Lab 2.4

Internet application development using PHP

Objectives

After completing this lab, you will be able to:

- Discuss basic concepts of PHP Internet application development.
- Appreciate the features of different types of Internet programming.
- Develop widgets for easy integration into other Web applications.

Synopsis

This is the third lab on Internet application development. In the previous two labs, you learned how versatile CGI can be when it comes to Internet programming. In this lab, rather than going further into CGI, we will introduce you to another approach for Internet programming: using PHP. PHP is a widely used open-source general-purpose scripting language that is especially suited for Web development. It can operate either as a CGI interpreter (similar to Perl in the previous lab) or as an integrated module of the Web server. The latter use has won PHP great popularity, earning it a place in the renowned Web development framework LAMP (i.e. Linux, Apache, MySQL and PHP). You will experience the ease of use and efficiency of PHP in this lab.

Unlike C or Perl, HTML tags can be embedded directly into a PHP script, so PHP code combines with HTML tags to produce a response to the browser which accesses the script. Furthermore, after a Web server, like our LabBook Support Server, is configured with a PHP module, a PHP script (with file extension `.php`) can be accessed by Web browsers without needing to change its permission to be executable. This is another distinguishing feature of PHP. Its features and capabilities, especially those distinguishing it from similar languages like C and Perl, will be highlighted throughout this lab.

In the final part of the lab, we will develop a very interesting Web *widget* which can display the latest weather information retrieved live from the Hong Kong Observatory and, more excitingly, which can be easily added to any webpage, as shown in the figure below.[1]

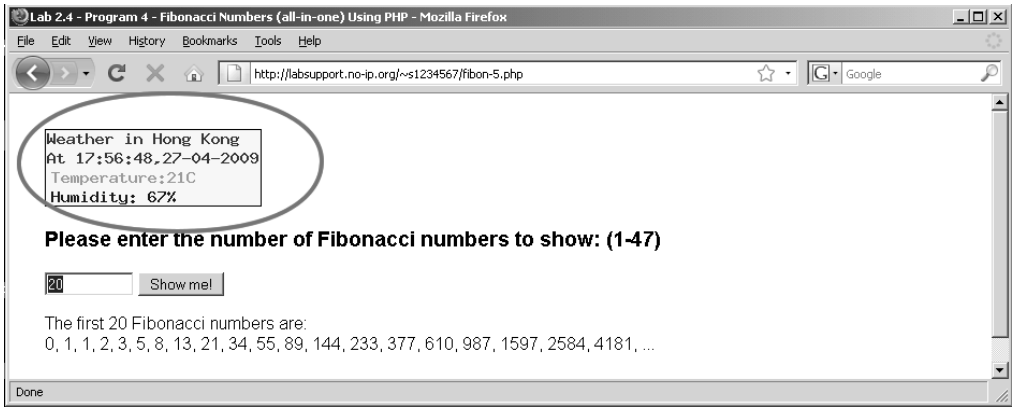

Figure 2.4.1

After completing this lab and the previous two labs, you should have ample understanding of Internet application development and the current landscape of this technical discipline.

Prerequisites

- You have a user account with remote login privileges on the LabBook Support Server, which has been configured to handle CGI scripts under your personal homepage directory.
- You are familiar with the use of an SSH client such as PuTTY.
- Your PC is connected to the Internet.
- You are familiar with basic UNIX commands.
- You have a basic understanding of Internet application development. (Otherwise, please complete or review Labs 2.2 and 2.3 before doing this lab.)

Background reading and preparation

Read the following online resources before doing this lab:

- 'A simple tutorial on PHP' — http://www.php.net/manual/en/tutorial.php

  You are expected to spend no more than 15 minutes reading the following three topics of this tutorial, which focus on the Web development process: 'Your first PHP-enabled page,' 'Something useful' and 'Dealing with forms.'

[1] A Web widget is a portable chunk of code that can be installed and executed within any separate HTML-based webpage by an end user without requiring additional compilation.

- 'A brief introduction to PHP' — http://dbs.cs.utk.edu/tutorial/phpintro.php

  You are expected to spend no more than ten minutes reading the following topics of this article, which focus more on the PHP language itself: 'Source code structure,' 'Operators,' 'Variables,' 'Control structures' and 'Arrays.'

