**NAME:** Pulin Prabhu

**BATCH:** SE COMPS, C

**ROLL NO.:** 45

***EXPERIMENT 6***

**AIM:** Implementing Server Side Scripting using PHP Basics.

**THEORY:**

PHP is an open source scripting language used to create dynamic web page, modify the database,collect data,send and receive cookies.PHP files can contain text, HTML, CSS, JavaScript, and PHP code. A PHP script can be placed anywhere in the document.

**Syntax:**

A PHP script starts with **<?php** and ends with **?>**

**Variables In PHP:**

Variables in PHP start with a **$** symbol followed by the variable name.The global keyword is used to access a global variable from within a function.

The **echo** and **print** statement are used to output data. **echo** has no return value while print has a return value of 1 so it can be used in expressions. **echo** can take multiple parameters while print can take one argument.

**Datatypes in PHP:**

PHP supports the following data types:

String, Integer, Float (floating point numbers - also called double),Boolean,Array,Object,NULL and Resource

The special resource type is not an actual data type. It is the storing of a reference to functions and resources external to PHP.

**PHP Constants:**

Constants in PHP can be created using the **define** keyword.

define(*name, value, case-insensitive*) where name: Specifies the name of the constant

*value*: Specifies the value of the constant

*case-insensitive*: Specifies whether the constant name should be case-insensitive. Default is false

***Constants are global automatically and can be used within the entire script***

**PHP Operators:**

The various operators in PHP include Arithmetic operators, Assignment operators, Comparison operators, Increment/Decrement operators, Logical operators, String operators, Array operators.

**===** is used to check for identity.Returns true if the variables have same value and similar data type.

**Dynamic Content:**

*Identifying Browser & Platform*

PHP creates some useful environment variables that can be seen in the phpinfo.php page that was used to setup the PHP environment.

One of the environment variables set by PHP is HTTP\_USER\_AGENT which identifies the user's browser and operating system.

PHP provides a function getenv() to access the value of all the environment variables. The information contained in the HTTP\_USER\_AGENT environment variable can be used to create dynamic content appropriate to the browser.

*Display Images Randomly*

The PHP rand() function is used to generate a random number.i This function can generate numbers with-in a given range. The random number generator should be seeded to prevent a regular pattern of numbers being generated. This is achieved using the srand() function that specifies the seed number as its argument.

*Using HTML Forms*

The PHP default variable $\_PHP\_SELF is used for the PHP script name and when you click "submit" button then same PHP script will be called and will produce following result −

The method = "POST" is used to post user data to the server script. There are two methods of posting data to the server script which are discussed in PHP GET & POST chapter.

*Browser Redirection*

The PHP header() function supplies raw HTTP headers to the browser and can be used to redirect it to another location. The redirection script should be at the very top of the page to prevent any other part of the page from loading.

The target is specified by the Location: header as the argument to the header() function. After calling this function the exit() function can be used to halt parsing of rest of the code.

**Control Structures and Conditional Iteration Constructs**

PHP uses the same control structures as any other language like Java or C, i.e.- **if…elseif...else** and **switch case**. The syntax for the same is also same as what we normally use in Java or C.

PHP provides conditional iteration constructs like **for loop**, **while loop**, **do-while loop** and **foreach loop.** The syntax for these constructs is also same as that of any other language.

**Arrays In PHP:**

We use the **array()** function to create an array in php.

In PHP, there are three types of arrays:

Indexed arrays - Arrays with a numeric index

Associative arrays - Arrays with named keys

Multidimensional arrays - Arrays containing one or more arrays

Indexed Array:$cars = array("Volvo", "BMW", "Toyota");

Associative arrays are arrays that use named keys that you assign to them.

