Integrating PHP with Embedded System

Contents

What is PHP?

- PHP stands for PHP: Hypertext Preprocessor
- PHP is a server-side scripting language
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source software
- PHP is free to download and use

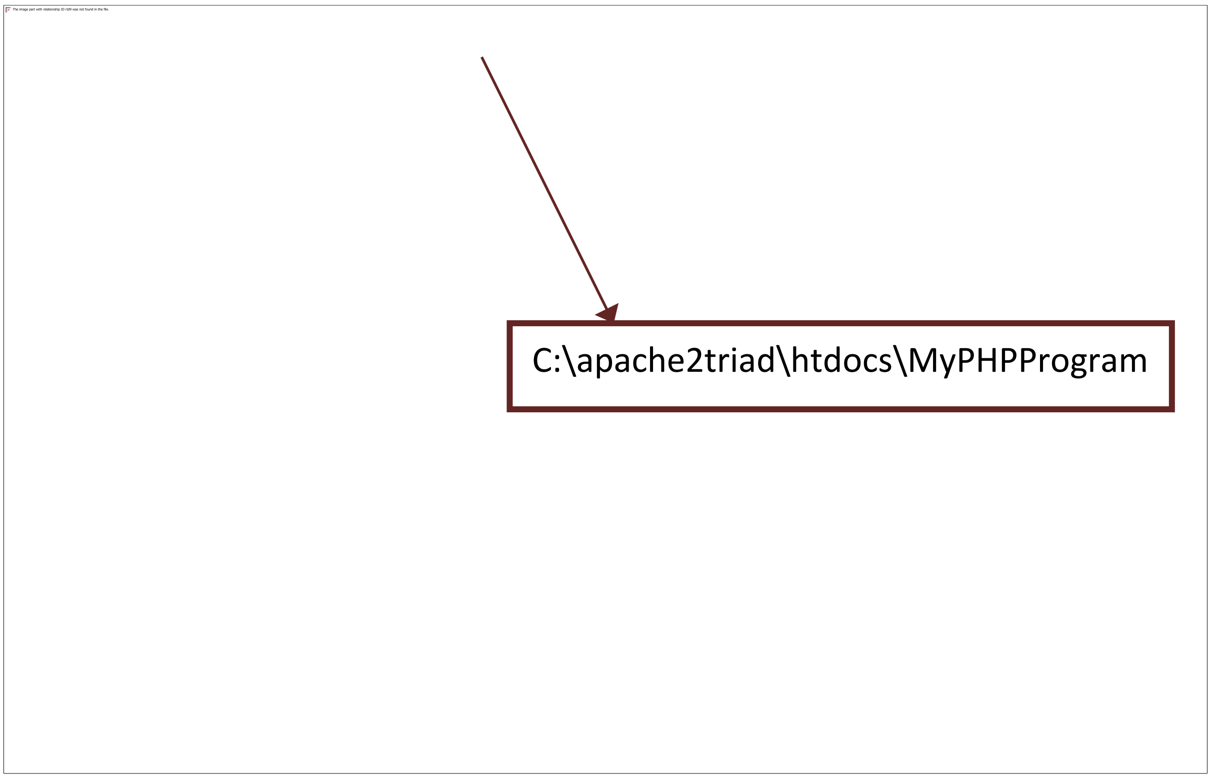
Basic PHP Syntax

- A PHP script starts with **<?php** and ends with **?>**
- The default file extension for PHP files is ".php"
- A PHP file normally contains HTML tags, and some PHP scripting code
- PHP statements are terminated by semicolon (;)
- In PHP, all user-defined functions, classes, and keywords (e.g. if, else, while, echo, etc.) are not case-sensitive

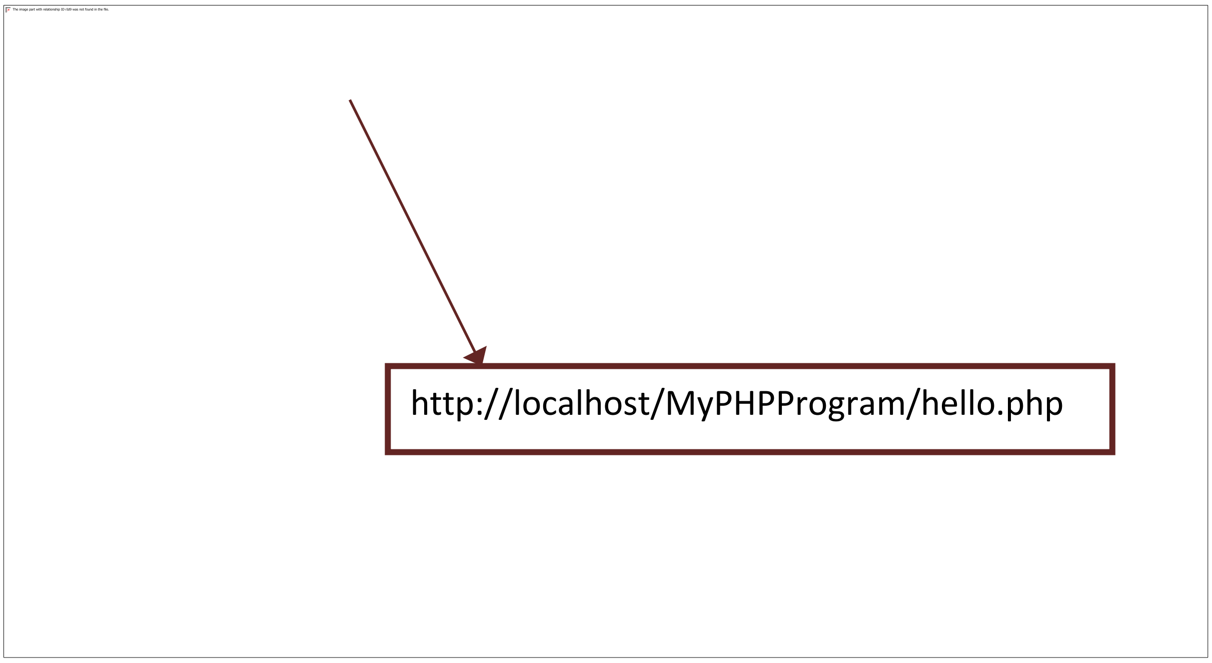
Hello World example

```
<html>
   <body>
       <?php
          // Use echo to print on console
          echo “Hello World!”;
       ?>
    </body>
</html>
```

Go to htdocs folder which is present in the apache2triad installed folder. There create a folder and save this program with .php extension such as Hello.php.


The image part with relationship ID rId9 was not found in the file.
C:\apache2triad\htdocs\MyPHPProgram

To execute hello world program, type in the address bar as follows:
http://localhost/MyPHPProgram/hello.php


The image part with relationship ID rId9 was not found in the file.
http://localhost/MyPHPProgram/hello.php

Error Management

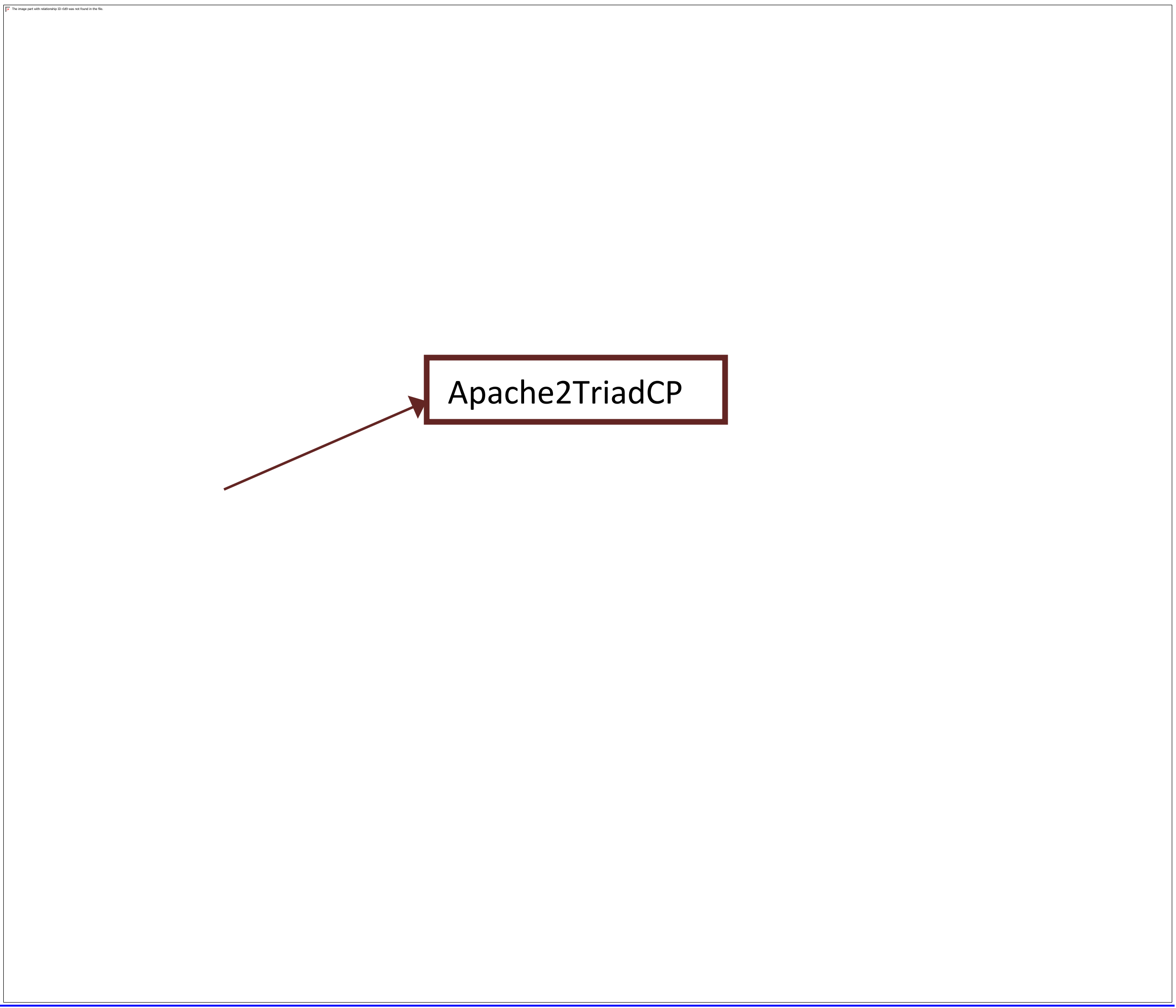
1. **Compile-time errors:**
   - Compile-time errors are detected by the parser while it is compiling a script.
   - The compile-time errors cannot be trapped from within the script itself
2. **Fatal errors:**
   - Fatal errors are the errors that halt the execution of a script.
   - The fatal errors cannot be trapped.
3. **Recoverable errors:**
   - Recoverable errors that represent significant failures, but can still be handled in a safe way.
4. **Warnings:**
   - Warnings are recoverable errors that indicate a run-time fault.
   - Warnings do not halt the execution of the script.
5. **Notices:**
   - Notices indicate that an error condition occurred, but is not necessarily significant.
   - Notices do not halt the execution of the script.

Finding errors present in the program

```
<html>
   <body>
     <?php
          echo “Hello World!”;
     // here ? is missing
     >
   </body>
</html>
```

To find the errors present in the program go to:

Start -> All programs -> Apache2triad -> Apache2TriadCP


The image part with relationship ID rId9 was not found in the file.
Apache2TriadCP

Then click on “PHP Error log”

The image part with relationship ID rId9 was not found in the file.

The list of errors in the program is displayed along with the line number where the error has occurred.

The image part with relationship ID rId9 was not found in the file.

```
[05-Jan-2007 05:21:11] PHP Parse error:  syntax error, unexpected T_STRING,
expecting ',' or ';' in C:\apache2triad\htdocs\MyPHPProgram\hello.php on line 4
```

Comments in PHP

- // Single line comment (C++ and Java-style comment)
- # Single line comment (Shell-style comments)
- /* Multiple line comment
  (C-style comments) */

PHP is a Loosely Typed Language

- In PHP, a variable does not need to be declared before adding a value to it
- PHP automatically converts the variable to the correct data type, depending on its value
- In PHP, the variable is declared automatically when you use it
- PHP variables must begin with a “$” sign
- Variables are used for storing values, like text strings, numbers or arrays
- The correct way of declaring a variable in PHP:

  $var_name = value;

PHP Variables Example

```
<html>
   <body>
         <?php
            $a = 25;         // Numerical variable
            $b = “Hello”;   // String variable
            $c = 5.7;       // Float variable
            echo “Number is : ”.$a.“<br/>”;
            echo “String is : ”.$b.“<br/>”;
            echo “Float value : ”.$c;
         ?>
  </body>
<html>
```

OUTPUT of the above given Example is as follows:

```
Number is : 25

String is : Hello

Float value : 5.7
```

Global and locally-scoped variables

- Global variables can be used anywhere
- Local variables restricted to a function or class

Example for Global and locally-scoped variables

```
<html>
   <body>
      <?php
          $x=24; // global scope

          // Function definition
          function myFunction() {
                $y=59; // local scope
               echo "Variable x is: $x <br>";
               echo "Variable y is: $y";
          }

          myFunction();// Function call

          echo "Variable x is: $x";
          echo "<br>";
          echo "Variable y is: $y";
        ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

Variable x is:

Variable y is: 59

Test variables outside the function:

Variable x is: 24
Variable y is:

Static Keyword in PHP

- Static keyword is used when you first declare the variable
- Each time the function is called, that variable will still have the information it contained from the last time the function was called

Static Keyword Example

```
<html>
   <body>
         <?php
                     // Function definition
               function myFunction() {
                           static $x=45;
                           echo $x;
                           echo "<br/>";
                           $x++;
               }
                     // Function call
               myFunction();
               myFunction();
               myFunction();
               myFunction();
               myFunction();
         ?>
       <body>
 <html>
```

OUTPUT of the above given Example is as follows:

```
45
46
47
48
49
```

ECHO and PRINT statements in PHP

- ECHO - It can output one or more strings
- PRINT – It can only output one string, and returns always 1
- ECHO is faster compared to PRINT as echo does not return any value
- ECHO is not a function and, as such, it does not have a return value
- If you need to output data through a function, you can use PRINT() instead:

  Example:

  ```
  echo 50;
  print (50);
  ```

PRINT Statement Example in PHP

```
<html>
   <body>
         <?php
              // Use ‘print’ to print on console
              print "Hello world!<br>***********";
         ?>
   </body>
 </html>
```

OUTPUT of the above given Example is as follows:

```
Hello world!
***********
```

String Functions in PHP

- strlen() function
- strpos() function

1. strlen() function

- The strlen() function returns the length of a string, in characters

```
<html>
     <body>
          <?php
               // Displays the length of the string
               echo strlen("Hello world!");
          ?>
     </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
12
```

2. strpos() function

- The strpos() function is used to search for a specified character or text within a string

```
<html>
     <body>
          <?php
               /* Displays the position of ‘world’ in the
               text ‘Hello world*/
              echo strpos("Hello world!","world");
          ?>
     </body>
 </html>
```

OUTPUT of the above given Example is as follows:

```
6
```

Constant in PHP

- define() function is used to set a constant
- It takes three parameters they are:
  1. Name of the constant
  2. Value of the constant
  3. Third parameter is optional. It specifies whether the constant name should be case-insensitive. Default is false

Constant string Example

```
<html>
     <body>
               <?php
                    /* Here constant name is ‘Hai’ and ‘Hello
                    Friend’ is its constant value and true
                    indicates the constant value is case-
                    insensitive */
                    define("Hai","Hello Friend",true);
                    echo hai;
               ?>
        </body>
 </html>
```

OUTPUT of the above given Example is as follows:

Hello Friend

PHP Example to calculate the area of the circle

```
<html>
     <body>
           <?php
                // defining constant value PI = 3.14
                define("PI","3.14");
                $radius=15;
                $area=PI*$radius*$radius;
                echo "Area=".$area;
           ?>
     </body>
 </html>
```

OUTPUT of the above given Example is as follows:

Area=706.5

Arithmetic Operators

- Arithmetic operators allow performing basic mathematical operations

Operator	Description	Example	Result
+	Addition	$a = 2 + 5;	$a=7
-	Subtraction	$a = 10 - 2;	$a=8
*	Multiplication	$a = 2 * 5;	$a=10
/	Division	$a = 15 / 5;	$a=3
%	Modulus	$a = 23 % 7;	$a=3.28
++	Increment	$a =5; $a ++;	$a=6
--	Decrement	$a =5; $a --;	$a=4

Arithmetic Operators Example

```
<html>
  <body>
        <?php
               // Add 20, 10 and sum is stored in $i
               $i=(20 + 10);
               // Subtract $i, 5 and difference is stored in $j
               $j=($i - 5);
               // Multiply $j, 4 and result is stored in $k
               $k=($j * 4);
               // Divide $k, 2 and result is stored in $l
               $l=($k / 2);
               // Devide $l, 5 and remainder is stored in $m
               $m=($l % 5);
               echo "i = ".$i."<br/>";
               echo "j = ".$j."<br/>";
               echo "k = ".$k."<br/>";
               echo "l = ".$l."<br/>";
               echo "m = ".$m."<br/>";
        ?>
  </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i = 30
j = 25
k = 100
l = 50
m = 0
```

Increment and Decrement Operators

Operator	Name	Description
++$a	Pre-increment	Increments $a by one, then returns $a
$a++	Post-increment	Returns $a, then increments $a by one
--$a	Pre-decrement	Decrements $a by one, then returns $a
$a--	Post-decrement	Returns $a, then decrements $a by one

Increment and Decrement Operators Example

```
<html>
      <body>
            <?php
                  $i=10;
                  $j=20;
                  $i++;
                  $j++;
                  echo $i."<br/>";
                  echo $j."<br/>";
                  // Post increment
                  $k=$i++;
                  // Pre increment
                  $l=++$j;
                  echo $k."<br/>";
                  echo $l;
            ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
11
21
11
22
```

Assignment Operators in PHP

- Assignment operator is used to write a value to a variable

Operator	Example	Is the same as
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

Assignment Operators Example

```
<html>
     <body>
          <?php
               $a=5;
               echo "a=".$a;
               echo "<br/>";

               $b=10;
               $b += 20;
               echo "b=".$b;
               echo "<br/>";
               $c=15;
               $c -= 5;
               echo "c=".$c;
               echo "<br/>";


               $d=20;
               $d *= 2;
               echo "d=".$d;
               echo "<br/>";
```

```
                        $e=25;
                        $e /= 5;

                        echo "e=".$e;
                        echo "<br/>";

                        $f=30;
                        $f %= 4;
                        echo "f=".$f;
                    ?>
                </body>
            </html>
```

OUTPUT of the above given Example is as follows:

```
a=5
b=30
c=10
d=40
e=5
f=2
```

String Operators in PHP

Operator	Name	Example	Result
.	Concatenation	$a = "Hello" $b = $a . " world!"	$b = "Hello world!"
.=	Concatenation Assignment	$a = "Hello" $a .= " world!"	$a = "Hello world!"

String Operators Example

```
<html>
   <body>
           <?php
                $a = "Hello";
                $b = $a . " Friend!";
                echo $b;
                echo "<br/>";

                $c="Good";
                $c .= " Day!";
                echo $c;
           ?>
    </body>
 </html>
```

OUTPUT of the above given Example is as follows:

```
Hello Friend!
Good Day!
```

The if Statement in PHP

- If statement executes some code only if a specified condition is true

Syntax:

```
if (condition) {
    code to be executed if condition is true;
}
```

The if Statement Example

```
<html>
      <body>
            <?php
                  $i=0;

                  /* If condition is true, statement is
                  executed*/

                  if($i==0)
                        echo "i is 0";
            ?>
      <body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is 0
```

The if...else Statement in PHP

- If...else statement executes some code if a condition is true and some another code if the condition is false

Syntax:

```
if (condition) {
        code to be executed if condition is true;
}

else {
        code to be executed if condition is false;
 }
```

The if...else Statement Example

```
<html>
     <body>
          <?php
               $i=1;

               /* If condition is true, statement1 is
               executed, else statement2 is executed*/

               if($i==0)
                    echo "i is 0";   //statement1
               else
                    echo "i is not 0";  //statement2
          ?>
     <body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is not 0
```

The if...elseif...else Statement in PHP

- If...elseif...else statement selects one of several blocks of code to be executed

Syntax:

```
if (condition) {
        code to be executed if condition is true;
}

elseif (condition) {
        code to be executed if condition is true;
}

 else {
        code to be executed if condition is false;
}
```

The if...elseif...else Statement Example (Comparing two numbers)

```
<html>
   <body>
         <?php
               $i=22;
               $j=22;
               /* If condition1 is true, statement1 is executed,
                if condition1 is false and condition2 is true,
                statement2 is executed, if both the conditions
                are false statement3 is executed */
               if($i>$j)
                     echo "i is greater";  //statement1
               elseif($i<$j)
                     echo "j is greater";  //statement2
               else
                     echo "numbers are equal";   //Statement3
         ?>
      <body>
</html>
```

OUTPUT of the above given Example is as follows:

```
numbers are equal
```

Switch Statement in PHP

- Switch statement selects one from multiple blocks of code to be executed

Syntax:

```
switch (n) {
        case label1:
                code to be executed if n=label1;
                break;
        case label2:
                code to be executed if n=label2;
                break;

        ...
        default:
                code to be executed if n is different from all labels;
}
```

Switch Statement Example

```
<html>
     <body>
          <?php
               $x=3;
               /* Expression value is compared with each case
               value. If it matches, statements following
               case would be executed. Break statement is
               used to terminate the execution of
               statement.*/
               switch ($x)
               {
                 case 1:
                          echo "Number 1";
                          break;
                 case 2:
                          echo "Number 2";
                          break;
                 case 3:
                          echo "Number 3";
                          break;
                 default:
                          echo "No number between 1 and 3";
               }
          ?>
     </body>
</html>
```

OUTPUT of the above given Example is as follows:

Number 3

For loop in PHP

- PHP for loop executes a block of code, a specified number of times

Syntax:

```
for (initialization; test condition; increment/decrement) {
        code to be executed;
}
```

For loop Example

```
<html>
      <body>
            <?php

                  echo "Numbers from 1 to 20 are: <br>";

                  /*in for loop, initialization usually declares
                  a loop variable, condition is a Boolean
                  expression such that if the condition is true,
                  loop body will be executed and after each
                  iteration of loop body, expression is executed
                  which usually increase or decrease loop
                  variable*/

                  for ($x=0; $x<=20; $x++) {
                        echo "$x  ";
                  }
            ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

Numbers from 1 to 20 are:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Declaring multiple variables in for loop Example

```
<html>
     <body>
          <?php

               /* Multiple variables can be declared in
                declaration block of for loop */

               for ($x=0,$y=1,$z=2;$x<=3;$x++) {
                    echo "x = $x,  y = $y,  z = $z <br>";
               }
          ?>
     </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
x = 0, y = 1, z = 2
x = 1, y = 1, z = 2
x = 2, y = 1, z = 2
x = 3, y = 1, z = 2
```

While Loop in PHP

- While loop, loops through a block of code as long as the specified condition is true

Syntax:

```
while (condition) {
    code to be executed;
}
```

While Loop Example

```
<html>
      <body>
            <?php
                  $i=1;

                  /* here <condition> is a Boolean expression.
                  Loop body is executed as long as condition is
                  true*/

                  while($i<5){
                        echo "i is = $i <br>";
                        $i++;
                  }
            ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

Do While loop in PHP

- Do while loop will always execute the block of code once, it will then check the condition, and if the condition is true then it repeats the loop

Syntax:

```
do {
    code to be executed;
} while (condition );
```

Do While loop Example

```
<html>
   <body>
         <?php
               $i=1;

               /* here <condition> is a Boolean expression. Please
               note that the condition is evaluated after executing
               the loop body. So loop will be executed at least
               once even if the condition is false*/

               do
               {
                     echo "i is = $i <br>";
                     $i++;
               }while($i<5);
         ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
i is = 1
i is = 2
i is = 3
i is = 4
```

User Defined Function in PHP

- Functions are group of statements that can perform a task

Syntax:

```
function functionName() {
        code to be executed;
}
```

User Defined Function Example

```
<html>
    <body>
         <?php

              // Function definition
              function myFunction()
              {
                    echo "Hello world";
              }

              // Function call
              myFunction();
         ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
Hello world
```

Swap Numbers PHP Example

```
<html>
   <body>
         <?php
               $num1=10;
               $num2=20;
               echo "Numbers before swapping:<br/>";
               echo "Num1=".$num1;
               echo "<br/>Num2=".$num2;

               // Function call
               swap($num1,$num2);

               // Function definition
               function swap($n1,$n2)
               {
                     $temp=$n1;
                     $n1=$n2;
                     $n2=$temp;
                     echo "<br/><br/>Numbers after
                                 swapping:<br/>";
                     echo "Num1=".$n1;
                     echo "<br/>Num2=".$n2;
               }
         ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
Numbers before swapping:
Num1=10
Num2=20

Numbers after swapping:
Num1=20
Num2=10
```

PHP Functions - Adding parameters

```
<html>
   <body>
         <?php
               // Function definition
               function writeName($fname)
               {
                     echo $fname . " Refsnes.<br />";
               }

               echo "My name is ";
               writeName("Kai Jim"); //Function call

               echo "My sister's name is ";
               writeName("Hege"); // Function call

               echo "My brother's name is ";
               writeName("Stale"); // Function call

         ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes.
My brother's name is Stale Refsnes.
```

PHP Functions - Return values

```
<html>
   <body>
         <?php
               // Function definition
               function add($x,$y)
               {
                     $total=$x+$y;
                     return $total;
               }
                                      // Function call
               echo "1 + 16 = " . add(1,16);
         ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

1 + 16 = 17

Break statement

- Break statement is used to terminate the loop
- After the break statement is executed the control goes to the statement immediately after the loop containing break statement

Break statement example

```
<html>
      <body>
            <?php
                  /* when $i value becomes 3, the loop is
                  Terminated*/
                  for($i=0;$i<5;$i++)
                  {
                        if($i==3)
                              break;
                        else
                              echo "$i ";
                  }
            ?>
      </body>
</html>
```

OUTPUT of the above given Example is as follows:
0 1 2

Continue statement

- There are cases in which, rather than terminating a loop, you simply want to skip over the remainder of iteration and immediately skip over to the next. Continue statement is used to skip a particular iteration of the loop.

