

# Responsive Design Lab

**Overview:** In this lab, you will create a small (3 page) web site that uses "single page design" and "roll your own" responsive design. You will learn how to create a HTML form to gather user input (and send it to a server side program). Your form will use some HTML5 validation (client side validation). You will learn how to use a bit of jQuery functionality (to implement a collapsible navigation bar).

- "Single page design" is where you have several virtual pages inside of a single physical page. The user can scroll up/down to go from virtual page to virtual page, or they can click in the fixed nav bar to navigate (internal links) to the virtual pages.
- "Responsive design" is when web pages adjust according to the width of the viewing device, e.g., one layout for desktop versus another layout for mobile phone. You test this by narrowing and widening your browser to see how the page elements realign themselves.
- HTML form tags have been around for a long time (for as long as you can remember being able to type input into a web page). Inside a form attribute you have one or more <input> tags plus a submit button. Whenever the user would click the submit button, all of the user's input values (that are in the same form as the submit button) are sent to the page that is specified by the form's "action" attribute. HTML5 (released not too long ago) introduced some new client side validation features.
- jQuery is an extensive library of JavaScript code. Its two main benefits are (1) you write less code than if you used just straight JavaScript, and (2) jQuery works well regardless of which browser (e.g., chrome vs. firefox vs. IE; old versions vs. new versions). jQuery tests which browser is being used and provides "graceful degradation of functionality" for older browser versions that do not support the "latest and greatest" features.

While Bootstrap is a great responsive design framework, we will not use any Bootstrap for this lab. Instead, you will learn the basics of what would be involved to write a framework like Bootstrap.

## Lab Requirements:

1. **Single Page Design.** You shall create a single file "02\_advLayout/index.html". This page will act like a small web site containing three virtual pages within the single physical page, using "single page design".
  - You may choose any topic for your "small web site" for this week, but it would be best if your topic is related to the topic you selected last week.
2. **Style Sheet.** Your single page shall reference an internal style sheet (and no external style sheet). The layout for this lab shall not affect your home or labs pages. Unless you are very new to HTML and CSS, come up with a totally new (and more exiting/professional) layout than what you had in your last lab.
3. **First Virtual Page - Landing Page.** The first virtual page (within "02\_advLayout/index.html") shall be a "landing page" (like a home page) that provides the user with an idea of what your "small web site" is about.
4. **Second Virtual Page - Informational Page.** The second virtual page (within "advLayout/index.html") shall be an informational page with at least 3 areas within it. Each of these areas shall have a title, an image, and some explanatory text. In desktop view, this virtual page shall display 2 or 3 of your areas side by side. In mobile view each area should be full width.
5. **Third Virtual Page - Form.** The third virtual page (within "02\_advLayout/index.html") shall be a form where users fill in some information and then submit it.
  - The form tag shall contain at least one of each of the following HTML input types: text box, text area, check box, select list, radio group, and number (new in HTML5). Each of these elements shall be "well named" so that when the form submits the email to you, the user input values will be labeled well.
  - The form tag shall contain a submit button.
  - At least two of the input tags shall be required. Your page shall implement this by using the HTML5 "required" attribute. If the user does not provide a value for a required field and they click submit, your page shall pop up a HTML5 validation bubble.
  - When the user clicks submit button (and if all the HTML5 validation conditions have been met), the user's input shall be sent to your Temple email address using the Temple mail program (see appendix).

6. **Fixed Title Nav.** "02\_advLayout/index.html" shall have a fixed Title/Nav bar which is always visible. This navigation bar shall have 4 links: 3 internal links (to the 3 virtual pages just described) and 1 link that gets you back to your main (current) index page that is stored up one folder, in your web root folder  
`<a href="../../index.html">`  
Remember that any fixed element has the potential to overlap with other elements. Apply a higher z-index to make your title nav be on top of other elements. When you link to each of your virtual pages, be sure that you "land" correctly at the top of that virtual page (not too high, not too low).
7. **Footer.** "02\_advLayout/index.html" shall have a footer with your name in it (can be fixed - always at the bottom - or visible only at the end of your last virtual page, as you wish).
8. **Responsive Design.** All three virtual pages shall employ responsive design, that is to say, all elements and components shall re-order and/or resize nicely according to the browser's screen size. Your grade will heavily depend on how well your webpage responds/looks at different screen sizes. Check the appendix to see an example that shows responsive design.
9. **Media Queries.** Your style shall have been designed "mobile first", which means you begin designing for mobile (most elements are "one up"). When that looks good, create a "desktop only" media query where you apply styles that have more than one area side by side and possibly different font sizes. You might then need a second media query for mobile only, but try to avoid this if possible.
10. **Collapsible Nav Bar.** The navigation bar shall be collapsible when in mobile view (using jquery) and always visible in desktop view.
11. **No Angular.** This lab shall not use Angular for code reuse (e.g., of title/nav or footer).
12. **Blog.** For this lab (and all labs), your labs page shall include a blog that describes and links to the work you did in the lab. In the blog, describe your experience doing the responsive design lab. Which aspects were easy? Which were hard? What were the most important things you learned in this lab?
13. **"For All Labs".** All code for this lab shall follow the requirements listed at the top of the 3344 labs page in a section entitled "Requirements for All Labs" (mostly just "good programming and design practices").