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

**String Functions**

There are many types of functions available for operating with strings:

chop() Removes whitespace or other characters from the right end of a string

chr() Returns a character from a specified ASCII value

convert\_uudecode() Decodes a uuencoded string

convert\_uuencode() Encodes a string using the uuencode algorithm

echo() Outputs one or more strings

explode() Breaks a string into an array

fprintf() Writes a formatted string to a specified output stream

implode() Returns a string from the elements of an array

join() Alias of implode()

print() Outputs one or more strings

printf() Outputs a formatted string

ltrim() Removes whitespace or other characters from the left side of a string

rtrim() Removes whitespace or other characters from the right side of a string

similar\_text() Calculates the similarity between two strings

sprintf() Writes a formatted string to a variable

sscanf() Parses input from a string according to a format

str\_getcsv() Parses a CSV string into an array

str\_ireplace() Replaces some characters in a string (case-insensitive)

str\_pad() Pads a string to a new length

str\_repeat() Repeats a string a specified number of times

str\_replace() Replaces some characters in a string (case-sensitive)

str\_shuffle() Randomly shuffles all characters in a string

str\_split() Splits a string into an array

str\_word\_count() Count the number of words in a string

strcasecmp() Compares two strings (case-insensitive)

strcmp() Compares two strings (case-sensitive)\

strcspn() Returns the number of characters found in a string before any part of some specified characters are found

strip\_tags() Strips HTML and PHP tags from a string

stripos() Returns the position of the first occurrence of a string inside another string (case-insensitive)

stristr() Finds the first occurrence of a string inside another string (case-insensitive)

strlen() Returns the length of a string

strnatcasecmp() Compares two strings using a "natural order" algorithm (case-insensitive)

strnatcmp() Compares two strings using a "natural order" algorithm (case-sensitive)

strncasecmp() String comparison of the first n characters (case-insensitive)

strncmp() String comparison of the first n characters (case-sensitive)

strpos() Returns the position of the first occurrence of a string inside another string (case-sensitive)

strrev() Reverses a string

**Regular Expressions**

PHP uses the **POSIX regular expression.**

**Brackets** ([]) - They are used to find a range of characters.

[0-9]: It matches any decimal digit from 0 through 9.

[a-z]: It matches any character from lower-case a through lowercase z.

[A-Z]: It matches any character from uppercase A through uppercase Z.

[a-Z]: It matches any character from lowercase a through uppercase Z.

**Quantifiers -** The frequency or position of bracketed character sequences and single characters can be denoted by a special character. Each special character having a specific connotation.

p+: It matches any string containing at least one p.

p\*: It matches any string containing zero or more p's.

p{N}: It matches any string containing a sequence of N p's

p$: It matches any string with p at the end of it.

^p: It matches any string with p at the beginning of it.

**Regexp POSIX Functions -**

ereg(): The ereg() function searches a string specified by string for a string specified by pattern, returning true if the pattern is found, and false otherwise.

ereg\_replace(): The ereg\_replace() function searches for string specified by pattern and replaces pattern with replacement if found.

split(): The split() function will divide a string into various elements, the boundaries of each element based on the occurrence of pattern in string.

preg\_match(): The preg\_match() function searches string for pattern, returning true if pattern exists, and false otherwise.

preg\_replace(): The preg\_replace() function operates just like ereg\_replace(), except that regular expressions can be used in the pattern and replacement input parameters.

preg\_split(): The preg\_split() function operates exactly like split(), except that regular expressions are accepted as input parameters for pattern.

preg\_grep(): The preg\_grep() function searches all elements of input\_array, returning all elements matching the regexp pattern.

**Fuctions in PHP**

While creating a user defined function we need to keep few things in mind:

1. Any name ending with an open and closed parenthesis is a function.
2. A function name always begins with the keyword function.
3. To call a function we just need to write its name followed by the parenthesis
4. A function name cannot start with a number. It can start with an alphabet or underscore.
5. A function name is not case-sensitive.