Continue statement example

```
<html>
     <body>
          <?php
               /* when $i value becomes 3, it will skip the
               particular of the loop*/

               for($i=0;$i<=5;$i++)
               {
                    if($i==3)
                         continue;
                    else
                         echo "$i ";
               }
          ?>
     </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
0 1 2 4 5
```

PHP Global Variables - Superglobals

- "Superglobals" are predefined variables in PHP
- They are always accessible, regardless of scope - and can access them from any function, class or file

The PHP superglobal variables are:

$GLOBALS
$_SERVER
$_REQUEST
$_POST
$_GET
$_FILES
$_ENV
$_COOKIE
$_SESSION

$GLOBALS

- $GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods)
- PHP stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable

Example:

```
<html>
   <body>
        <?php
           $a = 20;
           $b = 40;
           function addition()
           {
              $GLOBALS['c'] = $GLOBALS['a'] + $GLOBALS['b'];
           }
           addition();
           echo $c;
        ?>
    </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
60
```

$_SERVER

- $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations

Example

```
<html>
  <body>
   <?php
     echo $_SERVER['PHP_SELF'];
     echo "<br>";
     echo $_SERVER['SERVER_NAME'];
     echo "<br>";
     echo $_SERVER['HTTP_HOST'];
     echo "<br>";
      echo $_SERVER['HTTP_USER_AGENT'];
      echo "<br>";
      echo $_SERVER['SCRIPT_NAME'];
   ?>
  </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
/User/server.php
localhost
localhost
Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; GTB7.5; SLCC2;
.NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC
6.0; InfoPath.2; .NET CLR 1.1.4322)
/User/server.php
```

Array in PHP

- An array stores multiple values in one single variable
- In PHP, there are three kinds of arrays:
  - Numeric array
  - Associative array
  - Multidimensional array

Numeric Array in PHP

- Numeric array is an array with a numeric index

Numeric Array Example

```
<html>
   <body>
         <?php
               /* An array $flower_shop is created with three
               Values - rose, daisy,orchid */
               $flower_shop = array (
                                "rose",
                                "daisy",
                                "orchid"
               );
               /* Values of array $flower_shop is displayed based
               on index. The starting index of an array is Zero */
               echo "Flowers: ".$flower_shop[0].",
                     ".$flower_shop[1].", ".$flower_shop[2]."";
         ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
Flowers: rose, daisy, orchid
```

Associative array in PHP

- Associative array is an array where each ID key is associated with a value

Associative array Example

```
<html>
   <body>
         <?php
               /* Here rose, daisy and orchid indicates ID key and
               5.00, 4.00, 2.00 indicates their values respectively
               */
               $flower_shop = array (
                           "rose" => "5.00",
                           "daisy" => "4.00",
                           "orchid" => "2.00"
               );
               // Display the array values
               echo "rose costs
                           .$flower_shop['rose'].",daisy costs
                           ".$flower_shop['daisy'].",and orchild
                           costs ".$flower_shop['orchild']."";
         ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
rose costs 5.00,daisy costs 4.00,and orchild costs
```

Loop through an Associative Array

```
<html>
   <body>
         <?php
               $flower_shop=array("rose"=>"5.00",
                     "daisy"=>"4.00","orchid"=>"2.00");

               /* for each loop works only on arrays, and is used
                 to loop through each key/value pair in an array */
               foreach($flower_shop as $x=>$x_value) {
                     echo "Flower=" . $x .
                           ", Value=" . $x_value;
                     echo "<br>";
               }
         ?>
         </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
Flower=rose, Value=5.00
Flower=daisy, Value=4.00
Flower=orchid, Value=2.00
```

Multidimensional array in PHP

- Multidimensional array is an array containing one or more arrays

Multidimensional array Example

```
<html>
 <body>
  <?php
   /* Here $flower_shop is an array, where rose, daisy and orchid
   are the ID key which indicates rows and points to array which
   have column values. */
   $flower_shop = array(
    "rose" => array( "5.00", "7 items", "red" ),
    "daisy" => array( "4.00", "3 items", "blue" ),
    "orchid" => array( "2.00", "1 item", "white" ),
   );
   /* in the array $flower_shop['rose'][0], 'rose' indicates row
   and  '0' indicates column */
   echo "rose costs ".$flower_shop['rose'][0].
      ", and you  get ".$flower_shop['rose'][1].".<br>";

   echo "daisy costs ".$flower_shop['daisy'][0].
     ", and you get ".$flower_shop['daisy'][1].".<br>";

   echo "orchid costs ".$flower_shop['orchid'][0].
     ", and you get ".$flower_shop['orchid'][1].".<br>";
  ?>
 </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
rose costs 5.00, and you get 7 items.
daisy costs 4.00, and you get 3 items.
orchid costs 2.00, and you get 1 item.
```

PHP Forms

- Scripts will interact with their clients using one of the two HTTP methods. The methods are GET and POST
- When a form is submitted using the GET method, its values are encoded directly in the query string portion of the URL
- When a form is submitted using the POST method, its values will not be displayed the query string portion of the URL

The $_GET Function

- The built-in $_GET function is used to collect values from a form sent with method="get"
- Information sent from a form with the GET method is visible to everyone (it will be displayed in the browser's URL) and has limits on the amount of information to send (max. 100 characters)
- This method should not be used when sending passwords or other sensitive information. However, because the variables are displayed in the URL, it is possible to bookmark the page
- The get method is not suitable for large variable values; the value cannot exceed 100 characters

The $_POST Function

- The built-in $_POST function is used to collect values from a form sent with method="post"
- Information sent from a form with the POST method is invisible to others and has no limits on the amount of information to send
- However, there is an 8 Mb max size for the POST method, by default (can be changed by setting the post_max_size in the php.ini file)

The $_GET Function Example

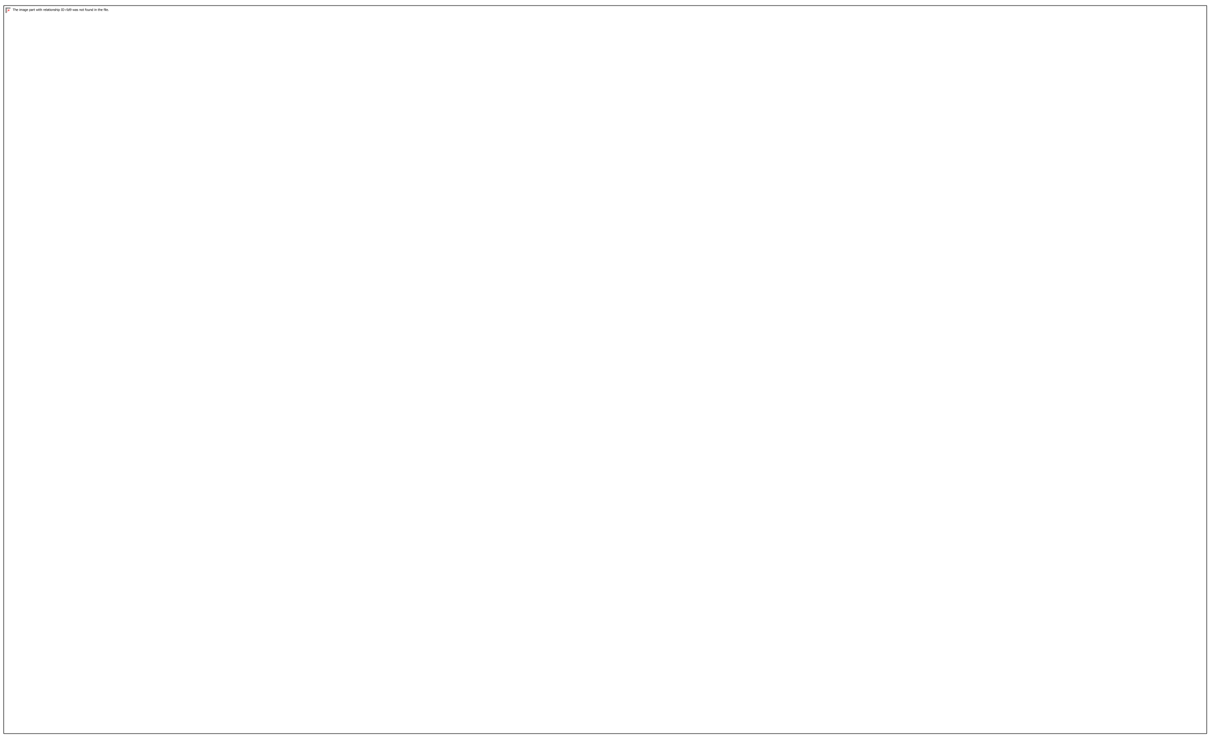
Form1.html

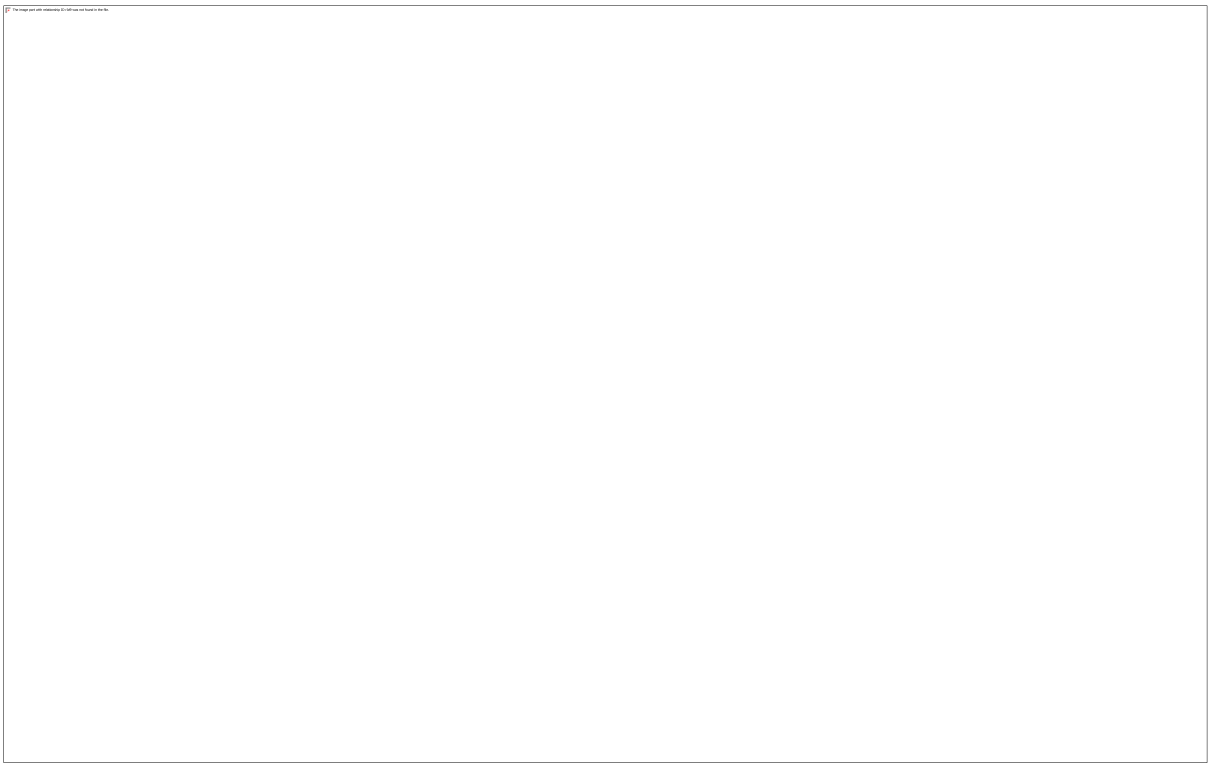
```
<html>
   <body>
         /* form submitted using ‘get’ method, action specifies
         next page which is to be loaded when button is clicked*/
         <form action="welcome.php" method="get">
                     // textbox is to take user input
               Name: <input type="text" name="fname" />
               Age: <input type="text" name="age" />
               // Submit button is to submit the value
               <input type="submit" />
         </form>
   </body>
</html>
```

welcome.php

```
<html>
   <body>
                  // $_GET to receive the data sent from Form1.html
         Welcome <?php echo $_GET["fname"]; ?>.<br />
         You are <?php echo $_GET["age"]; ?> years old!
   </body>
</html>
```

OUTPUT of the above given Example is as follows:





In this example, when you click on the “submit” button, the values entered in the textbox are encoded directly in the query string portion of the URL.

The $_ POST Function Example

form1.html

```
<html>
   <body>
         /* form submitted using ‘post’ method, action specifies
         next page which is to be loaded when button is clicked */
         <form action="welcome1.php" method="post">
                      // textbox is to take user input
               Name: <input type="text" name="fname" />
               Age: <input type="text" name="age" />
               // Submit button is to submit the value to next page
               <input type="submit" />
         </form>
   </body>
</html>
```

welcome1.php

```
<html>
   <body>
                  // $_GET to receive the data sent from form1.html
         Welcome <?php echo $_POST["fname"]; ?>.<br />
         You are <?php echo $_POST["age"]; ?> years old!
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

The image part with relationship ID rId9 was not found in the file.

The image part with relationship ID rId9 was not found in the file.

In this example, when you click on the "submit" button, the values will not be displayed the query string portion of the URL.

Another Example for PHP form

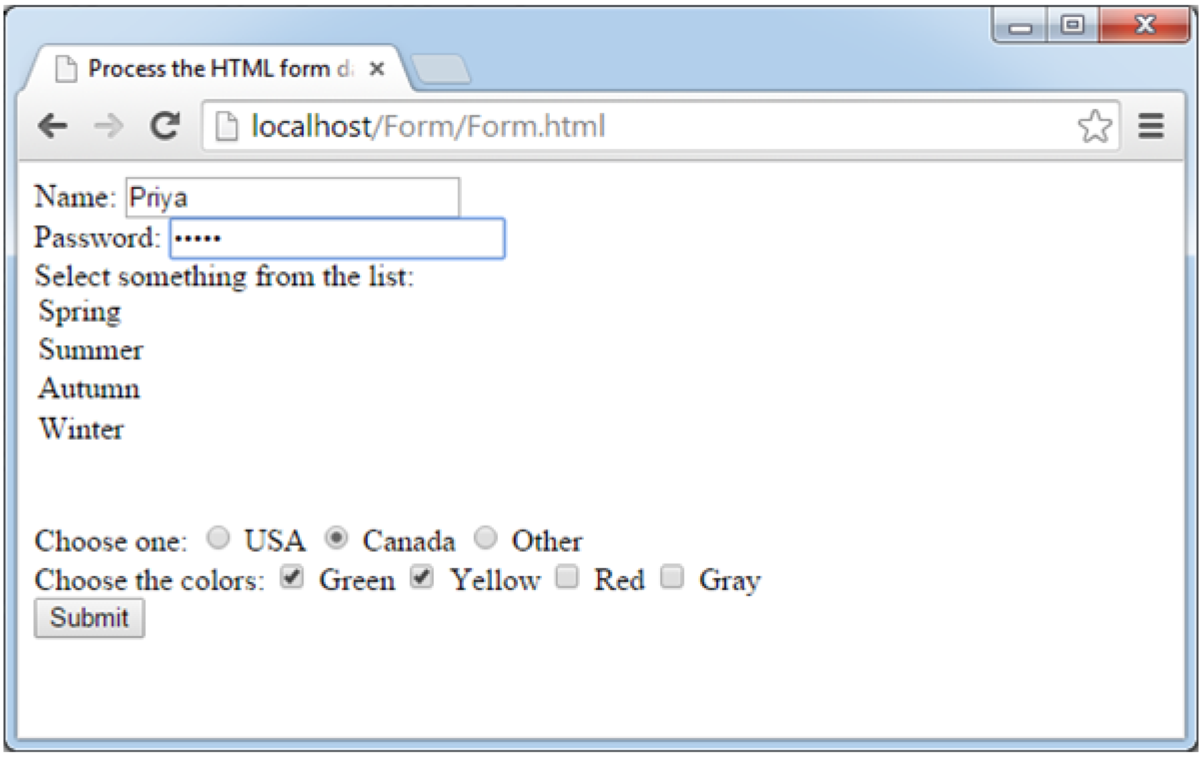
Form.html

```
<html>
 <head>
  <title>Process the HTML form data with the POST
         method</title>
 </head>
 <body>
   /* form submitted using ‘post’ method, action specifies next page
   which is to be loaded when button is clicked */
  <form name="myform" action="process.php" method="POST">

   // create an hidden textbox
   <input type="hidden" name="check_submit" value="1" />

      // textbox is to take user input
      Name: <input type="text" name="Name" /><br />
      Password: <input type="password" name="Password"
           maxlength="10" /><br />

      // Use ‘select’ tag to display the various options
      Select something from the list: <select name="Seasons">
      <option value="Spring"
        selected="selected">Spring</option>
      <option value="Summer">Summer</option>
      <option value="Autumn">Autumn</option>
      <option value="Winter">Winter</option>
      </select><br /><br />

      Choose one:
      //This will create radio buttons
      <input type="radio" name="Country" value="USA" /> USA
      <input type="radio" name="Country" value="Canada" />
       Canada
      <input type="radio" name="Country" value="Other" />
        Other
      <br />

      Choose the colors:
      //This will create checkbox
      <input type="checkbox" name="Colors[]" value="green"
       checked="checked" /> Green
      <input type="checkbox" name="Colors[]" value="yellow"
        /> Yellow
      <input type="checkbox" name="Colors[]" value="red" />
        Red
      <input type="checkbox" name="Colors[]" value="gray" />
        Gray
      <br />

        // Submit button is to submit the value to next page
        <input type="submit" />
      </form>
   </body>
</html>
```

Process.php

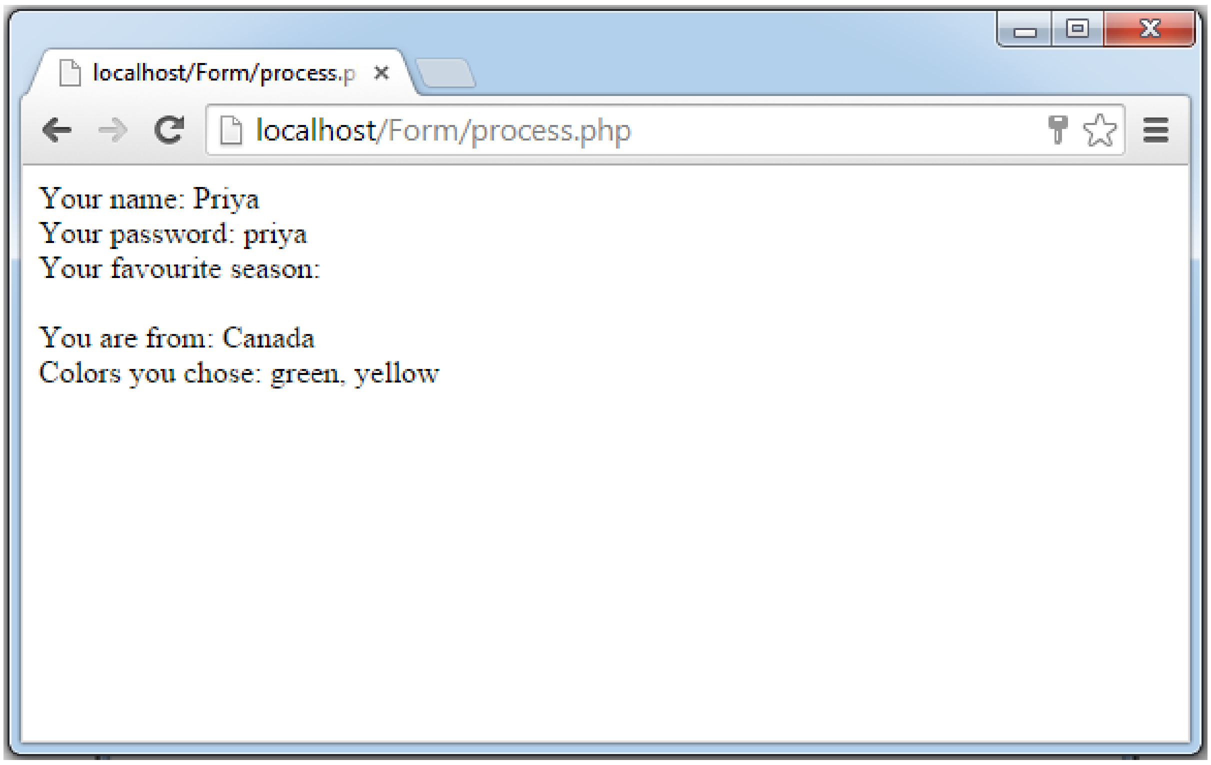
```
<html>
 <body>
  <?php
   if (array_key_exists('check_submit', $_POST)) {
    /*Converts the new line characters (\n) in the text
      area into HTML line breaks (the <br /> tag) */
    $_POST['Comments'] = nl2br($_POST['Comments']);
    //Check whether a $_GET['Languages'] is set
   if ( isset($_POST['Colors']) ) {
    $_POST['Colors'] = implode(', ', $_POST['Colors']);
    //Converts an array into a single string
   }

   //Let's now print out the received values in the browser
   echo "Your name: {$_POST['Name']}<br />";
   echo "Your password: {$_POST['Password']}<br />";
   echo "Your favourite season: {$_POST['Seasons']}
   <br/><br />";
   echo "You are from: {$_POST['Country']}<br />";
   echo "Colors you chose: {$_POST['Colors']}<br />";
  }
  else
  {
   echo "You can't see this page without submitting the
      form.";
  }
 ?>
</body>
</html>
```

OUTPUT of the above given Example is as follows:



Name: Priya
Password: •••••
Select something from the list:
Spring
Summer
Autumn
Winter

Choose one: USA [✓ Canada] Other
Choose the colors: [✓ Green] [✓ Yellow] Red Gray
Submit



```
Your name: Priya
Your password: priya
Your favourite season:

You are from: Canada
Colors you chose: green, yellow
```

Date() and time() function in PHP

- The PHP date() function formats a timestamp to a more readable date and time
- A timestamp is a sequence of characters, denoting the date and/or time at which a certain event occurred
- Some characters that are commonly used for date and time:
  - d - Represents the day of the month (01 to 31)
  - m - Represents a month (01 to 12)
  - Y - Represents a year (in four digits)
  - l (lowercase 'L') - Represents the day of the week
  - h - 12-hour format of an hour with leading zeros (01 to 12)
  - i - Minutes with leading zeros (00 to 59)
  - s - Seconds with leading zeros (00 to 59)
  - a - Lowercase Ante meridiem and Post meridiem (am or pm)

Example

