PoC: URL Shortener

Ghanshyam Singh(255)

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.

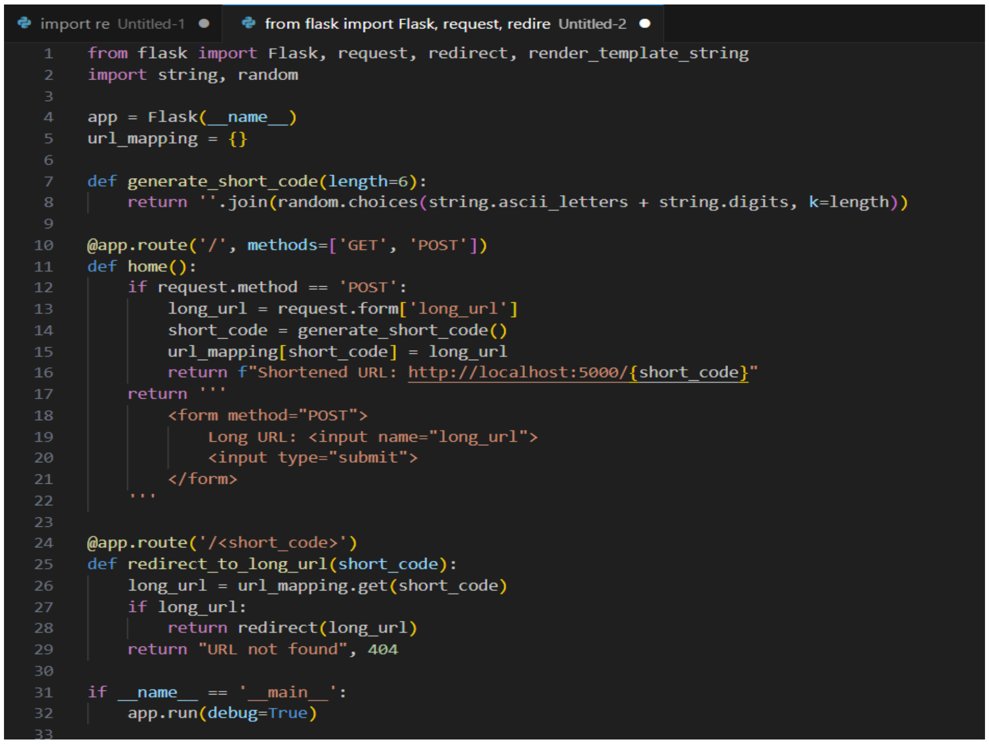
Tools & Technologies Used

- 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



```python
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.