## Submission:

1. After completing all the requirements, test locally (and syntax check – right click and View Source from Firefox), then publish and test what you published.
2. Then submit into blackboard a zip file of your whole web site (meaning all the work you have done in all the labs so far). Make sure to include all the necessary files and folders in your zip file.

## Grading:

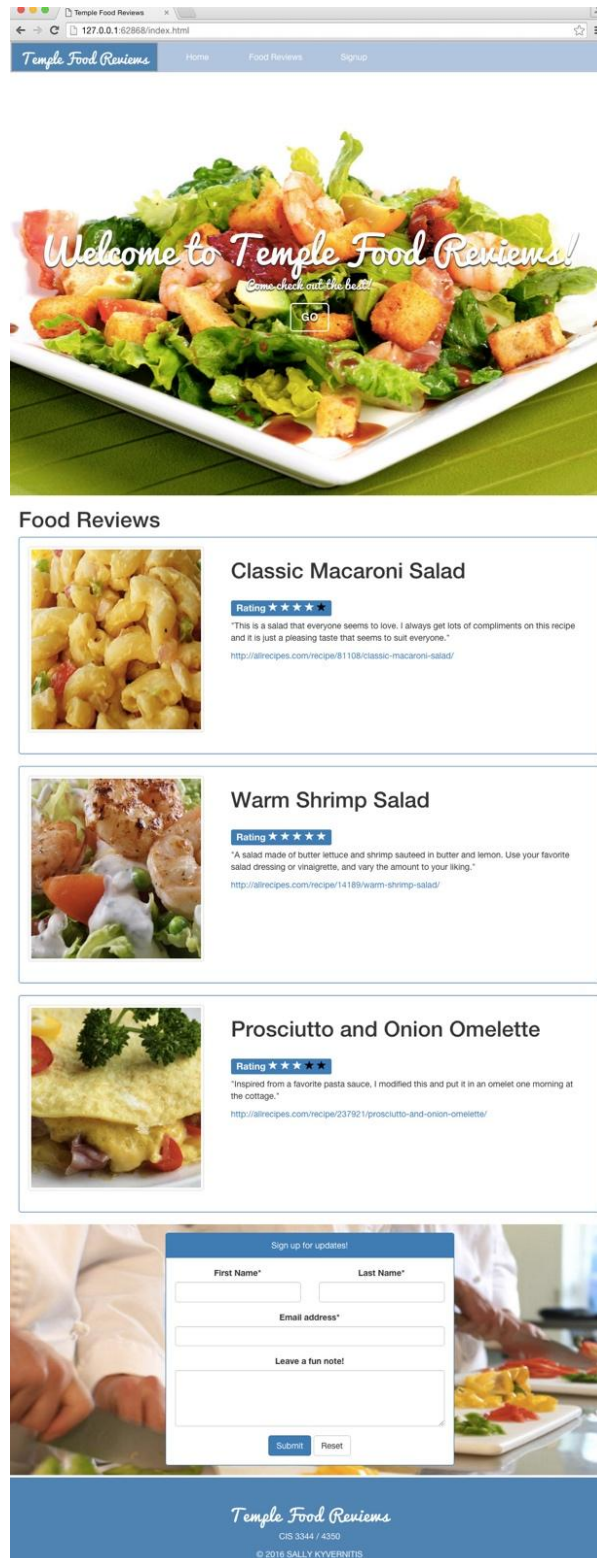
1. In **desktop width** browser, we will visit your labs page from your published web site. We will check that your **blog** for this week includes what was asked (what was easy? hard? important?). We will link from the blog to the landing area of the page you created for this week's lab.
2. We will scroll down top to bottom, checking if all three virtual pages look **professional**. Does the home page explain the (pretend) purpose of the 3 page website? Is there a clear visual division between each virtual page? Is each virtual page always at least as tall as the height of the browser?
3. We'll check that all links in the **nav bar** work and that the nav bar is always visible.
4. We'll check that the **informational virtual page** has at least 3 areas, each with picture, label and information. We'll check that (still in desktop view) these areas are displayed side by side (e.g., 2 up or 3 up).
5. We'll check that the **form** has the required input elements (text box, text area, check box, select list, radio group, number). We'll click submit without entering anything and see if we get a HTML5 validation bubble (at least two fields were to be required). After filling in all fields and clicking submit, we'll check the page that says an email that was sent to you. From this page, we'll check that you named your user input tags well.