You are expected to spend no more than 25 minutes on this preparation. Optionally, you may read on to finish the remaining topics of the above resources if time allows.

Expected duration

Approximately 90 minutes (including background reading and preparation)

Procedure

Step 1 Writing your first PHP program

Let's begin by writing a simple PHP program to let you get a general idea of how PHP works. Launch PuTTY (or another SSH client that you prefer) and connect to the LabBook Support Server as in Lab 1.1.

The simple program is shown in Listing 2.4.1 below. Use your favourite text editor to enter the program and save the file as `myfirst.php`.

```
1   <?php
2   # This is a simple PHP program
3   $name = "David";
4   print "Hello, $name!\n";
5   ?>
```

Listing 2.4.1 `myfirst.php`

Note: The line numbers in this lab's program listings are not part of the programs. They are shown for the convenience of explaining the programs.

The PHP code is enclosed in the special start and end processing instructions `<?php` and `?>`, which appear in lines 1 and 5 of the above listing. Similar to Perl, all variables in PHP are preceded by the `$` sign (see line 3). To execute the program, type `php myfirst.php`.

```
[s1234567@labsupport ~]$ php myfirst.php
Hello, David!
[s1234567@labsupport ~]$
```

Congratulations! You have finished your first PHP program. But more can be done with the program.

In Lab 2.2 Step 1, you verified the existence of the `public_html` directory under your home directory. There should also be a `cgi-bin` directory under `public_html`. If you have removed these two directories, create them again and set up the proper access permissions for them by following the instructions in Lab 1.8 Step 1.

Put the PHP program under the `~/public_html` directory and type `chmod +x myfirst.php` to make it readable by everyone. Then, access the program using your Web browser to go to the link below:

http://labsupport.no-ip.org/~s1234567/myfirst.php

Remember to replace s1234567 with your own account name. The result is shown below.

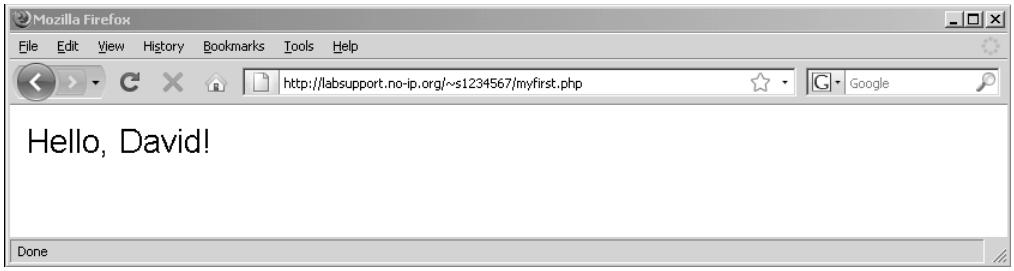

Figure 2.4.2

Not only can the PHP program work in console mode; it can also work in Web access mode. Amazingly enough, it needs no modification for such a switch. This shows that PHP is specially designed for Web application development. Also, we don't need to enable executable permission for a PHP program file, unlike with Perl or other script programs.

Note: A PHP program (with the file extension `.php`) can be accessed by Web browsers directly if the Web server, like our LabBook Support Server, has been configured with the PHP module.

Step 2 Developing a Fibonacci numbers program using PHP

Now, we will use PHP to develop a Fibonacci numbers program as we did in previous labs. Go back to Lab 2.2 to refresh your memory of Fibonacci numbers if necessary.

The program is shown in the listing below. Use your favourite text editor to enter the program and save the file as `fibon-2.php`.

```
1   <?php
2   # File: fibon-2.php
3   # Lab 2.4 - Program 2 - Fibonacci Numbers in PHP
4   # This program calculates the first MAX_N Fibonacci numbers and prints them out
5   #
6
7   define ("PROGRAM_NAME","Lab 2.4 - Program 2 - Fibonacci Numbers (PHP)");
8   define ("MAX_N",20);
9
10  $fibonacci = array(0,1);
11  $n = 1;
12
13  print PROGRAM_NAME."\n";
14  print "The first ".MAX_N." Fibonacci numbers are:\n";
15  while (sizeof($fibonacci) < MAX_N) {
16      $n = $fibonacci[sizeof($fibonacci) - 1] + $fibonacci[sizeof($fibonacci) - 2];
17      array_push($fibonacci, $n);
18  }
19
20  foreach ($fibonacci as $no) {
21      print $no.", ";
22  }
23  print "...\n";
24  exit(0);
25  ?>
```

