# Swayam Salunke (225)

# PoC: URL Shortener

## Objective

Design a basic web-based URL shortener that takes a long URL and returns a shortened version, similar to services like bit.ly or tinyurl.

## Tools & Technologies Used

- Python 3  
- Flask (Web framework)  
- Dictionary (in-memory data storage)  
- Random string generator

## How It Works

1. User inputs a long URL using a form (via Flask web page).  
2. The backend generates a unique short code (6-character).  
3. The long URL is stored in memory using a dictionary with the short code as the key.  
4. When someone accesses the short URL, they are redirected to the original long URL.

## Code Implementation

from flask import Flask, request, redirect, render\_template\_string  
import string, random  
  
app = Flask(\_\_name\_\_)  
url\_mapping = {}  
  
def generate\_short\_code(length=6):  
 return ''.join(random.choices(string.ascii\_letters + string.digits, k=length))  
  
@app.route('/', methods=['GET', 'POST'])  
def home():  
 if request.method == 'POST':  
 long\_url = request.form['long\_url']  
 short\_code = generate\_short\_code()  
 url\_mapping[short\_code] = long\_url  
 return f"Shortened URL: http://localhost:5000/{short\_code}"  
 return '''  
 <form method="POST">  
 Long URL: <input name="long\_url">  
 <input type="submit">  
 </form>  
 '''  
  
@app.route('/<short\_code>')  
def redirect\_to\_long\_url(short\_code):  
 long\_url = url\_mapping.get(short\_code)  
 if long\_url:  
 return redirect(long\_url)  
 return "URL not found", 404  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

## Sample Input & Output

Input:  
User enters: https://www.example.com/articles/how-to-learn-python  
  
Output:  
Shortened URL displayed: http://localhost:5000/Xy3fT9  
  
Accessing http://localhost:5000/Xy3fT9 will redirect to the original long URL.

## Future Improvements

- Store URLs in a persistent database (e.g., SQLite).  
- Add user authentication and URL history.  
- Track URL usage statistics (clicks).  
- Use base62 encoding of an auto-incrementing ID instead of random codes.