## CPSC 473 - Web Programming and Data Management

### Homework Assignment 4 - Spring 2016

#### Section 02 - due March 14, Section 01 - due March 16

Use the [class virtual machine](https://github.com/ProfAvery/node-box) to complete the following steps.

1. Download [db.json](https://gist.github.com/ProfAvery/cfad3a42c54fd974b93c) and place it in a new directory for this assignment.
2. Run

python -m [SimpleHTTPServer](https://docs.python.org/2/library/simplehttpserver.html)

to start a static web server on port 8000.

1. Run

[json-server](https://github.com/typicode/json-server) db.json

to start the RESTful web service on port 3000.

1. Using [Material Design Lite lists](https://www.getmdl.io/components/index.html#lists-section) as a starting point, create a web page in the same directory showing a list of actors. (See the section “List items with avatar and action.)
2. Add JavaScript code to load the list of actors from <http://localhost:3000/actors>. If the starred attribute is true, use the icon star. If starred is false, use the icon star\_border.
3. Add a form with a [text field](https://www.getmdl.io/components/index.html#textfields-section) and a [button](https://www.getmdl.io/components/index.html#buttons-section) to the page.
4. When the user clicks the button, POST a JSON object with the name from the text field and “starred”: false to <http://localhost:3000/actors>.
5. When the POST completes successfully, add the actor to the list. For the icon, use star\_border.
6. When the user clicks the star next to a name, toggle the icon text between star and star\_border and do a PUT to [http://localhost:3000/actors/*id*](http://localhost:3000/actors/id) with starred set to true or false.
7. Test your code by reloading the page and making sure that the names and star icons correctly reflect the current contents of db.json.

#### Working with Other Students

You may complete this exercise on your own, but you are encouraged to work together with another student.

If you choose to work with a partner:

* Submit only a single assignment.
* Include both names on the submission.
* Each student in a pair will receive the same grade.
* You may discuss the assignment with other pairs, but each pair must submit its own work.
* You may choose to work with a different partner on future assignments.

#### Submission

E-mail the following to [csuf.kenytt.net@gmail.com](mailto:csuf.kenytt.net@gmail.com) by 11:59p on the date indicated:

1. The URL of your GitHub repository containing all files for the application.

Include your name and your partner’s name (if you have one) in your e-mail.

Set the Subject: line of your e-mail to

[CPSC 473-02] Assignment 4  
or  
 [CPSC 473-01] Assignment 4

as appropriate. Monday night is Section 02; Wednesday night is Section 01.

You may submit multiple times before the deadline; I will only grade the most recent submission before the deadline, unless your e-mail indicates that I should do something else. Late work will not be accepted after the deadline.

#### Grading

This exercise is worth 10 points, one for each step show above. A point may be deducted each time you fail to follow the instructions as given.

#### Notes

* This assignment is particularly challenging, but when you complete it you will be well-prepared for the code in Project 1. Start early, and consider working with another student.
* Your code must pass JSHint with no errors or warnings using [this](https://gist.github.com/ProfAvery/c5db1692c457c526601c) [configuration](https://github.com/ProfAvery/node-box/blob/master/provisioning/files/jshintrc).
* Your code must be properly formatted. If you don’t know how to do this, use [JS Beautifier](http://jsbeautifier.org/).