Rich Internet Application Project II

WEBD2000 Rich Internet Applications

Evaluation : 30% of Final Grade
Due Date : Apr 18

Assignment Description

The power of Rich Internet Applications! Your task for this final project is to re-work the infamous Photo Album completed as Web Programming Project II (ASP.NET) as a rich internet application using AngularJS. The basic features remain of the original web application with some added bells and whistles.

The requirements are:

- This photo album rich internet application must include a client side (AngularJS), a set of server-sided handlers scripts (ASHX), data classes (VB/DLL), and a provided database (photos.accdb)
- The client side is to be implemented entirely through AngularJS with *multiple* Controllers.
- All content on the photo album web application is to be fed from an external Access database
- The photo album must contain the following features and functionality:
    - A next and previous button for scrolling through the photos. These buttons should enable and disable when appropriate.
    - A photo counter (Photo: ? of ?)
    - A title for each photo
    - A caption for each photo
    - If the user is on the last photo the next button should be disabled; vice versa for the previous button when on the first photo
    - A list of comments submitted by users for each photo; each comment includes an author and the comment itself
    - An “Add Comment” panel that can be opened / closed to add a new comment to the current photo by entering an author name and the comment itself. Both these fields are required.
    - A “Jump To” panel that can be opened that displays thumbnails of all photos in the photo album. The thumbnails should be arranged in rows. Each thumbnail has some sort of rollover effect to indicate they are clickable. When the user clicks on a thumbnail the photo album immediately jumps to that photo.
- All data communication between the client side and the server side is to be done via JSON
- ASP.NET handler scripts must take advantage of a pre-built custom data class (See Album.cs) plus you will need to develop a new one…what new object has been introduced to the system?
- Your AngularJS code should integrate at least one service containing the business logic of the app. Controllers should only act as a go between for the service and the view (HTML)
- The web app is required to implement Bootstrap in order to ensure it is responsive
- A loading screen (overlay) with an animated spinner should be displayed while all data communication is being carried out and all images are being downloaded to prohibit the user from tinkering with the controls of the web app until it is ready
- If there are no photos in the database, the photo album web app should inform the user of this and adjust the interface accordingly
- Comments must be included throughout your code to explain your logic
- There is no need to develop an administration end of the photo album for this project; although this would make a great addition to your portfolio
- See in class demonstration of finished app.
- There is no doubt that this application integrates loads of technology (HTML, CSS, AJAX, JSON, AngularJS, ASP.NET, C# Custom Classes, MS Access) and would make a great portfolio piece. Using MS Visio or any other design tool draw a diagram outlining the entire web application and all the technologies that are working together. Add small notes throughout explaining how it works. This diagram will be the perfect companion to the finished app in your portfolio!

See other side…

Requirements (Marks breakdown)

Client Side Development (AngularJS)	
Photo Content Population (counter, title, caption, etc) *Data communication via JSON, updating all content, layout and design, etc*	4
Next and Previous Button Navigation *Next and previous navigation, enabling and disabling, etc*	4
Commenting Population	2
Adding A New Comment Functionality *Data communication via JSON, appropriate refreshing of RIA, error check, etc*	7
Jump To Panel Functionality	4
Loading Screen Implemented	2
Handling Empty Database Functionality	2
Angular Controllers / Services / Code Design	3
Bootstrap Implemented	1
Internal Commenting	1
Server Side Development (ASHX)	
ASP.NET Handler Scripts *Retrieving/Receiving data via JSON, usage of Data Class(es), etc.*	8
Technical Diagram	
Technical diagram of working web app	2
TOTAL MARK	40

Other Notes

- Remember to test each layer of your app separately before integrating (as demonstrated throughout our lessons)
- Like any web application, make sure you test for all types of possible user interactive, database data, and input; you'd be surprise the number of ways you can break a web application that are unforeseen
- Remember to divide and conquer! Start with the back end and get your photo album contents to be structured in JSON via an ASP.NET handler. Then move on to incorporating comments. Then build the Angular client side that receives that JSON, etc…
- This project will be marked on the due date through code review. Note that this project is due the last day of the semester and therefore no late submissions can be accepted.