Proof of Concept (PoC) Report

Tool Name: Simple URL Shortener

Author: Sristi Dutta

Intern Id: 387

Executive Summary

The **Simple URL Shortener** is a Python-Flask based web application designed to generate short, easy-to-share links from long URLs. Users can input any valid web address, and the application will create a unique short code that redirects back to the original link. This PoC demonstrates how the tool manages URL mapping in a local SQLite database, ensures code uniqueness, and delivers an interactive, user-friendly interface.

This project validates the functionality of URL shortening systems, showing how they can improve usability, save space in communications, and track link usage in extended implementations.

Objective

The goal of this PoC is to validate the functionality of the Simple URL Shortener by:

- Accepting a long URL from the user.
- Generating a unique short code using Base62 encoding (letters and digits).
- Storing and retrieving URL mappings from an SQLite database.
- Redirecting short codes to their corresponding original URLs.
- Demonstrating a functional web interface with a clean design.

Scope

In Scope:

- Shortening of valid URLs.
- Persistent storage in SQLite database.
- Automatic prevention of duplicate short code generation for the same URL.
- Redirection from short URL to original URL.
- Basic HTML/CSS-based interface.

Out of Scope:

- Link analytics (click tracking, geolocation data).
- Expiration dates for links.

- Authentication or user account systems.
- Security hardening against malicious inputs (basic validation only).

Tool Overview

The application uses **Flask** as its backend framework, **SQLite** as its database, and a minimal front-end design via `style.css`.

The core functionality includes:

- Initializing the database with a urls table.
- Checking if a submitted URL already exists.
- Generating a random 6-character alphanumeric short code if it doesn't.
- Serving the front-end form via index.html.
- Redirecting requests from a short code to the stored long URL.

Key Features:

- Lightweight and fast local deployment.
- Persistent mapping storage in SQLite.
- Avoids duplicate entries for the same URL.
- Simple, mobile-friendly UI.

Requirements

Software Requirements:

- Python (latest version recommended)
- Flask (Python package)
- SQLite3 (comes pre-installed with Python)

Data Requirements:

- `database.db` — stores URL mappings.
- `app.py` — main application script.
- templates/`index.html` — HTML front-end.
- static/`style.css` — UI styling.

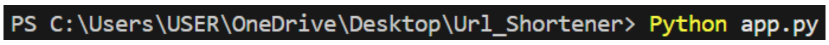
Steps to Run the Tool

1. **Install Python and Flask**

```
pip install flask
```

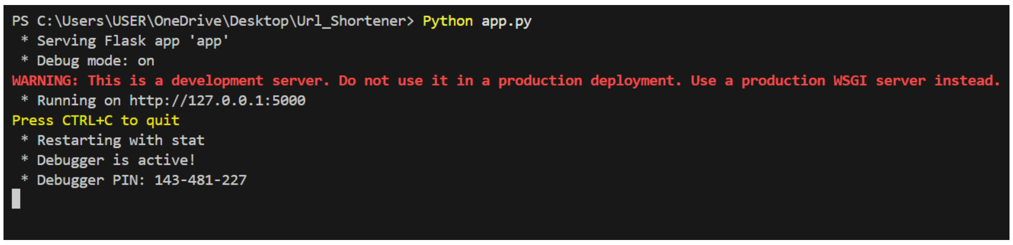
2. **Ensure `app.py` is configured correctly**

The `init_db()` function will automatically create the urls table in `database.db`.



```
PS C:\Users\USER\OneDrive\Desktop\Url_Shortener> Python app.py
```

