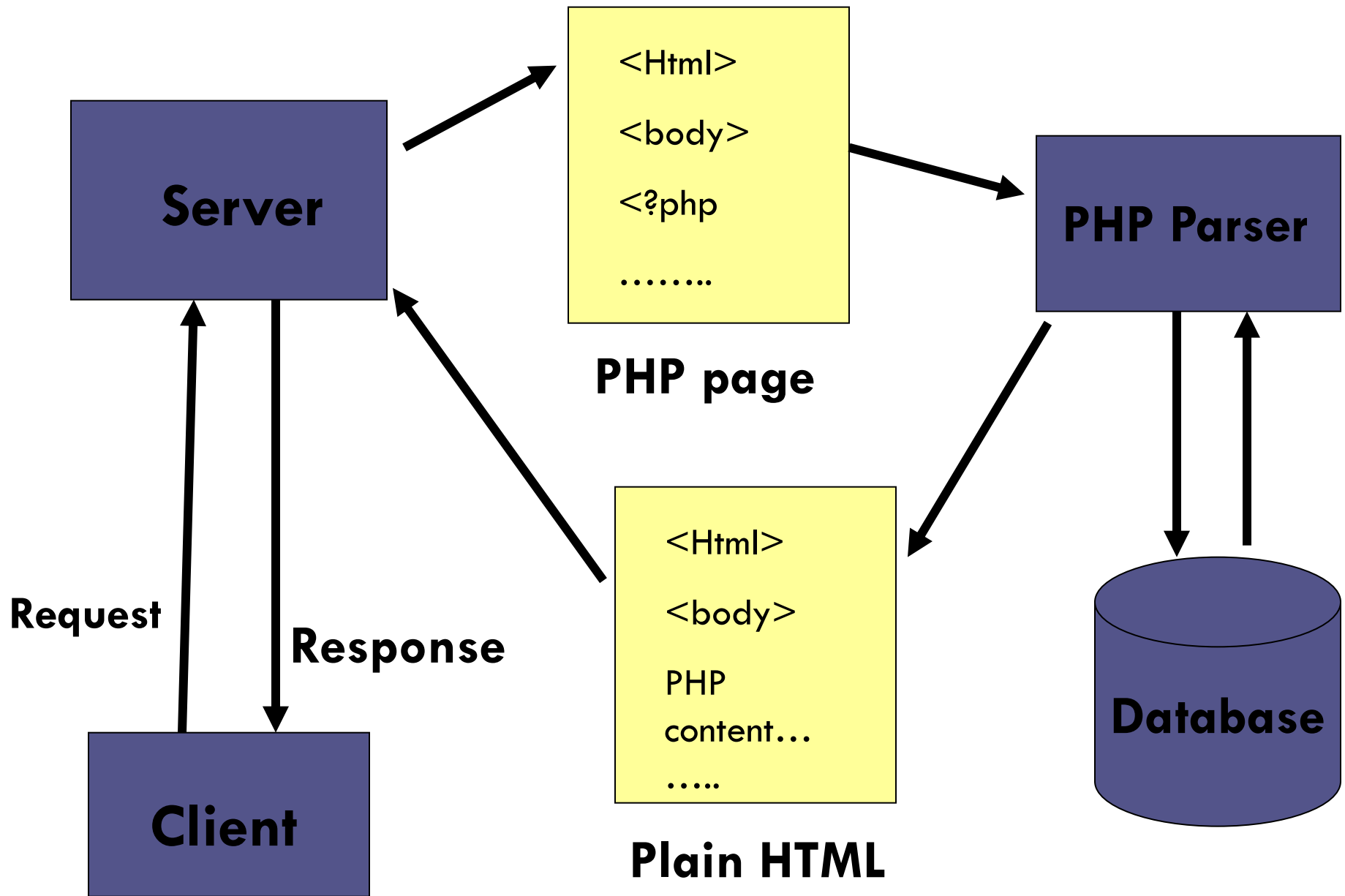


# PHP

## **PHP Hypertext Preprocessor**

- PHP is a **server-side** scripting language for creating dynamic and interactive web pages.
  - **Ie. Code is executed on the server.**
- A PHP page will generally contain PHP elements with HTML tags and other textual content.
- PHP is **case sensitive**.

