URL SHORTNER

DigiSuraksha

Parhari Foundation

Exploration and Proof of concept (POC)

Prathamesh Arun Kamble

04.08.2025

Intern Id-195

🔍 Tool : URL Shortner

📌 Objective:

The objective of this assignment is to develop a basic web-based URL shortener, similar to services like Bitly or TinyURL. The application should:

Accept a long URL as input

Generate a short, unique slug

Store the mapping between original and shortened URL

Redirect users from the short URL to the original link

🔧 Tools & Technologies

Tool	Purpose
Python 3	Core programming language
Flask	Web framework (REST + HTML)
HTML/CSS	Frontend form
GitHub	Version control and submission

🗂️ Project Structure

```
URL Shortner/
└── templates/
    └── index.html.      # HTML form
├── app.py               # Flask backend
```

🔄 How It Works

1. User accesses the homepage: http://localhost:5000

2. Submits a long URL via the HTML form

3. Backend:

Generates a random 6-character alphanumeric short code

Stores the original URL and short code in the SQLite database

4. User receives a short link like: http://localhost:5000/XyZ123

5. When accessed, this short link redirects to the original long URL

6. If the code is invalid, a 404 error is returned

✅ Step-by-Step Execution

# Step 1: Go to project folder

URL Shortner

# Step 2: Install Flask

pip install flask

# Step 3: Run the app

python app.py

Open your browser at: http://localhost:5000

Test the form by entering a valid long URL (e.g., https://www.wikipedia.org)

🔍 Tested Use Case

Action Output

Entered URL:

https://github.com/Prathameshkamble195350/Cybersecurity-2-month-Internship-Progra

m-2025	
Generated short URL:	http://127.0.0.1:5000/fD3Zpw
Redirection (on short URL visit):	GitHub page
Invalid short code tested:	Shows “Invalid short URL” (404 error)

🔒 Security Considerations (Future Scope)

This is a basic prototype. For production deployment:

Input validation and sanitization

Slug collision checking

URL reachability check

Access logging and rate-limiting

HTTPS and secure headers

URL expiry and analytics

📌 Outcome

This proof of concept demonstrates the complete working of a URL shortener web tool with:

Functional form input

Random slug generation

Database mapping

Redirection from short to long URL