```
<html>
   <body>
         <?php

               // display the date in the format YYYY/MM/DD
               echo "Today is " . date("Y/m/d") . "<br>";
               // ‘l’ is used to display the day
               echo "Today is " . date("l"). "<br>";
               // display the time in the format HH:MM:SS
               echo "The time is " . date("h:i:sa");

         ?>
   </body>
</html>
```

OUTPUT of the above given Example is as follows:

```
Today is 2014/08/19
Today is Tuesday

The time is 09:13:22am
```

How to connect to MYSQL database using PHP

To connect MYSQL using PHP go to: http://localhost//phpmyadmin

Enter the username and password

Give the database name in the field ‘create new database’

The image part with relationship ID rId9 was not found in the file.

Click on create button

Create a new table in the database by giving a table name and number of fields then click on Go

The image part with relationship ID rId9 was not found in the file.

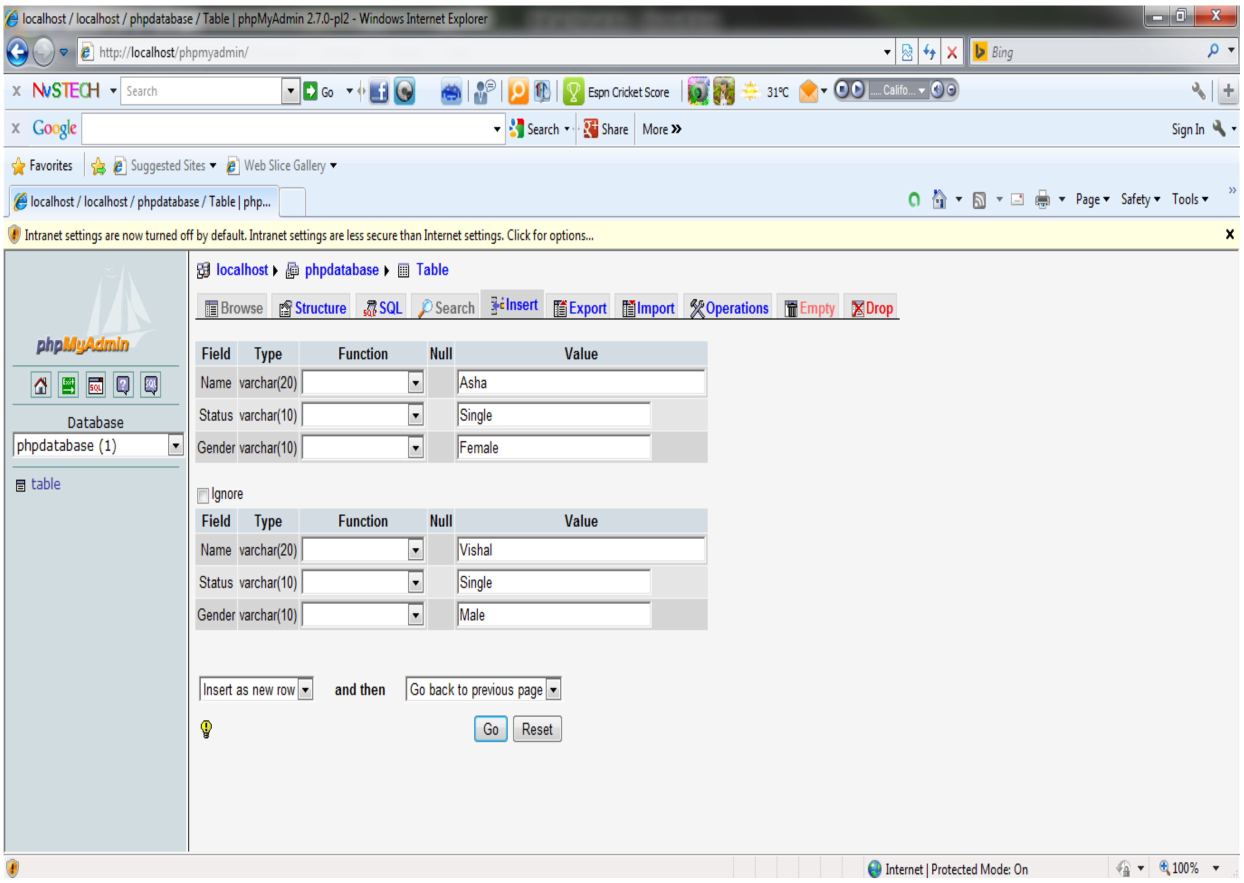
To give field name to the created table, write the field name in the 'field' column, select the data types for each fields, specify the length of each field then click on save to save the fields and click on Go

The image part with relationship ID rId9 was not found in the file.

When clicked on Go the table field details will be displayed

The image part with relationship ID rId9 was not found in the file.

To insert values in the field, go to insert and enter the values. Then click on Go



Field	Type	Function	Null	Value
Name	varchar(20)			Asha
Status	varchar(10)			Single
Gender	varchar(10)			Female

Ignore

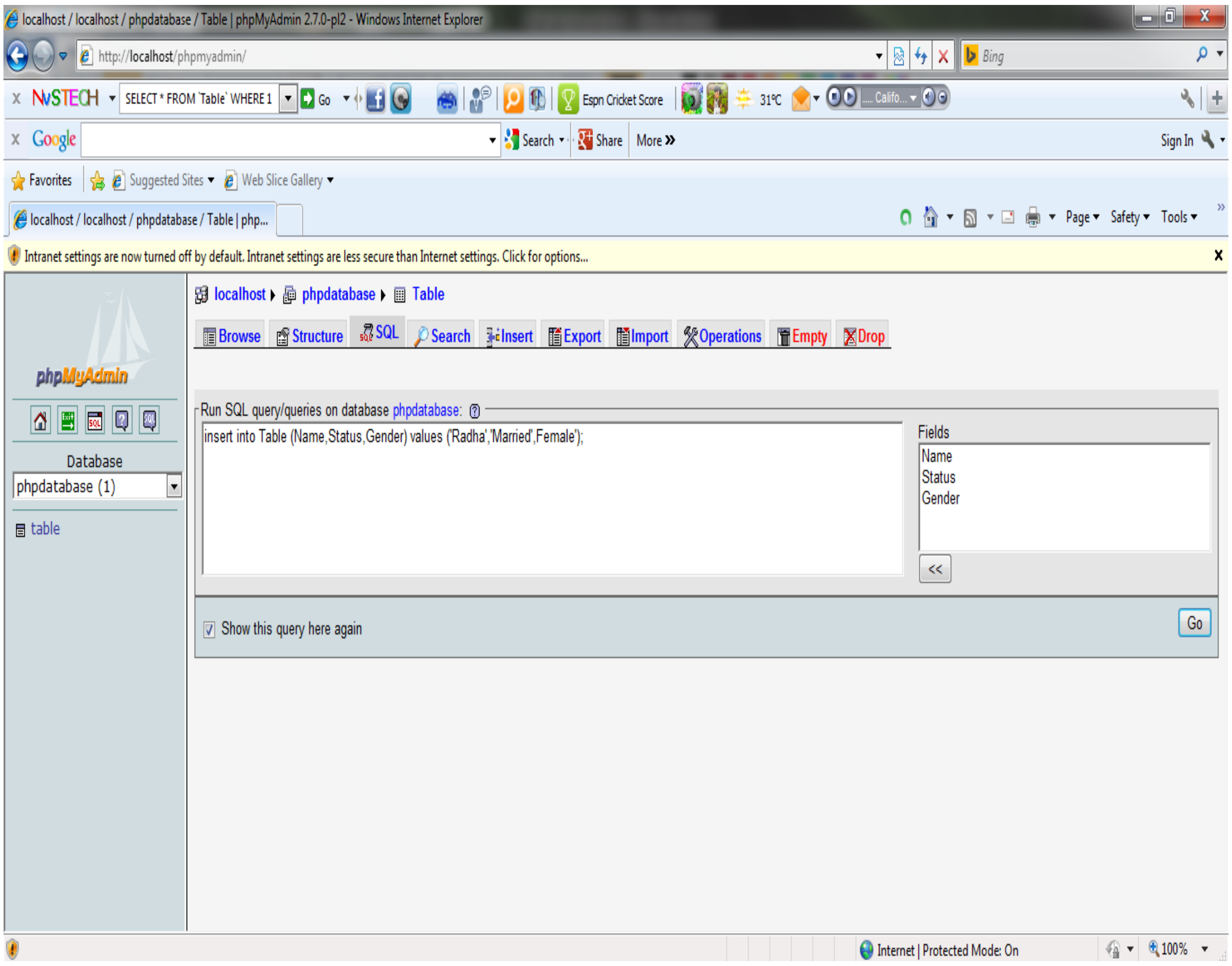
Field	Type	Function	Null	Value
Name	varchar(20)			Vishal
Status	varchar(10)			Single
Gender	varchar(10)			Male

Insert as new row and then Go back to previous page

localhost / localhost / phpdatabase / Table | phpMyAdmin 2.7.0-pl2 - Windows Internet Explorer
http://localhost/phpmyadmin/
localhost ▸ phpdatabase ▸ Table
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Database
phpdatabase (1)
table
Go
Reset
Internet | Protected Mode: On
100%

To view the created table, go to browse

To insert the values, go to SQL and write the query to insert the values and click on Go


localhost / localhost / phpdatabase / Table | phpMyAdmin 2.7.0-pl2 - Windows Internet Explorer
http://localhost/phpmyadmin/
localhost ▸ phpdatabase ▸ Table
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Database
phpdatabase (1)
table
Fields
Name
Status
Gender
Show this query here again
Go

Run SQL query/queries on database phpdatabase:

```
insert into Table (Name,Status,Gender) values ('Radha','Married','Female');
```

SQL query for insert:

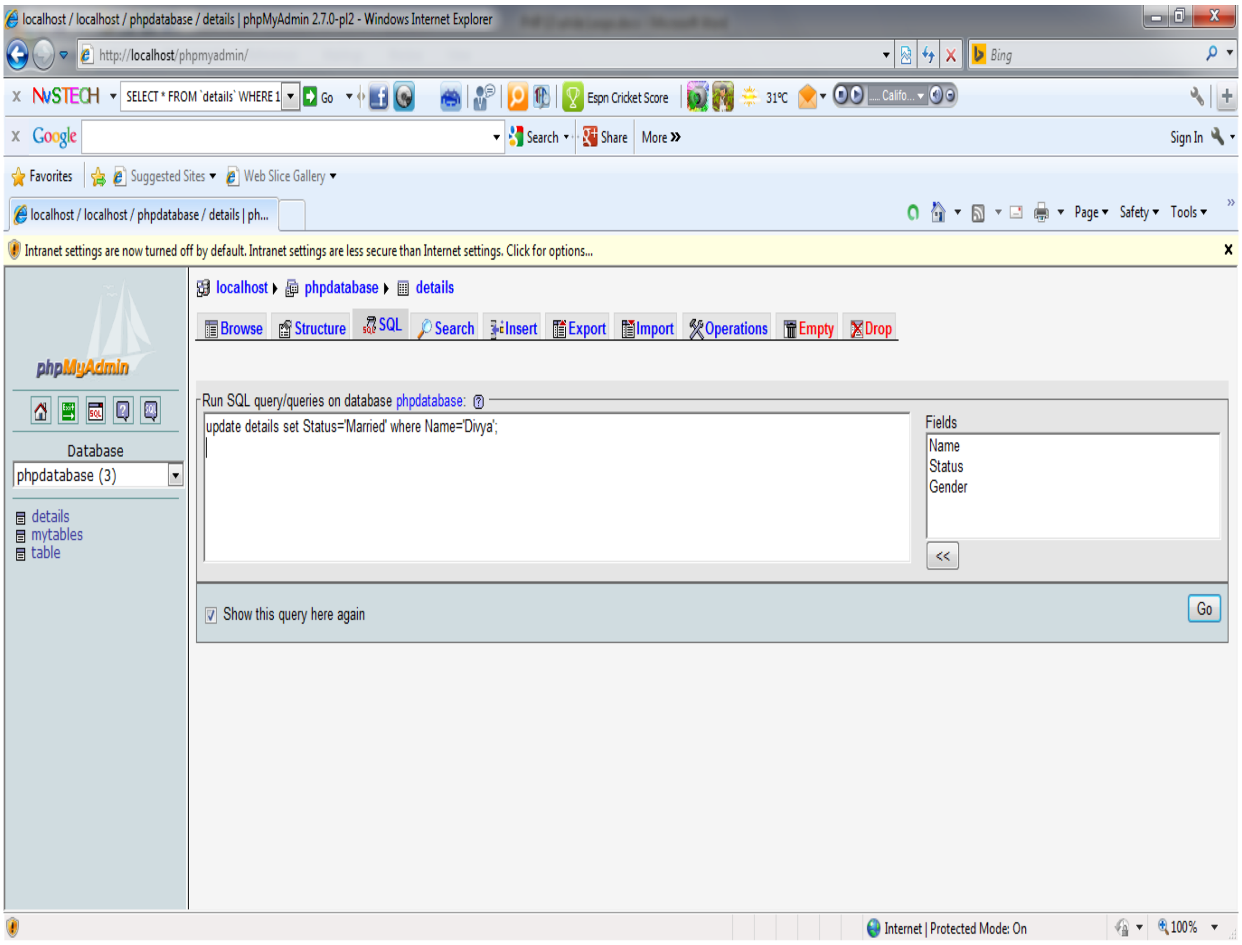
Syntax:

Insert into table_name values('value1','value2',…);

Example:

Insert into Login values('Radha','hello');

To update the values, go to SQL and write the query to update the values and click on Go


localhost / localhost / phpdatabase / details | phpMyAdmin 2.7.0-pl2 - Windows Internet Explorer
http://localhost/phpmyadmin/
localhost ▸ phpdatabase ▸ details
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Database
phpdatabase (3)
details
mytables
table
Fields
Name
Status
Gender
<<
Go
Internet | Protected Mode: On

Run SQL query/queries on database phpdatabase:

```
update details set Status='Married' where Name='Divya';
```

☑ Show this query here again

SQL query for update:

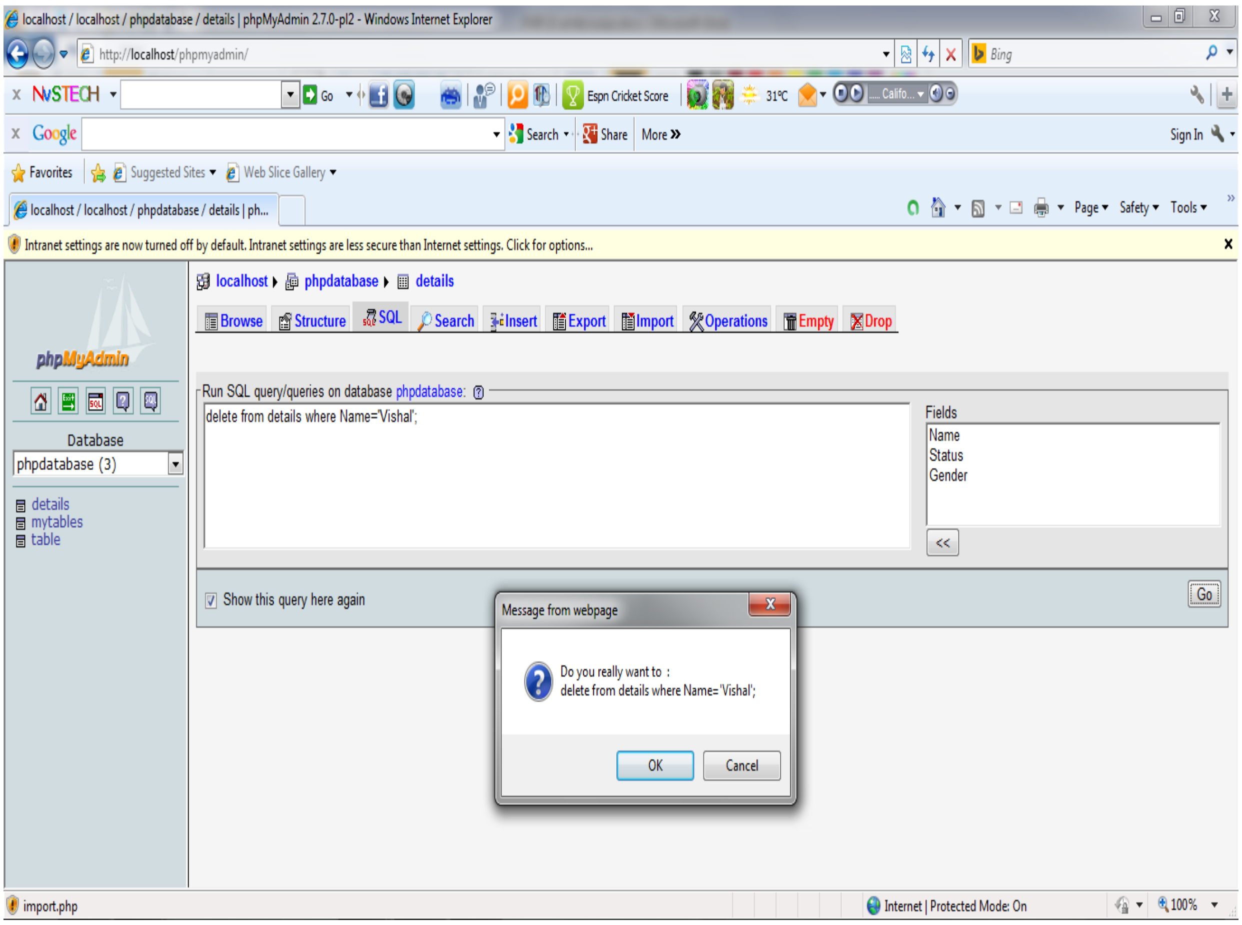
Syntax:

Update table_name set field_name=’value’ where field_name=’value’;

Example:

Update Login set password=’abcde’ where name=’Radha’;

To delete the values, go to SQL and write the query to delete the values and click on go


localhost / localhost / phpdatabase / details | phpMyAdmin 2.7.0-pl2 - Windows Internet Explorer
http://localhost/phpmyadmin/
localhost ▸ phpdatabase ▸ details
Browse
Structure
SQL
Search
Insert
Export
Import
Operations
Empty
Drop
Database
phpdatabase (3)
details
mytables
table
Fields
Name
Status
Gender
Show this query here again
Go
OK
Cancel

Run SQL query/queries on database phpdatabase:

```
delete from details where Name='Vishal';
```

Message from webpage

Do you really want to :
delete from details where Name='Vishal';

SQL query for delete:

Syntax:

Delete from table_name where field_name='value';

Example:

Delete from Login where name='Radha';

The functions used to connect web form to the MYSQL database:

mysql_connect():

This function opens a link to a MySQL server on the specified host (in this case it's localhost) along with a username (root) and password (q1w2e3r4/). The result of the connection is stored in the variable $db.

mysql_select_db():

This tells PHP that any queries we make are against the mydb database.

mysql_query():

Using the database connection identifier, it sends a line of SQL to the MySQL server to be processed. The results that are returned are stored in the variable $result.

mysql_result():

This is used to display the values of fields from our query. Using $result, we go to the first row, which is numbered 0, and display the value of the specified fields.

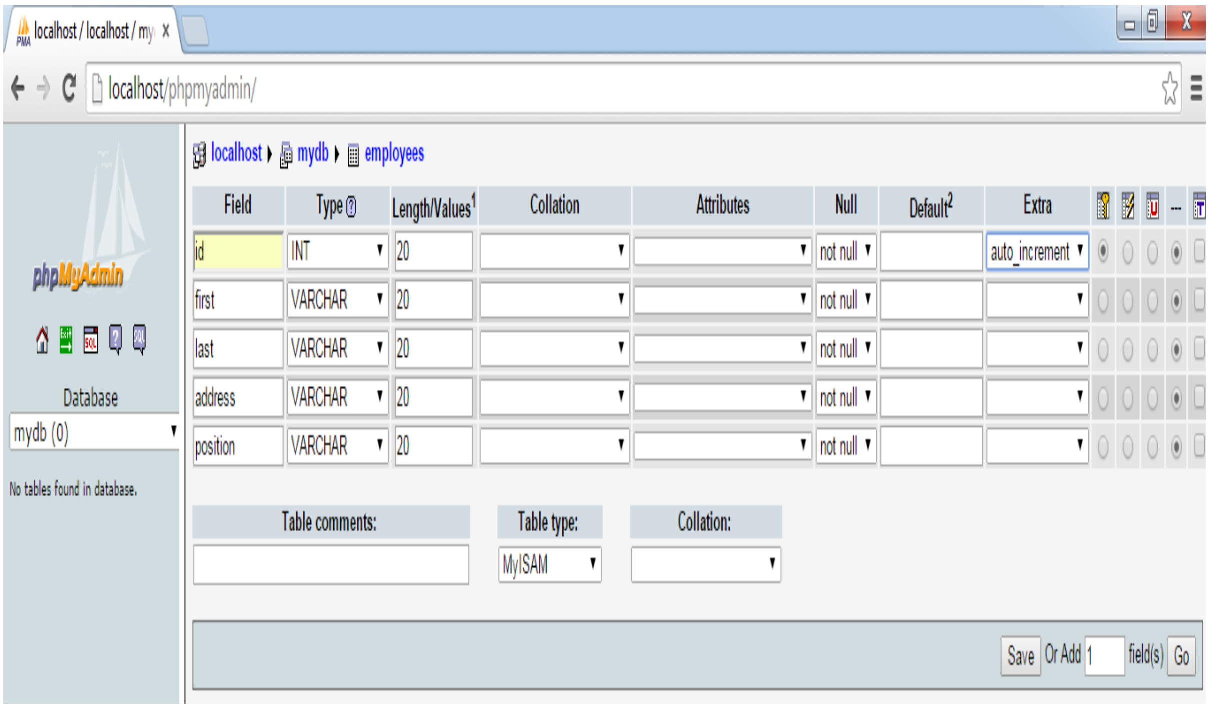
mysql_result($result,0,"position")):

This should be treated as a string and printed.

Display the data from MYSQL database in web form

