

**Web Development Course**

~ WEEK 6 DAY 2 ~

Assignment: A jQuery powered App

**Introduction**

**Goal:** Create a mini web application using jQuery

**Repository:** You will be creating your own project from scratch.

**Duration:** 4 - 8 hours.

**Reminders:**

* Projects and unix commands are to be managed and run inside your vagrant box. Remember to *vagrant ssh* into your box
* Work iteratively by creating very small runnable “experiments” that are on the path towards your actual goal or objective
* Commit and push your progress on a regular basis

Select and implement one of the app options below. Creativity is highly encouraged. If you have another idea instead of the 3 options below, we encourage it. But before starting on it, please run it by a teacher.

Focus on the functionality. Do not get caught up on the styling (CSS) of the app prior to implementing the JavaScript & jQuery functionality. Since we want to not focus on writing CSS for now, it makes a lot of sense to use a CSS framework like Bootstrap that gives us a bunch of stuff out of the box.

That said, since aesthetics are an important part of any portfolio, try to set aside some time at the end to stylize your app.

**Option 1: Photo Gallery**

Display a set of image thumbnails and a main image. When a user clicks a thumbnail, have the larger version of the thumbnail (not the thumbnail itself) replace the current main photo.

**Difficulty:** Medium

**Enhancements:**

Once you complete the initial requirements for the app, implement the following additional features, in order. Remember to commit and push after completing each one.

1. **Indicate selected image**When a thumbnail is selected for display, the thumbnail should change slightly, compared to the unselected ones. You can choose what you’d like to change (eg: border colour or opacity or perhaps a CSS3 transform)
2. **Overlay a caption coming from the thumbnail TITLE attribute**  
   Add a title attribute to each thumbnail in your gallery. This will be used as a caption for the main image so that when the main image is changed, it’s corresponding caption is also shown beside the main image.
3. **Low res thumbnail image**  
   Use a thumbnail version of the image for the smaller (clickable) images and a hi-res version for the main display image. This means the src of the thumbnail images will have to be low res versions and the main image’s src cannot just be replaced with the one from the thumbnail.
4. **Play/pause button toggle**Add a play button which when clicked, turns into a pause button. It should put the gallery into slideshow mode, where every few seconds the next image in the set is selected (as though the user clicked it). Clicking it again should deactivate the slideshow. **Bonus:** deactivate/pause the slideshow when the user clicks a thumbnail, since they likely mean to do so.

**Option 2: A Todo List**

Display a todo list (as an OL) with some fake todo items already there. Add a textbox and button that when used add the text entered into the list.

**Difficulty:** Medium-High

**Initial Requirements:**

1. **Clear text**  
   The text should clear from the textbox after you click the “Add” button
2. **Delete items**  
   Add a “Delete” or “X” button alongside each todo item that when clicked remove that item

**Enhancements:**

1. **Checkbox to mark completed**Create a checkbox to the left of todo items. If selected, the text in the todo item is marked with [line-through](http://www.w3schools.com/cssref/pr_text_text-decoration.asp" \t "_blank) (strike-through)
2. **Enter key**  
   Allow the user to submit their todo item by pressing enter when typing in the text area
3. **Editability**   
   Allow the user to edit each item. Add an edit button. Prompt the user for the new value in a modal, or inline. Do not attempt to use the same textbox that is used to add todos for this functionality. Pre-populate the edit textbox with the original text.
4. **Sortability**  
   Allow the user to reorder the todo list using drag and drop functionality. Tip: Leverage the jQuery UI library for this.
5. **Due date**  
   Allow the user to set a due date for each item in the list. only after creating it. Tip: Use a datepicker plugin like the one provided by jQuery UI.

**Option 3: Contact Cards**

Button to create new contact, when clicked presents user with some fields (Name, Number, Email, etc). When submitted, the information is added to a new contact card that is appended to a vertical list of contact cards.

The form to create a card should be within a <form> tag that should include a submit button. After a contact is created, the form fields should be cleared out so that the user can add new contacts without having to clear each field.

The Contact card can include fields such as:

* [A placeholder avatar image](https://www.google.com/search?site=&tbm=isch&source=hp&biw=1395&bih=717&q=avatar+png&btnG=Search+by+image&oq=vancouver+gastown&gs_l=img.3..0l2j0i5l2j0i24l6.154.1699.0.1822.17.13.0.4.4.0.114.927.11j1.12.0....0...1ac.1.26.img..1.16.939.Mzkjii1SdlU" \l "q=placeholder+avatar+png&tbm=isch" \t "_blank)
* First name
* Last name
* Phone number
* Email address
* Company name
* Salutation (input using <select>s aka dropdowns)
* Date of birth (input using 3 <select>s aka dropdowns)

**Difficulty:** High

**Enhancements:**

1. **Expandable contact card**  
   Make each existing contact div collapsible and expandable (animated). The expandable version should include additional, secondary information that the short version does not have room to display.
2. **Data Validation**  
   Do not allow the creation of contact cards that do not have basic information (name and email perhaps). Provide the user with feedback about their error by using an alert.
3. **Avatar / Photo**  
   Ability to add photo/avatar using HTML5 File API. Resources:
   1. [http://blog.teamtreehouse.com/reading-files-using-the-html5-filereader-api](http://blog.teamtreehouse.com/reading-files-using-the-html5-filereader-api" \t "_blank) (readAsDataURL)
   2. [http://www.html5rocks.com/en/tutorials/file/dndfiles/](http://www.html5rocks.com/en/tutorials/file/dndfiles/" \t "_blank) (try the demo under Reading Files)
4. **Delete cards**  
   Ability to delete contact cards by clicking an “X” button located on a corner of the card (eg: top right corner). Try to animate the removal of the card.

https://mail.google.com/mail/u/0/images/cleardot.gif

--

Cheers,

Chris Tulip