INFORCE

INFORCE .NET TASK

TECHNICAL STACK

- ASP.NET MVC (Framework or Core)
- Angular (Latest version), React
- EntityFramework CodeFirst approach - Any Framework for Unit tests **ACTUAL**

TASK

We need to build URL Shortener. The goals are to create an application that will shorten any URL and to have possibility to navigate by this short equivalent.
There are such views: Login view, Short URLs Table view, Short URL Info view, About view. Adding some Unit tests will be a huge plus.

LOGIN VIEW

On the Login View you should be able to enter Login, Password and Authorize yourself.
You need to have Admin and ordinary users.

SHORT URLS TABLE VIEW

This page has table with all URLs and their equivalent after shortening, deleting of the records is possible(only authorized users). You can view details about these URLs by navigating to Short URL Info view with correct Id.
Also it has "Add new Url" section, which is visible only to authorized users, where you can enter URL and convert it to a short representation, after that it should be added to the table automatically.

If such a URL already exists - error message should be shown.

Every authorized user can add new records("Add new Url" section) and view(redirects to the Short URL Info view), delete records created by themselves (URLs should be unique). Admin users can add new records("Add new Url" section) and view(redirects to the Short URL Info view), delete all existing records. Anonymous users can only see this table.

All changes should be visible right after they are done without reloading the page.
This page should be implemented with Angular.

Short URL Info (Anonymous can't access this page)

This page contains info about URL (CreatedBy, CreatedDate, any other fields).

About view

This should contain a description of your Url Shortener algorithm. Visible for everyone(even not authorize) but can be edited only by admin users. Just an ordinary Razor page with submit action.

Guidelines

Only spend up to two days working on this task.
If we have a technical interview, and I hope we do, we will focus on enhancing this application and discussing how you worked through some of these problems. It's important that we see your best work, if that means that you do not satisfy all of the requirements here. That is okay, we don't expect everyone to finish all parts at time.
If you have to choose between refactoring and making one piece of this perfect and implementing the next feature, choose refactoring because we want to see how your best work looks.
We want to see clear, correct code that uses the suitable data structures and patterns, and we want to see your style.