```
<html>
 <body>
  <?php
     // Open MYSQL server connection
     $db = mysql_connect("localhost", "root","q1w2e3r4/");
     // Select the database using MYSQL server connection
     mysql_select_db("mydb",$db);
     /* Using the database connection identifier, it sends
        a line of SQL to the MySQL server to be processed
        and the results are stored in the variable
        $result. */
     $result = mysql_query("SELECT * FROM employees",$db);
     // Displaying the details in a table
     echo "<table border=1>";
     echo "<tr><th>Name</th><th>Position</th></tr>";
     while ($myrow = mysql_fetch_row($result)) {
           printf("<tr><td>%s %s</td><td>%s</td></tr>",
           $myrow[1], $myrow[2],$myrow[4]);
     }
     echo "</table>";
  ?>
 </body>
</html>
```

OUTPUT of the above given Example would be:



localhost/phpmyadmin/

localhost ▸ mydb ▸ employees

Database: mydb (0)

No tables found in database.

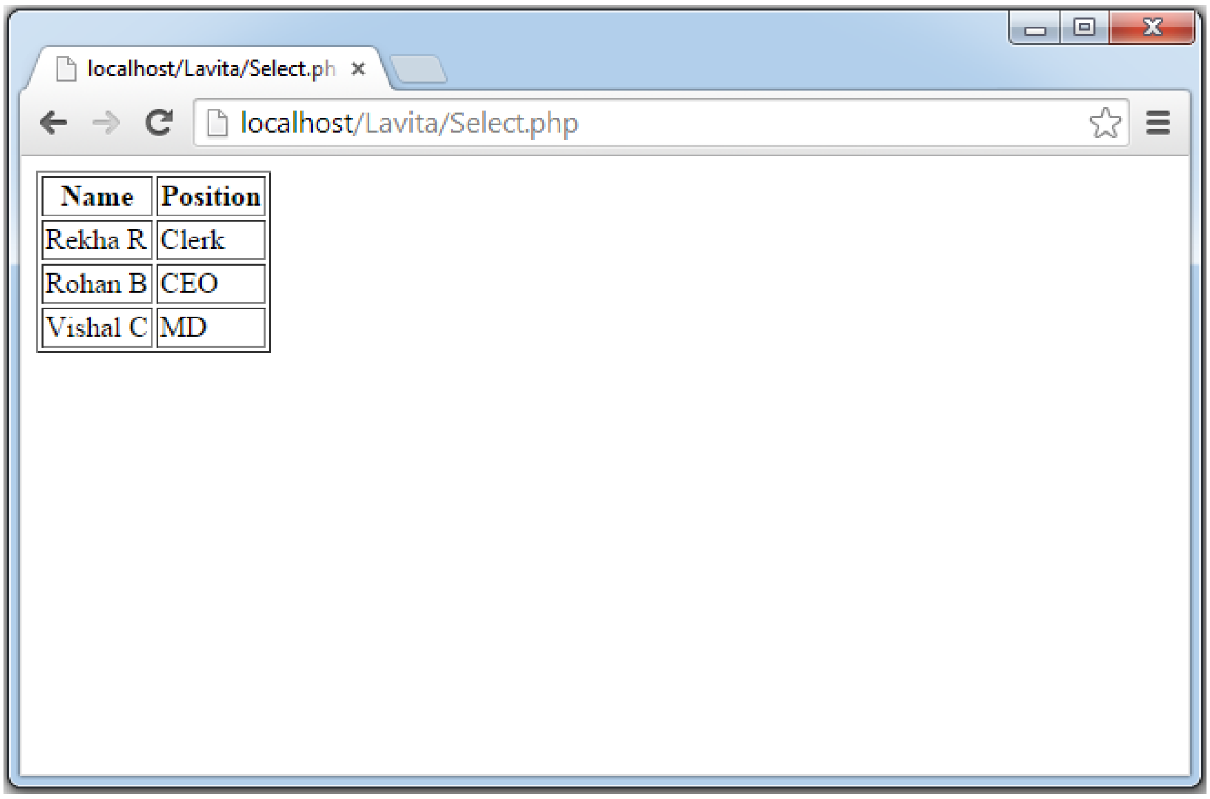
Field	Type	Length/Values[1]	Collation	Attributes	Null	Default[2]	Extra					
id	INT	20			not null		auto_increment	●	○	○	○	☐
first	VARCHAR	20			not null			○	○	○	●	☐
last	VARCHAR	20			not null			○	○	○	●	☐
address	VARCHAR	20			not null			○	○	○	●	☐
position	VARCHAR	20			not null			○	○	○	●	☐

Table comments:

Table type: MyISAM

Collation:

Save Or Add 1 field(s) Go



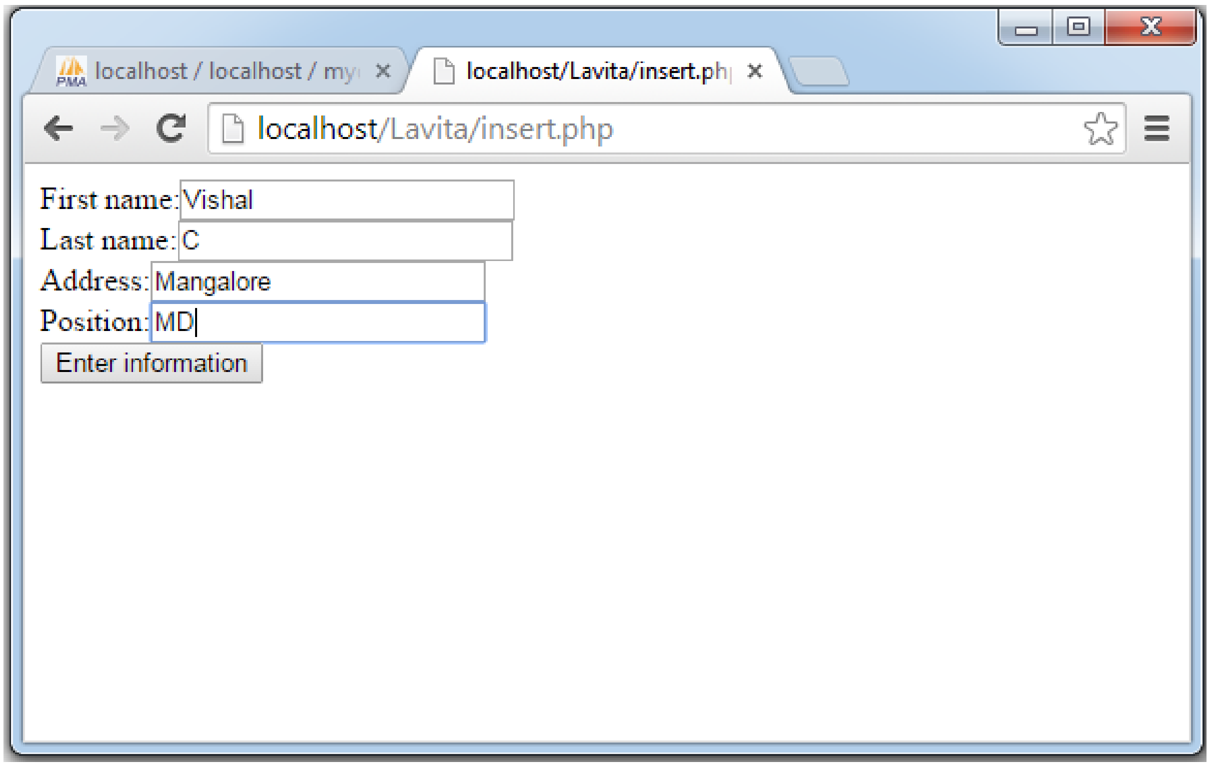
localhost/Lavita/Select.php

Name	Position
Rekha R	Clerk
Rohan B	CEO
Vishal C	MD

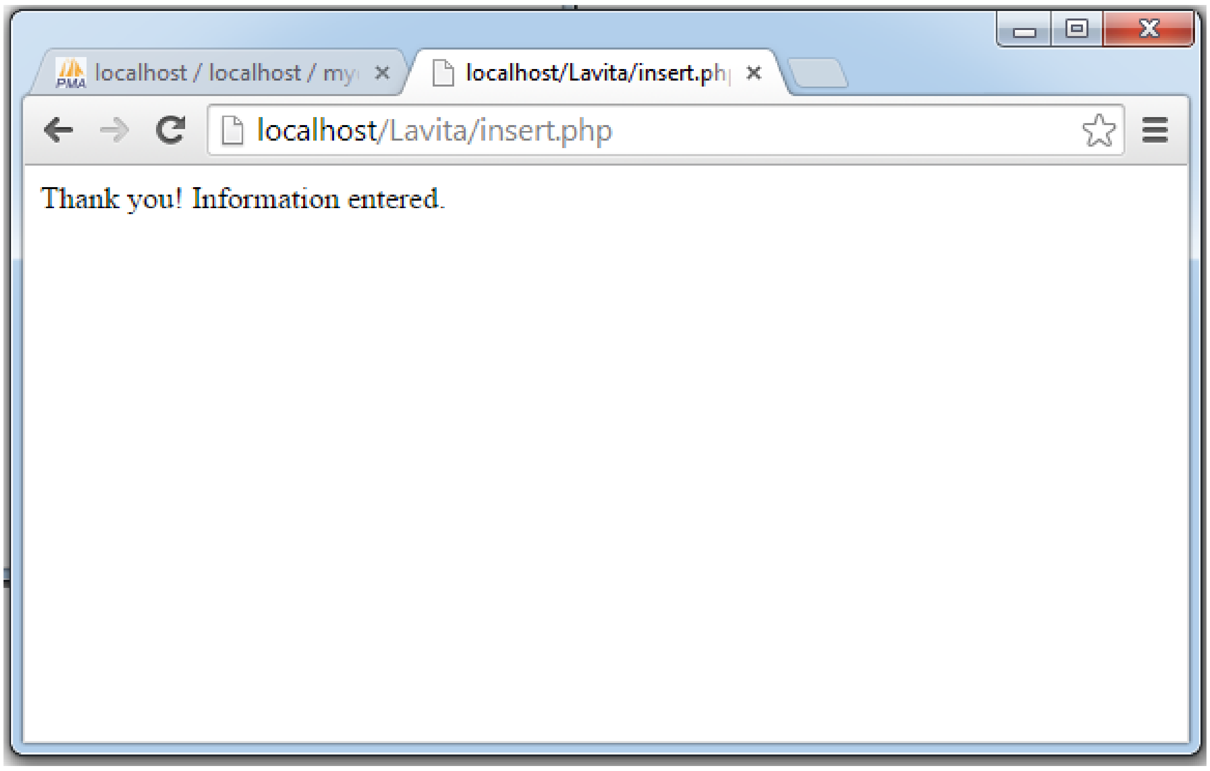
Insert the data into MYSQL database using web form

```
<html>
 <body>
  <?php
    if ($submit) {
     // Open MYSQL server connection
     $db = mysql_connect("localhost", "root","q1w2e3r4/");
     // Select the database using MYSQL server connection
     mysql_select_db("mydb",$db);
     /* Write insert query and assign the query in $sql
     Variable */
     $sql = "INSERT INTO employees (first,last,address,position)
      VALUES('$first','$last','$address','$position')";
     // Execute the query
     $result = mysql_query($sql);
     echo "Thank you! Information entered.";
    }
    else
    {
      // display form
 ?>
     <form method="post" action="<?php echo $PHP_SELF?>">
      First name:<input type="Text" name="first"><br>
      Last name:<input type="Text" name="last"><br>
      Address:<input type="Text" name="address"><br>
      Position:<input type="Text" name="position"><br>
      <input type="Submit" name="submit" value="Enter
        information">
     </form>
  <?php
    } // end if
  ?>
 </body>
</html>
```

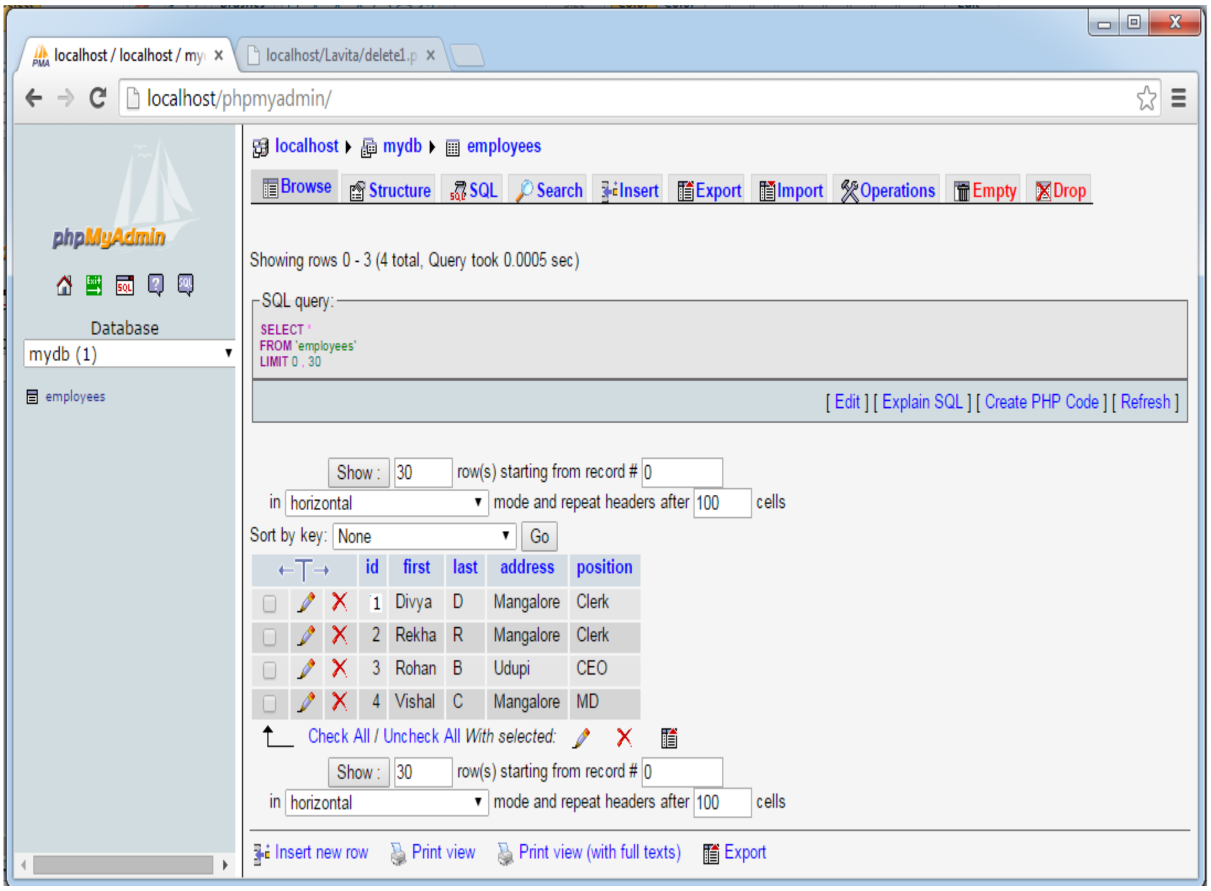
OUTPUT of the above given Example would be:


localhost/Lavita/insert.php

First name: Vishal
Last name: C
Address: Mangalore
Position: MD
Enter information


localhost/Lavita/insert.php

Thank you! Information entered.



Showing rows 0 - 3 (4 total, Query took 0.0005 sec)

SQL query:

```
SELECT *
FROM `employees`
LIMIT 0 , 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None Go

id	first	last	address	position
1	Divya	D	Mangalore	Clerk
2	Rekha	R	Mangalore	Clerk
3	Rohan	B	Udupi	CEO
4	Vishal	C	Mangalore	MD

Check All / Uncheck All *With selected:*

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Insert new row   Print view   Print view (with full texts)   Export

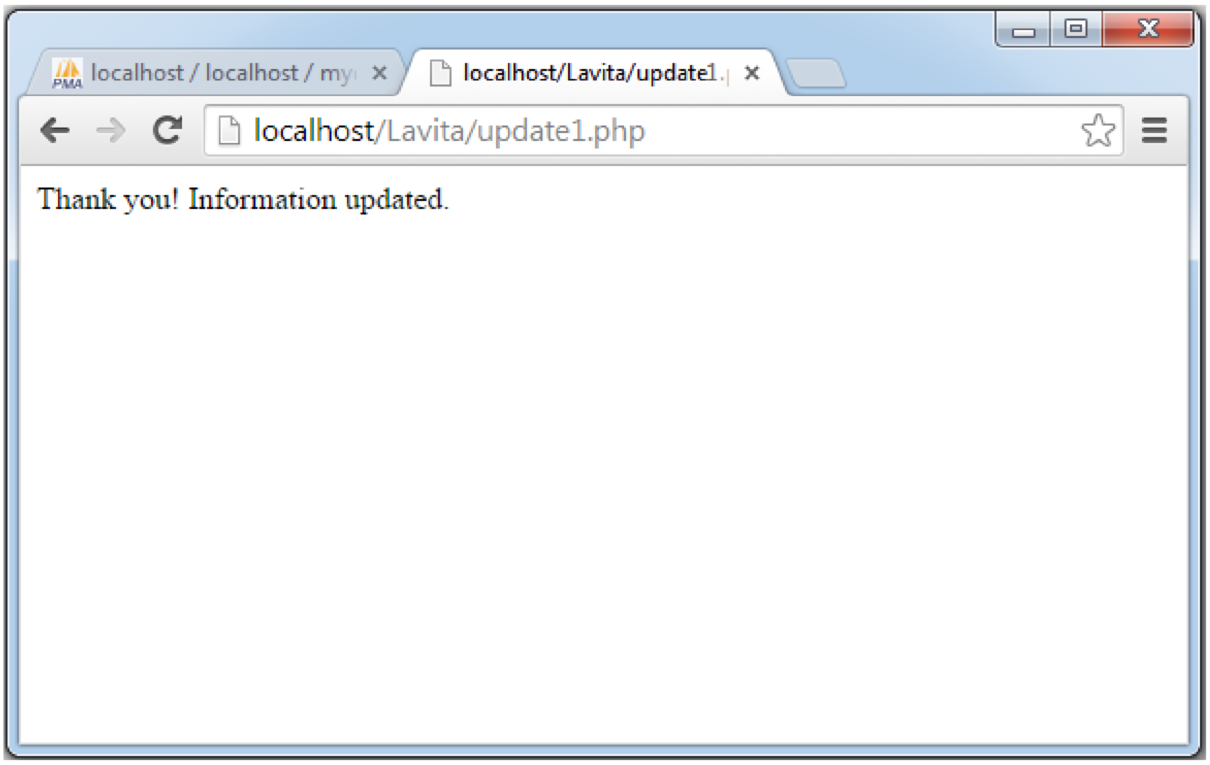
Update the data present in MYSQL database using web form

```
<html>
 <body>
  <?php
    // Open MYSQL server connection
    $db = mysql_connect("localhost", "root","q1w2e3r4/");
    // Select the database using MYSQL server connection
    mysql_select_db("mydb",$db);
    if ($id) {
      if ($submit) {
         // Write UPDATE query and assign to $sql Variable
        $sql = "UPDATE employees SET
                 first='$first', last='$last',
                 address='$address',
                position='$position'
                 WHERE id=$id";
          // Execute the query
          $result = mysql_query($sql);
         echo "Thank you! Information updated.";
     }
     else
     {

          // Write query to SELECT data from table
           $sql = "SELECT * FROM employees WHERE id=$id";
           // Execute the query
           $result = mysql_query($sql);
           // Fetch the values
            $myrow = mysql_fetch_array($result);
 ?>
     <form method="post" action="<?php echo $PHP_SELF?>">
      <input type=hidden name="id" value="<?php echo
        $myrow["id"] ?>">
     First name:<input type="Text" name="first"
        value="<?php echo $myrow["first"] ?>"><br>
     Last name:<input type="Text" name="last"
        value="<?php echo $myrow["last"] ?>"><br>
     Address:<input type="Text" name="address"
        value="<?php echo $myrow["address"]?>"><br>
     Position:<input type="Text" name="position"
        value="<?php echo $myrow["position"]?>"><br>
     <input type="Submit" name="submit" value="Enter
        information">
     </form>
  <?php
   }
  }
  else
  {
   // display list of employees
   $result = mysql_query("SELECT * FROM
                           employees",$db);
   while ($myrow = mysql_fetch_array($result)) {
      printf("<a href=\"%s?id=%s\">%s %s</a><br>",
         $PHP_SELF, $myrow["id"],$myrow["first"],
         $myrow["last"]);
   }
  }
 ?>
 </body>
</html>
```

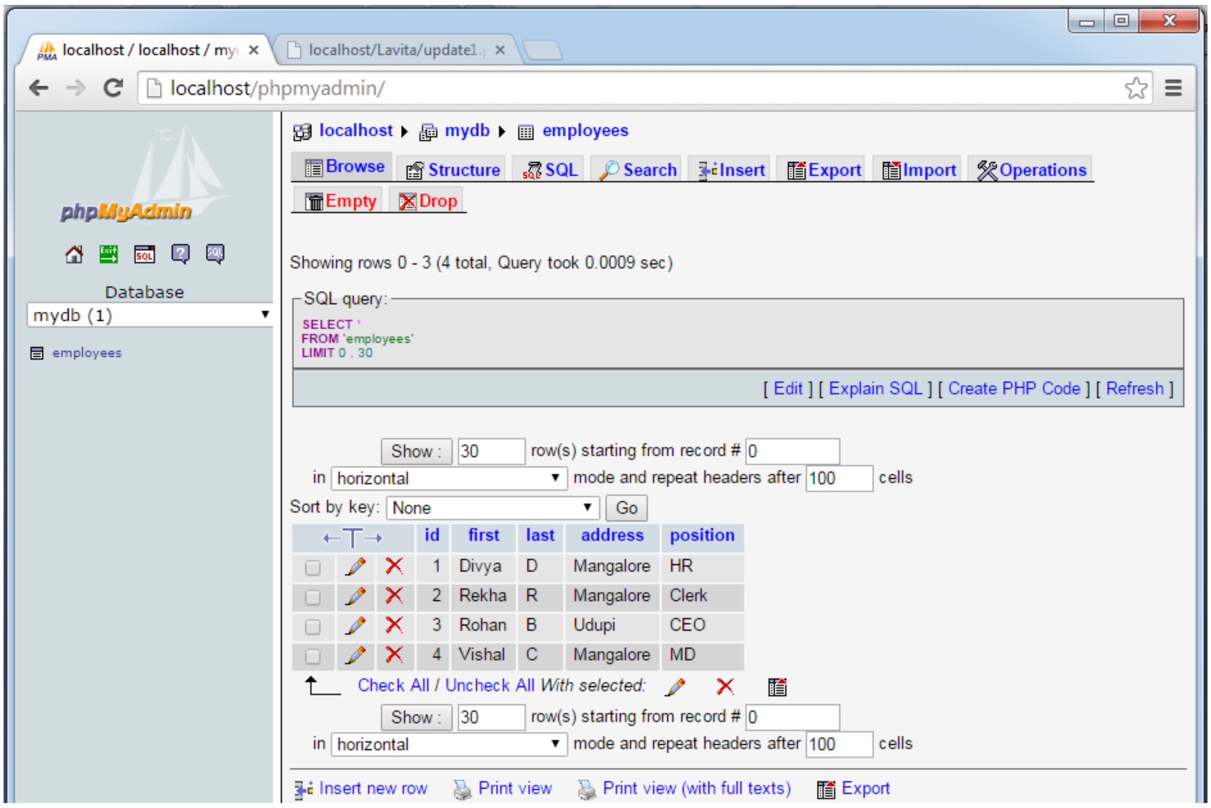
OUTPUT of the above given Example would be:

The image part with relationship ID rId9 was not found in the file.

The image part with relationship ID rId9 was not found in the file.


localhost / localhost / my
localhost/Lavita/update1.
localhost/Lavita/update1.php

Thank you! Information updated.



