E:\url.py

```python
import string
import random

class URLShortener:
    def __init__(self):
        self.url_mapping = {}
        self.base_url = "https://example.com/"  # Replace with your base URL

    def generate_short_code(self):
        """Generate a random 6-character short code."""
        characters = string.ascii_letters + string.digits
        short_code = ''.join(random.choice(characters) for _ in range(6))
        return short_code

    def shorten_url(self, original_url):
        """Shorten a given URL and return the short URL."""
        if original_url in self.url_mapping:
            return self.base_url + self.url_mapping[original_url]

        short_code = self.generate_short_code()
        self.url_mapping[original_url] = short_code
        return self.base_url + short_code

    def expand_url(self, short_url):
        """Expand a short URL back to the original URL."""
        short_code = short_url.split("/")[-1]
        for original_url, code in self.url_mapping.items():
            if code == short_code:
                return original_url

        return None  # If short URL is not found in mapping

# Example usage:
if __name__ == "__main__":
    shortener = URLShortener()

    # Shorten URLs
    original_url1 = "https://www.example.com/articles/how-to-build-an-url-shortener"
    short_url1 = shortener.shorten_url(original_url1)
    print(f"Shortened URL 1: {short_url1}")

    original_url2 = "https://www.example.com/products/item1234"
    short_url2 = shortener.shorten_url(original_url2)
    print(f"Shortened URL 2: {short_url2}")

    # Expand shortened URLs
    expanded_url1 = shortener.expand_url(short_url1)
    print(f"Expanded URL 1: {expanded_url1}")

    expanded_url2 = shortener.expand_url(short_url2)
    print(f"Expanded URL 2: {expanded_url2}")

```