## L20 HW –Web Form PHP Validation

## Professor Navarro

## COP3834 – Web Application Development

### **Background**

Forms in html only offer basic validation tools to prevent mistyped user input from being sent to the back office for storage or processing.

In this assignment, you will create a form and logic in PHP to accept student Lark submission files. A sample form is provided for you to update.

### **Procedure**

# Part 1 – Lark Form Set-up

1. Create or reuse the Unix LARK Submission form provided. All of the fields and javascript validation on the form are required.
2. Make the "Student Email" Input required to be submitted.

# Part 2 - Make a larkform php and use it to display

1. The larkform.php should be used to display the form and interface with the user. LarkForm.php should not contain functions or logic. Add all functions definitions, logic, class definitions on other php pages.

# Part 3 Php form Validations

1. Add php form validation for year published, ensure that It Is a 4 digit number.
2. Add php form validation for Medium, ensure that It Is alpha characters.
3. Add php form validation for Student email, ensure that It has alpha/numeric/underscore/dash or dots, then an @ symbol followed by alpha or numeric.

# PhP file upload Button

1. Create a file upload button.

# PHP file validations

1. Use php to validate that a file ending with .tgz Is attempting to be uploaded If "Type" Input has selected "Linux Bash". If the user selects "WebGL" from Type, then use PHP to validate the file being uploaded ends In .zip. If file type validates successful, upload to the uploads directory, If not, notify user to correct.

# Input error

1. Create an error handler to Notify user of errors for Input file Is unusable or any other errors.

# Persistent cookie

1. With php, Create a persistent cookie with an expiry date of 2 days or use a session variable. Populate the value from the "student email" Input. If user returns to page with cookie set, the update html to display "Welcome back [name of student email]". Be sure to use super global variables

# File Notification and Making a log

1. Notify user of successful form completion by closing form or opening new page.
2. Create a log of form submissions. Write all form Input to an ascii text file In the uploads directory with php. Use an append to prevent overwriting previous text. Each line should start with a date followed by all of the form data. Consider using 'local storage' and/or arrays.

# Submission -

1. Zip all web related files together and submit as single zipped directory.

|  |  |
| --- | --- |
| **L20 HW - Web Form PHP Validation Rubric** |  |
| **Assessment** | **Points** |
| Submit larkform and associated php, html, css, js files in a win zipped directory. | 10 |
| Cookie or session variables are handled correctly | 10 |
| PHP, Html, CSS, JS is easy to read, structured well with appropriate comments. Appropriate use of supporting php pages. | 10 |
| Successful submission results in form closing. | 10 |
| Form validation meets requirements and includes appropriate php | 20 |
| Appropriate error handling with php | 10 |
| File upload mechanism works correctly. | 20 |
| Submission upload log implemented correctly. | 10 |
| Sub total | 100 |
| Extra Credit – Add logic to prevent file upload from writing over file with same name. If it has the same name, modify the file name (ie add a number) to make it unique. | 5 |