



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.

1. 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.

The screenshot shows a web-based application for managing permissions. On the left, there's a sidebar with a list of groups: Admins, General Managers, Managers - Tech, Manager - Billing, Managers - Sales, Support - Tech, Support - Billing, and Support - Sales. Below this is a button for "Add New Group". The main area is titled "Roles" and contains a list of roles with checkboxes: Pizza editor (checked), Dust sniffer (checked), Poker cheater (checked), and Drug user (checked). At the top right, there's a search bar with the placeholder "search" and a "Show All" dropdown. A navigation bar at the very top has tabs for "Groups" and "Roles", with "Roles" being the active tab. At the bottom right, there are three small numbered boxes: 1, 2, and 3.

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;



2. 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

3. Back-end Task (.Net app)

- Write a "proxy" that will return the modified text of habr.com;
- 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}/info/about/ should show the content of the page

<https://habr.com/info/about/> with changed words that were 6 characters long.

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

Publishing the test task

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