

Assignment Web 4

Due Date: November 30

Purpose

You are part of the web development team at *Yippee!*, a huge Internet company that is rapidly losing ground to Google. You have been charged with implementing their newest, cloud-based service: *FotoFan*. To create this web app, you will use all of your HTML, CSS, JavaScript, jQuery, JSON, and AJAX knowledge.

You may work on this project with one (1) partner, if you wish.

Problem

Yippee! has many web services, including email, news, and now a cloud-based photo service. You are to create a web site that includes:

1. the *Yippee!* splash page that shows the various services the company has to offer
2. the *FotoFan* page that enables users to store and view their photos

Yippee! Page

The splash page should highlight the company's various web services, setting the tone for the rest of the website. Each service should have a link to a page devoted to that content. The FotoFan link should bring the user to a new page; all the other links should bring up a page with an internal Yippee! message: "In development by other web teams" or the like. The splash page should also include a clear way to login to the services. If the user clicks on the FotoFan page before logging in, then a login screen should pop up. *The user **must** login before entering the FotoFan page.*

FotoFan Page(s)

The FotoFan page is the main implementation page of the project. This page should look more like a web app than a typical web page, but should still have a way to navigate back to the splash page. The main goal of the page is to allow the user to view photos in a large scale as well as choose a single photo from a set of thumbnails. The page should be as dynamic as possible, changing as the user chooses various options; many of these dynamic elements are listed below.

Functionality should include (but is not limited to):

- a clear/consistent location to view a photo in high resolution
- a clear/consistent location to view and choose a photo from a set of thumbnails
- an user interface (UI) to create a slideshow of the current thumbnails
- a UI to choose a single thumbnail to view in high resolution
- a dropdown menu or other UI with pre-loaded choices from the user's JSON formatted image metadata.

The metadata for each photo includes at least:

- year taken. Note that this is the only metadata guaranteed for each photo.
- country

- state (if the country is the US)
- location (e.g., city, park, or other location name)
- description

Not all metadata should be included in the interface noted above. For example, it is not practical to show all photo descriptions in a dropdown menu. However, the year certainly is a common item to have in such a menu.

- search capability (for the above metadata)
- thumbnails should be dynamically loaded with the set of photos that match the menu choice or search results.
- a way to logout from the service. This should bring the user back to the splash page.
- a non-working link/interface to upload photos. (This may be part of Web 5...)

Note there are many design choices you have to make. For example, what should be shown in the main photo window before the user has made a choice? What should you do if there are more photos in a category than can fit in the thumbnail section? What should be done if a search fails?

Specifics

- The splash page should follow web page design guidelines. You can look at Yahoo's, Google's, and AOL's pages (among others) for drastically different approaches to this page. The design "look and feel" is up to you, but once chosen, should be followed throughout the site.
- The login information should be hard coded at this point. The two users that currently have "accounts" are Joe and Jane. Their passwords are "joe123" and "jane999," respectively. Each user should have his/her own database.
- All actions should be smoothly animated. For example, choosing a thumbnail should give some kind of nice feedback so the user knows it is working. Images should not just suddenly appear.
- The splash page should be stored in `index.html` in a folder called `yippee`. Create a link to this page from your home page.
- The FotoFan page should be named `FotoFan.html` and be stored in the `yippee` folder.
- CSS, PHP, images, etc. should be stored in appropriate folders/files. The site should be well organized.
- Build your web pages with HTML5 and CSS. CSS rules should be stored in an external style sheet.
- Functionality is to be provided with JavaScript and PHP. Store JavaScript code in an external file(s).
- JSON must be used to store image URLs and metadata. You will need two such JSON formatted data sets (one each for Joe and Jane). These should be stored in a separate file(s). However, to begin development (we have not yet seen how to store these externally), you can store them as strings in your JavaScript code.

- Access the JSON formatted data stored on the server through AJAX.
- Use jQuery as much as possible to simplify modifications of the DOM and communication with the server storing the JSON data.
- Plugins are allowed.
- Use proper JavaScript/jQUERY event handler conventions.
- You can use any web page editor to create your pages.

Notes

Joe's photos should include all of the images and metadata on my Anti-Research page. Jane's photos and metadata are up to you. Don't go crazy with quantity for the latter.

Upload all of your web pages to the server before 11:59:59 PM on the due date.

If your words or images are not on point, making them dance in color won't make them relevant.

– Edward Tufte