URL Shortner

Name : Khushi Patwa

Intern Id : 234

1. Objective

The purpose of this PoC is to design and implement a basic URL shortener system that:

- Converts long URLs into short, shareable links.
- Redirects short links to their original URLs.
- Demonstrates core functionality with minimal setup and code.
- Serves as a foundation for future enhancements like analytics, authentication, and persistence.
-

2. Technology Stack

Component	Technology
Programming Language	Python
Framework	Flask (Lightweight Web API)
Storage	In-memory Dictionary
Hashing	hashlib (MD5 or SHA256)
Runtime	Python 3.x
Tools (optional)	Postman, curl, Browser

3. Features

- Generate short URLs from long URLs
- Redirect to original URLs using short links
- In-memory storage for quick testing
- Minimal API with clean JSON input/output
- Easy to extend and customize

4. How It Works

1. **User sends POST request** with a long URL to the /shorten endpoint.
2. The service hashes the long URL (MD5) and creates a **6-character short code**.
3. The mapping of short_code → long_url is stored in memory.
4. A short URL (e.g., http://localhost:5000/abc123) is returned.
5. When someone visits the short URL, it **redirects** to the original URL using the /abc123 endpoint.

5. Limitations

Limitation	**Description**
No persistent storage	Data is lost when the app restarts
No custom short code support	All short codes are hash-based
No authentication	Open API without access control
Limited collision handling	Possible rare collisions with hash method
Not scalable	In-memory store not suitable for production

6. Example Output

POST /shorten

Request:

json

CopyEdit

```
{
  "long_url": "https://www.example.com/very/long/path"
}
```

Response:

json

CopyEdit

```
{
  "short_url": "http://localhost:5000/9a8f1c"
}
```

7. Instructions to Run

Prerequisites

- Python 3.x installed
- Flask module (pip install flask)

Run the App

bash

CopyEdit

```
python app.py
```

Flask will run on:
http://localhost:5000

Test Endpoints

Shorten a URL:

bash

CopyEdit

```
curl -X POST http://localhost:5000/shorten \
-H "Content-Type: application/json" \
-d '{"long_url": "https://www.example.com"}'
```

Open Short URL in Browser:

Go to http://localhost:5000/<short_code>

8. what poc inteneds to achieve

The PoC is **not** a full production system. It is a **prototype** that proves the idea **can work**, and it provides a **starting point** for building a scalable, secure, and feature-rich URL shortening service.