Showing rows 0 - 3 (4 total, Query took 0.0009 sec)

SQL query:

```
SELECT *
FROM `employees`
LIMIT 0 , 30
```

[Edit] [Explain SQL] [Create PHP Code] [Refresh]

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells
Sort by key: None Go

id	first	last	address	position
1	Divya	D	Mangalore	HR
2	Rekha	R	Mangalore	Clerk
3	Rohan	B	Udupi	CEO
4	Vishal	C	Mangalore	MD

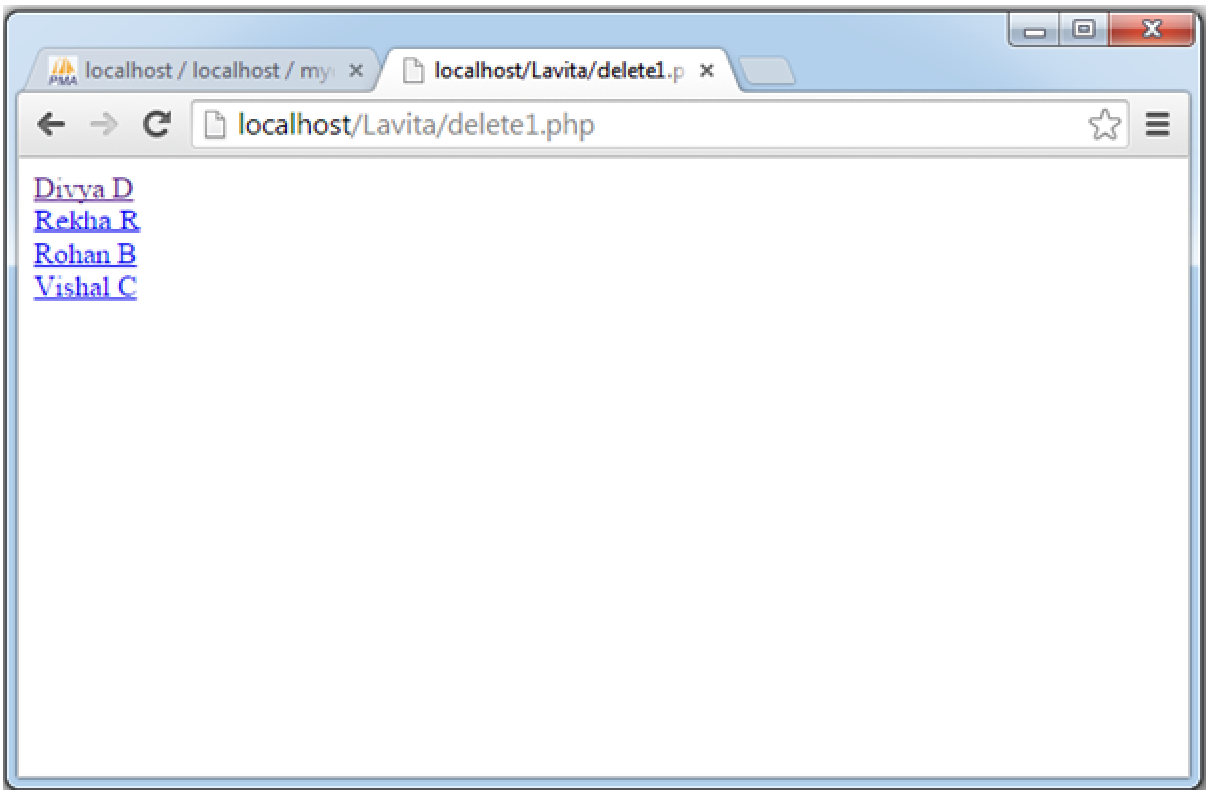
Check All / Uncheck All *With selected:*

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

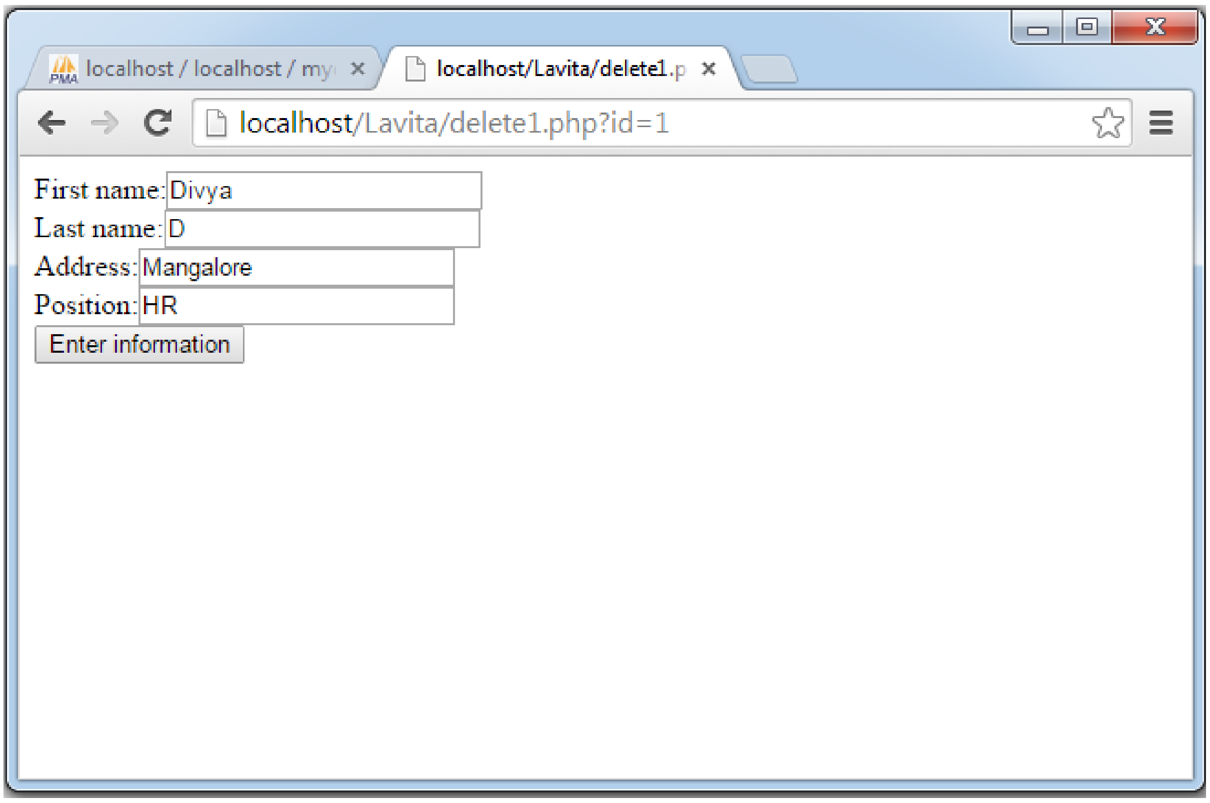
Delete the data from MYSQL database using web form

```
<html>
 <body>
  <?php
   // Open MYSQL server connection
   $db = mysql_connect("localhost", "root","q1w2e3r4/");
   // Select the database using MYSQL server connection
   mysql_select_db("mydb",$db);
   if ($id) {
    if ($submit) {
      // Write DELETE query to delete data from table based on ID
     $sql = "DELETE FROM employees WHERE id=$id";
     // Execute the query
     $result = mysql_query($sql);
      echo "Thank you! Information deleted.";
    }
    else
    {
     // Write SELECT query to select data from table based on ID
     $sql = "SELECT * FROM employees WHERE id=$id";
     $result = mysql_query($sql);
     $myrow = mysql_fetch_array($result);
  ?>
     <form method="post" action="<?php echo $PHP_SELF?>">
       <input type=hidden name="id"
             value="<?php echo $myrow["id"] ?>">
        First name:<input type="Text" name="first"
             readonly="readonly"
             value="<?php echo $myrow["first"] ?>"><br>
         Last name:<input type="Text" name="last"
             readonly="readonly"
             value="<?php echo $myrow["last"] ?>"><br>
         Address:<input type="Text" name="address"
            readonly="readonly"
            value="<?php echo $myrow["address"]?>"><br>
         Position:<input type="Text" name="position"
            value="<?php echo $myrow["position"]?>"><br>
         <input type="Submit" name="submit"
            value="Delete information">
     </form>
  <?php
   }
  }
  else
  {
     // display list of employees
     $result = mysql_query("SELECT * FROM
            employees",$db);
     while ($myrow = mysql_fetch_array($result)) {
        printf("<a href=\"%s?id=%s\">%s %s</a><br>",
           $PHP_SELF, $myrow["id"],$myrow["first"],
           $myrow["last"]);
    }
   }
  ?>
 </body>
</html>
```

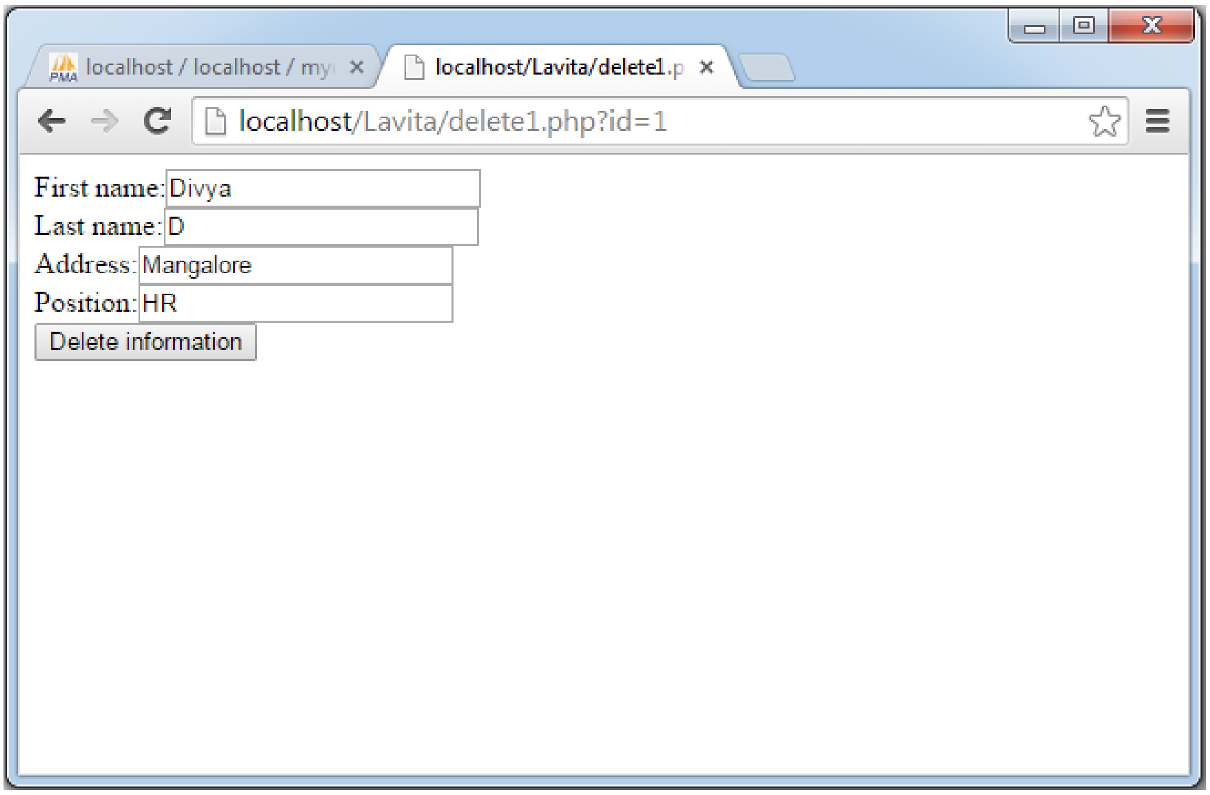
OUTPUT of the above given Example would be:


localhost / localhost / my
localhost/Lavita/delete1.p
localhost/Lavita/delete1.php

Divya D
Rekha R
Rohan B
Vishal C


localhost / localhost / my
localhost/Lavita/delete1.p
localhost/Lavita/delete1.php?id=1

First name: Divya
Last name: D
Address: Mangalore
Position: HR
Enter information


localhost / localhost / my
localhost/Lavita/delete1.p
localhost/Lavita/delete1.php?id=1

First name: Divya
Last name: D
Address: Mangalore
Position: HR
Delete information

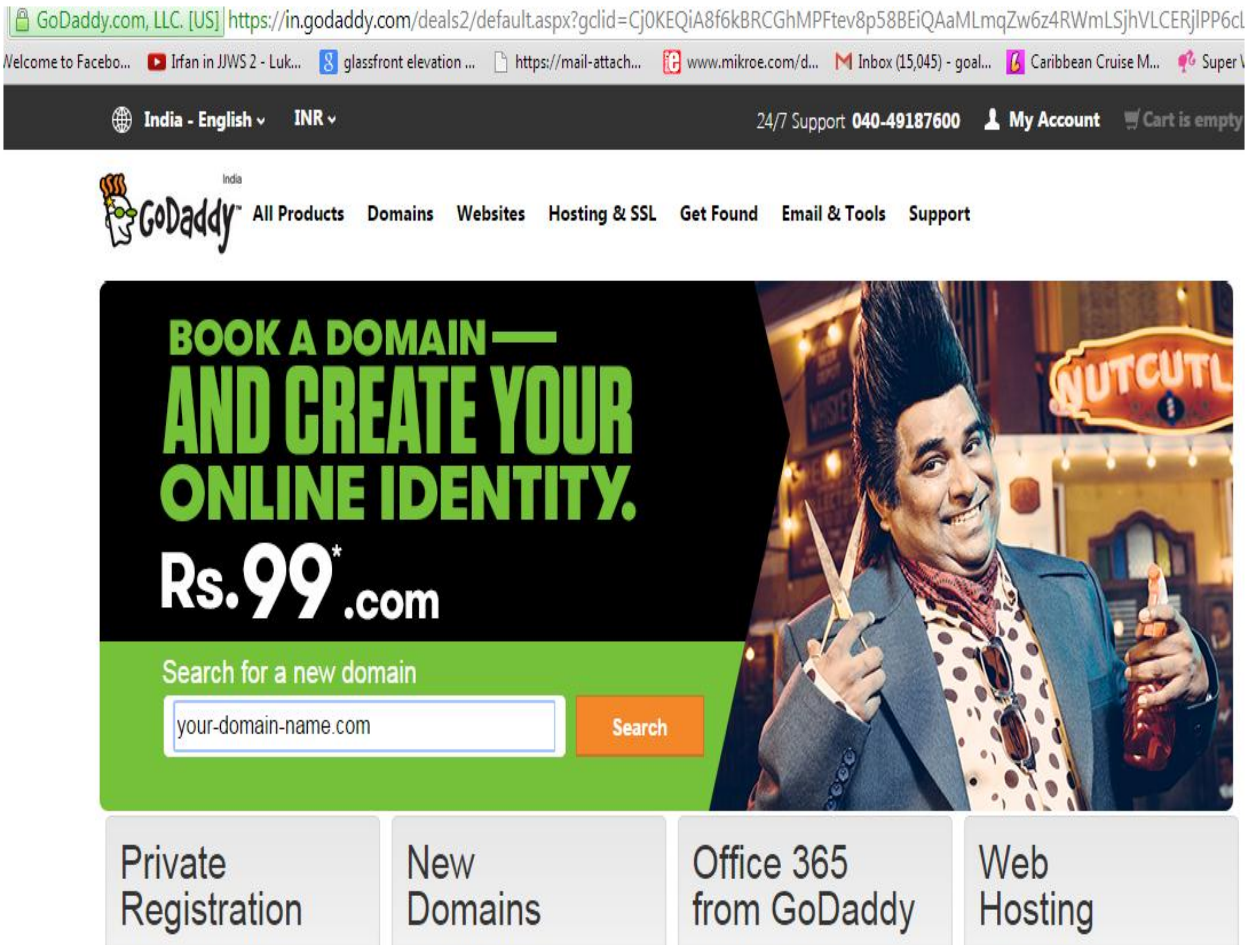
The image part with relationship ID rId9 was not found in the file.

The image part with relationship ID rId9 was not found in the file.

Hosting your domain using cPanel

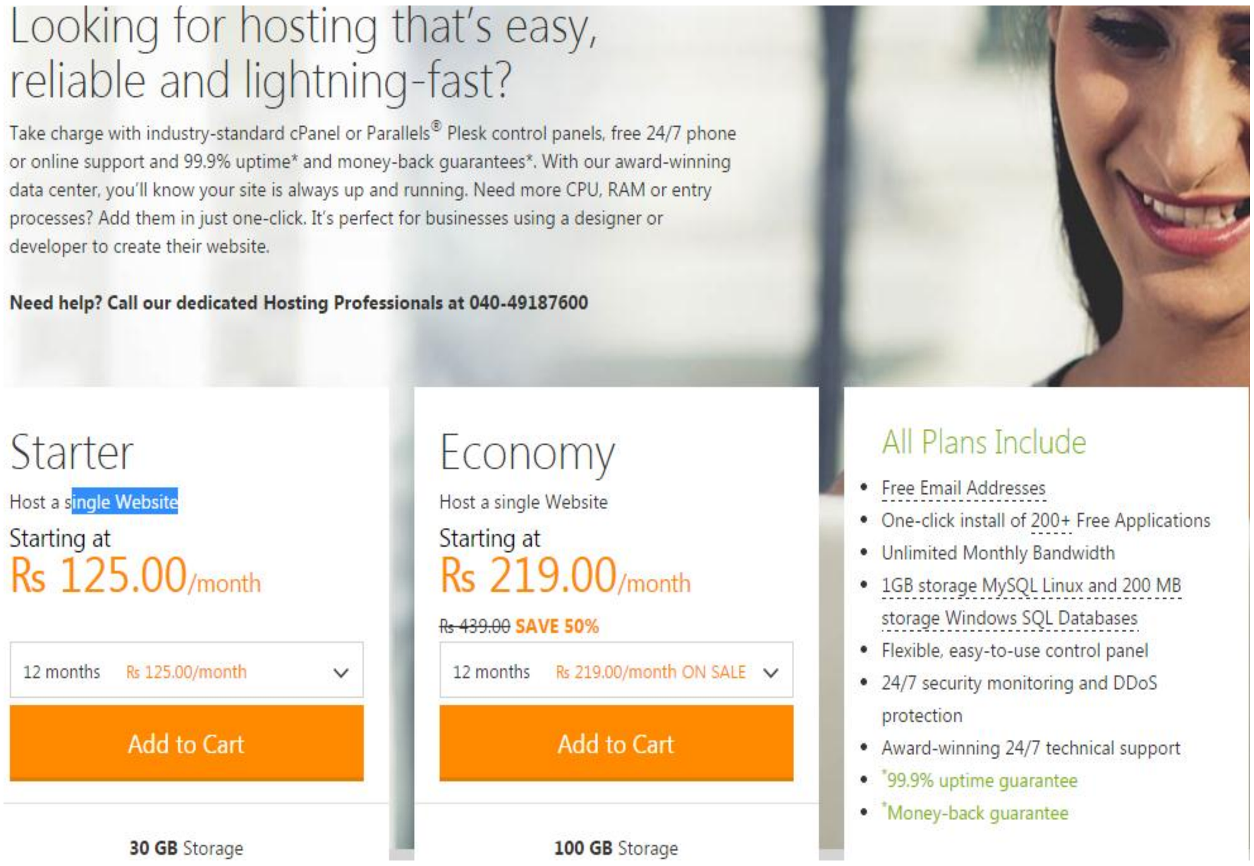
Step 1:

Register your required Domain name by visiting godaddy.com



Step 2:

Buy required domain Space to upload your code



Looking for hosting that's easy, reliable and lightning-fast?

Take charge with industry-standard cPanel or Parallels® Plesk control panels, free 24/7 phone or online support and 99.9% uptime* and money-back guarantees*. With our award-winning data center, you'll know your site is always up and running. Need more CPU, RAM or entry processes? Add them in just one-click. It's perfect for businesses using a designer or developer to create their website.

Need help? Call our dedicated Hosting Professionals at 040-49187600

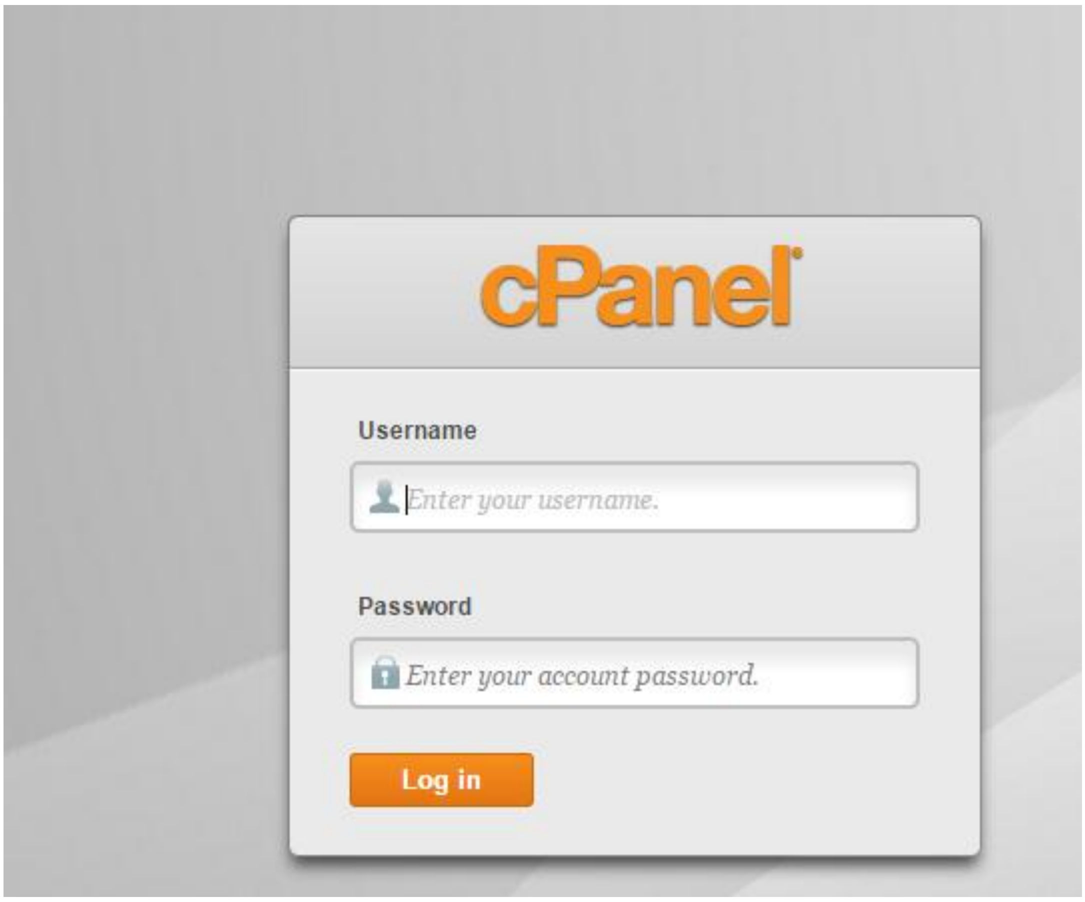
Starter
Host a single Website
Starting at
Rs 125.00/month
12 months Rs 125.00/month
Add to Cart
30 GB Storage

Economy
Host a single Website
Starting at
Rs 219.00/month
~~Rs 439.00~~ SAVE 50%
12 months Rs 219.00/month ON SALE
Add to Cart
100 GB Storage

All Plans Include
- Free Email Addresses
- One-click install of 200+ Free Applications
- Unlimited Monthly Bandwidth
- 1GB storage MySQL Linux and 200 MB storage Windows SQL Databases
- Flexible, easy-to-use control panel
- 24/7 security monitoring and DDoS protection
- Award-winning 24/7 technical support
- *99.9% uptime guarantee
- *Money-back guarantee

Step 3:

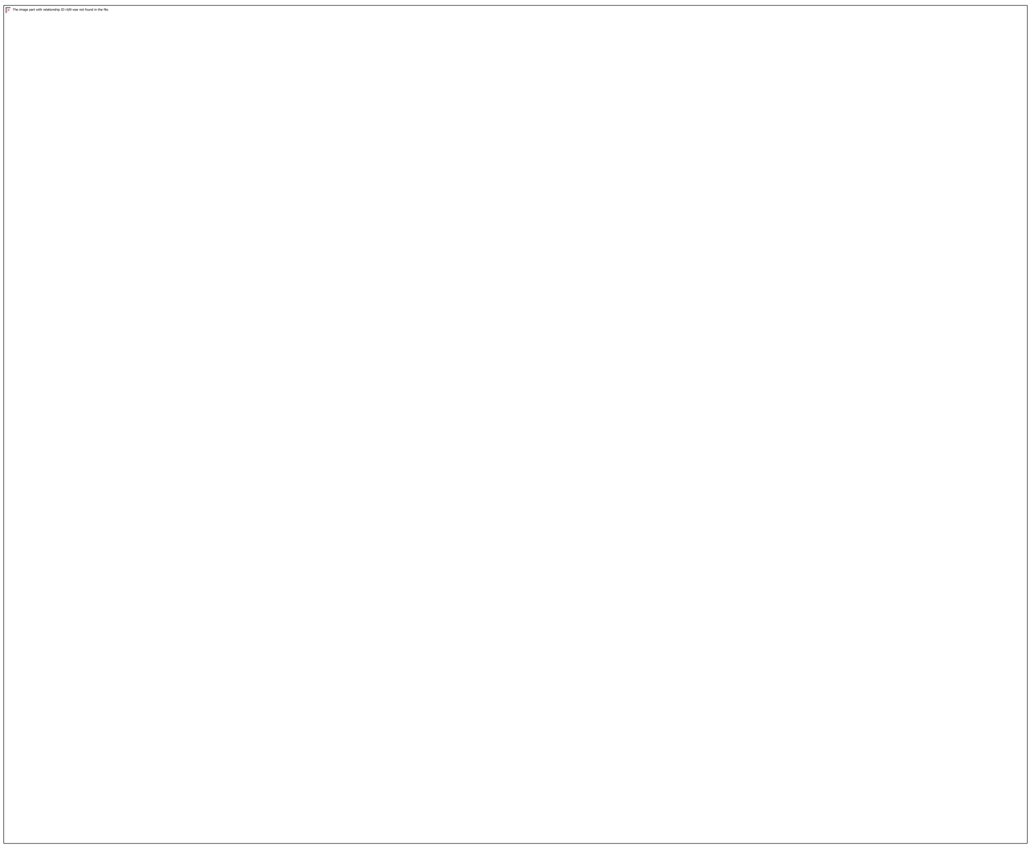
Launch your cPanel – your-domain-name.com/cPanel



Provide your cPanel username and password click on login

Step 4:

Click on file Manger


The image part with relationship ID rId9 was not found in the file.

