Name: ABHIJEET SUNIL MANKANI

Roll No. 22065004

Branch: Civil Engineering

URL SHORTENER SERVICE

Introduction

This report outlines the design and implementation of a URL shortener service. The service allows users to submit long URLs and receive shortened versions. The project utilizes a database for storing URL mappings, Node.js for backend development, and Next.js for frontend development.

Technical Overview

Database

We have selected Firebase Firestore as our database. Firestore provides a NoSQL document-based database that efficiently stores URL mappings. Its real-time data synchronization capabilities are beneficial for updating URL mappings.

Backend Development

The backend is built using Express.js, a lightweight and flexible Node.js framework. Express.js facilitates the creation of RESTful APIs for URL shortening and redirection.

Frontend Development

The frontend is developed using Next.js, a React-based framework that enables server-side rendering and static site generation. This choice ensures a responsive and efficient user interface for submitting URLs.

System Flow

URL Submission:

Users submit long URLs through the Next.js frontend.

The frontend sends a request to the Express.js backend to shorten the URL.

URL Shortening:

The backend generates a unique shortened URL using a hash function.

The mapping between the original and shortened URLs is stored in Firebase Firestore.

URL Redirection:

When a user clicks on a shortened URL, the request is sent to the backend.

The backend retrieves the original URL from Firestore based on the shortened URL and redirects the user.

APIs

URL Creation API: This API endpoint accepts a long URL and returns a shortened version.

URL Redirection API: This API handles requests for shortened URLs and redirects users to the original URLs.

Optimization Strategies

Performance: Use caching mechanisms to reduce database queries and improve response times.

Scalability: Implement efficient data retrieval and storage practices to handle increased traffic.

Error Handling: Implement robust error handling using try-catch blocks and error logging to ensure system reliability.

Conclusion

This URL shortener service effectively integrates Firebase Firestore, Express.js, and Next.js to provide a robust solution. The system's design ensures efficient URL shortening and redirection, making it suitable for a wide range of applications. Future enhancements could include additional features to further enhance user experience and system reliability.