# 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.

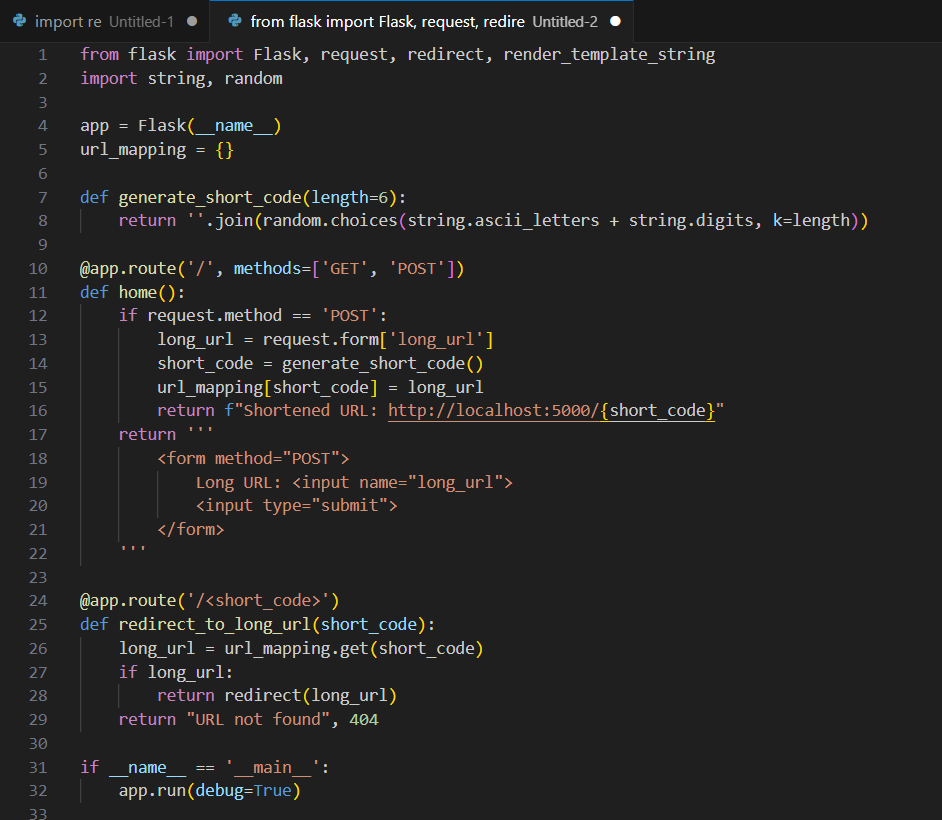
## Tools & Technologies Used

- 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



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.