- Files are saved with “.php” extension.
- Every PHP line terminates with a semicolon.
- PHP is an Open Source Software (OSS)



- When a web browser requests a PHP page from the server, the server calls up the PHP parser. The parser executes all the PHP script instructions on the page, and returns the file to the browser as plain HTML. During this process the PHP parser may also retrieve information from a database.

In order to develop and run PHP web pages three main components are needed:

- **Web Server**

- PHP will work with many of the Web Servers such as IIS. But is most often used with the freely available Apache server.

- **Database**

- PHP supports many database software such as MYSQL, Oracle, Informix, Sybase etc.

- **PHP Parser**

- A parser is required to process PHP script instructions and generate HTML output that can be sent to the web browser.

- All PHP code must be included inside one of the following markup tags:
  - `<?php php code ?>`
  - `<script language="php"> php code </script>`
  - `<? php code ?>`



## Example 1

- Write a PHP page to print a message

```
<html>
<body>
<h1>My first PHP page</h1>
<?php
echo "<h2>Hello world</h2>";
?>
</body>
</html>
```



Print "Hello World"

```
echo "<h2 align=center><font color=Blue face =  
Arial>Hello World</font></h2>";
```

- Comments

Comments can be added inside a PHP code block by using `//` and `#` between single line comments and `/*` and `*/` between multi-line comments.

```
<html>
```

```
<body>
```

```
<?php
```

```
/* Here is an introduction to  
Multi line comments */  
//Here is single line comment  
# Here is another single line comment  
echo "<h2>Hello world</h2>";
```

```
?>
```

```
</body>
```

```
</html>
```

# Variables

- All variables must begin with a **dollar (\$)** **sign** followed by a meaningful name.
- The name can contain any **letter, number** or the **underscore**, but the name cannot begin with a number.
  - Eg: \$abc  
\$my\_abc  
\$var123

## **Example:**

- Write a PHP program to print the value of a variable as HTML

```
<html>
```

```
<body>
```

```
<h1> Use of Variables</h1>
```

```
<?php
```

```
$today="Thursday";
```

```
echo "<h2>Today is $today</h2>";
```

```
?>
```

```
</body>
```

```
</html>
```



- Note:

A variable in PHP can store any type of value  
(eg: Integer, floating pt, string, boolean etc)

# Arrays

- The **array()** function can be used to define an array.
- **Numeric Arrays:**
  - A numeric array stores each array element with a numeric index.

- There are two methods to create a numeric array:

```
<html><body>
<h1>Numeric Arrays</h1>
<?php
//Method 1
$color=array("red","blue","green");
echo "My favorite color is: $color[1] <br>";
//Method 2
$course[1]="Software Eng";
$course[2]="Networks";
$course[3]="Comp Security";
echo "I am following the $course[1] course at NSBM"; ?>
</body></html>
```

- **Associative Arrays**

An array can also be defined using variable names as array elements.

- **Example:**

```
<html><body>
<h1>Associative Arrays</h1>
<?php
//Method 1
$prov=array("west"=>"Colombo","south"=>"Galle");
$p=$prov["west"];
echo "I live in $p <br><br>";
//Method 2
$fruit["red"]="Apple";
$fruit["orange"]="Orange";
$fruit["yellow"]="Pineapple";
$f=$fruit["yellow"];
echo "I like $f"; ?>
</body></html>
```

## Listing all the array elements

- All the array elements can be retrieved by using the PHP in-built function **foreach()**.

```
foreach(array as variable)
```

```
foreach(array as key =>value)
```

```
<html><body>
```

```
<h1>Associative Arrays</h1>
```

```
<?php
```

```
echo "Method 1 <br>";
```

```
$color=array("red","blue","green");
```

```
foreach($color as $col)
```

```
{echo "<li>I Like $col</li>";}
```

```
echo "<br>Method 2 <br>";
```

```
$prov=array("west"=>"Colombo","south"=>"Galle");
```

```
foreach($prov as $key => $val)
```

```
{echo "<li>Key: $key - Value: $val</li>"; } ?>
```

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

# **Handling Form Data**

- A web browser can send data to a web server in different ways such as using a URL, hyperlink or a form.



