1. Customize navigation
   1. Comment out the old dropdown
   2. Add a link to Web.sitemap
   3. Change Application Name to whatever you want displayed instead, just above the commented-out code.
   4. Right click WebApp, add, sitemap, keep name default
   5. Mess with the code in Web.sitemap to make navigation work how you want, you’ll need to copy/paste MenuID and SiteMapDataSource stuff.
2. You’ll want a SamplePages folder or something along those lines to put pages in that you want to link to, about default and contact can just be left in the webapp folder.
   1. To create a new page, you right click the sample pages folder, add, new item, web form with master page.
3. Add a css page to use with CSS-Grid, and make sure to link to it in site.master
   1. Right click content, add, stylesheet, name it Customize.css
   2. Add “<link href="Content/Customize.css" rel="stylesheet" />” to the head of Site.Master so we can use the stuff we put in Customize.css
4. Open toolbox, and pin to left side. Drag elements from here instead of typing code manually
   1. Label: change text property, add AssociatedControlID property and match it to whatever it is a label for.
   2. Text Box: change ID
      1. OPTIONAL: add ToolTIp, add MaxLength (if required).
   3. Check Box: change ID,
      1. OPTIONAL: add a text property
   4. Button: change ID, change Text, add an OnClick property if you want it to be functional
      1. Make sure you set “causes validation” pproperty to be false on a cancel button
5. Right click the web forms page you are working on, view code to get to the back end of the page. You need to add objects for each button here, which replaces all the if/else nonsense we needed to use in razor. One object for submit button, one object for clear button.
   1. YOU NEED TO HAVE A PROPERTY ON THE BUTTON (IN THE .ASPX, NOT IN THE CODE BEHIND AKA ASPX.CS) ONCLICK =”BUTTONNAME”
   2. Create a button in .aspx, go to designer, double click the button. This will add the required OnClick property, as well as create an object in the .aspx.cs!
6. Validator types we use regularly, grab from toolbar as usual:
   1. Required field validator – self explanatory
   2. Regular expression validator – contains a property “ValidationExpression” that has requirements for the content of the text field being validated. Can set constraints to make sure the user has proper formatting for an email, postal code, phone number, etc.
   3. Range validator
   4. Compare validator – compare the value in the field being validated to either a value we choose (“ValueToCompare” property) or to another field
7. Use validation summary to display results from validators.
8. After validation messages are finished, you still need to setup an if statement on the aspx.cs that checks if the validation has worked before processing, this should be done for any button that requires validation.
   1. Also note that using validation on a checkbox is stupid, so you probably just want to use an if/else statement to check if the checkbox has been checked, and if so we can do stuff and if not send an error message.
9. Now that we are done validation, we need to create a class that can process data
   1. Right click webapp, add, class, give the class an appropriate name (in my example we are making a class for the user page, I named it UserClass so I know what it is).
   2. We need to create a property for each field that contains data (properties are CapitalCase and have a get/set)
   3. Default constructor and greedy constructor must be created. Within the greedy constructor, create variables and use them to assign data to the properties created in step b.
10. We are storing data! Yay! But we need to display it… lets throw that shit into a WebGrid! The code for populating a webgrid is in the aspx.cs stuff. Need to create a new list <t> and then use the class we just created in step 9 to fill it.

Questions for don:

* Why does the p tags around the buttons fix the formatting?
* How does the last part of customize.css (“input[type=checkbox] + label”) work?
* Is there a good way to fix the alignment of the text property of a checkbox?
* If I’m comparing two email fields I only need the compare validator on the second field right?
  + To go along with this, do I still need the expression validator on both?
* Is there a good way to format the error message for the checkbox not being checked? All the validator stuff looks great, this message is bleh.
* Get clarification on how constructors work, see practice forms
* Get clarification on the “variables” with get/set at the top of the class, these are properties? How does get/set work in this example? Online I always see get with some actual code after it.
* Why does if (terms) work? We don’t need if terms == true?
* **I ran into an annoying problem… my WebGrid was not being created in the designer, so I couldn’t access it. How can I stop this from happening?**