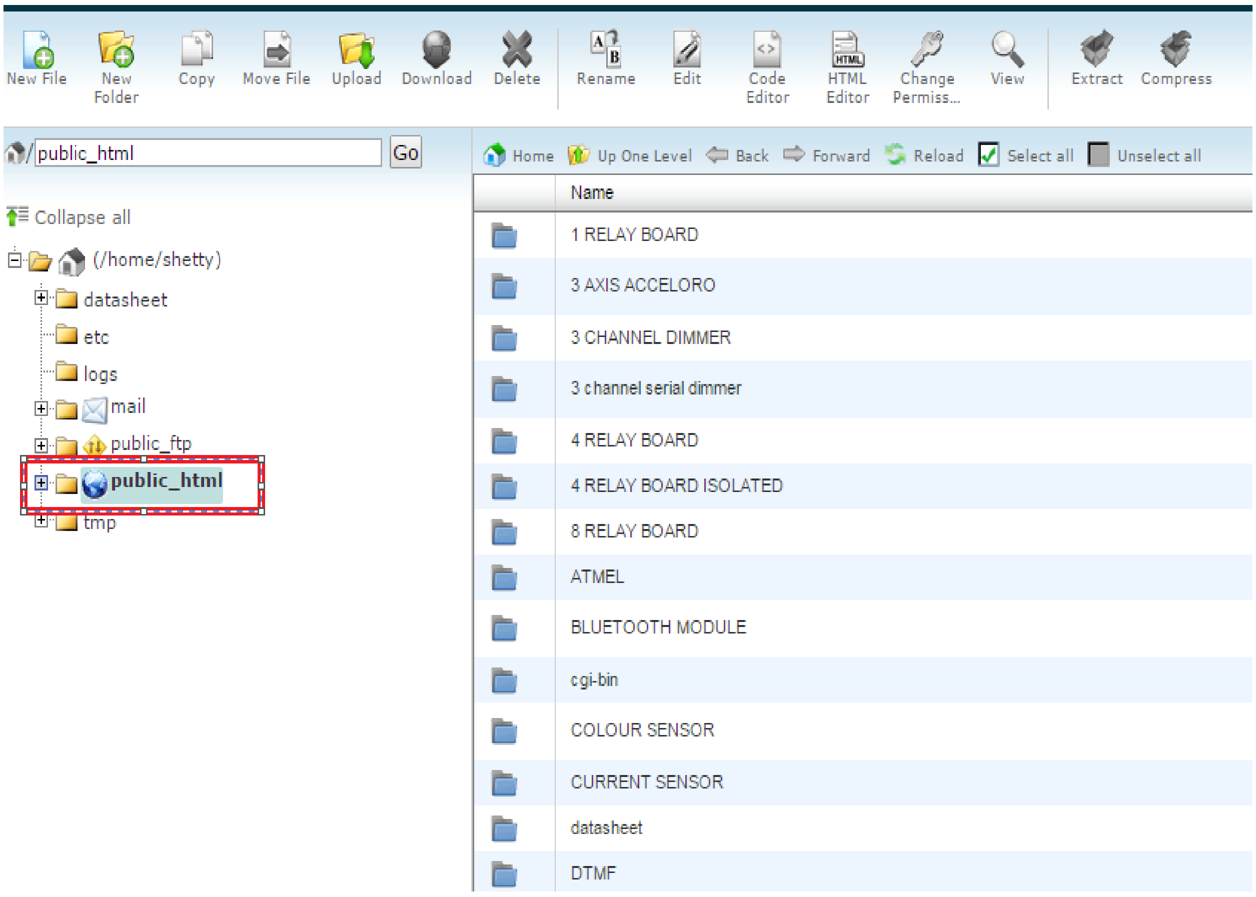
Step 5:

Click on Web root and Go

The image part with relationship ID rId9 was not found in the file.

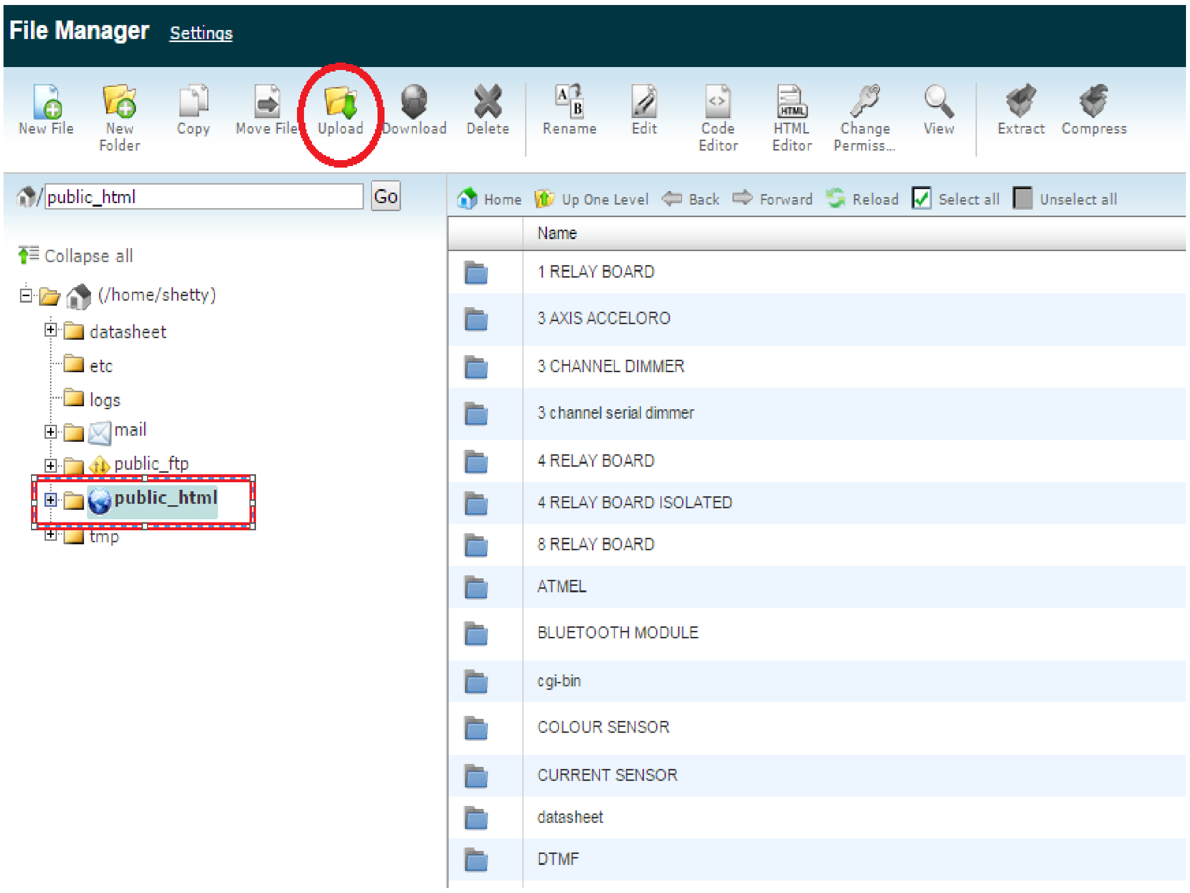
Step 6:

Select the Public Folder_html



Step 7:

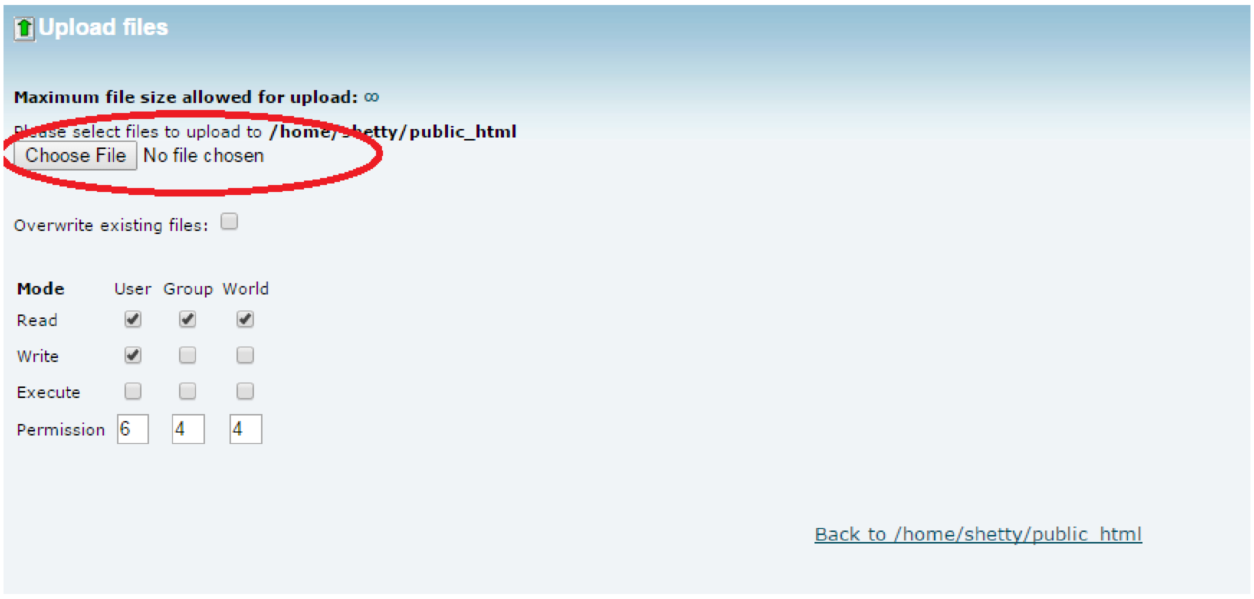
Click on Upload



Name
1 RELAY BOARD
3 AXIS ACCELORO
3 CHANNEL DIMMER
3 channel serial dimmer
4 RELAY BOARD
4 RELAY BOARD ISOLATED
8 RELAY BOARD
ATMEL
BLUETOOTH MODULE
cgi-bin
COLOUR SENSOR
CURRENT SENSOR
datasheet
DTMF

Step 8:

Click on the choose file to upload your code



Upload files

Maximum file size allowed for upload: ∞

Please select files to upload to /home/shetty/public_html

Choose File No file chosen

Overwrite existing files: ☐

Mode	User	Group	World
Read	✓	✓	✓
Write	✓	☐	☐
Execute	☐	☐	☐
Permission	6	4	4

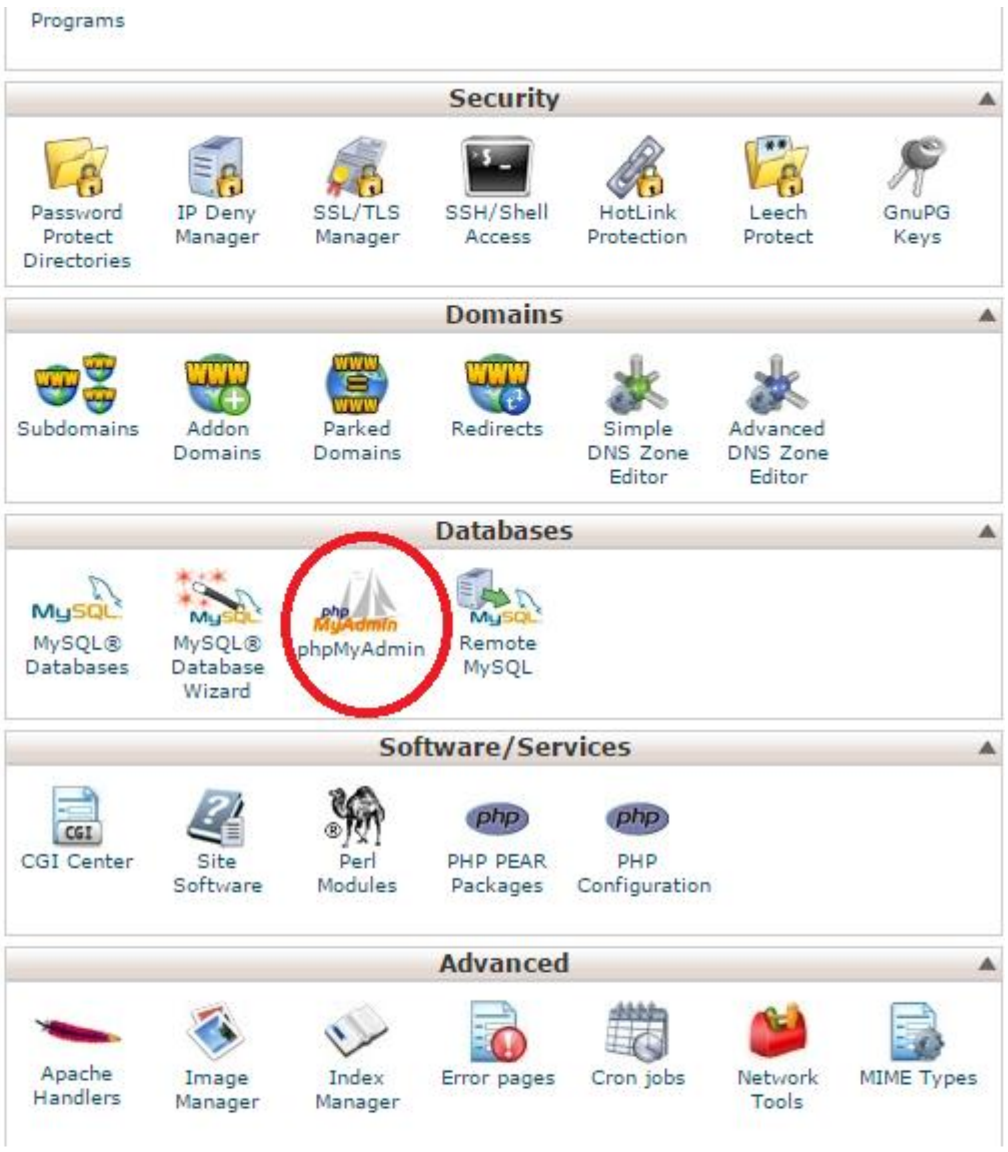
Back to /home/shetty/public_html

Note:

Once the code is uploaded your website will be live on internet. That you can check with open the browser with your-domain-name.com

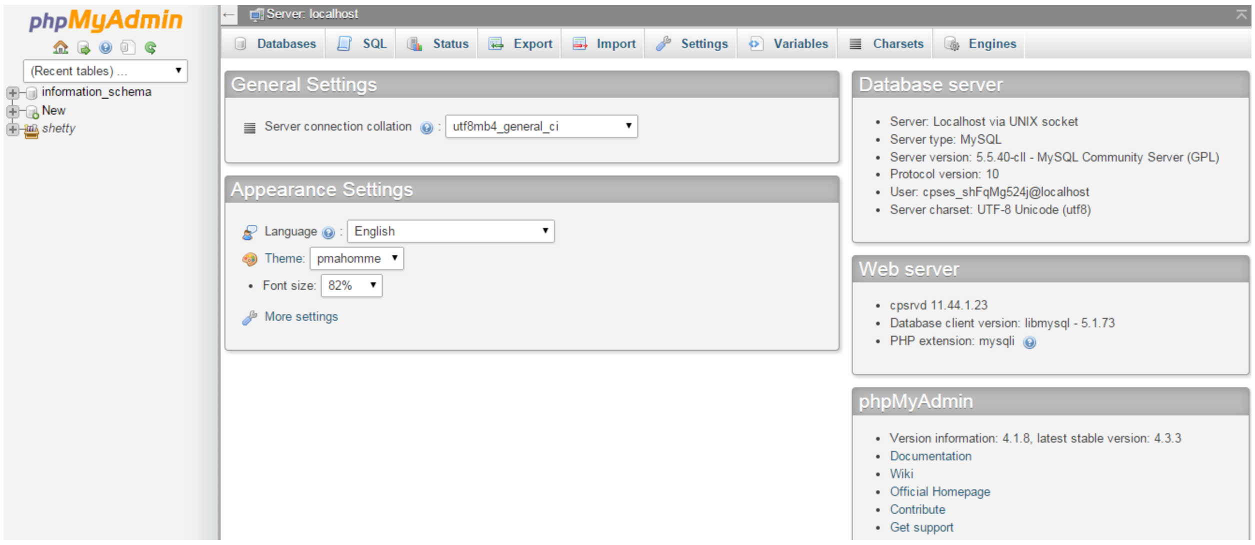
Step 9:

To create database – click on the phpMyAdmin



Step 10:

Manage your Tables and database on phpMyAdmin console



General Settings

Server connection collation : utf8mb4_general_ci

Appearance Settings

Language : English

Theme: pmahomme

- Font size: 82%

More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server version: 5.5.40-cll - MySQL Community Server (GPL)
- Protocol version: 10
- User: cpses_shFqMg524j@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

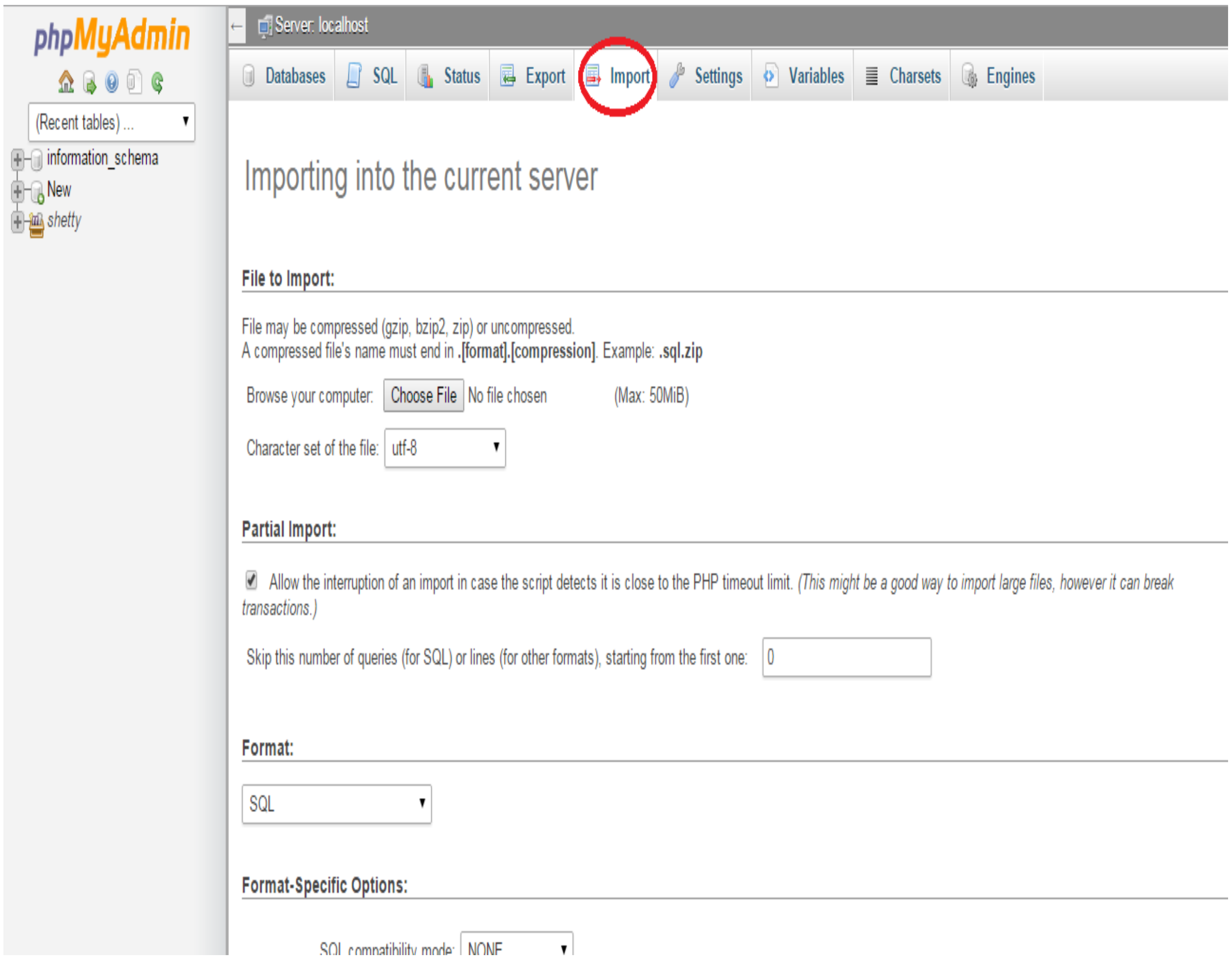
- cpsrvd 11.44.1.23
- Database client version: libmysql - 5.1.73
- PHP extension: mysqli

phpMyAdmin

- Version information: 4.1.8, latest stable version: 4.3.3
- Documentation
- Wiki
- Official Homepage
- Contribute
- Get support

Step 11:

Importing Database or table by clicking on import



Importing into the current server

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: Choose File No file chosen (Max: 50MiB)

Character set of the file: utf-8

Partial Import:

☑ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one: 0

Format:

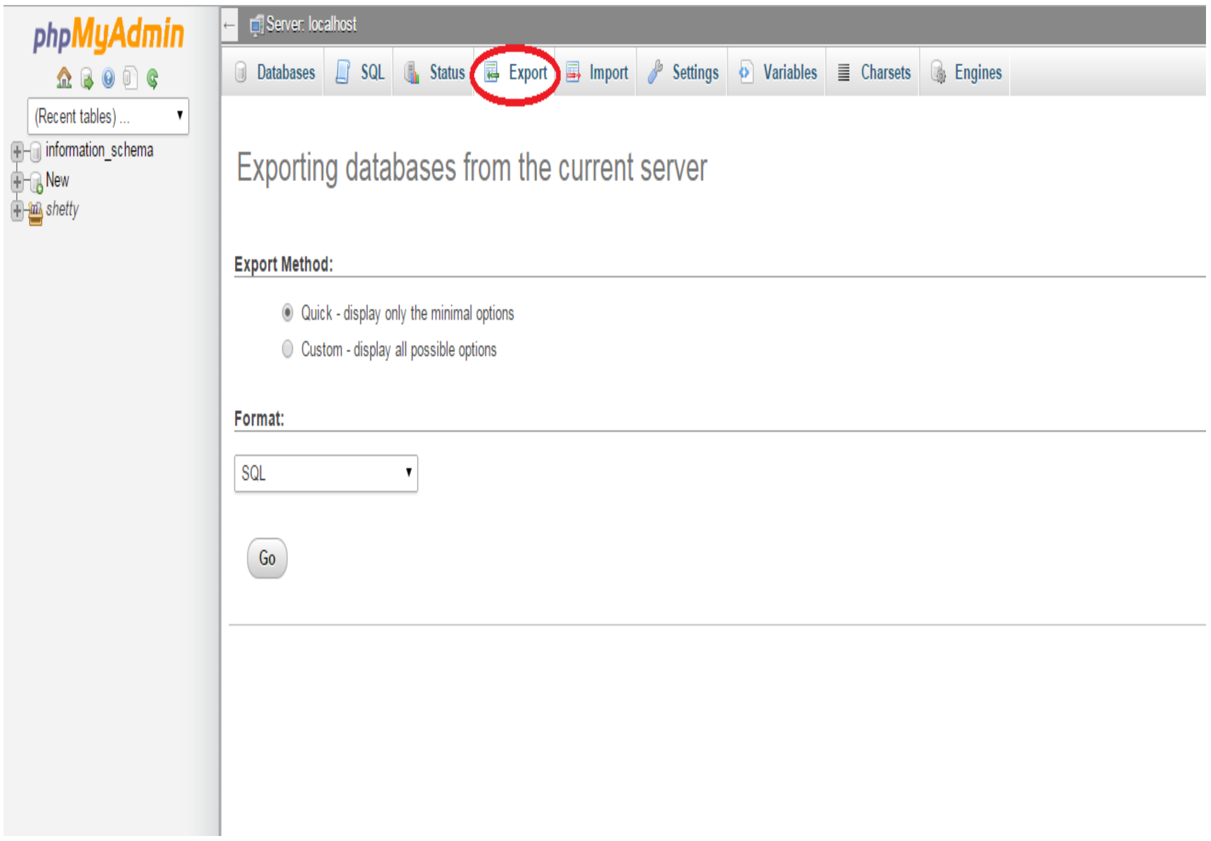
SQL

Format-Specific Options:

SQL compatibility mode: NONE

Step 12:

Exporting your database to local machine



Exporting databases from the current server

Export Method:

● Quick - display only the minimal options

○ Custom - display all possible options

Format:

SQL

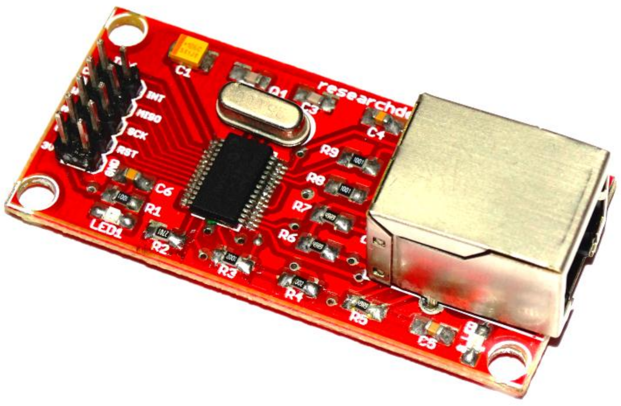
EMBEDDED SYSTEM WITH PHP

OVERVIEW

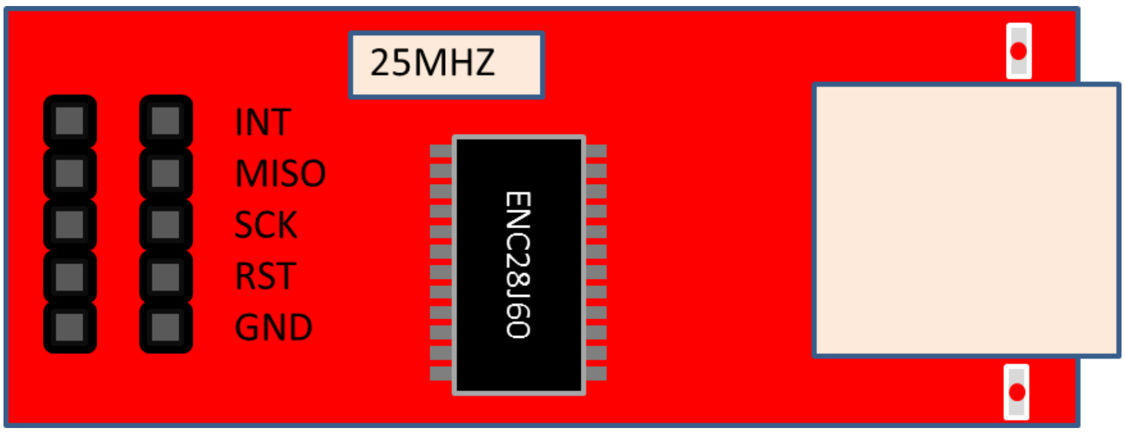
This Ethernet Breakout-Module is simplest way to add LAN connectivity to your microcontroller based products and projects. Use this module to enable Ethernet interface for your product. It works with any microcontroller operating at 3.3V or 5V. This module works at 3.3V and is compatible with 5V interface lines.