- PHP stores data coming from the browser in built-in variables:
  - **\$\_GET[]** – When the request is made using the GET method.
  - **\$\_POST[]** - When the request is made using the POST method.
  - **\$\_REQUEST[]** - When the request is made using the GET/POST method.

# Example

- Create an HTML page with a hyperlink to send the following data to the server. Create a PHP page to access this data.

Userid = 1001

Username = abc

Status = off

## Sender.html

```
<html>
```

```
<body>
```

```
<h1>Sending Data to the Server</h1>
```

```
<hr>
```

```
<a href =
```

```
"http://localhost/data1.php?userid=1001&username  
=abc&status=off"> Click here to send the data
```

```
</a>
```

```
</body>
```

```
</html>
```

```
<html><body>  
<h1>Receiving data</h1>  
<?php
```

```
$uid = $_REQUEST["userid"];  
$uname = $_REQUEST["username"];  
$status = $_REQUEST["status"];  
echo "User ID =$uid<br>";  
echo "User name =$uname<br>";  
echo "Status =$status";  
?>  
</body>  
</html>
```

- Example 2:

Create an HTML page with two hyperlinks to send the following data to the web server.

Create a PHP page to access this data and print them in HTML as given below:

## Data:

Type = Subject

Choice1 = English

Choice 2 = Maths

Type = Sports

Choice1 =Cricket

Choice 2 = Chess

## Output:

Favorite Subjects:

1. English

2. Maths

Or

Favorite Sport:

1. Cricket

2. Chess

```
<html><body>
```

```
<h1> Sending Data</h1>
```

```
<hr>
```

```
<a
```

```
  href="http://localhost/data2.php?type=subject&choice1=
  English&choice2=Maths">Your Favorite Subject
```

```
</a>
```

```
<hr>
```

```
<a
```

```
  href="http://localhost/data2.php?type=sport&choice1=Cri
  cket&choice2=Chess">Your Favorite Sport
```

```
</a>
```

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

```
<html><body>
<h1>Receive Data</h1>
<?php
$link = $_REQUEST["type"];
$choice1 = $_REQUEST["choice1"];
$choice2 = $_REQUEST["choice2"];
?>
<h2>Favorite <?php echo $link; ?></h2>
<ol>
<li><?php echo $choice1; ?></li>
<li><?php echo $choice2; ?></li></ol>
</body></html>
```



# Attributes of the Form Tag:

- **Name:**

An optional name assigned to the form.

- **Method:**

GET or POST- These are two commands that can be used to send and receive documents to and from web servers.

- **Action:**

Either the URL that identifies the web server and the program that is to receive the form's contents or send form data in the form of an email message.

## Example 3

- Create the following form in HTML. When the user clicks the login button, display the name and password on a separate page.

### **Enter your Data**

Name:

Password:

Submit form

```
<html><body>
<h1>Enter your Data</h1>
<form name="form1" method="POST"
  action="http://localhost/login.php" >
Name: <Input type="text" name="uname"><br>
Password: <Input type="password" name="pword">
  <br>
<input type="submit" value="Submit form">
</form>
</body> </html>
```

```
<html><body>
<h1>Displaying the Data Sent</h1>
<?php
$name = $_REQUEST["uname"];
$password = $_REQUEST["pword"];
echo "<h3>Name: $name</h3> <br>";
echo "<h3>Password: $password</h3>";
?>
</body></html>
```

- **Example 4**

Modify this example so that the user name and password will be displayed on the same page below the form.