Listing 2.4.2 `fibon-2.php`

The program above may not be that exciting, since it resembles the equivalent Perl version we worked with in Lab 2.3 (i.e. `fibon-2.pl`). You probably won't have much difficulty understanding the program. If necessary, refer to the background reading for explanations of the functions (e.g. `define()`, `array()`, etc.) and control structures (e.g. `foreach`) used above. However, it does have features which are different from its Perl counterpart. For instance, it can be accessed by Web browsers without any modification as long as it is placed under the `~/public_html` directory.

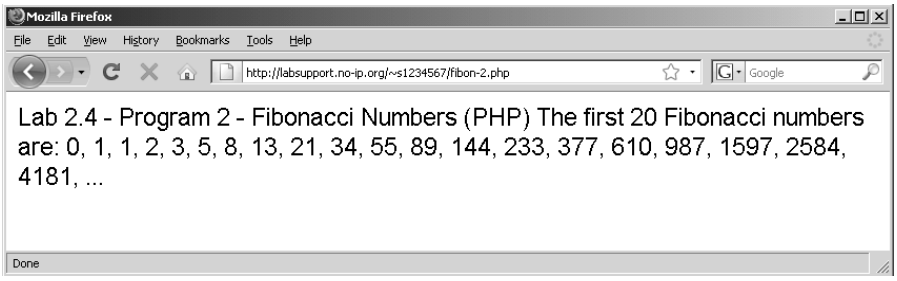

Figure 2.4.3

We modify the program further as below to show a striking difference between the PHP and Perl languages.

```
1   <!—
2   # File: fibon-3.php
3   # Lab 2.4 - Program 3 - Fibonacci Numbers in PHP
4   # This program calculates the first MAX_N Fibonacci numbers and prints them out
5   -->
6
7   <H3>Lab 2.4 - <font color=blue>Program 3</font> - Fibonacci Numbers (PHP)<H3>
8
9   <?php
10  define ("MAX_N",20);
11  $fibonacci = array(0,1);
12  $n = 1;
13  ?>
14
15  <p>The first <? print MAX_N ?> Fibonacci numbers are:</p>
16  <? while (sizeof($fibonacci) < MAX_N) {
17      $n = $fibonacci[sizeof($fibonacci) - 1] + $fibonacci[sizeof($fibonacci) - 2];
18      array_push($fibonacci, $n);
19  }
20
21  foreach ($fibonacci as $no) {
22      print $no.", ";
23  }
24  ?>
25  ...</p>
```

Listing 2.4.3 `fibon-3.php`

As you can see from the above listing, the PHP program contains some HTML tags to produce output to Web browsers while it also contains PHP code to carry required logic (i.e. calculating Fibonacci numbers in this case). In previous labs, we saw that C and Perl do not allow a mix of HTML tags and C and Perl program code respectively; and output to browsers has to be made through some function calls (i.e. `printf()` in C and `print()` in Perl). In PHP, program code is enclosed in the special start and end processing instructions `<?php` and `?>`, which allow you to hop into and out of PHP mode flexibly. This is the predominant usage form among PHP-based applications.

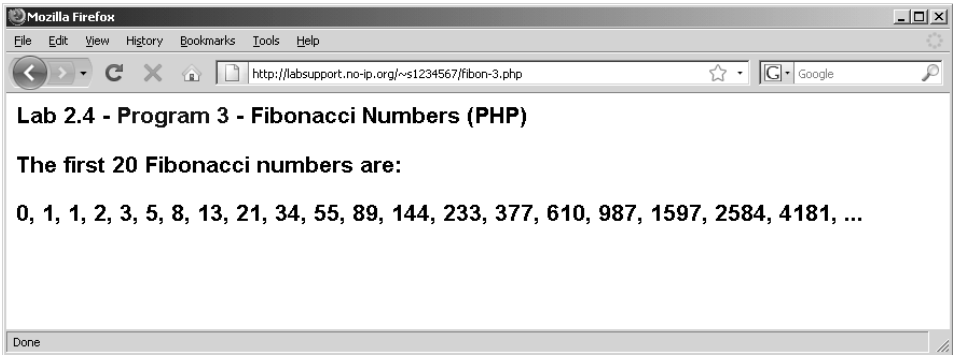

Figure 2.4.4

Other Web programming languages like ASP (Active Server Pages) of the .NET Framework and JSP (JavaServer Pages) of the Java framework also support the usage form shown above for PHP.

