**URL Shortener**

**Problem**

The main problem is the long number of characters in the sharing links. It’s make links hard to embedded in email ,slack or privat message.

**Goal**

Our goal is to limit the number of characters in the URL by shortening it and making them much easier to share.

**Solution**

The solution is to build a service, composed of two parts back-end and front-end.

Front-end provides a simple interface connecting users with the backend making URL generating much easier for a user.

Backe-end is responsible for generating short URLs and storing original URLs in a database. The second feature is to redirect users who clicked on the share link to the original route.  
URL generating algorithm creates 10 characters long URL id using 64 characters, so hypothetically speaking if we generate 1000 urls for every hour probability of at least one collision is ~17 years.

**Consumer**

We are building application for a company employees with basic computer skills. It means we have to provide a friendly interface with simple errors communication.

**Value**

The value of the application is storing and making connections between long URLs and short ones. But after all the main value is redirecting users to the original routes.

**High availability**

As the application provides access to important reports, we must implement the HA design. Unavailability of the application can cause financial losses to the company if employees lose access to important files. We are not implementing it in MVP.

**MVP Technologies**

For a backend we chose Express.js, LowDB and Nanoid.

For a frontend we used a React.js, styled-components,

**App extension**

After testing MVP good idea is to use a cloud database, and implement HA with Kubernetes + Nginx ingress.

In the future, we can add topics to links to make them more readable for a humans. Links would looks like this ‘/financial/URL\_id’