Exercise 17 – Advanced Bootstrap – Forms and Components

Objective

In the first part of this exercise you will build a mobile-first responsive form using Bootstrap’s default stylings and helper classes.

In the second part of this exercise you will take a second pass at the form using Bootstrap’s classes to improve the experience for our users.

In the final part of this exercise you will create a Meet the Team page.

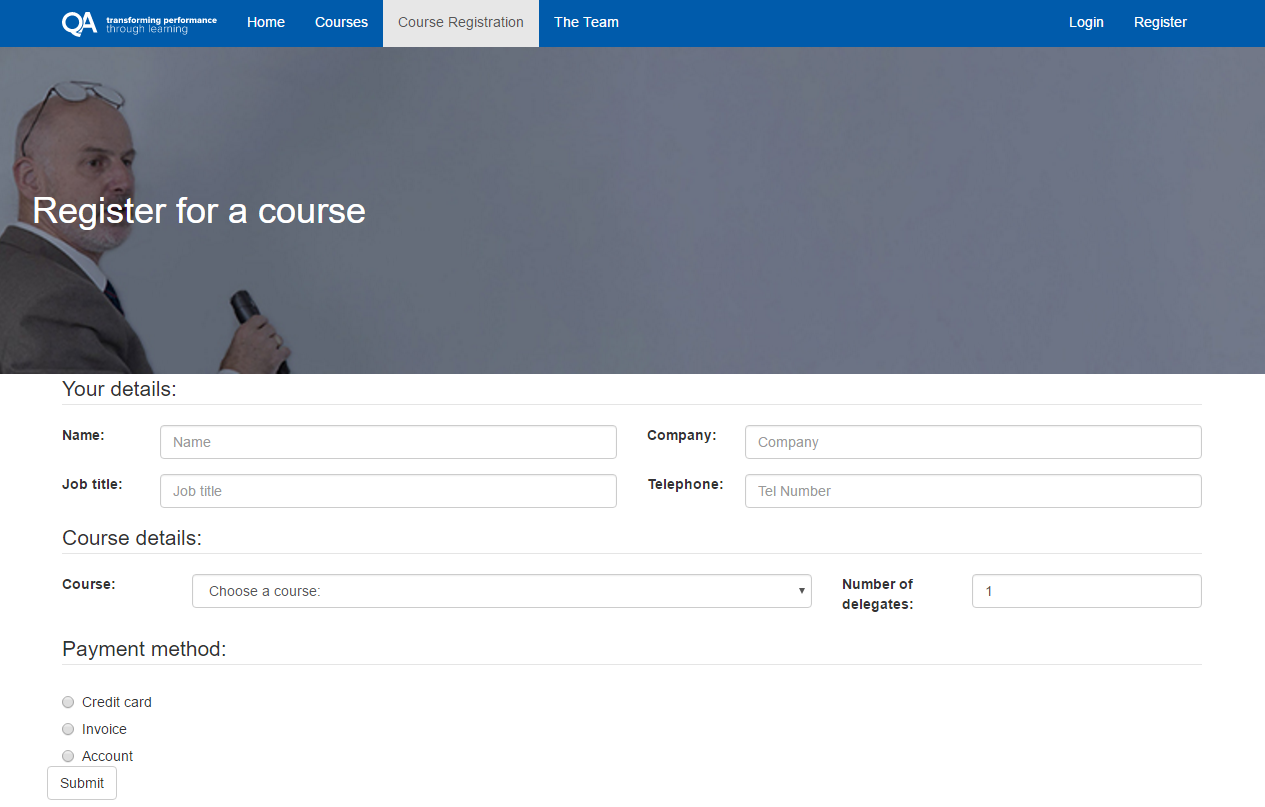
Part 1 – Building our Course Registration page

In the starter folder you will find the beginnings of a new website built with Bootstrap, currently we only have a very simple home page. We are going to now build out the Course Registration page using the tools we learnt about in this chapter.

