DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
Discover. Learn. Empower.

Experiment No:1

Student Name: Sumanyu Singh
UID:23BCS10801
Branch: BE CSE
Section/Group:KRG-3B
Semester: 6th
Date of Performance:07/01/2026
Subject Name: System Design
SubjectCode:23CSH-314

Aim: To design a scalable, highly available **URL Shortener system** that converts long URLs into unique short URLs and efficiently redirects users to the original URLs, while meeting low-latency, high-availability, and scalability requirements.

Objective:

- To analyze and define the **functional and non-functional requirements** of a URL Shortener system.
- To design **RESTful APIs** for URL creation and redirection.
- To ensure **unique short URL generation** with minimal collisions.
- To achieve **low latency** and **high availability** using scalable architecture.
- To design an efficient **database schema** for storing URL mappings.

Procedure/Algorithm/Pseudocode:

1. Accept the long URL (optional custom URL and expiration date) from the user.
2. Generate a unique short URL using a counter/Base62 encoding and validate uniqueness.
3. Store the short URL–long URL mapping with expiration details in the database/cache.
4. On accessing the short URL, retrieve the long URL and redirect the user, else return an error.

Functional Requirement :

1. The system shall generate a short URL from a given long URL.
2. The system shall allow users to create a custom short URL (optional).
3. The system shall support URL expiration (default and user-defined).
4. The system shall redirect users from the short URL to the original long URL.
5. The system shall ensure uniqueness of every generated short URL.
6. The system shall handle URL creation and redirection requests via REST APIs.

Non - Functional Requirements :

1. The system shall provide high availability (24×7 access) for URL creation and redirection.
2. The system shall ensure low latency (≈200 ms) for both shortening and redirect operations.
3. The system shall be scalable to support millions of users and billions of URLs.
4. The system shall maintain data consistency with eventual consistency guarantees.
5. The system shall be fault tolerant, avoiding single points of failure.
6. The system shall ensure security and reliability of stored URL data.

Learning outcomes:

1. Understand the system design principles of a URL Shortener.
2. Learn to define functional and non-functional requirements.
3. Gain knowledge of REST API design and request handling.
4. Understand URL shortening techniques and collision handling.
5. Learn scalability, caching, and load balancing concepts.

HLD of URL Shortener (Final Reliable System Design):