## Example 5:

- Create the following form to enter the name and the marks of a student.

**Enter your marks**  
  
Name :   
  
**Marks:**  
  
Subject 1 :   
  
Subject 2 :

Create a PHP page to access this data and display:

- Name of the student
- The Total Marks
- Marks Average
- Status of the Student - if the student has got an average of 50 or above, display the status as "You have Passed" ,else display the status as "You have Failed".



```
<html><body>
<h2>Marks Details</h2>
<?php
    $name=$_REQUEST["sname"];
    $s1=$_REQUEST["s1"];
    $s2=$_REQUEST["s2"];
    echo "<h3>Name: $name</h3>";
    $sum=$s1+$s2;
    $avg=$sum/2;
    echo "Total Marks: $sum <br>";
    echo "Average Marks: $avg <br>";
    if($avg >= 50)
        echo "<b>Status : You have Passed</b>";
    else
        echo "<b>Status : You have failed</b>"; ?>
</body></html>
```

- Example 6:

Create the following form using HTML:

## **Guest Book**

Name

Email

Your Comments:

Sign the Guest Book

Cancel

- When the user presses the “sign the guest book” button display the following table with the data entered:

Thank you for signing the guest book

Your Name:	ABC
E-mail:	abc@yahoo.com
Comments:	Your site is great

```
<html><body>
<h2><u>Thank you for signing the guest book!!</u></h2>
<?php
$name=$_REQUEST["name"];
$mail=$_REQUEST["mail"];
$comment=$_REQUEST["comment"]; ?>
<table border="1" cellpadding="20">
<tr><th align="left">Your Name:</th>
<td><?php echo $name; ?></td></tr>
<tr><th align="left">Email:</th>
<td><?php echo $mail; ?></td></tr>
<tr><th align="left">Comments</th>
<td><?php echo $comment; ?></td></tr>
</table>
</body></html>
```

## Example 7:

- Write a PHP program to display the data entered into the following form:

### Room Reservation

Name:

Address:

Telephone:

No of rooms:

Type of room: Sea side ☐ Hill side ☐

Extras:

Meals

☐

Pool

☐

TV

☐

Reserve

Cancel

```
<html><body>
<h1> Reservation Details</h1>
<?php
$name=$_REQUEST["uname"];
$add=$_REQUEST["address"];
echo "Name: $name<br>";
echo "Address: $add<br>";

echo "Number of Rooms Reserved:";
$rm=$_REQUEST["rooms"];
echo "$rm <br>";

$type=$_REQUEST["type"];
echo "Type of Room Selected: $type<br>";
```

```
echo "<u>Extras Selected: </u><br>";  
if(isset($_REQUEST["ex1"]))  
{  
$ex1=$_REQUEST["ex1"];  
echo "$ex1 <br>";  
}  
if(isset($_REQUEST["ex2"]))  
{  
$ex2=$_REQUEST["ex2"];  
echo "$ex2 <br>";  
}  
if(isset($_REQUEST["ex3"]))  
{  
$ex3=$_REQUEST["ex3"];  
echo "$ex3 <br>";  
}  
echo "<h2>Thank you for Reserving!!!</h2>"; ?>  
</body></html>
```

## Example 8:

- Write a PHP program to display the data entered into the following form. Display the data below the form:

### Menu

What would you like to have for dinner?

Fried Rice ☐   Hoppers ☒   Bread ☐

What would you like to drink?

