

Project Web 4

Due Date: November 24

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 web service: *FotoFan*. To create this web app, you will use all of your HTML, CSS, JavaScript, jQuery, PHP, and MySQL knowledge. This means this is a *full-stack* development project.

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 (look-n-feel) 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 (because your group is not responsible for *everything!*). The splash page should include a clear way to login to the services, in which case the user is already logged in if they then go to the *FotoFan* page. If the user clicks on the *FotoFan* page before logging in, then the same login screen should first pop up. *The user must login before entering the FotoFan page.*

For a sense of how this should work, check out the yahoo.com (or similar) splash page. It includes a login link plus a link for mail. You must be logged in before you can access your mail. I also expect a nicer, cleaner splash page than yahoo's, which I find scattered and full of ads (which you don't need).

FotoFan Page(s)

The *FotoFan* page (or pages) is the main implementation portion 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 or a new set of photos from their library (your database). 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
- a 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 image metadata.

You will store your own set of photos in your MySQL database on the server. The metadata for each photo includes at least:

- a link to each photo. I suggest storing all of the photos on the server; then you are guaranteed that they will still be there when it comes time to grade your project. Alternatively, you can use URLs for photos on the interwebs if you're confident they won't change.
- year taken. Note that this is the only metadata (besides the link to the photo itself) guaranteed for each photo.
- country
- state (if the country is the US)
- brief description; this could include additional location data

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 represents another part of the project being completed by a different group in the company.

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's¹ 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 "jane321," respectively. Each user should have his/her own database. You may add additional users if you wish.
- All actions should have good and consistent feedback and/or immediate results.
- Actions for appropriate options 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.

¹AOL still exists!

- The splash page should be stored in `index.html` in a folder called `yippee`. Create a link to this page from your home page(s).
- 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 within `yippee`. 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).
- The `mySQL` database must be used to store image URLs/links and metadata. You will need at least two tables, one each for the users Joe and Jane.
- Use jQuery as much as possible to simplify modifications of the DOM and communication with the server storing the data.
- You may want to store metadata in JSON format. Such data is then easy to parse with JavaScript in the browser.
- Plugins are allowed, but must work with your own database.
- Use proper JavaScript/jQuery event handler conventions.
- You can use any web page editor to create your pages.

Notes

Joe's photos should include at least six of the image sections 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. No hard copy is necessary to turn in.

Much of the grade will depend on: Does it work? However, a well-designed site is also essential. Adding extra functionality will give you more points in the usual "creativity" section.

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

– Edward Tufte