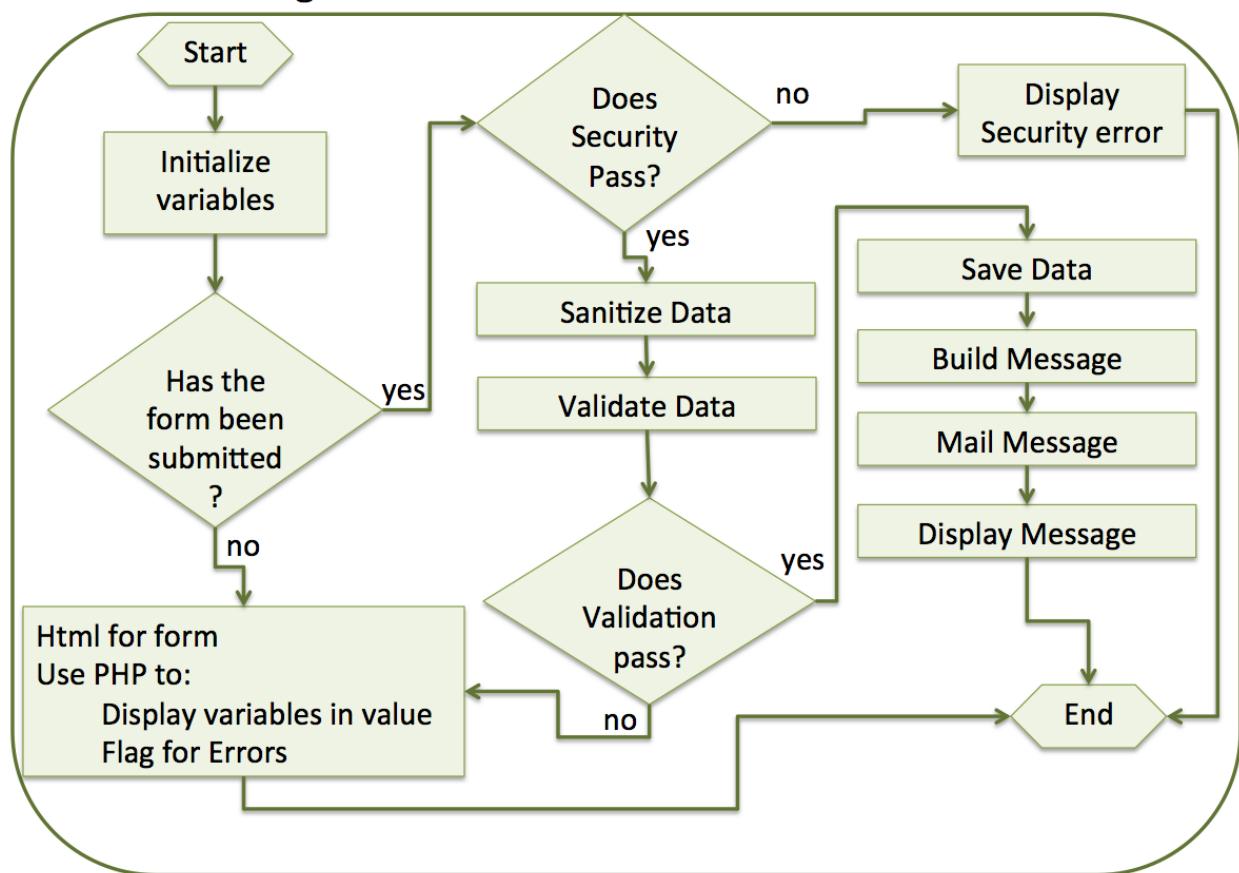


Form Processing Flow Chart



Once our data has passed validation we want to save it. We are going to save our data in a csv file. It is more common to save in a database (take CS 148) but a csv file works great for our purposes and is actually pretty easy and at this point mostly of things we have already done. The code can be found in GitHub commit 8: PHP Save data to csv file (and 8.1 since I forgot a part)

The first step to saving our data is to create an array to hold our data. I chose the name `dataRecord`. I create an empty array in Section 1.

```
$dataRecord = array();
```

