Rubric Detail

*A rubric lists marking criteria that instructors use to evaluate student work. Your instructor linked a rubric to this item and made it available to you. Select **Grid View** or **List View** to change the rubric's layout.*

Name: **Assignment 1 Rubric**

Exit

Grid View   List View

	Novice	Competent	Proficient
Submission	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Submission is via a GitHub repository. You must add your tutor to the repository (if private). Documentation should also be in the repository as a README.md file using Markdown.
GitHub	**0** (0%)	**.5** (1.67%)	**1** (3.33%) GitHub history shows frequent commits during the development of the software with appropriate messages. GitHub layout should be logical and clean.
Website Design	**0** (0%) Little effort given to website look and feel.	**.5** (1.67%) Website is logical and functional	**1** (3.33%) Website looks professional

	Novice	Competent	Proficient
User Login	**0** (0%)	**.5** (1.67%)	**1** (3.33%) In assignment 1 there is no user authentication. However the system must check that the user exists and remembers them while they are logged in. Logging out user no longer provides access for user.
Local storage	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Local storage is used to remember the logged in username. Logging out should clear the username from local storage.
User Administration	**0** (0%)	**2.5** (8.33%)	**7** (23.33%) Super Admins and Group Admins have the ability to create new users by specifying the username and email address. Once created users can log in. A Super Admin can also remove a user. A user can not be created with a username that already exists.

	Novice	Competent	Proficient
Group Administration	**0** (0%)	**2.5** (8.33%)	**8** (26.67%) Group Admins can create new groups and add existing users to groups. Channels can also be added to a group. Users can not see the channels within a group without also being explicitly added to a channel. Groups can be removed and users can be removed from groups and channels.
JSON Serialisation	**0** (0%)	**1** (3.33%)	**2** (6.67%) Data is converted to JSON and stored in a text file regularly (preferably after every change to the data). Data is loaded from the JSON file when the server starts.
Documentation - Git	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Describe the layout of your git repository and the approach you took for version control.
Documentation - Data Structures	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Describe the main data structures used in the program. For example, how the users and groups are represented.

	Novice	Competent	Proficient
Documentation - REST API	**0** (0%)	**1** (3.33%)	**2** (6.67%) The Angular front end should communicate with the Node.js server using a REST API. Describe each route provided, parameters, return values, and what it does.
Documentation - Angular Architecture	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Describe your Angular architecture in terms of components, services, and models.
Comments and Naming	**0** (0%)	**1** (3.33%)	**2** (6.67%) All functions commented. Logical file, function, and variable names. Consistent naming, indenting, and syntax.
Validation and Error Checking	**0** (0%)	**.5** (1.67%)	**1** (3.33%) Fields are checked for appropriate values. Errors are displayed in a user friendly manner. User can recover from the error in a user friendly manner.

Name:**Assignment 1 Rubric**

Exit