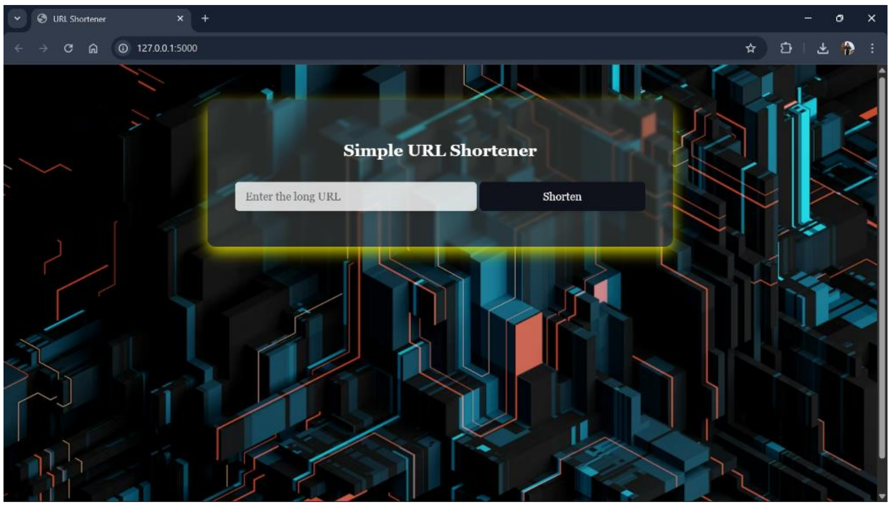
3. **Run the Application**



```
PS C:\Users\USER\OneDrive\Desktop\Url_Shortener> Python app.py
 * Serving Flask app 'app'
 * 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: 143-481-227
```

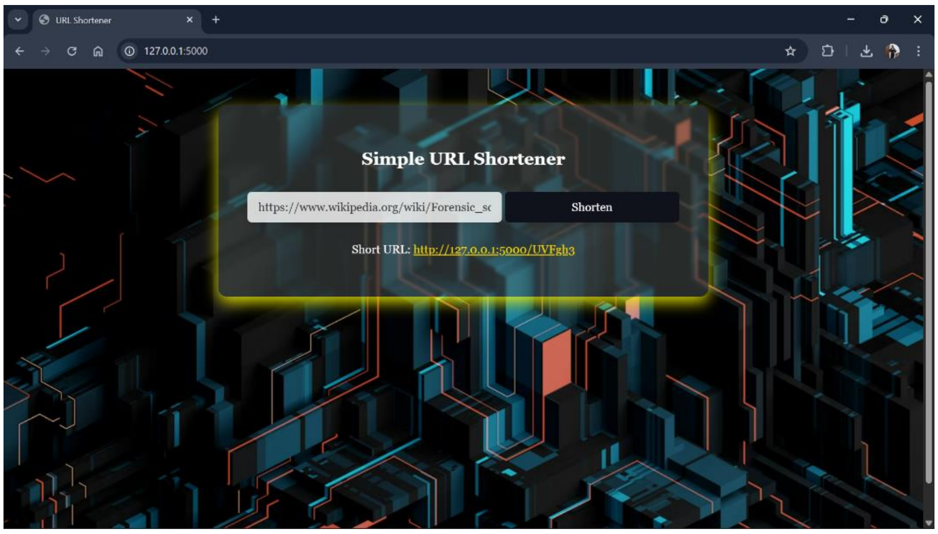
This will start the Flask development server at `http://127.0.0.1:5000/`.

4. **Access the Web Interface**



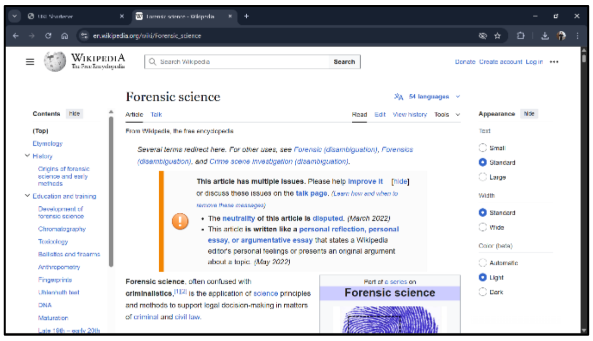
- Open your browser and navigate to `http://127.0.0.1:5000/`.
- Enter a long URL and click **Shorten**.
- Copy and use the generated short link.