Step 3 Developing an Internet application using PHP

Now we will enhance `fibon-3.php` to enable it to receive and validate input from Web browsers. Use the listing `fibon-4.php` below to prepare the program under your homepage directory `~/public_html`.

```
1   <!--
2   # File: fibon-4.php
3   # Lab 2.4 - Program 4 - Fibonacci Numbers in PHP with browser input and validation
4   -->
5
6   <html>
7   <head>
8   <script language="JavaScript">
9      function setFocus() {
10     document.FibonacciForm.max_n.select();
11     document.FibonacciForm.max_n.focus();
12  }
13  </script>
14  </head>
15  <title>Lab 2.4 - Program 4 - Fibonacci Numbers (all-in-one) Using PHP</title>
16  <body style="padding: 25px" onload="setFocus()">
17
18  <form name="FibonacciForm" method="post" action="fibon-4.php ">
19     <h3>Please enter the number of Fibonacci numbers to show: (1-47)</h3>
20     <input name="max_n" value="20" size="10" maxlength="2" />
21     <input name="submit" type="submit" value="Show me!" />
22  </form>
23
24  <?php
25
26  if (empty($_POST))
27     exit (0);
28
29  $max_n = $_POST['max_n'];
30
31  if ($max_n < 1 || $max_n > 47) {
32      print "Your input: ".$max_n." is invalid. Please enter an integer between 1 - 47";
33      exit(1);
34  }
35
36  $fibonacci = array(0,1);
37  $n = 1;
38  ?>
39
40  <p>The first <?php print $max_n  ?> Fibonacci numbers are:<br>
41  <?php while (sizeof($fibonacci) < $max_n ) {
42      $n = $fibonacci[sizeof($fibonacci) - 1] + $fibonacci[sizeof($fibonacci) - 2];
43      array_push($fibonacci, $n);
44  }
45
46  foreach ($fibonacci as $no)
47      print $no.", ";
48  ?>
49  ...</p>
50
```

Listing 2.4.4 `fibon-4.php`

`fibon-4.php` takes the all-in-one approach so that one program file can both provide a form for users to input data and process the submitted form data (including validation). Functionally, it is equivalent to its Perl counterpart `fibon-5.pl` in the last lab. However, the programming in `fibon-4.php` features the embedding of HTML tags, which is not allowed in Perl. Go back to review `fibon-5.pl` to see this difference. The result of `fibon-4.php` is shown in the figure below.