Tea ☐   Coffee ☐   Fruit drink ☒



```
<html><body>
```

```
<form method="POST">
```

```
<h2>What would you like to have for dinner?</h2>
```

```
Fried Rice <input type="checkbox" name="ch1" value="Fried-  
rice">
```

```
Hoppers <input type="checkbox" name="ch2" value="Hoppers"  
checked>
```

```
Bread <input type="checkbox" name="ch3" value="Bread"><br>
```

```
<h2>What would you like to drink?</h2>
```

```
Tea <input type="checkbox" name="ch4" value="Tea">
```

```
Coffee <input type="checkbox" name="ch5" value="Coffee">
```

```
Fruit Drink <input type="checkbox" name="ch6" value="Fruit-  
Drink"><br>
```

```
<input type="SUBMIT" value="Order">
```

```
<input type="RESET" value="Cancel"></form>
```

```
<?php
echo "<h2>Your Order</h2>";
echo "You like to have for dinner<br><br>";
if(isset($_REQUEST["ch1"]))
{$ch1=$_REQUEST["ch1"];
echo "$ch1 <br>";}
if(isset($_REQUEST["ch2"]))
{$ch2=$_REQUEST["ch2"];
echo "$ch2 <br>";}
if(isset($_REQUEST["ch3"]))
{$ch3=$_REQUEST["ch3"];
echo "$ch3 <br>";}
echo "You like to drink<br><br>";
```

```
if(isset($_REQUEST["ch4"]))
{$ch4=$_REQUEST["ch4"];
echo "$ch4 <br>";}
if(isset($_REQUEST["ch5"]))
{$ch5=$_REQUEST["ch5"];
echo "$ch5<br>";}
if(isset($_REQUEST["ch6"]))
{$ch6=$_REQUEST["ch6"];
echo "$ch6<br>";}
?>
</body><html>
```

## Example 9:

- Write a PHP program to display the data entered into the following form in a table:

### Order Form

Name:

Email:

Registration password:

Country  ▼

Do you want our newsletter?    Yes ☒ No ☐

Any Comments?

```
<html><body>
<h1>Order Details</h1>
<?php
$name=$_REQUEST["cname"];
$mail=$_REQUEST["mail"];
$pw=$_REQUEST["pword"];
$ct=$_REQUEST["country"];
$news=$_REQUEST["nl"];
$comm=$_REQUEST["comment"]; ?>
```

```
<table border="1" cellpadding="15">
<tr><th align="left">Customer Name:</th>
<td><?php echo $name; ?></td></tr>
<tr><th align="left">Email Address:</th>
<td><?php echo $mail; ?></td></tr>
<tr><th align="left">Password:</th>
<td><?php echo $pw; ?></td></tr>
<tr><th align="left">Registration Country:</th>
<td><?php echo $ct; ?></td></tr>
<tr><th align="left">Newsletter Required?:</th>
<td><?php echo $news; ?></td></tr>
<tr><th align="left">Customer Comments:</th>
<td><?php echo $comm; ?></td></tr>
</table></body></html>
```

## Example 10:

- Write a PHP program to display the data entered into the following form in a table:

List values:

Mahogany

Pine

Maplewood

Teak

### Registration Form

Name:

Address:

Password:

Select a Size:

☐ King ☐ Twin ☒ Small

Extras:

☐ Drawers ☒ Mattress ☐ Casters

Type of Wood:

Please Share your comments with us:

Comments.....

# **File Uploads**

- Creating an upload form

Example 1:

create the following form using HTML to upload a file.

## **Uploading a file**

Select the file to be uploaded

<input type="text"/>	<input type="button" value="Browse..."/>
<input type="button" value="Send File"/>	



- Note:

Only **POST** requests can be used for file uploads.

```
<html><body>
```

```
<h3>Uploading a file</h3>
```

```
<h4>Select the file to be uploaded</h4><br>
```

```
<form action="http://localhost/uploadfile.php"  
  method="POST" enctype="multipart/form-  
data">
```

```
<input type="file" name="file1" size="50">
```

```
<br>
```

```
<input type="SUBMIT" value="Send File">
```

```
</form>
```

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

## Reading Uploaded Files

- Once the files are uploaded the properties of the uploaded file can be accessed using the array **`$_FILES[]`**.

## Example:

- `$_FILES["file1"]["name"]`

Returns the **actual name** of the uploaded file.

