ASSIGNMENT 4

Aim : Validations using PHP

Problem Statement : Validate URL, Email, Required using functions empty, preg-match, filter - var in PHP

Theory :

1) What Is PHP ?
   - PHP is an acronym for "PHP: Hypertext Preprocessor". PHP is widely used open-source scripting language. PHP scripts are executed on the server. PHP is free to use and download. PHP file can contain text, html, CSS, Javascript and PHP code. PHP code is executed on server and the result is returned to the browser as plain HTML. PHP files have extension ".php"
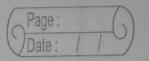   - PHP can generate dynamic page content. It can create, open, read, write, delete and ~~create~~ close files on the server. PHP can collect from data. It can send and recieve cookies. PHP can add, delete, modify data in database and can be used to control user-access. It can encrypt data.

2) Advantages of PHP
   - Extremely flexible : PHP is highly flexible whether it is during an ongoing project or after completeing the project. PHP has the ability to make changes even after starting the project and it saves valuable time.
   - Eassly integration and compatiblity : PHP can easily run on different platforms including UNIX, Solaris and linux. It can be integrated without effort with other technologies such as Java, etc.

- Efficient Performance: Depending on how the web developer codes, PHP has the potential to turn in an efficient language.
- Cost - efficient: PHP is an open source web language hence it is completely free.
- Gives web developer more control

3) PHP validations for email, URL

- In PHP to check whether an email address is valid to use PHP's filter_var() function. To validate email_address in php use the filter_validate_email which is a PHP email validation filter.
- These PHP filters are used to validate and filter data coming from insecure sources like user input. As of PHP 5.2.0, the filter functions are enabled by default. There is no installation needed to use these functions. filter_var - filters a variable with a specified filter

eg:
```
<?php
    $email_address = " me @ example . com ";
    if (filter_var ($email_address . filter_validate_email))
    {   // email address is valid }
    else  { // email address is invalid }
?>
```
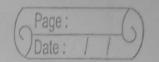
The preg_match() function searches a string for pattern returning true if the pattern exists and false otherwise.

eg:
```
$name = test_input ( $POST ["name"]);
    if (! preg_match ("/^ [a-zA-Z]* $/" $name)) {
        $nameErr = ' only letter & white space allowed ";  }
```

The code below shows a way to check if a URL address syntax is valid (this regular expression also allows dashes in the URL). If the URL address syntax is not valid the store an error message

```
$website = test_input ( $_POST ["website"]);
if (! preg_match ("/\b (?: (?: https ? | ftp ) : \/ \/ | www \.)
[a-z 0-9 +& @ # \ / %? = ~ _ | !: , . ;] * [a-z 0-9 +
& @ # \/ % = ~ _ |] /i ", $website ))
{ $websiteErr = " Invalid
```

It also accepts a regular expression followed by an array that is filled with results of match after
     The empty () accepting a regular expression followed by string
     The empty () determines whether a variable is empty
syntax: bool empty ($var)

Conclusion — I studied various features of PHP and used PHP for form validation.