1. Start off by creating a new HTML file called registration.html with all the starter code we will need to build with Bootstrap. You’ll also need to link to the app.css file provided.
2. Copy the nav element and all its contents from the index.html and put it inside the body of your new registration.html page. We want consistent navigation across the whole site.
3. Move the ‘active’ class from the Home list item to Course Registration.
4. Immediately after the nav element create an h1 element with the content “Register for a course” and give it a class of “trainer”
5. Add a main element with the bootstrap container class and inside this create a form.
6. We’re taking a mobile first approach to this form so we’ll want each element to stack top to bottom and taking the width of the page. Let’s gather some delegate information – create a series of appropriate label and input elements to gather the name, job title, company and contact telephone number for our delegate.
   1. Don’t forget to use the ‘form-group’ class to create appropriate wrappers around your label+input pairs.
   2. Each input element will require a class of ‘form-control’
7. Related form elements should really be wrapped in a fieldset, so go ahead and wrap your form-groups in a fieldset and create a legend of ‘Your details:’
8. Add a submit button at the end of your form. Your notes should help if you’ve forgotten what classes this button will need.
9. Marvel at your form in the browser. Isn’t it glorious?
10. Let’s gather some more details from our delegate.
11. Create a new fieldset for “Course details” and within that add a select element with the following options:
    1. Programming with Javascript
    2. Developing Responsive Websites
    3. Developing Web Applications using HTML5
    4. Leveraging the power of jQuery
    5. Building MVC Web Applications using AngularJS
12. Below the select box create an input element to take the number of delegates.
13. Create a third fieldset for “Payment method” and within this create three divisions with a class of ‘radio’. Within each division add a label and radio button: one for credit card, one for invoice, and one for account.
14. Take a look at your form in the browser and see how well it particularly suits mobile devices.

Part 2 – Improving our form for both mobile and desktop experiences

Our form is looking pretty good, but we’re wasting a lot of space on mobiles by having those labels and we’re not utilising the space afforded to us by desktops. Let’s fix that now.

1. Some of our inputs could benefit from the HTML5 placeholder attribute. Place an appropriate one on each of the “text”, “number” and “tel” input type elements.
2. We can apply a similar technique to the select element by adding in a disabled option to the top of the list. Something along the lines of “Choose a course” would do nicely here.
3. The great thing with placeholders is that we can now turn off labels for mobile devices. Use the classes “hidden-xs” and “hidden-xm” to hide labels for the elements we’ve just added placeholders for.
4. We’re now not wasting so much space on our mobile form, let’s have a look at the larger views where we’re not currently making the most of the additional width.
5. Add a class of “form-horizontal” to the form element
6. Each form-group acts as a row so we needn’t add row classes. Go ahead and add a class of “col-md-2” to each of our labels and then wrap its paired element (e.g. input or select element) in a div with a class of “col-md-10”. This creates a horizontally aligned label/input pairing which looks better on a desktop browser but we still see the benefits of our mobile-first design on small screens.
7. We can go yet another step further and utilise the Bootstrap grid to improve space use for even larger screens. Wrap all of the form-groups within the first fieldset (Your details) in a div with a class of row. Use the appropriate wrappers to make the 1st two form-group elements take up half the width and the 2nd two form-group elements take up the other half for large screens only.
8. Use a similar technique to the previous step to put the Course and Number of Delegates elements on the same row for large screens. You’ll likely have to change the share of space within this row for the number of delegates label and input – have a play with the numbers until it looks right to you. Your page should end up looking something like the below:

Part 3 – Building the team page

We have a link through to The Team but no page currently, so we’re going to use Bootstrap and some of its built in components to create one.

1. Create a new Bootstrap page with the filename team.html
2. Copy in the nav and its whole contents to the top of the body in our new page. Be sure to move the ‘active’ class from Course Registration to The Team.
3. Add a h1 below the nav with the content ‘Meet the team’ and a class of ‘’theTeam”
4. Add a main element below the h1 and make it a Bootstrap container. Create a new row and a div within taking 12 columns across all sized devices.
5. Within our column create a div with a class of ‘page-header’ and inside add a paragraph with a class of ‘lead’ – find the contents of the paragraph in theteam.txt within the resources folder.
6. Create a new row.
7. Within this row we will have 4 containers. On an extra small screen they will take up the full width of the row, on small screens they will take up half the width, on a medium screen a third, and on a large screen a quarter of the width.
8. Within the first column we’re going to create our first team member profile. Create a div with a class of “thumbnail” and put the image of Adrian inside it.
9. Underneath the image create a div with a class of caption with a h3 that reads “Adrian Jakeman” followed by a paragraph with his profile which can be found in the profiles.txt file in the resources folder. He is the first paragraph.
10. Now create profiles for David Walker, Laurence Sharp and Phil Stirpe in the remaining 3 columns you created.
11. Hopefully you will have something that looks like the below on a large screen:

