Software Requirements Specification (SRS)

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements of the URL Shortener Web App. This application allows users to generate short links from long URLs, set optional expiry and custom shortcodes, and view analytics.

1.2 Scope

This system consists of:

- A frontend web application built with React and Material UI.

- A backend microservice built with Node.js and Express.

- Persistent storage using MongoDB.

- Logging through a custom middleware.

1.3 Intended Audience

- Developers

- Testers

- Evaluators and Hiring Panel

2. Overall Description

2.1 Product Perspective

The product is a standalone web-based application that interfaces with a backend microservice via RESTful APIs.

2.2 User Classes

- General user: Can shorten up to 5 URLs, view analytics, delete entries locally.

2.3 Design/Theme

- Header: #004030

- Accent: #4A9782

- Background: #FFF9E5

Software Requirements Specification (SRS)

- Cards: #DCD0A8

3. Functional Requirements

3.1 Frontend URL Shortening

- Accepts up to 5 long URLs with optional validity and custom shortcode.

3.2 Backend URL Shortening

- POST /shorturls: Validates input, stores expiry and shortcode.

3.3 Redirection

- GET /:shortcode redirects to long URL if not expired.

3.4 Statistics Page

- GET /shorturls/:code returns original URL, expiry, click count and metadata.

3.5 Local State

- Saved in localStorage. Each entry deletable.

4. Non-Functional Requirements

4.1 UI/UX

- Material UI only. Consistent theme colors.

4.2 Logging

- No console.log. Custom logging middleware.

4.3 Platform

- Frontend: http://localhost:3000

- Backend: http://localhost:5000

5. System Interface

/shorturls [POST] - Shorten URL

/shorturls/:shortcode [GET] - Get stats

/:shortcode [GET] - Redirect

6. Validation Rules

Software Requirements Specification (SRS)

URL: Valid URL

Validity: Integer > 0

Shortcode: Alphanumeric, 4?10 chars

Expiry: Defaults to 30 min

7. Limitations

- No authentication

- History is localStorage-based

- Location via IP (coarse)

8. Future Enhancements

- User login

- QR code

- Public dashboards

- Export to CSV