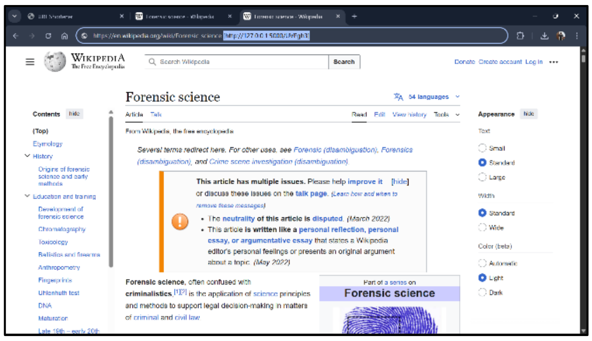
PoC Test Execution



https://www.wikipedia.org/wiki/Forensic_sc

Short URL: http://127.0.0.1:5000/UVFgh3


URL Shortener
Forensic science - Wikipedia
en.wikipedia.org/wiki/Forensic_science
Search Wikipedia
Search
Donate Create account Log in


URL Shortener
Forensic science - Wikipedia
Forensic science - Wikipedia
https://en.wikipedia.org/wiki/Forensic_science http://127.0.0.1:5000/UVFgh3
Search Wikipedia
Search
Donate Create account Log in

Forensic science

54 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

Several terms redirect here. For other uses, see Forensic (disambiguation), Forensics (disambiguation), and Crime scene investigation (disambiguation).

This article has multiple issues. Please help improve it or discuss these issues on the talk page. (Learn how and when to remove these messages)

- The neutrality of this article is disputed. (March 2022)
- This article is written like a personal reflection, personal essay, or argumentative essay that states a Wikipedia editor's personal feelings or presents an original argument about a topic. (May 2022)

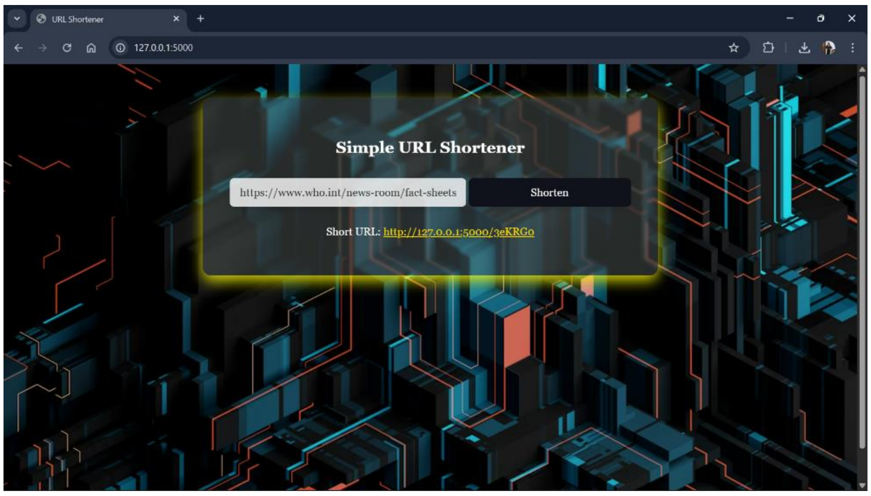
Forensic science, often confused with **criminalistics**,[1][2] is the application of science principles and methods to support legal decision-making in matters of criminal and civil law.