- `$_FILES["file1"]["type"]`

Returns the **type of file** uploaded.

- `$_FILES["file1"]["size"]`

Returns the **size of file** uploaded.

- File1 -> name of the file control, which is used to refer to the file uploaded.

- Example 1:

Write a PHP page to display actual name, file type and size of an uploaded file.

```
<html><body>
<h2>File information</h2>
<hr>
<?php
    $fname=$_FILES["file1"]["name"];
    $type=$_FILES["file1"]["type"];
    $size=$_FILES["file1"]["size"];
    echo "File name: $fname <br>";
    echo "File type: $type <br>";
    echo "File size: $size <br>";
    ?>
</body></html>
```

- **Note:**

when a file is uploaded, PHP stores the file in a temporary folder in the server, with a temporary file name. As soon as the request is over, this file gets deleted from the server.

To keep the file in the server, the file should be moved to a different location in the server.



- The temporary filename of the file stored in the server, can be accessed by using:  
**`$_FILES["file1"]["tmp_name"]`**

# **To move uploaded files to a different location**

```
move_uploaded_file (temp file name, “new location”);
```

## **Note:**

If the file is moved successfully a “true” value is returned.

Else a “false” value is returned.

- Example 2:

Write a PHP page to move to a new location an uploaded file.

```
<html><body>
```

```
<h1><center>Copying uploaded  
file</center></h1>
```

```
<?php
```

```
$fname=$_FILES["file1"]["name"];
```

```
$ftmp=$_FILES["file1"]["tmp_name"];
```

```
$flag=move_uploaded_file($ftmp,"D:/$fname");
```

```
if($flag)
```

```
    echo "File moved successfully";
```

```
else
```

```
    echo " Sorry couldn't move";
```

```
?>
```

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

# Opening and Closing Files

- To read or write to a file, first the file should be opened. In PHP, the function **fopen()** is used to open the temporary file.

```
$fp = fopen(filename, "mode");
```

- `fopen()` returns a **file pointer** (an integer value).  
This value can be used, to work with files.

Mode	Description
r	Open for reading only

- **fgets()** – Read one line of text.
- **fclose()** – Close the temporary file
- **feof()** -

It is a function that can be used to determine whether the program has reached the end of the file.

- **Example 3**

Write a program to open and read a file using PHP.



```
<html><body>
```

```
<h2>Reading a File</h2>
```

```
<?php
```

```
$fp=fopen($_FILES["file1"]["tmp_name"],"r");
```

```
while(!feof($fp))
```

```
{ $text = fgets($fp);
```

```
echo "$text <br>"; }
```

```
fclose($fp);
```

```
?>
```

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

- **Example 4**

Create the following form using HTML.

**Application Form**

Name:

E-mail:

CV:

When the above form is submitted, write a PHP page to display the name, e-mail address and the contents of the CV.

```
<html><body>
<h2><u>Application Form</u></h2>
<form method="POST" action="http://localhost/file2.php"
  enctype="multipart/form-data">
Name:<input type="text" name="aname"><br>
E-mail: <input type="text" name="email"><br>
CV:<input type="file" name="file1" size="30"><br>
<input type="SUBMIT" name="SUBMIT" value="Send
  Application">
<input type="RESET" name="RESET" value="Reset">
</form>
</body></html>
```

```
<html><body>
<h2>Data of Applicant</h2>
<table border=1>
<tr><th>Name:</th>
<td><?php $name=$_REQUEST["aname"];
echo $name ?></td></tr>
<tr><th>E-mail:</th>
<td><?php $mail=$_REQUEST["email"];
echo $mail ?></td></tr>
</table><br>
<h3><u>CV Details:</u></h3>
<?php $handle = fopen($_FILES["file1"]["tmp_name"],"r");
while(!feof($handle))
{$text = fgets($handle);
echo "$text <br>"; }
fclose($handle); ?>
</body></html>
```