Experiment No: 01

Student Name: Amol Saurav
UID: 23BCS10067
Branch: BE-CSE
Section/Group: KRG_2B
Semester: 6th
Date of Performance: 07/01/26
Subject Code: 23CSH-314
Subject Name: System Design

Aim: To design, implement, and evaluate a simple URL redirection system that converts long URLs into short URLs.

Objective:

1. To understand the basic working of a URL shortening service.
2. To implement a functional URL redirection mechanism.
3. To design the HLD and LLD using **draw.io**.
4. To analyze the impact of design choices on system performance.

Procedure:

1. Study the working principles of a URL shortening service and identify its core components.
2. Design the High-Level Design (HLD) representing system architecture and request flow using draw.io.
3. Create the Low-Level Design (LLD) showing database schema, APIs, and component interactions using draw.io.
4. Implement the URL shortening logic and store the short-to-long URL mappings in the database.
5. Implement the URL redirection mechanism to redirect users from short URLs to original URLs.
6. Simulate multiple user requests to generate load on the system.
7. Measure system latency and throughput under different load conditions.
8. Analyze the collected performance metrics and relate them to the system design decisions.

Functional Requirements:

1. Accept long URLs from users.
2. Generate a unique short URL.
3. Store short–long URL mappings.
4. Redirect short URLs to original URLs.
5. Track access count for short URLs.
6. Display if the URL already exists

Non Functional Requirements:

1. Low latency(less than 20ms).
2. High concurrent request handling.
3. Horizontal scalability.
4. High availability.
5. Data consistency and reliability.
6. Secure URL handling.

Outcome:

1. The URL shortening system was successfully designed and implemented based on the defined HLD and LLD.
2. The system correctly generated unique shortened URLs and performed accurate redirection to the original URLs.
3. The integration of an in-memory cache such as Redis/Memcached significantly reduced database lookups for frequently accessed URLs.
4. Cache usage resulted in lower redirection latency and improved throughput under high read traffic.
5. Performance testing under simulated user loads showed stable response times at low and moderate traffic levels.
6. Throughput scaled effectively with increased load until hardware or resource limits were reached.
7. The system handled concurrent requests reliably while maintaining consistency and availability.

High Level Design:-