

Design Document - React URL Shortener Web App

1. Architecture

The application is a React-based frontend project that implements a URL Shortener with analytics. It uses React Router for navigation, Material UI for styling, and localStorage for client-side persistence. Logging is implemented through a custom middleware function instead of console.log.

2. Data Model

Each shortened URL is stored in localStorage in the following structure:

```
{
  "longUrl": "string",
  "shortCode": "string",
  "expiry": "number (timestamp)",
  "clicks": [
    { "timestamp": "string", "source": "string", "location": "string" }
  ]
}
```

3. Key Decisions

- 1 React was chosen for component-based development and scalability.
- 2 Material UI was used for consistent, modern styling.
- 3 React Router was used for navigation between pages.
- 4 localStorage was used as a mock backend for persistence due to the test constraints.
- 5 A logger middleware was implemented for controlled logging instead of using console.log directly.

4. Routing Strategy

- 1 / - ShortenerPage: Users can shorten URLs, specify validity, and custom codes.
- 2 /stats - StatsPage: Displays analytics of all shortened URLs including click data.

5. Error Handling

The application validates URLs before shortening. Invalid URLs (not starting with http/https) are rejected with an alert. Expiry defaults to 30 minutes if not specified. Custom shortcodes are validated to avoid duplicates.

6. Assumptions

- 1 All API requests are assumed to be pre-authorized (no login required).
- 2 localStorage is used for persistence since a backend service is not required in this evaluation.
- 3 The app runs only on http://localhost:3000 as per instructions.