Internship Project Report: URL Shortener Service

Submitted By: Farman Davda

GitHub Repository: https://github.com/FarmanDavda/URLSORTNER

Internship Organization: [Your Internship Company Name]

Duration: [Insert Internship Duration]

# 1. Project Title

URL Shortener Service using Spring Boot, H2 Database, and Swagger UI

# 2. Introduction

The URL Shortener Service is a web-based application developed using Spring Boot, designed to convert long URLs into compact, easily shareable links. It provides a backend API system that performs URL shortening, redirection to the original URL, and tracking click counts for analytics purposes. The application is lightweight and utilizes an in-memory H2 database for simplicity and testing.

# 3. Objective

* - Learn backend development with Java Spring Boot.
* - Build and deploy a RESTful API-based microservice.
* - Implement real-world concepts like redirection and analytics tracking.
* - Document and test APIs using Swagger UI and Postman.

# 4. Abstract

This project tackles the common problem of lengthy and unattractive URLs. The service generates a unique short string (Base62 encoding) corresponding to each long URL. When accessed, the service redirects users to the original URL and updates a click count for basic analytics. All data is stored temporarily in an in-memory H2 database, and the service APIs are well documented using Swagger.

# 5. Technology Stack

Language: Java 17

Framework: Spring Boot

ORM: Spring Data JPA

Database: H2 (In-Memory)

API Testing: Postman

API Documentation: Swagger UI

Version Control: Git & GitHub

Build Tool: Maven

# 6. System Features

* - Shorten any valid long URL
* - Redirect short URL to original URL
* - Track number of visits per short link
* - Store data in H2 memory database
* - Test endpoints with Swagger UI & Postman

# 7. Project Modules

1. URL Entity: Stores original URL, short code, and click count.

2. Repository Layer: Uses Spring Data JPA to interact with the H2 database.

3. Service Layer: Generates unique short URLs (Base62 encoding), increments click count, and processes URLs.

4. Controller Layer: Provides REST endpoints for shortening, redirecting, and viewing stats.

5. Swagger Configuration: Enables live documentation and testing of APIs.

# 8. Application Flow

1. User submits a long URL.

2. Backend generates a Base62 encoded short key.

3. URL & key are saved in H2 database.

4. When the short URL is hit:

- The original URL is retrieved.

- User is redirected.

- Click count is incremented.

# 9. Testing & Validation

* - Postman used to validate REST endpoints.
* - Swagger UI integrated for live API interaction.
* - H2 Console enabled for in-memory database inspection.

# 10. Conclusion

This URL Shortener project demonstrates practical use of Spring Boot, REST API design, and lightweight database integration. It reflects core backend development skills and the ability to deliver real-world utility-based solutions. The project is modular, scalable, and ready for extension (e.g., user login, custom aliases, expiry time).