FEATURES

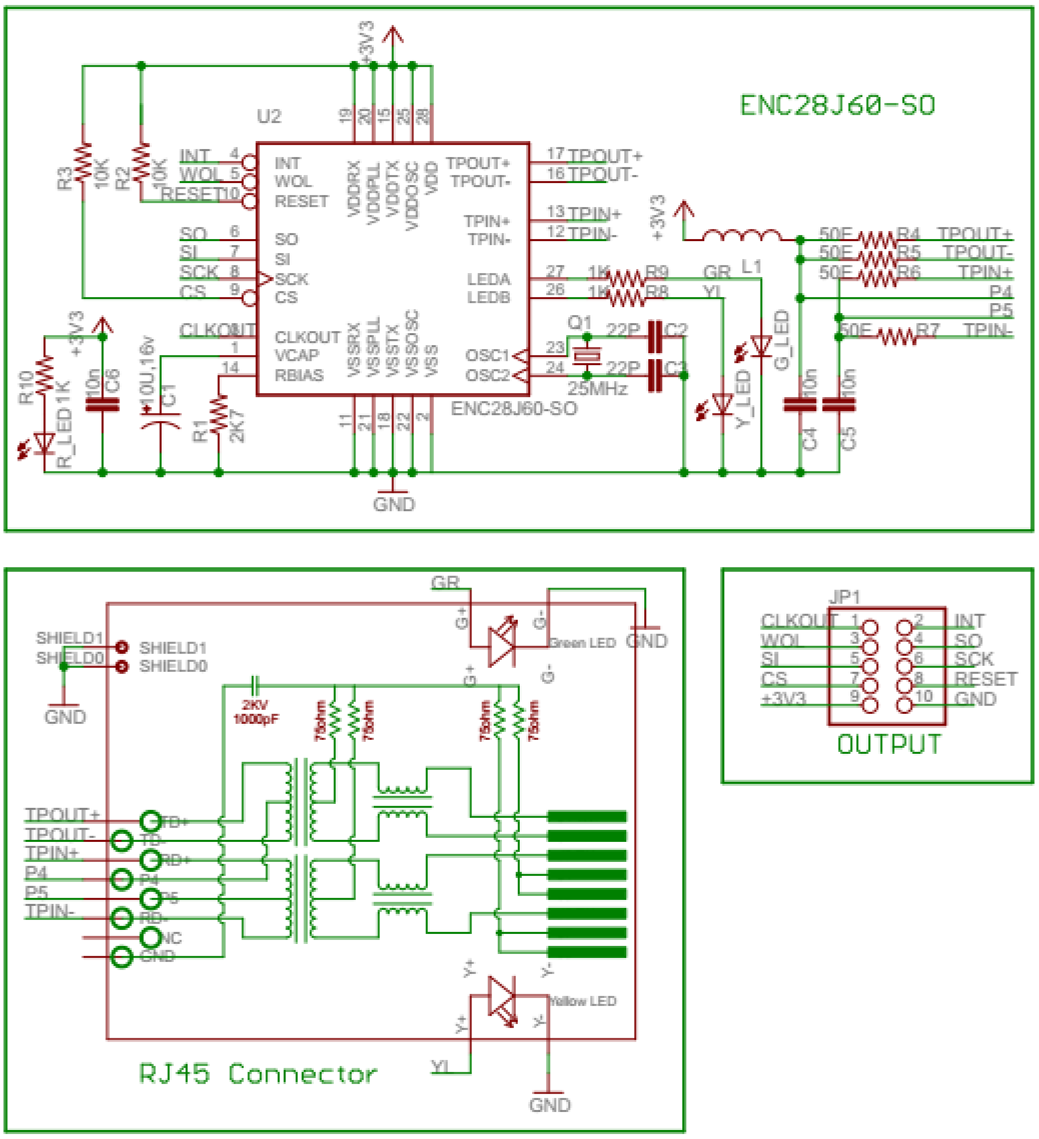
- Brand new and high quality.
- Chip board ENC28J60-I/SO.
- The board 25MHZ crystal.
- The network interface board HR911105A.
- 3.3 V power supply pin.
- Size: 6cm x 3.2cm - 2.4inch x 1.28inch.
- High quality PCB FR4 Grade with FPT Certified



APPLICATION DIAGRAM



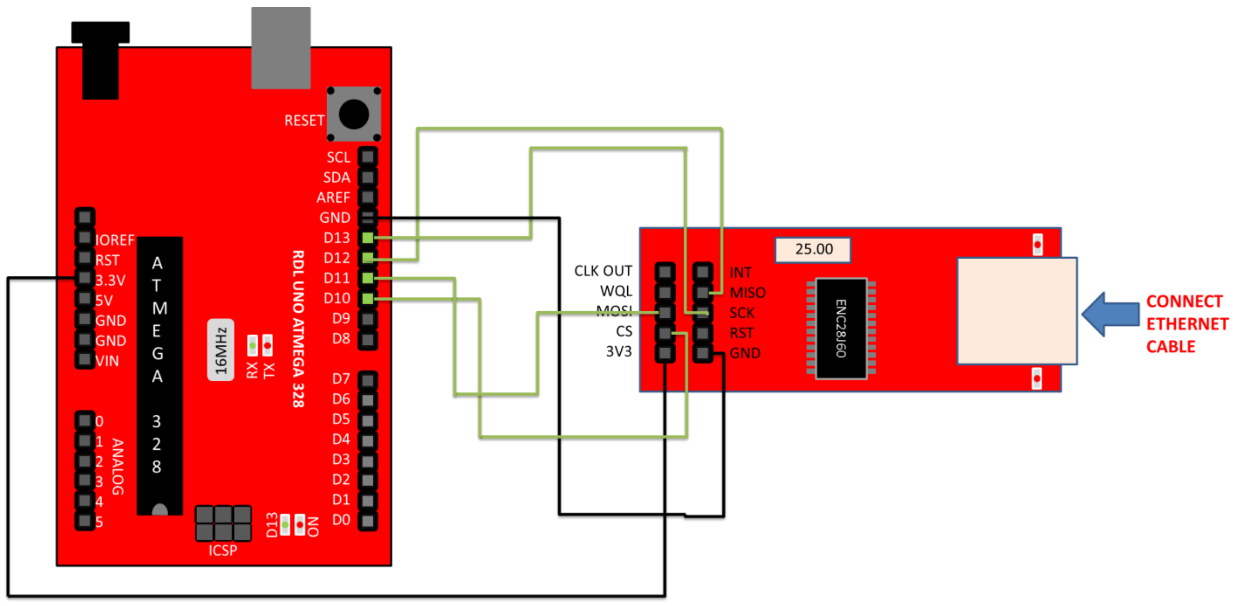
CIRCUIT DIAGRAM



PIN CONNECTION

ETHERNET MODULE PIN	RDL UNO PIN
MISO	12
SCK	13
GND	GND
3V3	3.3V
CS	10
MOSI	11

INTERFACE



LIBRARY

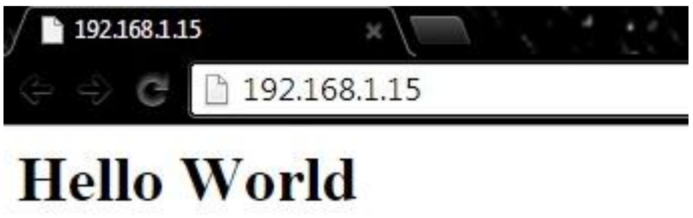
(*add this library on your arduino folder in program files*)

https://drive.google.com/file/d/0BzrGD4zr88GnZFRRalNjU1E0UmM/view

CODE

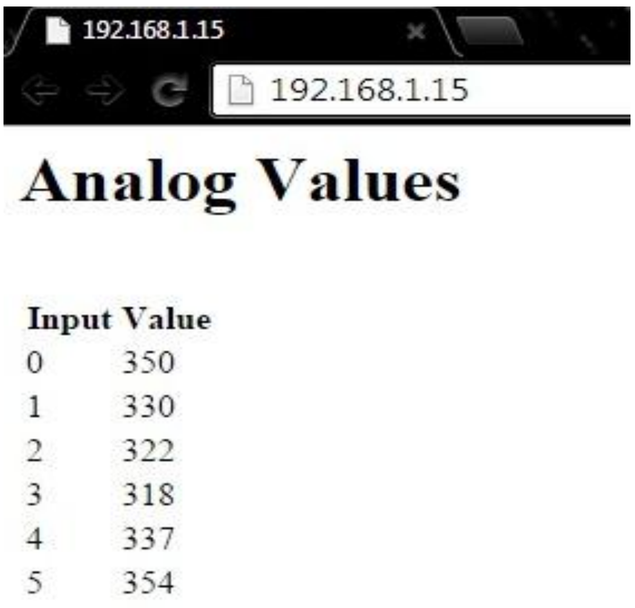
- **With Arduino**

1. ADD hello world example from Examples



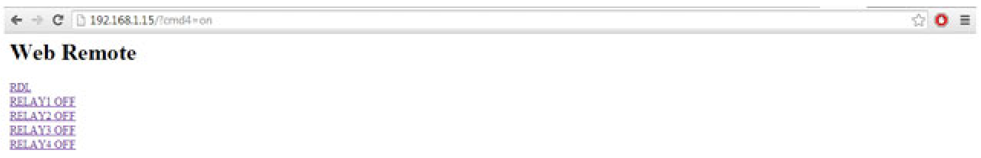
```
Hello World
```

2. Add Server Reading Example



```
Analog Values

Input Value
0     350
1     330
2     322
3     318
4     337
5     354
```

3. relay interface(Web Remote)



Web Remote

RDL
RELAY1 OFF
RELAY2 OFF
RELAY3 OFF
RELAY4 OFF

code:

```
#include "etherShield.h"
#include "ETHER_28J60.h"

int outputPin1 = 2;
int outputPin2 = 3;
int outputPin3 = 4;
int outputPin4 = 5;

static uint8_t mac[6] = {0x54, 0x55, 0x58, 0x10, 0x00, 0x24}; // this just needs to be unique
for your network,
// so unless you have more than one of these boards
// connected, you should be fine with this value.

static uint8_t ip[4] = {192, 168, 1, 15}; // the IP address for your board. Check your home
hub
// to find an IP address not in use and pick that
// this or 10.0.0.15 are likely formats for an address
// that will work.

static uint16_t port = 80; // Use port 80 - the standard for HTTP

ETHER_28J60 e;
char flag1=0,flag2=0,flag3=0,flag4=0;
void setup()
{
```

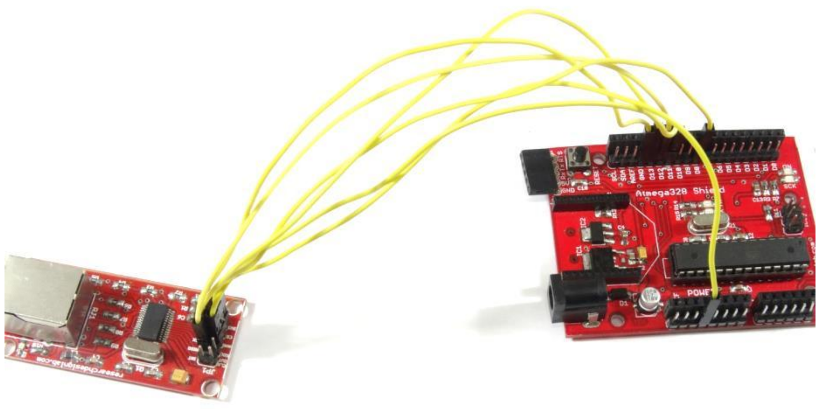
```
e.setup(mac, ip, port);
pinMode(outputPin1, OUTPUT);
pinMode(outputPin2, OUTPUT);
pinMode(outputPin3, OUTPUT);
pinMode(outputPin4, OUTPUT);

}

void loop()
{
char* params;
if (params = e.serviceRequest())
{
e.print("<H1>Web Remote</H1>");
e.print("<A HREF='?cmd1=off'>RDL</A></BR>");
// dispaly();
if (strcmp(params, "?cmd1=on") == 0)
{
digitalWrite(outputPin1, HIGH);
flag1=1;
dispaly();
}
else if (strcmp(params, "?cmd1=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin1, LOW);
flag1=0;
dispaly();
}
if (strcmp(params, "?cmd2=on") == 0)
{
digitalWrite(outputPin2, HIGH);
flag2=1;
dispaly();

}
else if (strcmp(params, "?cmd2=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin2, LOW);
flag2=0;
dispaly();
}
if (strcmp(params, "?cmd3=on") == 0)
{
digitalWrite(outputPin3, HIGH);
flag3=1;
dispaly();

}
else if (strcmp(params, "?cmd3=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin3, LOW);
flag3=0;
dispaly();
}
if (strcmp(params, "?cmd4=on") == 0)
{
digitalWrite(outputPin4, HIGH);
flag4=1;
dispaly();

}
else if (strcmp(params, "?cmd4=off") == 0) // Modified -- 2011 12 15 # Ben Schueler
{
digitalWrite(outputPin4, LOW);
flag4=0;
dispaly();
}
e.respond();
}
}
void dispaly()
{

if(flag1==0)
e.print("<A HREF='?cmd1=on'>RELAY1 ON</A></BR>");
else
e.print("<A HREF='?cmd1=off'>RELAY1 OFF</A></BR>");

if(flag2==0)
e.print("<A HREF='?cmd2=on'>'>RELAY2 ON</A></BR>");
else
e.print("<A HREF='?cmd2=off'>RELAY2 OFF</A></BR>");


if(flag3==0)
e.print("<A HREF='?cmd3=on'>'>RELAY3 ON</A></BR>");
else
e.print("<A HREF='?cmd3=off'>RELAY3 OFF</A></BR>");

if(flag4==0)
e.print("<A HREF='?cmd4=on'>'>RELAY4 ON</A></BR>");
else
e.print("<A HREF='?cmd4=off'>RELAY4 OFF</A></BR>");

}
```

