Experiment No.: 09

Title: Write a program to validate a form using PHP.

Objectives:

1. To learn form handling in php using get and post method.
2. To understand empty () & preg_match () functions in PHP.

Theory:

HTML forms are required to collect the data from user, customer, etc. The HTML <form> tag is used to create a HTML form. The HTML form contains form elements. Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons and many more.

<input > Element-

Most important form element. It can be displayed in several ways, depending on type attribute.

Action Attribute-

It defines the action to be performed when the form is submitted. Normally, the form data is sent to a web page on server when user clicks on submit button. If action attribute is omitted, the action is set to the current page.

Method Attribute-

The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.

<form action="script" method="get">
<form action="script" method="post">

When to Use GET?

The default method when submitting the form data is GET. When GET is used, the submitted form data will be visible in the page address field.

Notes-

- Appends form-data into the URL in name/value pairs.
- The length of a URL is limited.
- Never use GET to send sensitive data.
- Useful for form submissions where a user wants to bookmark the result.
- GET is better for non-secure data, like query strings in Google.

When to Use POST?

Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.

Notes-

- POST has no size limitations & hence can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked.

Proper Validation of Form data is important to protect Form from hackers and spammers. Mostly when Form is submitted, the Form data is sent with POST method.

What is $_SERVER[“PHP_SELF”]?

It is a super global variable that returns the filename of currently executing script. It sends submitted Form data to page itself, instead of jumping to different page. In this case, user will get error messages on same page as the Form.

PHP Form Security -

$_SERVER[“PHP_SELF”] variable can be used by hackers. If PHP_SELF is used in page, then hacker can enter slash (/) & then some cross site scripting (XSS) commands are executed. The Cross site scripting (XSS) is insertion of malicious code into a page. XSS is a type of computer security vulnerability typically found in web application. XSS enables attackers to inject client side script into web page viewed by other user.

How to avoid $_SERVER["PHP_SELF"] Exploits?

By using htmlspecialchars () function, which converts special characters to HTML entities. It converts special characters to HTML entities i.e. it replaces HTML characters like < and > with < and > This prevents attackers from injecting HTML or JavaScript code (Cross-site Scripting Attack) in forms.

PHP Required Field Validation -

These fields cannot be empty & must be filled out. Add some new error variables to hold error messages for required fields. Add if else statement for each $_POST variable. This checks $_POST variable is empty or not (using PHP’s empty () function). If it is empty, error message is stored in different error variables & if not empty, it sends data.

Display Error Messages -

In HTML form, we add script after each required field which generates correct error message if needed.

PHP Regular Expression Validation -

Simple way to check if name field only contain letters & whitespaces by using preg_match () function. It searches string for pattern, returns true if pattern exists & false otherwise.

Syntax-

preg_match (pattern, variable)

Key concepts: PHP_SELF, htmlspecialchars (), empty (), preg_match ()