Assignment – Mobile and Responsive Design  
COS318 – Web Programming

Your third Survivor season is about to begin. (The show Survivor has been on for more than 40 seasons; and yes, I’ve seen them all.) In this assignment you will finish your development of the Cloud Storage and Api Versioning project by making it compatible with mobile and responsive design. That means supporting different screen sizes and making elements behave as the user would expect when actions on the page are loading.

1. **(40 Points) CSS Media Query**
   1. Start with your completed project, API Versioning.
   2. Add a media query to your CSS that will activate when the screen width is 1000 or less.
   3. All elements of the page must flow correctly at the new screen size.
      1. Use your browser to control the screen size, or use the device size toolbar in the developer console to target specific device sizes.
2. **(40 Points) Loading States**
   1. Update all of the buttons (and sliders if you did previous assignment stretch levels) so that anytime the page is loading, they are disabled from input.
      1. This includes all form of loading, initial page load of the images, as well as when an image is being uploaded.
      2. Buttons should update their text to display loading messages while they are disabled.
      3. Ensure that the buttons return to normal when either the upload has completed or there was an error.
   2. During upload of the images, display a toast message that indicates the current step or progress of the image upload.
      1. There are three distinct steps to have a message for here; creating the image on the server, uploading to Azure (this could have a progress bar if you did previous assignment stretch levels), and finally calling upload complete on the image.
3. **(20 Points)** Code style, formatting, completeness, and quality.

Stretch Levels

If you already have a lot of experience with mobile development, or if you just played Survivor for the third time, try to complete these stretch levels for a reputation bonus. If you try for the stretch levels, make sure to type it in the comments on Moodle so I don’t miss it.

**Parvati Level**

Add a confirmation modal to the page when the user clicks Purge to confirm if they want to purge the images or not. This modal should block all other input on the page and have two options, “Cancel” and “Purge.” Hint: Remember rule 2, this should not be using anything with javascript alert. This modal should not pause any other javascript running on the page.

**Tony Level**

Preserve the api-version that the user has selected “1.0” or “1.1” so that if the page is refreshed, it uses whatever version was last used on that computer. Hint: it doesn’t need to be preserved across different computers; use the local storage of the web browser.) If you have done the image size from the previous stretch levels, preserve that value too for an additional bonus.

The Rules

1. No inline styles or inline javascript.
2. Error messages must be “in-page” i.e. no pop-ups or alerts.
3. Any resources not created by you (images, javascript libraries, etc.) must be referenced using a CDN or URL, not directly included in your assignment submission.
4. All requests that submit a body to your server must have their entities validated with appropriate annotations, such as MinLength, Range, or Required.
5. The root path of your server must display the main page of your application.
6. Service/data/model classes must not have any http, request, or response references.
7. Controller entity classes must not be used directly to store data on the server; translate them into a model (data storage) class before saving the data. Conversely, controllers must not send any model classes to the user; translate them into controller entity classes before sending the response.
8. All service class instances must be obtained using dependency injection.
9. You may not use any synchronous methods in your C# code wherever there is an async option.
10. All controllers (and their corresponding entities) must enforce the usage of an api version. Your namespace and folder structure for controllers and entities must contain the api version.