Design:

Back-End - Flask

Endpoints:

GET	**/**	-	main page (index.html)
POST	**/add_url**	-	add long url to make it short
GET	**/<short_url>**	-	redirect from the short url to the long one
GET	**/stats**	-	get statistics about redirects count, url creations and errors.

Front-End – Vue.js

Database Model:

we have 3 tables.

1. URL – the main one, store urls and map between long to short url.

2. Redirect – log table for redirects (short to long url)

3. Error – log table for http errors (its better to use logger, but according to the task requirements we need to provide statistics about errors)

Short url technique:

I thought about 2 methods:

1. Generate random 6 digits hash from the alphabet (a-z,A-Z,0-9).
   the idea is good but the query to find Url by its short hash, is slower than the second method.
   we will have 64^6 addresses, it should be ok.

2. save Url object in the DB, than, take its ID and encode it to Base 62(because we have 62 characters.
   this encoded string will be the short url.
   when we get redirect request we will decode the url and it will be the ID in the DB, this is why it should be faster.

finally I chose the first one, due to time constrains and small problem I had

with the second method (padding small numbers with dummy characters)