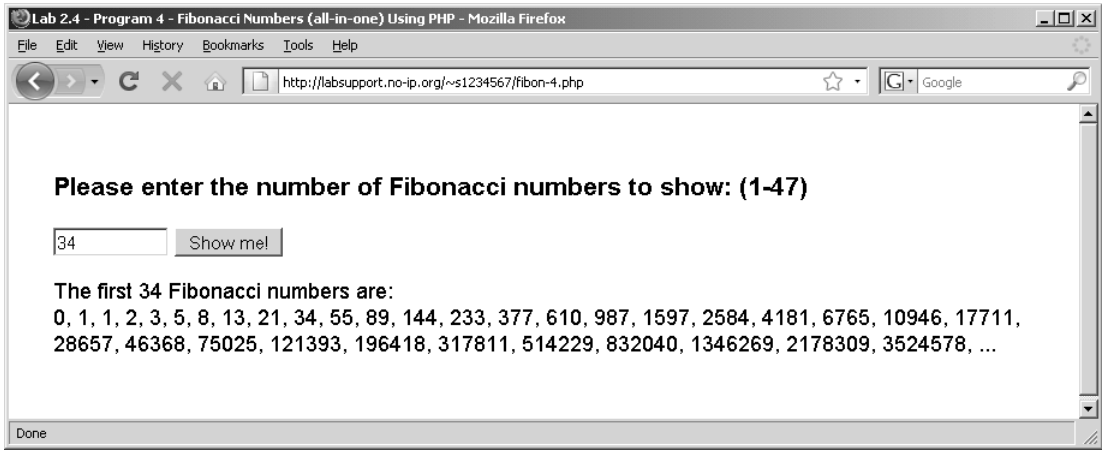

Figure 2.4.5

Step 4 Developing a Web widget using PHP

The previous steps put us in a good position to learn some more Internet programming. One current trend is to develop Web widgets, which are small programming components that can be easily integrated with other Web applications to provide added functionality without rewriting the code. We are now going to develop a Web widget that can display the latest weather information for Hong Kong. This information will be retrieved directly from the website of the Hong Kong Observatory. The following is the program listing.

```
<?php
// File: weather-widget.php
// Lab 2.4 -  A web widget that can display the latest weather information of Hong Kong.
//  The information is retrieved directly from the Hong Kong Observatory's web site at
//

// Retrieve information freshly from Hong Kong Observatory

$file = fopen("http://pda.hko.gov.hk/wxreporte.htm", "r") or exit("Unable to open file!");

//Extract the key information: temperature and relative humidity

while(!feof($file)){
  $line = fgets($file);
  if (preg_match("/(?:AIR TEMPERATURE : )(\d+) /", $line, $matches)) {
    $temp = $matches[1];
      } elseif (preg_match("/(?:RELATIVE HUMIDITY : )(\d+) /", $line, $matches)) {
    $humid = $matches[1];
    break;
   }
}
fclose($file);

//Generate the output as an image for easily integration with other web applications

header("content-type: image/png");
$img=ImageCreate(200,70);
$bgcolor2=ImageColorAllocate($img,245,245,245);
Imagefilledrectangle($img, 1, 4, 285, 50, $bgcolor2);
$red=ImageColorAllocate($img,255,0,0);
$green=ImageColorAllocate($img,30,205,10);
$blue=ImageColorAllocate($img,0,10,255);
ImageString($img,5,1,0,"Weather in Hong Kong",$red);
ImageString($img,5,1,18,"At ".date('H:i:s,j-m-Y'),$red);
ImageString($img,5,5,36,"Temperature:".$temp."C",$green);
ImageString($img,5,5,54,"Humidity: ".$humid."%",$blue);
Imagepng($img);
ImageDestroy($img);

?>
```

Listing 2.4.5 `weather-widget.php`

Use the listing `weather-widget.php` above to prepare the weather widget under your homepage directory `public_html`. We can make a copy of `fibon-4.php` into `fibon-5.php` under the directory `public_html` using the command:

```
cp fibon-4.php fibon-5.php
```

To integrate the weather widget, use vi or any other text editor to edit `fibon-5.php` as follows:

- Insert the following line of code into line 17:

```
<img src="weather-widget.php" width="200" height="70" id="image" border="1" />
```

- Replace line 18 with the following:

```
<form name="FibonacciForm" method="post" action="fibon-5.php ">
```

It is assumed that `weather-widget.php` is located in the same directory as `fibon-5.php`. Otherwise, you should change the value of `src` to the location of your widget (e.g. `src = 'http://labsupport.no-ip.org/~s1234567/weather.php'`). The result should be like the figure below.