**Syntax -**

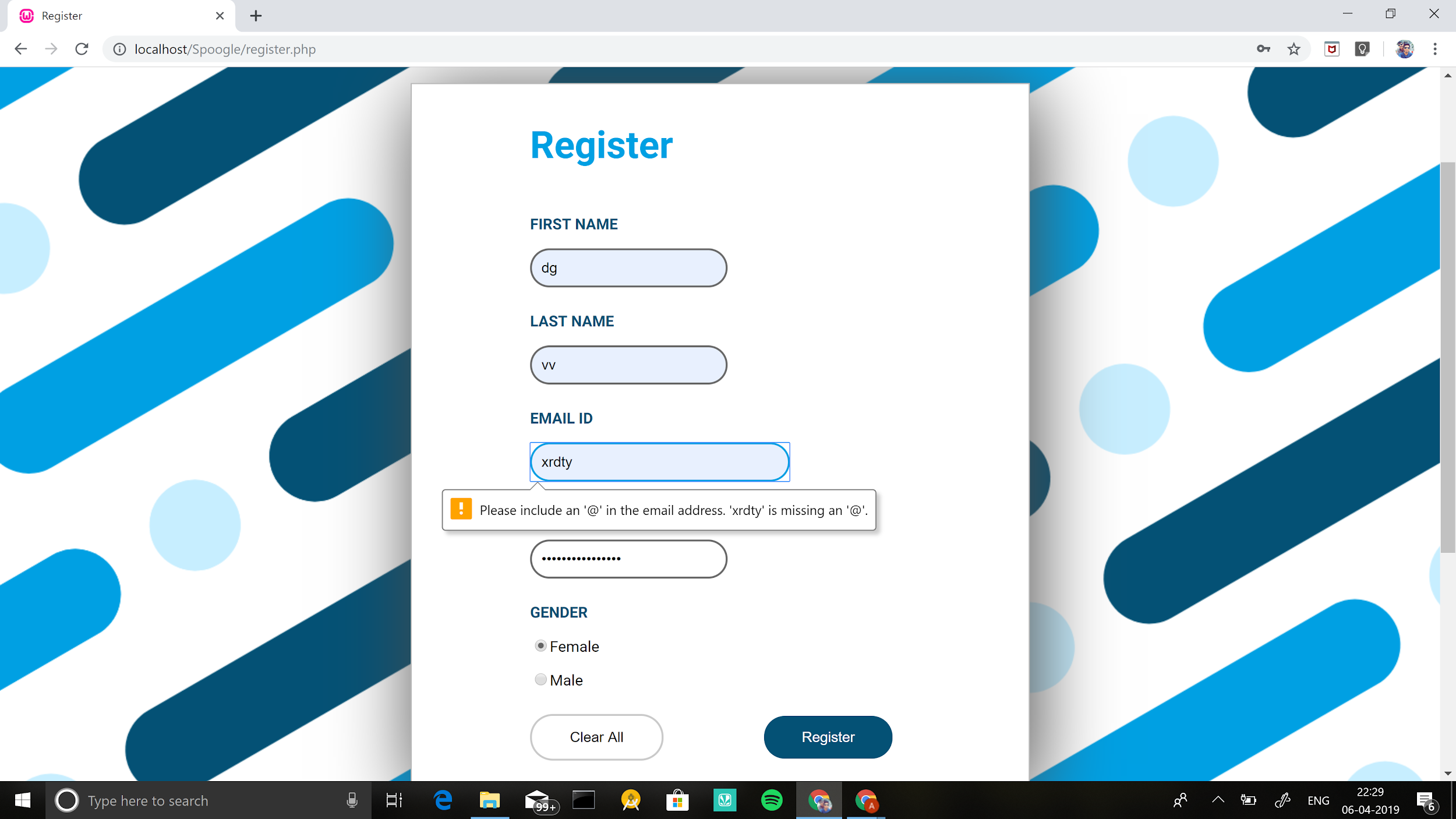
function function\_name(){

executable code;

}

**Reusability of code -** If we have a common code that we would like to use at various parts of a program, we can simply contain it within a function and call it whenever required. This reduces the time and effort of repetition of a single code. This can be done both within a program and also by importing the PHP file, containing the function, in some other program.

**SCREENSHOTS:**

****

**CONCLUSION:** From this experiment, we have now implemented basic elements of php in our website and also made our website available on the localhost server.