# TEST ASSIGNMENT FOR CANDIDATES FOR A FULL-STACK DEVELOPER POSITION

Test assignment consists of 3 parts: Front end task, data structures and a backend task.

## Frontend Task

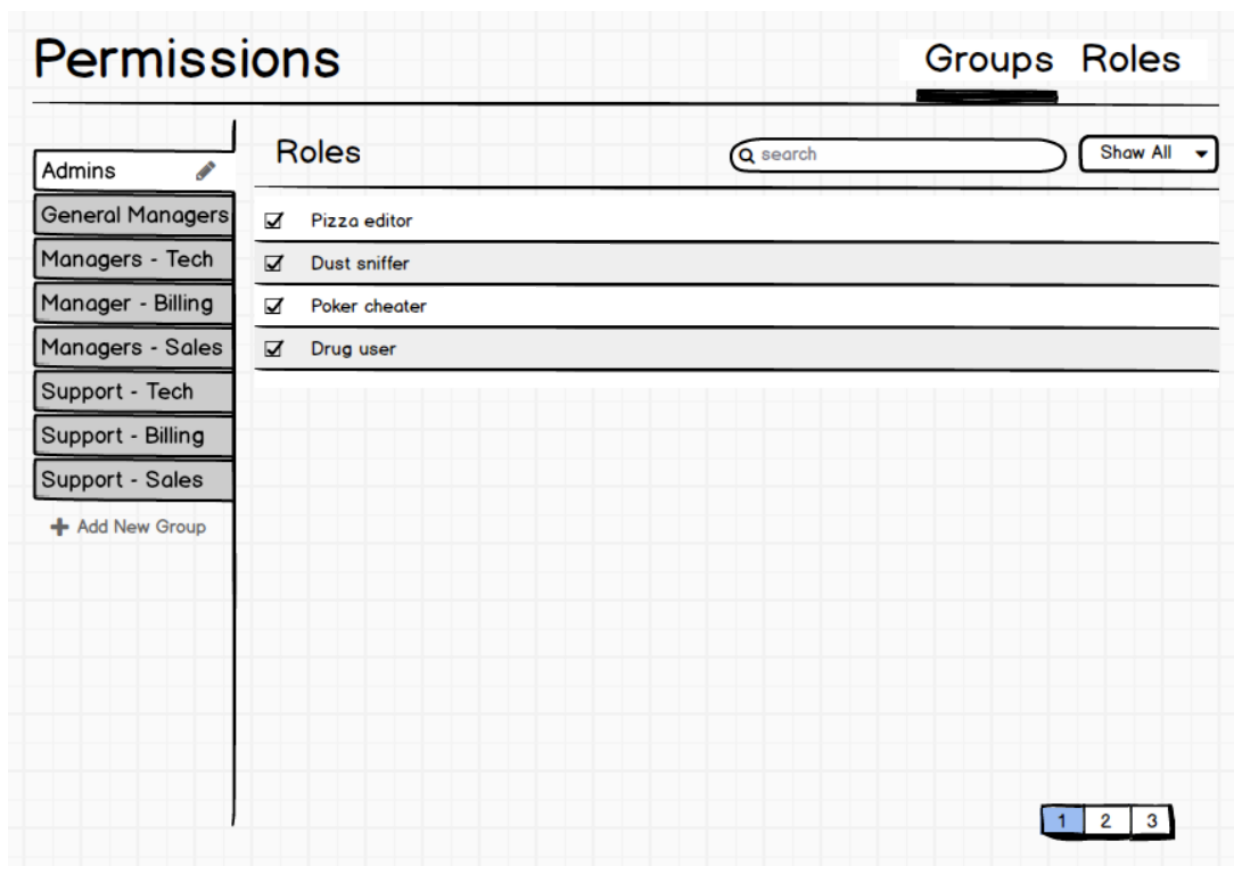
Need to create an application to edit the list of Groups and Roles. Roles are combined into groups.

**Group list:**

* you can create a new group;
* you can edit the name of existing groups;

**Form to edit the list of roles included in the group:**

* list of all roles;
* ability to mark the roles required for the group;
* filter:
  + display all roles, marked roles, filter by label;
  + list search;
  + page navigation (pagination).

The mockup shows an example of editing access groups. The interface for editing roles looks similar, only the left part contains a list of roles and the right part contains a list of access roles.   
  


**Recommendations**

It is not necessary to implement the server part, it is enough to start-up the webpack dev server.   
You can store the data (access rights) can be stored in a JSON-file.

**Technologies**

Webpack (v2/3) - for building. The project must have a file describing how to start the project

(readme.md).

Angular (v2/4) - for developing client-side logic. You can also use components.

**UX**

* Responsive (up to and including 320px in width).

**CSS**

* Do not use CSS framework;
* use BEM methodology for naming classes;
* use preprocessor (PostCSS);
* support latest versions of browsers;

## Data structures

* Create a relational database model on the topic "Selling railroad tickets";
* How detailed the description of the subject area will be, you decide
* It is desirable to use MSSQL Server
* The result of the task can be a dump of the created database (\*.bak file), \*.dacpac schema description or SQL script that creates the database

## Back-end Task (.Net app)

* Write a "proxy" that will return the modified text of reddit.com (or any other site of your choice);
* To each word, which consists of six letters, you must add a symbol "™";
* You can use any .Net technology and libraries in the C# language;
* The functionality of the original site must not be altered;
* All internal navigation links of the site must be replaced by the address of the proxy-server.

That is, site navigation must be handled by a Proxy without taking the client back to the original site.

Example. A request to, say, {proxy address}/r/popular/ should show the content of the page

https://www.reddit.com/r/popular/ with changed words that were 6 characters long.

And all the site navigation to sections of the site should go through Proxy.

Hint: There is no SLOC limit for your solution. But, if your proxy solution contains more than 300 SLOC, then probably you do something wrong.

**Publishing the test task**

The project must be published in a **private** repository on github.com or bitbucket.org.