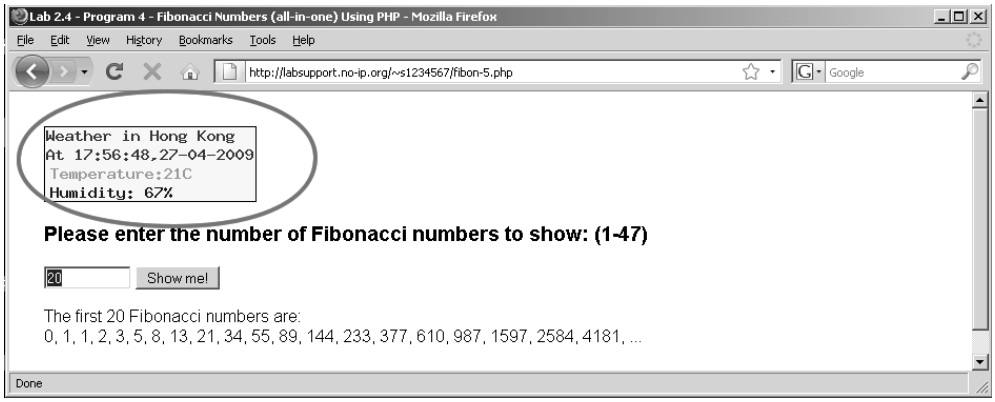

Figure 2.4.6

Every time you press the submit button, the widget will try to get the most up-to-date weather information and the current time for displaying. Referring to `weather_widget.php`, you can see the following useful techniques:

- Line 9 retrieves data from an external Web homepage as easily as opening a file.
- Lines 13–20 extract useful information using the regular expression technique.
- Lines 26–38 generate a graphical output based on the extracted information.

You have successfully finished this lab.

Summary

In this lab, you learned a different Internet programming paradigm, using PHP, in which you can combine HTML tags with PHP code to deliver the intended outcome to users at their Web browsers. In the previous two labs, we used C and Perl to implement Internet applications riding on the CGI mechanism. These programs do not allow the kind of code mixing that PHP does. Besides, PHP is fully geared to Internet application development, and we have seen in this lab that PHP makes development work very efficient. As in the previous lab, we have gone through the steps for carrying out form processing and server-side validation, which are common to Internet application development.

We also developed a useful weather information widget which can be integrated into other Web applications with the addition of just one line of code. After completing this and the previous two labs, you should have a good sense of the current landscape of Internet application development.

Further reading and useful resources

- 'PHP: Language reference' — http://www.php.net/manual/en/langref.php (for more about PHP syntax, data types, operators, control structures, predefined variables, functions, etc.)

Questions and exercises

1 Describe three main differences between Perl and PHP as far as Internet application development is concerned.

2 Find two websites with useful materials for learning PHP.

3 Referring to `fibon-4.php`, what is the function of lines 26 and 27? They are repeated below for easy reference:

```
if (empty($_POST))
    exit (0);
```

4 With reference to lines 15 and 17 of `weather-widget.php` and the weather report from the Hong Kong Observatory (http://pda.hko.gov.hk/wxreporte.htm), an instance of which is extracted below for reference, write a single line of code to extract the data about the `INTENSITY OF UV RADIATION` from the report using the `preg_match()` function.

```
...
<p align="center"><b><u>Current Weather</u></b></p>
   AT
   3 P.M.

   AT THE HONG KONG OBSERVATORY :<br>
   AIR TEMPERATURE : 25 DEGREES CELSIUS<br>
   RELATIVE HUMIDITY : 79 PER CENT
   <p></p>DURING THE PAST HOUR
   THE MEAN UV INDEX RECORDED AT KING'S PARK : 1<br><br>
   INTENSITY OF UV RADIATION : LOW<br><br><p>THE AIR TEMPERATURES AT OTHER PLACES
WERE:</p>
<table width="220" border="0" cellspacing="0" cellpadding="5">
...
```

Mini-project

Note: Be prepared to spend up to a few hours (perhaps even more) doing this mini-project. Good luck!

Referring to Step 4, the widget developed will refresh itself only when you press the submit button. Please modify `fibon-4.php` as `fibon-5.php` such that it will refresh the widget every five seconds automatically. Here are some hints:

- JavaScript should be used to inscribe the required functionality into `fibon-5.php`.
- Use the `SetInterval("doWork()", 5000)` function of JavaScript to involve another, i.e. `doWork()` every 5,000 milliseconds (that is, five seconds).
- Assign the `SetInterval()` function to the `onLoad` event so that it is triggered once `fibon-4.php` is loaded, e.g. `<body onload="setInterval('doWork()', 5000);">`.
- Define `myfunction()` within the `<script></script>` block to load the weather information as follows:

```
function doWork() {
var d=new Date().getTime();
 var imgsrc = "weather-widget.php"
   document.getElementById('image').src  = imgsrc +'?d='+d;
   return;
}
```