B.C.A. (Sem – VI)

B.C.A. - 601

Building Application Using PHP

Purushottam Singh

Unit:- 4

Unit-4

Email, Web Services, XML: -

E-Mail:-

Understanding E-Mail: -

- PHP must be configured correctly in the php.ini file with the details of how your system sends email. Open php.ini file available in /etc/ directory and find the section headed [mail function].
- Windows users should ensure that two directives are supplied.
- The first is called SMTP that defines your email server address.
- The second is called sendmail_from which defines your own email address.
- The configuration for Windows should look something like this –

```
[mail function]
; For Win32 only.
SMTP = smtp.secureserver.net

; For win32 only
sendmail_from = name@ispname.com
```

- Linux users simply need to let PHP know the location of their sendmail application. The path and any desired switches should be specified to the sendmail_path directive.
- The configuration for Linux should look something like this -

```
[mail function]
; For Win32 only.
SMTP =

; For win32 only
sendmail_from =

; For Unix only
sendmail_path = /usr/sbin/sendmail -t -i
```

Sending E-Mail: -

- PHP makes use of mail() function to send an email.
- This function requires three mandatory arguments that specify the recipient's email address, the subject of the the message and the actual message additionally there are other two optional parameters.

Syntax: -

```
mail( to, subject, message, headers, parameters );
```

- Here is the description for each parameter.

Sr.No	Parameter & Description
1	**To** Specifies the receiver / receivers of the email
2	**Subject** Specifies the subject of the email. This parameter cannot contain any newline characters
3	**Message** Defines the message to be sent. Each line should be separated with a LF (\n). Lines should not exceed 70 characters
4	**Headers** (Optional) Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\r\n)
5	**Parameters** (Optional) Specifies an additional parameter to the send mail program

Example: -

```
<html>

  <head>
    <title>Sending HTML email using PHP</title>
  </head>

  <body>

    <?php
      $to = "xyz@somedomain.com";
      $subject = "This is subject";

      $message = "<b>This is HTML message.</b>";
      $message .= "<h1>This is headline.</h1>";

      $header = "From:abc@somedomain.com \r\n";
      $header .= "Cc:afgh@somedomain.com \r\n";
      $header .= "MIME-Version: 1.0\r\n";
      $header .= "Content-type: text/html\r\n";

      $retval = mail ($to, $subject, $message, $header);

      if( $retval == true ) {
        echo "Message sent successfully...";
      }else {
        echo "Message could not be sent...";
      }
    ?>

  </body>
</html>
```

