Frontend task for NinjaRMM

Introduction:

Implement a client web application to work with devices. One device has the following properties:

- Id
- System Name
- Type
- HDD Capacity

There are only 3 possible types for devices on this app:

- Windows Workstation
- Windows Server
- Mac

Feature Requirements:

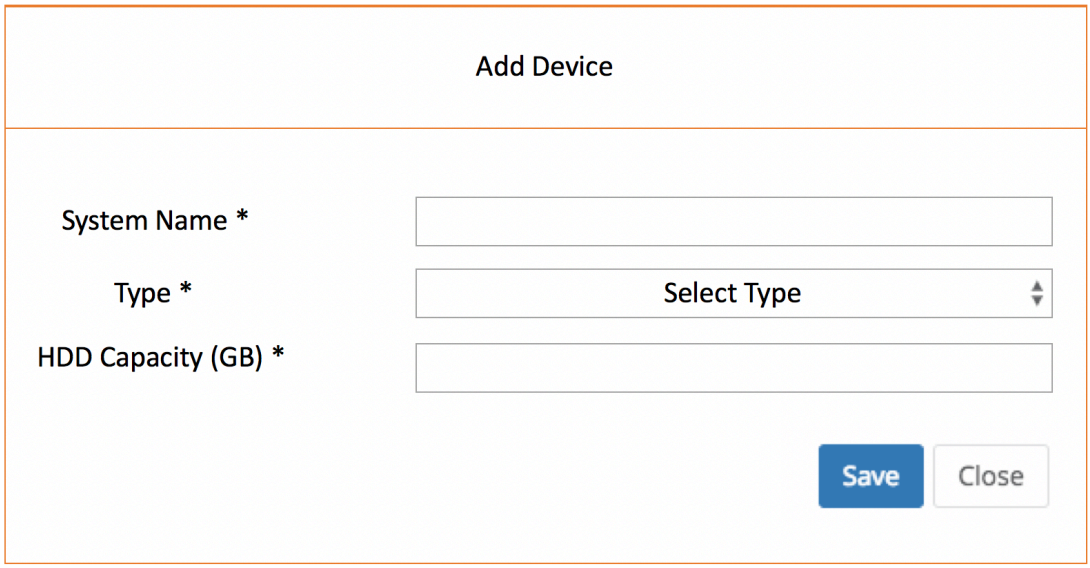
1- The main view is a dashboard where user can see all devices listed.

2- User can filter devices by one Type.
For example, if user select “Window Server” from Device Type filter, the only device displayed will be the third one.

3- User can sort devices by system name.
If user select “System Name” from Sort By, the devices should be sorted in alphabetical order according to system name.

4- User can sort devices by HDD Capacity from low to high.
There is an example of devices sorted by hard drive capacity in the image above.

5- User can Add, Update or Delete devices.
A second view or modal allows to add new devices or update an existent one.



- Add new device.
  - Create one button in dashboard view in order to go to the Add Device Form.
  - All three properties are required (System Name, Type and HDD Capacity).
  - HDD Capacity should only accept a number for gigabytes.
- Update device.
  - Create one button at the right side of every device in dashboard in order to edit the device properties. This button should open the Edit Device Form.
  - Apply same requirements for fields as Add Device.
- Delete device.
  - Create one button at the right side of every device in dashboard in order to delete it.

Important notes:

- You are not forced to use the design showed in the images above. You can create your own. We are also evaluating good css skills.
- Implement unit testing and/or integration testing may have you extra points if you do it well. But it is not a requirement.
- If your device type filter allows many device types, you will receive extra points. But just one is required.
- Good documentation in your readme file may give you extra points.
- You will be evaluated for the functionalities requested in this document. Other initiatives won't be taken into consideration to improves your final score.

Project Requirement and Tools:

- You should create your project from scratch and save it inside a folder named “devices-clientapp”.
- The available framework to complete this task is ReactJS.
- Feel free to use any dependencies that you need to build your app.
- You should use git as source version control and create a repository in your github account.
- You have to send an email with a link to this repository.
- Please provide documentation in your readme file explaining how to run your project.

Server Application

A backend application will be provided. You can perform some requests to it in order to do all needed operations with devices. You do not need to make any changes on serverapp.
This app does not have data persistency implemented. So, any change in the data (creation, update or delete) will be lost if you kill the server. A new start will have the data in the initial state.

These are examples for the requests available:

```
### GET devices
GET http://localhost:3000/devices

### POST device
POST http://localhost:3000/devices
Content-Type: application/json

{
  "system_name": "my-mac",
  "type": "MAC",
  "hdd_capacity": "64"
}

### GET device
GET http://localhost:3000/devices/e7ocoQ2n3

### PUT device
PUT http://localhost:3000/devices/e7ocoQ2n3
Content-Type: application/json

{
  "system_name": "my-win-server",
  "type": "WINDOWS_SERVER",
  "hdd_capacity": "500"
}

### DELETE device
DELETE http://localhost:3000/devices/e8okoP2l5
```

Please download or clone this repository in order to use the server app. https://github.com/NinjaMSP/devicesTask_serverApp