Name: Ankita Kolapte
Intern ID: 270

Proof Of Concept: URL Shortener with QR Code Support

Objective

Create a web-based URL shortener that:
- Accepts a long URL from the user.
- Generates a unique 7-character alphanumeric short code.
- Stores the mapping between short code and original URL in SQLite.
- Redirects users from short URL to the original URL.
- Generates a QR code for each short URL for easy scanning on mobile devices.

Technology

Stack

- Backend: Python with Flask framework
- Database: SQLite
- Frontend: HTML with Flask templating (Jinja2) - Libraries:
- Flask → Web framework
- sqlite3 → Lightweight database
- qrcode → Generate QR codes
- string & random → Unique short code generator

Features

1. Form input: Users paste a long URL.
2. Short code generation: 7-character random alphanumeric string.
3. Storage: Uses SQLite to store mappings.
4. Redirection: Accessing / redirects to the original URL.
5. QR code generation: Automatically generates a QR code for each short URL.

Code Overview

- app.py handles Flask routes:
- / serves the form and generates short URLs.
- / redirects to the original URL.

- Database initialized with SQLite.
- Random short code generator ensures uniqueness.
- QR code generator creates an image file in static/ directory for each new short URL.

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 and saves the mapping in the database.
4. App creates a QR code for the short URL and saves it in the static/ folder.
5. The shortened URL and QR code are displayed to the user.
6. Visiting the short link or scanning the QR code redirects to the original URL.

Limitations

- QR codes are generated locally and may need hosting for remote access.
- Current implementation stores data only locally.
- No analytics or click tracking implemented.

Output

Input URL:
https://www.example.com/very/long/url

Output:
Short URL: http://127.0.0.1:5000/XyZ123a
QR Code: Saved as static/XyZ123a.png

Instructions

1. Clone or copy the code to your local machine.
2. Install dependencies: pip install flask qrcode
3. Run database setup script to create urls.db.
4. Start the Flask app: python app.py
5. Open browser at http://127.0.0.1:5000
6. Paste any long URL and generate a short link with a QR code.
7. Test by clicking the link or scanning the QR code.

Conclusion

This PoC demonstrates an enhanced URL shortener that not only shortens URLs but also provides QR codes for each link, making it easier to share and access on mobile devices. It focuses on combining traditional short link functionality with modern QR code convenience for real-world usability.