Proof Of Concept: URL SHORTNER

Objective

Create a web-based URL shortener that:
- Accepts a long URL from the user.
- Generates a unique short code.
- Stores the mapping between short code and original URL in SQLite.
- Redirects users from short URL to the original URL.
- Displays a fake domain name in the shortened URL link for presentation, while maintaining real local links for functionality.

Technology Stack

- Backend: Python with Flask framework
- Database: SQLite
- Frontend: HTML with Flask templating (Jinja2)

Features

1. Form input: Users paste a long URL.
2. Short code generation: 6-character random alphanumeric string.
3. Storage: Uses SQLite to store mappings.
4. Redirection: Accessing /<short_code> redirects to the original URL.
5. Fake domain display: The displayed short URL uses a fake domain (e.g., https://google.com/abc123), but the actual clickable link uses the local Flask server URL (e.g., http://127.0.0.1:5000/abc123).

Code Overview

- app.py handles Flask routes:
  - / serves the form and generates short URLs.
  - /<short_code> redirects to the original URL.
- Database initialized with SQLite.
- Random short code generator ensures uniqueness.

How it works

1. User opens the app on local server (http://127.0.0.1:5000).
2. User inputs a long URL and submits.
3. App generates a short code, saves mapping in DB.
4. App displays the shortened URL with fake domain text but clickable link points to local

server.
5. User clicks short URL, app looks up the original URL and redirects.

Limitations

- The fake domain is only for display purposes.
- Clicking the fake domain URL will not work unless hosted with proper DNS setup.
- Current implementation stores data only locally (no persistence across servers).

Example Output

Input URL:
https://www.youtube.com/watch?v=dQw4w9WgXcQ

Output:
Displayed link: https://google.com/aB9xY2 (fake domain shown)
Actual clickable link: http://127.0.0.1:5000/aB9xY2

Instructions to Run

1. Clone or copy the code to your local machine.
2. Install Flask: pip install flask
3. Run app: python app.py
4. Open browser at http://127.0.0.1:5000
5. Paste any long URL and generate a short link.
6. Click on the short link to test redirection.

Name: Shreya Bishwas Pandey

InternId: -120

What this POC intends to achieve:

This PoC demonstrates a basic URL shortener that takes a long URL, creates a unique short code, stores the mapping, and redirects users to the original URL. It also shows a fake domain name for the short URL to enhance presentation while keeping actual redirects functional locally**.**