugs) for long URLs and stores them in memory.
When someone visits the short URL, they are instantly redirected to the original long URL — similar to services like Bit.ly or TinyURL.

Key Features

1. **Web Form Interface** — Enter long URLs easily.
2. **Random Slug Generation** — 6-character alphanumeric short codes.
3. **In-Memory Storage** — Fast URL mapping (upgradable to SQLite).
4. **Instant Redirection** — Short link sends the user to the long link.
5. **Minimal Dependencies** — Just Flask and Python standard library.
6. **Lightweight** — Runs on any OS with Python installed.

Types / Modules Available

- Web form (Flask HTML rendering)
- Random slug generator (string + random)
- URL mapping dictionary
- HTTP redirect endpoint

How Will This Tool Help?

- Quickly create short, shareable links for testing or personal use.
- Learn core backend web development concepts.
- Demonstrate Flask-based REST-like functionality.

Input & Output Example

Input (Long URL):

https://www.example.com/very/long/path?with=query&params=1

Output (Short URL):

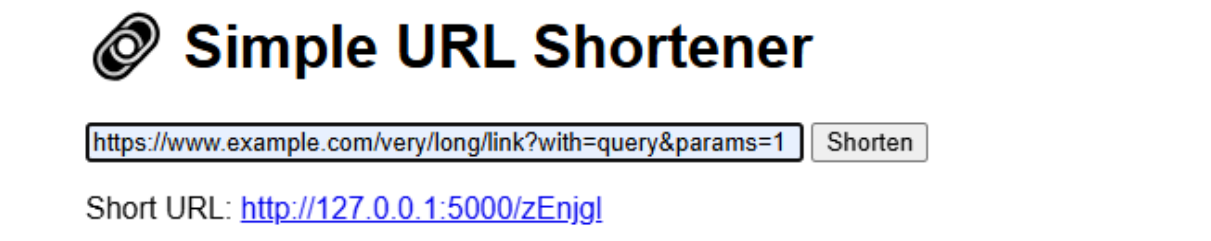
http://127.0.0.1:5000/aB9xY2

Command to Run

python url_shortener.py

Sample Output in Browser

```
PS C:\Users\Admin\Desktop\shortnerlink> python url_shortener.py
 * Serving Flask app 'url_shortener'
 * 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: 177-863-532
127.0.0.1 - - [08/Aug/2025 17:14:27] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [08/Aug/2025 17:14:27] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [08/Aug/2025 17:14:49] "POST /shorten HTTP/1.1" 200 -
```



A Brief Summary

1. User visits homepage (/) → enters long URL.
2. Form submits to /shorten → generates slug + saves mapping.
3. User gets a short link to share.
4. Anyone visiting /slug is redirected to the original URL.

Best Case Scenarios

- Shortening URLs in internal reports.
- Quickly generating test links for demos.
- Personal bookmarking with clean short codes.

Limitations / Suggestions

- Currently uses in-memory storage (data lost on restart).
- Could add SQLite/PostgreSQL support for persistence.
- Could implement custom slugs.
- Could track click counts for analytics.

Good About the Tool

- Very easy to run and understand.
- Simple structure — good for learning.
- Cross-platform.

Summary

This URL Shortener tool is a small yet functional web app built with Python and Flask.
It covers core backend concepts — form handling, random code generation, data mapping, and HTTP redirects.