4. web integration

PIN CONNECTION

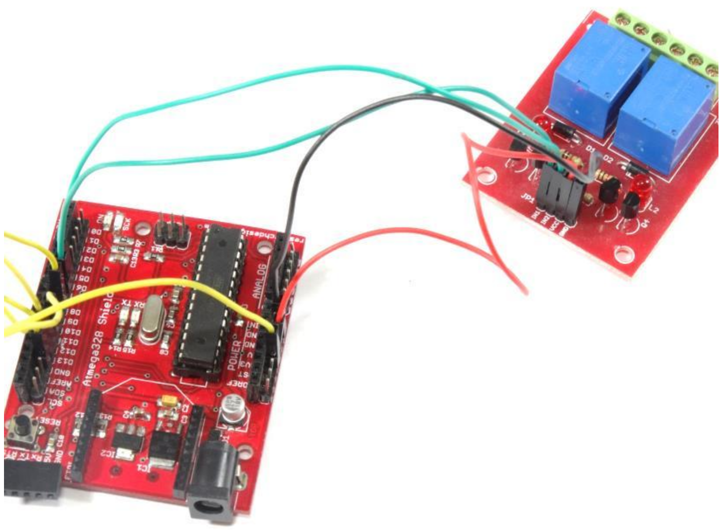
WITH ETHERNET MODULE

ETHERNET MODULE PIN	RDL UNO PIN
MISO	12
SCK	13
GND	GND
3V3	3.3V
CS	8
MOSI	11

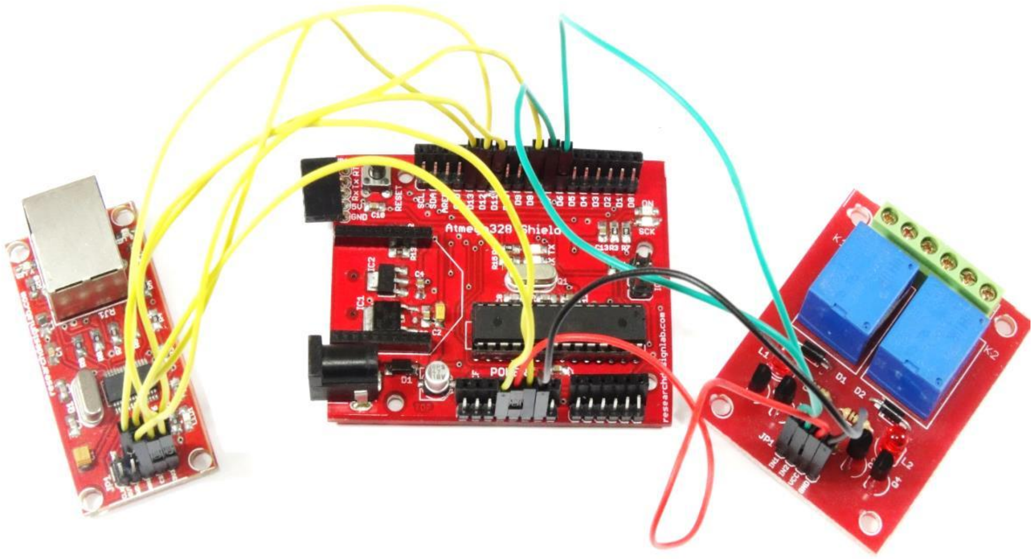


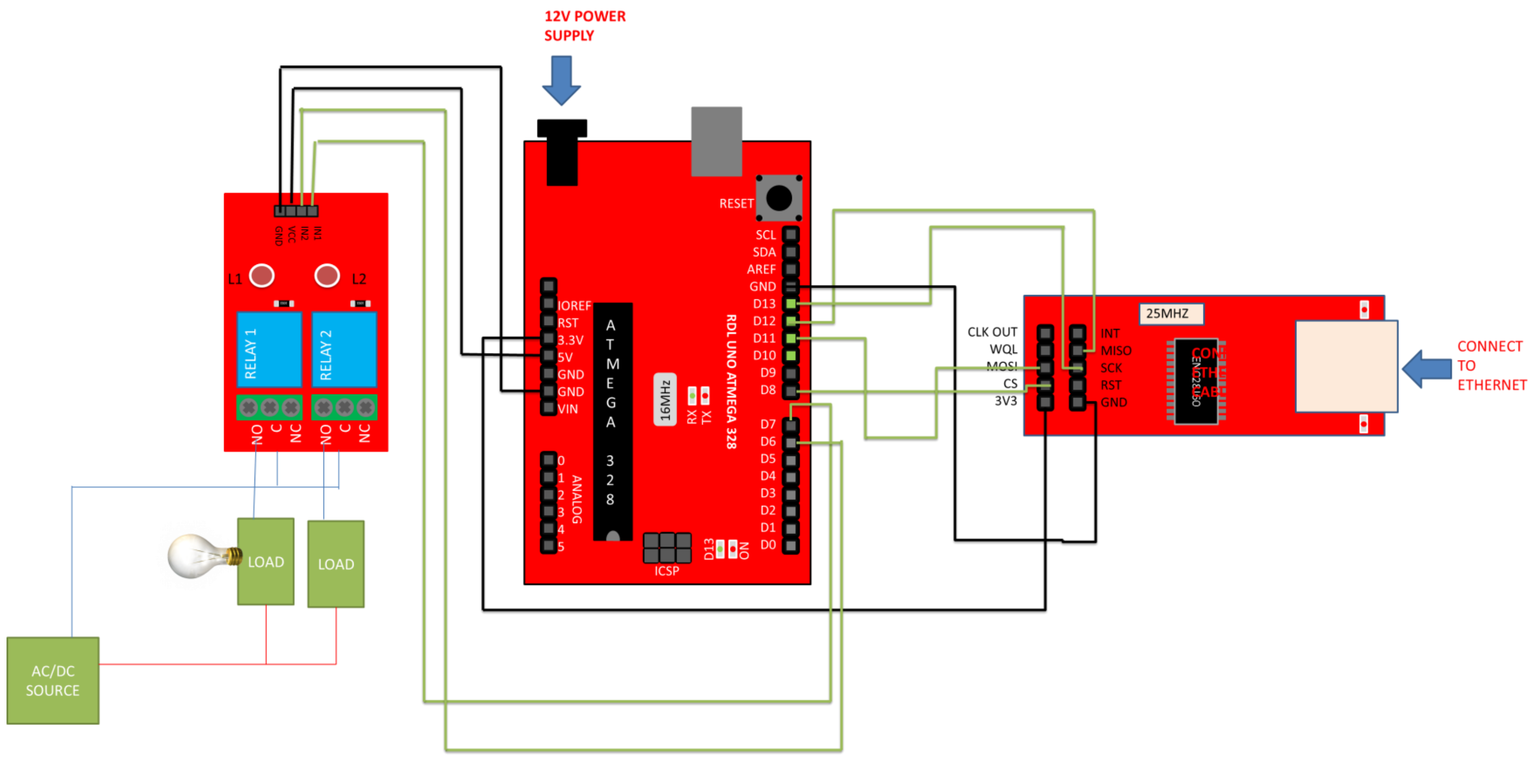
WITH RELAY

ETHERNET MODULE PIN	RELAY PIN
D7	IN1
D6	IN2
5V	VCC
GND	GND



Connection Diagrams





CODE

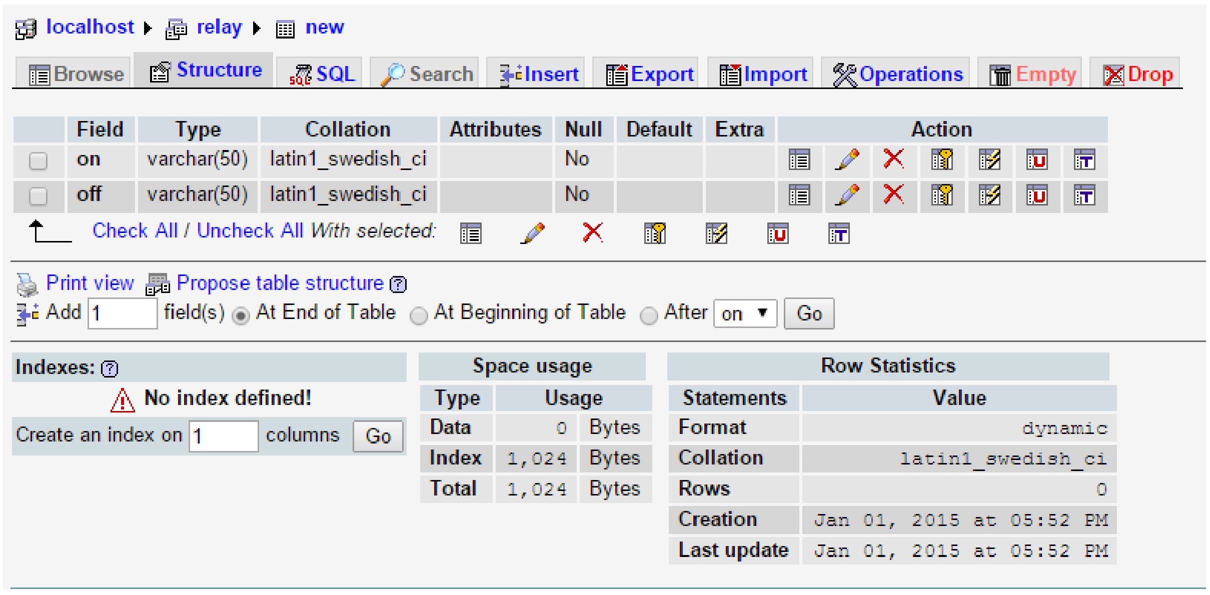
```
#include <EtherCard.h>


#define REQUEST_RATE 5000 // milliseconds


// ethernet interface mac address

static byte mymac[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };

// remote website name

const char website[] PROGMEM = "passport.eu5.org";
```

```
byte Ethernet::buffer[700];

static long timer;

int i,val;

char RL1=0,RL2=0;

// called when the client request is complete

static void my_result_cb (byte status, word off, word len) {

  Serial.print("<<< reply ");

  Serial.print(millis() - timer);

  Serial.println(" ms");

 // Serial.println((const char*) Ethernet::buffer + off);

   i=150;

  val=0;

      while(val!='R')

  {

   val=Ethernet::buffer[i];

   i++;

  }

   while(val!='=')

  {

   val=Ethernet::buffer[i];

   i++;


  }

  RL1=Ethernet::buffer[i];


      while(val!='R')
  {
   val=Ethernet::buffer[i];
   i++;

  }
   while(val!='=')
  {
   val=Ethernet::buffer[i];
   i++;

  }
  RL2=Ethernet::buffer[i];

  if(RL1=='1')
 digitalWrite(7, HIGH);
  else if(RL1=='0')
  digitalWrite(7, LOW);
  if(RL2=='1')
  digitalWrite(6, HIGH);
  else if(RL2=='0')
digitalWrite(6, LOW);



}

void setup () {
  Serial.begin(9600);
```

```
  Serial.println("\n[getDHCPandDNS]");


  if (ether.begin(sizeof Ethernet::buffer, mymac) == 0)
    Serial.println( "Failed to access Ethernet controller");


  if (!ether.dhcpSetup())
    Serial.println("DHCP failed");


  ether.printIp("My IP: ", ether.myip);
  // ether.printIp("Netmask: ", ether.mymask);
  ether.printIp("GW IP: ", ether.gwip);
  ether.printIp("DNS IP: ", ether.dnsip);


  if (!ether.dnsLookup(website))
    Serial.println("DNS failed");
  ether.printIp("Server: ", ether.hisip);


  timer = - REQUEST_RATE; // start timing out right away
   pinMode(7, OUTPUT);
    pinMode(6, OUTPUT);
}


void loop () {


  ether.packetLoop(ether.packetReceive());


  if (millis() > timer + REQUEST_RATE) {
```

```
  timer = millis();

  Serial.println("\n>>> REQ");

  ether.browseUrl(PSTR("/new/check.php"), " ", website, my_result_cb);

 }



}
```

PHP: (LOCAL HOST)



	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	on	varchar(50)	latin1_swedish_ci		No			
☐	off	varchar(50)	latin1_swedish_ci		No			

Check All / Uncheck All *With selected:*

Print view Propose table structure
Add 1 field(s) ● At End of Table ○ At Beginning of Table ○ After on Go

Indexes:
No index defined!
Create an index on 1 columns Go

Space usage		
Type	Usage	
Data	0	Bytes
Index	1,024	Bytes
Total	1,024	Bytes

Row Statistics	
Statements	Value
Format	dynamic
Collation	latin1_swedish_ci
Rows	0
Creation	Jan 01, 2015 at 05:52 PM
Last update	Jan 01, 2015 at 05:52 PM

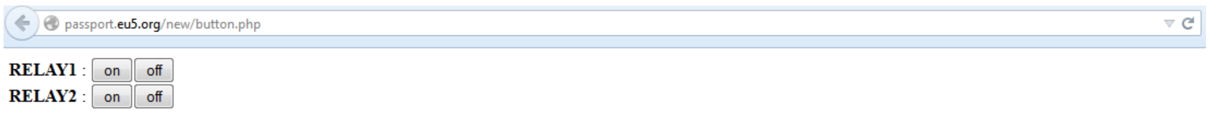
(*for hosting detail please refer page no 74*)

PHP CODE:

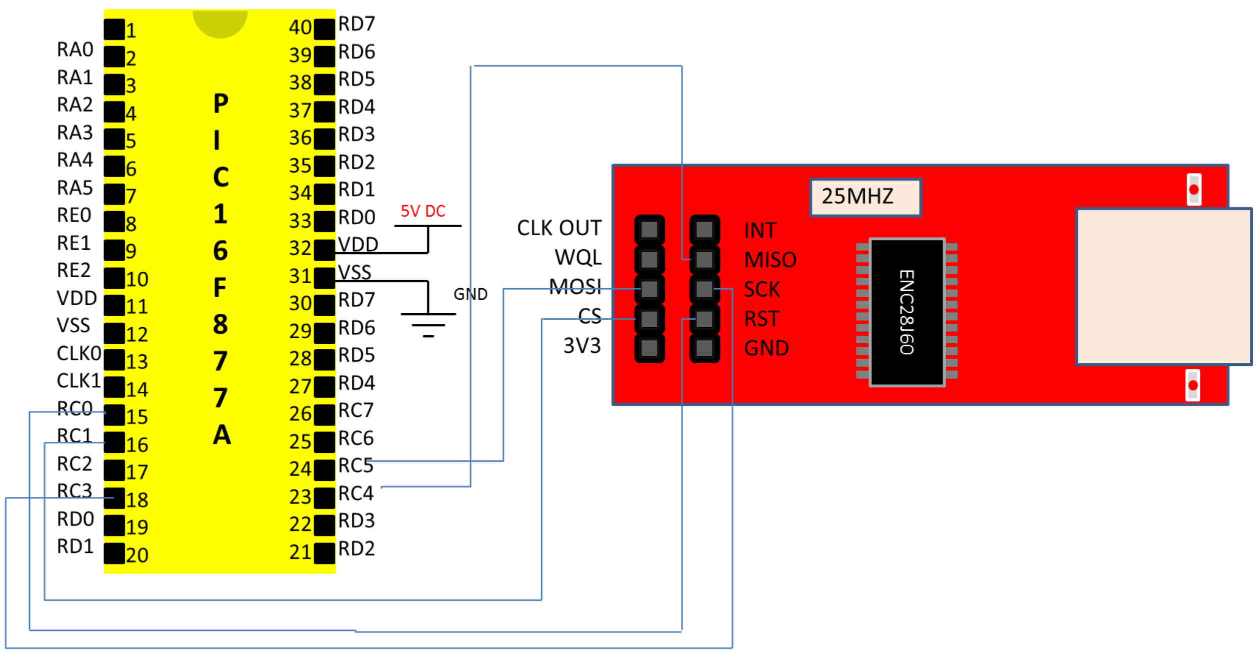
For buttons:

```
<html>
<body>
<?php
if(isset($_POST['first'])){
            include "relay.php";
$sql="SELECT * FROM reg WHERE ONNN='0' ";
$result=mysql_query($sql);
            //$query = mysql_query("SELECT * FROM reg )"

$sql1="UPDATE reg SET ONNN='1' ";

            $result2 = mysql_query($sql1);
            //$query1=mysql_query("UPDATE reg SET OFFF='0')"
//$sql2="UPDATE reg SET OFFF='0' ";
//$result3 = mysql_query($sql2);

}
if(isset($_POST['last'])){
include "relay.php";
$sql="SELECT * FROM reg WHERE ONNN='1' ";
$result=mysql_query($sql);
//$sql1="UPDATE reg SET ONNN='0' ";
//          $result2 = mysql_query($sql1);
$sql2="UPDATE reg SET ONNN='0' ";
$result3 = mysql_query($sql2);
}

if(isset($_POST['second'])){
            include "relay.php";
$sql="SELECT * FROM reg WHERE ONNN='0' ";
$result=mysql_query($sql);
            //$query = mysql_query("SELECT * FROM reg )"

$sql1="UPDATE reg SET OFFF='1' ";

            $result2 = mysql_query($sql1);
```

```
        //$query1=mysql_query("UPDATE reg SET OFFF='0')"
//$sql2="UPDATE reg SET OFFF='0' ";
//$result3 = mysql_query($sql2);
}
if(isset($_POST['last2'])){
include "relay.php";
$sql="SELECT * FROM reg WHERE ONNN='1' ";
$result=mysql_query($sql);
//$sql1="UPDATE reg SET ONNN='0' ";
//        $result2 = mysql_query($sql1);
$sql2="UPDATE reg SET OFFF='0' ";
$result3 = mysql_query($sql2);
}
?>

<form method="post" action="<?php echo $PHP_SELF?>">
<b>RELAY1</b> : <input type="submit" name="first" value="on"><input type="submit"
name="last" value="off"><br>
<b>RELAY2</b> : <input type="submit" name="second" value="on"><input type="submit"
name="last2" value="off"><br>


</form>

<?php
?>






</body>
</html>
```


passport.eu5.org/new/button.php
RELAY1 : on off
RELAY2 : on off

3. With Pic 16F877A



PIN CONNECTION

Ethernet module	Pic 16f877a
RST	RC0
CS	RC1
SCK	RC3
MOSI	RC5
MISO	RC4
GND	GND

code

```
/*
 * Project Name:
    httpserver_example (Ethernet Library http server demo for ENC28J60 mcu)
 * Copyright:
    (c) Mikroelektronika, 2005-2010.
 * Revision History:
    2007/12/10:
      - initial release; Author: Bruno Gavand.
    2010/12/20:
      - modified for PRO compilers (FJ);
    2012/10/19:
    Modifier pour usage personnel

 * description  :
 *     this code shows how to use the Spi_Ethernet mini library :
 *           the board will reply to ARP & ICMP echo requests
 *           the board will reply to HTTP requests on port 80, GET method with pathnames :
 *                 /             will return the HTML main page
 *                 /s            will return board status as text string
 *                 /t0 ... /t7     will toggle PD0 to PD7 bit and return HTML main page
 *                 all other requests return also HTML main page
 * Test configuration:
    MCU:             PIC16F877A
                     http://ww1.microchip.com/downloads/en/DeviceDoc/41291D.pdf

    Dev.Board:         PIC-Ready1
                     http://www.mikroe.com/products/view/177/pic-ready-prototype-board/
                     http://www.mikroe.com/products/view/305/pic-ready1-board/

    Oscillator:        External Clock 8.0000 MHz
    Ext. Modules:      ac:Serial_Ethernet_board
                     http://www.mikroe.com/eng/products/view/14/serial-ethernet-board/

    SW:              mikroC PRO for PIC
                     http://www.mikroe.com/en/compilers/mikroc/pro/pic/
 * NOTES:
    - Connect Serial Ethernet Board on PortC (Board and MCU Specific)
    - Since the ENC28J60 doesn't support auto-negotiation, full-duplex mode is
      not compatible with most switches/routers.  If a dedicated network is used
      where the duplex of the remote node can be manually configured, you may
      change this configuration.  Otherwise, half duplex should always be used.
    - External power supply should be used due to Serial Ethernet Board power consumption.
 */
/*
Pour voir l'états des variables tapper : 192.168.0.170/s
Pour faire une commande d'interrupteur faire :
192.168.0.170/dXo ouvre(OFF)
192.168.0.170/dXf ferme(ON)

avec X pour valeur : 0 a 7
*/
// duplex config flags
#define Spi_Ethernet_HALFDUPLEX     0x00  // half duplex
#define Spi_Ethernet_FULLDUPLEX     0x01  // full duplex

// mE ehternet NIC pinout
sfr sbit SPI_Ethernet_Rst at RC0_bit;
sfr sbit SPI_Ethernet_CS  at RC1_bit;
sfr sbit SPI_Ethernet_Rst_Direction at TRISC0_bit;
sfr sbit SPI_Ethernet_CS_Direction  at TRISC1_bit;
// end ethernet NIC definitions

typedef struct
{
  unsigned canCloseTCP: 1;  // flag which closes TCP socket (not relevant to UDP)
  unsigned isBroadcast: 1;  // flag which denotes that the IP package has been received via subnet
broadcast address (not used for PIC16 family)
} TEthPktFlags;

/************************************************************
 * ROM constant strings
 */
const unsigned char httpHeader[] = "HTTP/1.1 200 OK\nContent-type: " ;  // HTTP header
const unsigned char httpMimeTypeHTML[] = "text/html\n\n" ;             // HTML MIME type
const unsigned char httpMimeTypeScript[] = "text/plain\n\n" ;          // TEXT MIME type
unsigned char httpMethod[] = "GET /";
/*
 * web page, splited into 2 parts :
 * when coming short of ROM, fragmented data is handled more efficiently by linker
 *
 * this HTML page calls the boards to get its status, and builds itself with javascript
 */
 //*************************************************************************
 // Change the IP address of the page to be refreshed

const char *indexPageHEAD = "<meta http-equiv='refresh' content='10;url=http://192.168.1.15/'>\
<HTML><HEAD></HEAD><BODY>\
<h1>RESEARCH DESIGN LAB,<p>  </h1>\
<p><a href=\"http://olivier.fournet.free.fr/e.html\">NOTICE</a><p>\
<p><a href=/>Reload</a><p>\
<script src=/s></script>";

const char *indexPageBODY =  "<table><tr><td valign=top><table border=1 style='font-size:20px;
font-family: terminal;'>\
<tr><th colspan=2>ADC</th></tr>\
<tr><td>AN2</td><td><script>document.write(AN2);</script></td></tr>\
<tr><td>AN3</td><td><script>document.write(AN3);</script></td></tr>\
</table></td><td><table border=1 style='font-size:20px; font-family: terminal;'>\
<tr><th colspan=2>PORTB (IN) : <script>document.write(PORTB);</script></th></tr>\
<script>\
var str,i;\
str='';\
for(i=0;i<8;i++)\
{\
str+='<tr><td bgcolor=#cccccc>BUTTON #'+i+'</td>';\
if(PORTB&(1<<i))\
{str+='<td bgcolor=green>ON';}\
else\
{str+='<td bgcolor=red>OFF';}\
str+='</td></tr>';}\
document.write(str);\
</script>";

const char *indexPageBODY2 =  "</table></td><td>\
<table border=1 style='font-size:20px; font-family: terminal;'>\
<tr><th colspan=3>PORTD (OUT) : <script>document.write(PORTD);</script></th></tr>\
<script>\
var str,i;\
str='';\
for(i=0;i<8;i++)\
{\
str+='<tr><td bgcolor=#cccccc>LED #'+i+'</td>';\
if(PORTD&(1<<i))\
{\
str+='<td bgcolor=yellow>ON';\
}\
else\
{\
str+='<td bgcolor=#999999>OFF';\
}\
str+='</td></tr>';}\
document.write(str);\
</script>\
</table></td></tr></table>\
<p>This is HTTP request #<script>document.write(REQ)</script><p>\
</BODY></HTML>";

//*************************************************************************


/***********************************
 * RAM variables
 */
unsigned char   myMacAddr[6] = {0x54, 0x55, 0x58, 0x10, 0x00, 0x24};   // my MAC address
unsigned char   myIpAddr[4]  = {192, 168, 1, 15};                  // my IP address
unsigned char   getRequest[15];                               // HTTP request buffer
unsigned char   get_Request, digit_getRequest, etat_interrupteur;
unsigned char   dyna[30];                               // buffer for dynamic response
unsigned long   httpCounter = 0;                            // counter of HTTP requests

/*******************************************
 * functions
 */

/*
 * put the constant string pointed to by s to the ENC transmit buffer.
 */
/*unsigned int    putConstString(const char *s)
        {
        unsigned int ctr = 0 ;

        while(*s)
                {
                Spi_Ethernet_putByte(*s++) ;
                ctr++ ;
                }
        return(ctr) ;
        }*/
/*
 * it will be much faster to use library Spi_Ethernet_putConstString routine
 * instead of putConstString routine above. However, the code will be a little
 * bit bigger. User should choose between size and speed and pick the implementation that
 * suites him best. If you choose to go with the putConstString definition above
 * the #define line below should be commented out.
 *
 */
#define putConstString  SPI_Ethernet_putConstString

/*
 * put the string pointed to by s to the ENC transmit buffer
 */
/*unsigned int    putString(char *s)
        {
        unsigned int ctr = 0 ;

        while(*s)
                {
                Spi_Ethernet_putByte(*s++) ;

                ctr++ ;
                }
        return(ctr) ;
        }*/
/*
 * it will be much faster to use library Spi_Ethernet_putString routine
 * instead of putString routine above. However, the code will be a little
 * bit bigger. User should choose between size and speed and pick the implementation that
 * suites him best. If you choose to go with the putString definition above
 * the #define line below should be commented out.
 *
 */
#define putString  SPI_Ethernet_putString

/*
 * this function is called by the library
 * the user accesses to the HTTP request by successive calls to Spi_Ethernet_getByte()
 * the user puts data in the transmit buffer by successive calls to Spi_Ethernet_putByte()
 * the function must return the length in bytes of the HTTP reply, or 0 if nothing to transmit
 *
 * if you don't need to reply to HTTP requests,
 * just define this function with a return(0) as single statement
 *
 */
unsigned int  SPI_Ethernet_UserTCP(unsigned char *remoteHost, unsigned int remotePort,
                        unsigned int localPort, unsigned int reqLength, TEthPktFlags *flags)
{
  unsigned int    len = 0 ;                 // my reply length
  unsigned int    i ;                    // general purpose integer

  // should we close tcp socket after response is sent?
  // library closes tcp socket by default if canClose flag is not reset here
  // flags->canClose = 0; // 0 - do not close socket
                      // otherwise - close socket

  if(localPort != 80)
  {                   // I listen only to web request on port 80
   return(0) ;
  }

  // get 10 first bytes only of the request, the rest does not matter here
  for(i = 0 ; i < 10 ; i++)
  {
   getRequest[i] = SPI_Ethernet_getByte() ;
  }

  getRequest[i] = 0 ;

  if(memcmp(getRequest, httpMethod, 5))
  {     // only GET method is supported here
   return(0) ;
  }

  httpCounter++ ;   // one more request done

  get_Request = getRequest[5];  // s , d

  if(get_Request == 's')  // utiliser pour <script src=/s></script>
  {
   // if request path name starts with s, store dynamic data in transmit buffer
   // the text string replied by this request can be interpreted as javascript statements
   // by browsers
   len = putConstString(httpHeader);     // HTTP header
   len += putConstString(httpMimeTypeScript); // with text MIME type

   // add AN2 value to reply
   IntToStr(ADC_Read(2), dyna);
   len += putConstString("var AN2=");
   len += putString(dyna);
   len += putConstString(";");

   // add AN3 value to reply
   IntToStr(ADC_Read(3), dyna);
   len += putConstString("var AN3=");
   len += putString(dyna);
   len += putConstString(";");

   // add PORTB value (buttons) to reply
   len += putConstString("var PORTB=");
   IntToStr(PORTB, dyna);
   len += putString(dyna);
   len += putConstString(";");

   // add PORTD value (LEDs) to reply
   len += putConstString("var PORTD=");
   IntToStr(PORTD, dyna);
   len += putString(dyna);
   len += putConstString(";");

   // add HTTP requests counter to reply
   IntToStr(httpCounter, dyna);
   len += putConstString("var REQ=");
   len += putString(dyna);
   len += putConstString(";");
  }
  else
  {
   //
   if(get_Request == 'd') // Commande PORTD
   {
   if( isdigit(getRequest[6]) )
   {
    digit_getRequest  = getRequest[6] - '0'; // numéro de port 0 à 7

    if( getRequest[7] == 'o' )  // Contact Ouvert (OFF)
     etat_interrupteur = 0;

    if( getRequest[7] == 'f' )  // Contact Fermer (ON)
     etat_interrupteur = 1;

    switch(digit_getRequest)
```

```
    {
      case 0: PORTD.B0 = etat_interrupteur; break;
      case 1: PORTD.B1 = etat_interrupteur; break;
      case 2: PORTD.B2 = etat_interrupteur; break;
      case 3: PORTD.B3 = etat_interrupteur; break;
      case 4: PORTD.B4 = etat_interrupteur; break;
      case 5: PORTD.B5 = etat_interrupteur; break;
      case 6: PORTD.B6 = etat_interrupteur; break;
      case 7: PORTD.B7 = etat_interrupteur; break;
    }
   }
  }
  //
 }

 if(len == 0)
 {         // what do to by default
   len =  putConstString(httpHeader);       // HTTP header
   len += putConstString(httpMimeTypeHTML); // with HTML MIME type
   len += putConstString(indexPageHEAD);    // HTML page first part
   len += putConstString(indexPageBODY);    // HTML page second part
   len += putConstString(indexPageBODY2);   // HTML page second part
 }

 return(len) ;                          // return to the library with the number of bytes to transmit
}

/*
 * this function is called by the library
 * the user accesses to the UDP request by successive calls to Spi_Ethernet_getByte()
 * the user puts data in the transmit buffer by successive calls to Spi_Ethernet_putByte()
 * the function must return the length in bytes of the UDP reply, or 0 if nothing to transmit
 *
 * if you don't need to reply to UDP requests,
 * just define this function with a return(0) as single statement
 *
 */
unsigned int  SPI_Ethernet_UserUDP(unsigned char *remoteHost, unsigned int remotePort,
                              unsigned int destPort, unsigned int reqLength, TEthPktFlags *flags)
{
  return 0;            // back to the library with the length of the UDP reply
}

/*
 * main entry
 */
void main()
{
 //ANSEL = 0x0C ;         // AN2 and AN3 convertors will be used
 //C1ON_bit = 0;          // Disable comparators
 //C2ON_bit = 0;
 PORTA = 0 ;
 TRISA = 0xff ;        // set PORTA as input for ADC

 //ANSELH = 0;           // Configure other AN pins as digital I/O
 PORTB = 0 ;
 TRISB = 0xff ;        // set PORTB as input for buttons

 PORTD = 0 ;
 TRISD = 0 ;           // set PORTD as output

 /*
  * starts ENC28J60 with :
  * reset bit on RC0
  * CS bit on RC1
  * my MAC & IP address
  * full duplex
  */
 SPI1_Init();
 SPI_Ethernet_Init(myMacAddr, myIpAddr, Spi_Ethernet_FULLDUPLEX) ;

 while(1)
 {                     // do forever
   /*
    * if necessary, test the return value to get error code
    */
   SPI_Ethernet_doPacket() ;   // process incoming Ethernet packets

   /*
    * add your stuff here if needed
    * Spi_Ethernet_doPacket() must be called as often as possible
    * otherwise packets could be lost
    */
 }
}
```

The image part with relationship ID rId9 was not found in the file.