Once you have the array you need to put the data into the array so you can save it. We can do this in Section 2 and I chose to put the data into the array just after I sanitize it. We append the data to the end of the array, doing this for each item. The order that you put them into your array would correspond to the column in excel. So if you put the email address into the array first, then email will be in column A

```
$dataRecord[] = $email;
```

Once we have confirmed that our data is valid we are ready to save our array. Refer back to chapter 3 on opening a csv file as the process is the same except we are going to open the file for append this time. If you use a data folder (like I did) be sure the data folder is already created as the fopen command will only create the file but will create a folder.

```
$file = fopen($filename, 'a');
```

By opening the file for append any data that has already been saved will still be there. Once the file is opened we can save the array that has our data in it by using the fputcsv command

```
fputcsv($file, $dataRecord);
```

Keep in mind this will save the record (will be a new row in your csv file) on the server and not on your local machine. To see the contents of the data file you will need to download it from the web. The variable \$filename contains the folder and name of the file, for example: data/registration.csv. Look at the url for your form, for example:

<https://rerickso.w3.uvm.edu/projects/form/commit-8/form.php>

Of course with any file once we are done we should close it:

```
fclose($file);
```

The process for saving data is easier than for reading a file. Here is the complete code:

```
// SECTION: 1e misc variables
// array used to hold form values that will be written to a CSV
file
$dataRecord = array();

// SECTION: 2b Sanitize (clean) data

$email = filter_var($_POST["txtEmail"], FILTER_SANITIZE_EMAIL);
$dataRecord[] = $email;

//@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
//
// SECTION: 2e Save Data

// This block saves the data to a CSV file.
$myFolder = 'data/';
```

```

$myFileName = 'registration';
$fileExt = '.csv';
$filename = $myFolder . $myFileName . $fileExt;
if ($debug) print PHP_EOL . '<p>filename is ' . $filename;

// now we just open the file for append
$file = fopen($filename, 'a');

// write the forms information
fputcsv($file, $dataRecord);

// close the file
fclose($file);

```

To download the data file just delete the file name form the url and add the filename for the data file:

<https://rerickso.w3.uvm.edu/projects/form/commit-8/data/registration.csv>

You should put a link on your sitemap to the csv file for the class. Normally though you would only have the link in an administrators page.

A few things to note. We never saved any labels so our data will begin in row 1. The other csv files we worked with had a header row so if you open this csv file with php code do not read in any headers. On the other hand if you would like to add headers you can. The steps to do this are:

1. Create an array to hold the header row. For example: `$labels = array;`
2. Add the labels to the array. For example: `$labels[ ] = 'Email Address';`
3. Repeat above for each label. Note first label will be in column A, second in column B etc.
4. After opening the file and before the fputcsv to save the \$dataRecord write (fputcsv) the headers. For example: `fputcsv($file, $labels);`
5. Save your data record.
6. Close your file.

Summary

Our data is valuable so once valid you should save it. Key php code:

- `dataRecord = array();`
- `fopen($filename, 'a');`
- `fputcsv`

Self Test Questions

1. Random Number File Writer

Write some php code that writes a series of 10 random numbers to a file. Each random number should be in the range of 1 through 500.

2. File reading and saving

Write some php code that opens a file that you've created called "name.txt" and saves your name to the end of it. It should not delete other names in the file. (Your name could be stored as a variable or you could create a form to submit the name.) Then display the list of names in the file on the page.

3. Two Files

Write some php code that writes the numbers 1 through 100 to a file (one number per line) and then reads this file and sums all the numbers in it, and saves the sum to a second file. Then print the sum on screen and the list of numbers.

4. Edit a file

Create a file named *students.csv* which contains a student's name and their grade, as follows:

Bob, A,
Kathy, B,
Jim, A,
Bill, C,

Write some php code that opens the file, changes Jim's grade to C and saves the file. (Note: don't solve this problem by simply accessing that index, assume Jim's name could appear on any line of the file.)

5. Random Alphabet Files

Write some php code that generates a random string of twenty letters from the set [A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,Y,X,Z] and saves it to a file. If a file with the same name already exists, the code shouldn't overwrite the file but should create a new one. (Optional: make a form with just a button that generates a new file each time it's pressed.)