***DUE SEPTEMBER 19TH 2024***

Design and develop a form that asks the user to enter their personal info and fill out a short survey.  The form will be valid HTML, contain no tables, and have content completely separate from presentation by using external CSS rules.  The form will also have logic that will allow the user to go back and make changes if they find that they entered the wrong info.

Specifications

Implementation:

General

* Code is indented to show tag parent/child relationships.
* Your page should contain proper semantic formatting of content.
* The root directory for this assignment should be ~/public\_html/dig3134/assignment02
* The image directory for this assignment should be ~/public\_html/dig3134/assignment02/image
* The CSS directory for this assignment should be ~/public\_html/dig3134/assignment02/css
* Your HTML pages should be located in the root directory for this assignment and should be called*form\_entry.php*, *form\_preview.php*, and *form\_confirmed.php*
* The <title>’s of *form\_entry.php*, *form\_preview.php*, and *form\_confirmed.php* should be “*Assignment 2 – first\_name last\_name*”.
* All directories and filenames should contain no spaces or uppercase letters.
* All links need to be functional.
* All of your .php files need to have corresponding .txt symbolically linked files in the root directory.

HTML

* Web page validates as HTML 5 ([http://validator.w3.orgLinks to an external site.](http://validator.w3.org/)).

CSS

* The CSS directory for this assignment should be ~/public\_html/dig3134/assignment02/css
* All styles must be documented in an external CSS file called styles.css and linked to the document using the @import url rule or the <link> element.
  + All presentational HTML attributes should be replaced with CSS rules.
  + Use classes and/or id’s where appropriate.

PHP Functional Spec:

General

* No global variables should be used.
* Your logic should check if variables/superglobal arrays exist before trying to access values stored inside.
* Form submission should use the POST method
* You should mix standard HTML and PHP wherever possible (as opposed to outputting all of your HTML with PHP echo statements).

Form Processing

* Page 1 (**form\_entry.php**)
  + A page that contains blank form fields that the user will fill in.  The submit button should be labeled “Preview Answers”.
* Page 2 (**form\_preview.php**)
  + Once the user hits submit, the data they entered on Page 1 will be presented back to them with label information (for example: First Name – Dan Novatnak).
  + There will also be two buttons on the page.  One button will be labeled “Edit”, and the other button will be labeled “Finish”.
  + Edit Button
    - If the user hits the Edit button on Page 2, they will be sent to Page 1, and all of the data that they entered will appear as pre-filled values of each form field.  These values should be stored using Cookies and not hidden form fields (or any other more advanced method).
    - Once back on Page 1, functionality will continue as described in “Page 1” above.
  + Finish Button
    - If the user hits the Finish button on Page 2, they should be sent to Page 3.
* Page 3 (**form\_confirmed.php**)
  + The user should be shown a message that says “Thank you, your data has been submitted”
* Your form should be divided visually into at least 2 different styled fieldsets with legends.
* Questions should have obvious visual dividers.

Presentation

Content

The user should be shown two fieldsets:

* The first fieldset will contain text input fields to enter a first name, last name, e-mail, phone number, and students server address.
* The second fieldset will contain 5 questions of your choosing.
  + **Use common sense: Avoid obscene, offensive, or otherwise inappropriate questions.**