**Description**

**GoodMats**

JavaScript Applications Exam, 4 October 2016

**Assignment**

Create a SPA application for working with learning materials. Each material has **title**, **description**, optional **img**. Registered users can leave comments on materials and add materials to their profile in a categoty **want-to-watch**, **watched** or **watching**

**Users pages/functionality**

* **Users** can **Register**, **Login** and **Logout** in the application
  + Users provide username and password
* **All users** (even not logged in) can
  + **View** other users profile page
  + **View** list of materials, that are created by any user
    - The title, createdOn, description and img are shown
    - Clicking on a title should take the user to the details page of that material
  + **Search** by title, user or description
  + **View** all comments of a material
  + **View** their and other users profile page
  + **View** the details page of the material
    - Shows the title, full description, img, username of the creator of the material and a list with all comments
      * Clicking on the username (who created the material) should open that user's profile page (#/users/:username)
* **Registered and Logged-in users** can:
  + **Have** a profile page on /users/:username
  + **Create** a new material
    - Providing title, description and optionally img
  + **Leave** comments on existing materials
    - Prodiving commentText
  + Assign a **Status** to a material
    - Status can be want-to-watch, watched, watching

**User Profile pages/functionality**

* Shows the current users details by username
* Show want-to-watch, watched, watching and lists materials in that state for the current user

**Materials pages/functionality**

* **View all materials**
  + List all materials from the server
* **Search materials**
  + **View only filtered materials** that contain the **search phrase**
  + Redirects to **Search results page**
* **Details page**:
  + Shows the title, full description, user, img and list with all comments
    - Clicking on the user should open that user's profile page (#/users/:username)
  + Logged-in users can assign a **Status** to a material
    - Status can be want-to-watch, watched or watching

**Application routes**

Implement at least the following routes in your app:

* #/
  + Redirects to #/home
* #/home
  + Shows all materials
  + Available to all users, logged-in or not
* #/home?filter=XXXXXX or #/home/XXXXXX
  + Shows the materials that match the search phrase in the *Search Page*
  + Available to all users, logged-in or not
* #/materials/:id
  + Shows the materials the *Details page* of the material with the provided id, with full description
    - Logged in users only should be able to change the status of the material (want-to-watch, watched, watching)
  + Available to all users, logged-in or not
* #/profiles/:username
  + Shows the profile of user with the username passed in the URL
  + #/profiles/john should show the profile of the user john
  + Available to all users, logged-in or not

**Other routes are also OK. You are not limited to only the routes above**

**Server routes**

See the [SERVER\_ROUTES.md](file:///D:\Desktop\Exam\SERVER_ROUTES.md) file

**Data object**

* Create a data object containing methods for each operation to the server
  + *Examples:*
    - data.getMaterials() to get all materials
    - data.login(username, password) to login users
    - And more...

**Unit tests**

* Write unit tests with Mocha, Chai and Sinon for each method of the data object
  + Test if the methods work correctly with correct data
  + Test if the methods throw/return approriate exceptions/results

**Application requirements**

**Mandatory** provide a good **User Experience(UX)**

* No need to be pretty, just usable
  + Design the application in an intuitive way, so that your colleagues aren't confused how to use and test your application

**Mandatory** use extensively the following libraries:

* jQuery
  + For AJAX
  + For DOM manipulation
* Handlebars.js
  + For view templates

**Do NOT** use any of the following:

* AngularJS
* KendoUI framework
  + Data-binding, routing, ViewModels, etc...
* Durandal
* Backbone.js
* Ember.js
* Batman.js

You can use libraries for:

* Notifications
  + toastr or any other
* UI components
  + jQueryUI, KendoUI UI components, Twitter Bootstrap
* Utils frameworks:
  + Moment, Underscore/Lodash
  + Libs for encryption
  + etc...
* Routers
* Module loading

**Starting the node server**

1. Restore the packages by running npm install in this directory.
2. Run npm start. You should see a message App running at http://localhost:3001/ in the terminal.
3. *Hint*: the server stores the data in db/localDb.json. You can look there :)

**Deliverables - read carefully!**

* **ZIP (no rars)**, all your code, including the server
  + **Remove the node\_modules and bower\_components folder**
  + **Make sure that you have a package.json and a bower.json file in the root directory of your application**
    - If you don't, you can run npm init and/or bower init - this will create the files
      * Don't forget to edit your name out of both files.
  + **If you install packages with bower or npm, do not forget to add --save to the end of the install command**
    - *Example*: bower install jqueryui --save
    - **This is required so your colleagues can restore the packages you installed and run your application!!**
* Send the **ZIP** file on the page, provided at <https://telerikacademy.com>

**Constraints and validation**

* Implement client-side validation in your application. In case a validation fails, notify the user with an appropriate message. **The only validations necessary are described below. Other validations are not required**.
* **User**
  + **username**
    - A string
    - Has length between 6 and 30
    - Can contain only Latin letters, digits and the characters '\_' and '.'
  + **password**
    - A string
    - Has length between 6 and 30
    - Can contain only Latin letters, digits
* **Material**
  + **title**
    - A string
    - Has length between 6 and 100
    - Can contain any characters
  + **description**
    - A string
    - Has no length restrictions
    - Can contain any characters