

## Interview Task - URL Shortener

Write a simple URL shortener service. The site should allow a user to enter a valid URL and give back a shortened version using a human readable word or words as a substitute for their URL. For example, running on localhost, this might shorten <https://www.activehousing.co.uk/products-services> to <http://localhost/aardvark>.

### Specification

- The site should allow duplicate URLs to be shortened, but should not allow duplicate shortened versions to be generated, i.e. <https://www.activehousing.co.uk/products-services> could be shortened many times, but the resulting short URL must always be unique. How you handle collisions when generating the shortened form is at your discretion, but we would be interested to hear your rationale for your given approach.
- A user should optionally be able to enter a short (140 character) description of their URL.
- When a shortened URL is visited, a counter for that URL should be incremented to indicate how many times the shortened link has been used.
- The main page for the site should show a listing of the 10 most recent shortened URLs, along with their descriptions.

### Design / Branding

Download the following assets to use in the project:

- [Mockup of the system layout.](#)
- [Document detailing the required colour scheme.](#)
- [A logo for the system.](#)

### Constraints

- The word list used to shorten urls **MUST** be from the EFF's short wordlist ([https://www.eff.org/files/2016/09/08/eff\\_short\\_wordlist\\_2\\_0.txt](https://www.eff.org/files/2016/09/08/eff_short_wordlist_2_0.txt)).
- Frameworks (e.g. Laravel, Symfony) **SHOULD** be used; our preference is the Laravel framework, but we will accept submissions in other suitably modern frameworks.
- Libraries, be they server side or client side, **MAY** be used so long as they do not effectively complete the entire task by themselves.

### Response

Send responses either as compressed archived (zip, tar.gz etc) or via a GitHub or Bitbucket link to [recruitment@hallnet.co.uk](mailto:recruitment@hallnet.co.uk)

### Marking Criteria

The main way that we grade candidates is by looking at the code you send us. Specifically we will assess the following:

- How well you've used the framework / CMS that you have chosen
- Demonstrating a good understand of the challenges presented by the problem
- Handling of errors and edge cases
- The efficiency of the code that you have written
- How well written and tested the code is