6. Back in your Single Page, we will slowly narrow the browser until your media query changes your layout to **mobile view**. We'll check that all elements on all three pages adjusted nicely (typically all the areas "one up", some fonts/padding/margins may have changed).
  - a. We'll check that the **nav bar collapsed** and opens/closes when clicked (without causing the content to realign). We will check that all three internal links (from the mobile nav bar) work and correctly position to the right spot at the top of each virtual page.
7. We will **view source** to check for syntax errors, good naming, code indentation, neat code with nothing that's unnecessary.
8. We will use the final link to get **back to your main home page** (one folder up).

### Suggested Approach:

1. Decide the **topic** of your single page web site for this week's lab. Try to align it with your overall web site theme.
2. Find three **images** that are related to the topic. If the size of any image file is more than 500K or so, use the picture editing techniques (from the "Web Design" tutorial page) to make it smaller (or pick a smaller image to start with – google images allows you to filter by file size). Find or create a paragraph of text for each image and come up with a label for each image.
3. There are several sample pages referenced by this lab – single page design, responsive design, jquery (collapsible nav bar), and forms. Your job is to combine the code from all of these examples. I would start by taking the **responsive design example** (simplest one with Paris, London, Tokyo) and substitute your landing page info, the informational info, and add a form at the end.
4. Using "**Mobile First**" **design** approach, work in narrow browser width and set your style in the top area (that's not inside a media query). Create your landing page and your informational page (3 images each with label/title and explanatory paragraph). Each page needs to be visually separated from the other. This can be by alternating colors or adding borders etc.
5. **Design a form** that makes sense for your small web site's purpose. Use all the different input types that were asked. Name each one well. Make sure that the action attribute of the form tag references your own temple email address. Test the form to see if it sends you an email. Work on aligning the form Test that the form emails you the information of the form.
6. Once the physical page looks pretty good, widen the browser and make sure it looks good also in **wide mode** (desktop).
7. Add in a **fixed nav bar (desktop mode)** (using the single page design example) with internal links and test that. You might have to mess around with where you place the anchor tags (that are referenced) and/or how you style things so that you get to the right spot with each link.
8. Add in the jquery code to make your **nav bar collapse** (using the jquery example)
9. Test everything as we will do when we grade.

## Appendix: Example Design



**Desktop View (large screen)**




**Mobile View**

## Appendix: HTML Form and Input Example

### How to Send User Input to a Server Side Program (action attribute of HTML form)

If you want to get input from a user and have that input sent to a server side program, you need to use the HTML `<form>` tag. As shown on the right, your form can include various input elements, such as text boxes, text areas (larger, scrollable), drop-down lists, check boxes, radio buttons (select one and the others become unselected), and submit button. **Whenever the user clicks on a submit button, all the form's input values are sent to the server side program that is specified in the form action attribute (see red below).**

Here is the code (see below) for the HTML form on the right.



```
<html>
<head>
  <style>
    body {background-color:khaki}
  </style>
</head>
<body>
  <form name="myForm" method="post" action="http://www.temple.edu/cgi-bin/mail?sallyk@temple.edu">
    What is Your Name? <input type="text" name="userName" /> <br/> <br/>
    What is your Sex?   <input type="radio" name="userSex" value="M" />  Male
                      <input type="radio" name="userSex" value="F" />  Female <br/> <br/>
    Are you a Temple University Student? <input type="checkbox" name="isCisMajor" /> <br/> <br/>
    What Class are you in?
    <select name="collegeClass" >
      <option value="Fr" >Freshman</option>
      <option value="So">Sophomore</option>
      <option value="Jr">Junior</option>
      <option value="Sr" selected="selected">Senior</option>
      <option value="NaS">Not a Student</option>
    </select>   <br/> <br/>
    Comments or Suggestions? <br/>
    <textarea name="inputcomments" cols="45" rows="5">Type your comments here</textarea> <br/> <br/>
    <input type="submit" value="Click Here" />
  </form>
</body>
</html>
```

These radio button options are grouped together because they have the same name

If there are no errors in your form, when you test it (for example, with the input below left), you should see a confirmation page like the one shown below on the right. The confirmation page is generated by the CGI mail program – the server side program that was specified in the action attribute of the form tag.

What is Your Name?

What is your Sex? ☐ Male ☒ Female

Are you a Temple University Student? ☒

What Class are you in?

Comments or Suggestions?

**www.temple.edu Email Sent**

The following message has been sent.

---

Reply-to: sallyk@temple.edu (WebMonitor mail)  
From: sallyk@temple.edu (WebMonitor mail)  
To: sallyk@temple.edu  
Subject: FORM results

(userName) Sally Kyverntis  
(userSex) F  
(isCisMajor) on  
(collegeClass) Gd  
(inputcomments) I really like web development.

For more about HTML forms and input, check out W3Schools.com  
[http://www.w3schools.com/html/html\\_forms.asp](http://www.w3schools.com/html/html_forms.asp)