Part of a series on
Forensic science

Contents hide: (Top), Etymology, History, Origins of forensic science and early methods, Education and training, Development of forensic science, Chromatography, Toxicology, Ballistics and firearms, Anthropometry, Fingerprints, Uhlenhuth test, DNA, Maturation, Late 19th – early 20th

Appearance hide: Text — Small, Standard, Large; Width — Standard, Wide; Color (beta) — Automatic, Light, Dark

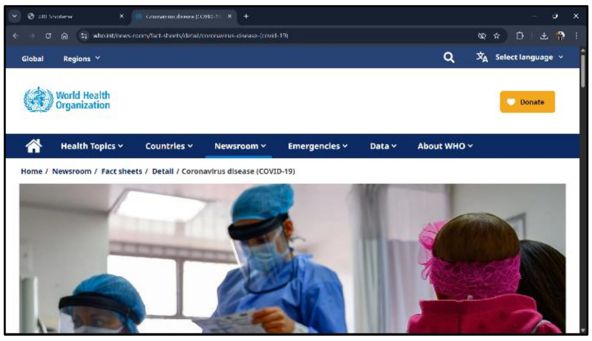
Wikipedia Short Link

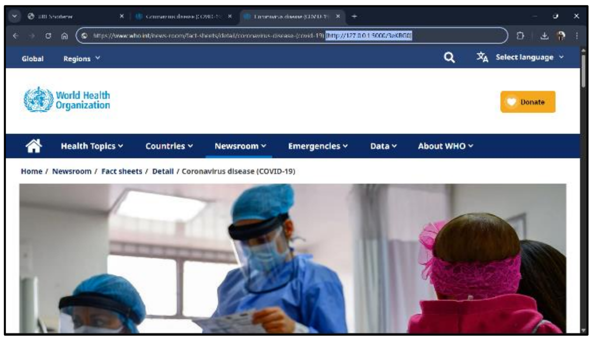


Simple URL Shortener

https://www.who.int/news-room/fact-sheets | Shorten

Short URL: http://127.0.0.1:5000/3eKRGo


URL Shortener
Coronavirus disease (COVID-19)
who.int/news-room/fact-sheets/detail/coronavirus-disease-(covid-19)
Global
Regions
Select language
World Health Organization
Donate
Health Topics
Countries
Newsroom
Emergencies
Data
About WHO
Home / Newsroom / Fact sheets / Detail / Coronavirus disease (COVID-19)


URL Shortener
Coronavirus disease (COVID-19)
Coronavirus disease (COVID-19)
https://www.who.int/news-room/fact-sheets/detail/coronavirus-disease-(covid-19) http://127.0.0.1:5000/3eKRGo
Global
Regions
Select language
World Health Organization
Donate
Health Topics
Countries
Newsroom
Emergencies
Data
About WHO
Home / Newsroom / Fact sheets / Detail / Coronavirus disease (COVID-19)

WHO Short Link

Use Case Scenario

A content creator wants to share a long YouTube playlist link on social media. Instead of pasting the lengthy URL, they use this tool to generate a short, clean link. The audience can then easily click and access the original video playlist without clutter in the post.

Advantages

- **Simplicity:** Easy to set up locally.
- **Persistent Storage:** All mappings are saved until explicitly removed.

- **Reusability:** Same link always returns the same short code.
- **UI-Friendly:** Clean, minimalistic interface styled with modern CSS.

Threat Impact Analysis

Potential Risks Without This Tool:

- Sharing long, unattractive URLs that may discourage clicks.
- Risk of breaking links when sent via character-limited platforms.

Possible Security Concerns (if unaddressed):

- **Phishing abuse** — attackers could mask malicious sites under short links.
- **Spam** — automated submissions without rate limits.

Impact Reduction Measures (Future Work):

- Add URL scanning before shortening.
- Implement analytics and abuse detection.
- Allow password-protected short links.

Future Enhancements

- **Custom Short Codes** — allow users to specify their own.
- **Click Analytics** — track usage and sources.
- **Link Expiry** — automatically delete after a set date.
- **API Support** — enable automated shortening via HTTP requests.
- **Security Validation** — integrate with phishing/malware scanners.

Conclusion

This PoC confirms that the **Simple URL Shortener** successfully converts long URLs into short, shareable links and stores them for persistent access. With its lightweight Flask backend, SQLite storage, and simple interface, the tool is functional for small-scale use cases. By adding analytics, security features, and customization, it can scale into a more robust public-facing URL shortening service.