Sending attachments with E-Mail: -

- To send an email with mixed content requires to set Content-type header to multipart/mixed. Then text and attachment sections can be specified within boundaries.
- A boundary is started with two hyphens followed by a unique number which can not appear in the message part of the email.
- A PHP function md5() is used to create a 32 digit hexadecimal number to create unique number.
- A final boundary denoting the email's final section must also end with two hyphens.

```
<?php
  // request variables // important
  $from = $_REQUEST["from"];
  $emaila = $_REQUEST["emaila"];
  $filea = $_REQUEST["filea"];

  if ($filea) {
    function mail_attachment ($from , $to, $subject, $message, $attachment){
      $fileatt = $attachment; // Path to the file
      $fileatt_type = "application/octet-stream"; // File Type

      $start = strrpos($attachment, '/') == -1 ?
        strrpos($attachment, '//') : strrpos($attachment, '/')+1;

      $fileatt_name = substr($attachment, $start,
        strlen($attachment)); // Filename that will be used for the
        file as the attachment

      $email_from = $from; // Who the email is from
      $subject = "New Attachment Message";

      $email_subject =  $subject; // The Subject of the email
```

```
        $email_txt = $message; // Message that the email has in it
        $email_to = $to; // Who the email is to

        $headers = "From: ".$email_from;
        $file = fopen($fileatt,'rb');
        $data = fread($file,filesize($fileatt));
        fclose($file);

        $msg_txt="\n\n You have recieved a new attachment message from $from";
        $semi_rand = md5(time());
        $mime_boundary = "==Multipart_Boundary_x{$semi_rand}x";
        $headers .= "\nMIME-Version: 1.0\n" . "Content-Type: multipart/mixed;\n" . "
          boundary=\"{$mime_boundary}\"";

        $email_txt .= $msg_txt;

        $email_message .= "This is a multi-part message in MIME format.\n\n" .
          "--{$mime_boundary}\n" . "Content-Type:text/html;
          charset = \"iso-8859-1\"\n" . "Content-Transfer-Encoding: 7bit\n\n" .
          $email_txt . "\n\n";

        $data = chunk_split(base64_encode($data));

        $email_message .= "--{$mime_boundary}\n" . "Content-Type: {$fileatt_type};\n" .
          " name = \"{$fileatt_name}\"\n" . //"Content-Disposition: attachment;\n" .
          //" filename = \"{$fileatt_name}\"\n" . "Content-Transfer-Encoding:
          base64\n\n" . $data . "\n\n" . "--{$mime_boundary}--\n";

        $ok = mail($email_to, $email_subject, $email_message, $headers);

        if($ok) {
          echo "File Sent Successfully.";
          unlink($attachment); // delete a file after attachment sent.
        }else {
          die("Sorry but the email could not be sent. Please go back and try again!");
        }
      }
      move_uploaded_file($_FILES["filea"]["tmp_name"],
        'temp/'.basename($_FILES['filea']['name']));

      mail_attachment("$from", "youremailaddress@gmail.com",
        "subject", "message", ("temp/".$_FILES["filea"]["name"]));
    }
  ?>

  <html>
    <head>

      <script language = "javascript" type = "text/javascript">
        function CheckData45() {
          with(document.filepost) {
            if(filea.value ! = "") {
              document.getElementById('one').innerText =
                "Attaching File ... Please Wait";
            }
          }
        }
      </script>

    </head>
    <body>

      <table width = "100%" height = "100%" border = "0"
        cellpadding = "0" cellspacing = "0">
```

```
<tr>
  <td align = "center">
    <form name = "filepost" method = "post"
    action = "file.php" enctype = "multipart/form-data" id = "file">

      <table width = "300" border = "0" cellspacing = "0"
        cellpadding = "0">

        <tr valign = "bottom">
          <td height = "20">Your Name:</td>
        </tr>

        <tr>
          <td><input name = "from" type = "text"
            id = "from" size = "30"></td>
        </tr>

        <tr valign = "bottom">
          <td height = "20">Your Email Address:</td>
        </tr>

        <tr>
          <td class = "frmtxt2"><input name = "emaila"
            type = "text" id = "emaila" size = "30"></td>
        </tr>

        <tr>
          <td height = "20" valign = "bottom">Attach File:</td>
        </tr>

        <tr valign = "bottom">
          <td valign = "bottom"><input name = "filea"
            type = "file" id = "filea" size = "16"></td>
        </tr>

        <tr>
          <td height = "40" valign = "middle"><input
            name = "Reset2" type = "reset" id = "Reset2" value = "Reset">
          <input name = "Submit2" type = "submit"
            value = "Submit" onClick = "return CheckData45()"></td>
        </tr>
      </table>

    </form>

    <center>
      <table width = "400">

        <tr>
          <td id = "one">
          </td>
        </tr>

      </table>
    </center>

  </td>
 </tr>
</table>

</body>
</html>
```

XML:-

What is XML: -

- The XML language is a way to structure data for sharing across websites.
- XML is easy to create. It looks a lot like HTML, except that you make up your own tags.
- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.
- XML plays an important role in many different IT systems.
- XML is often used for distributing data over the Internet.
- It is important (for all types of software developers!) to have a good understanding of XML.

Example: -

```
<html>
  <body>

    <?php
      $note=<<<XML

      <note>
        <to>Gopal K Verma</to>
        <from>Sairamkrishna</from>
        <heading>Project submission</heading>
        <body>Please see clearly </body>
      </note>

      XML;
      $xml=simplexml_load_string($note);
      print_r($xml);
    ?>

  </body>
</html>
```

XML Document and DTD: -

- An XML document with correct syntax is called "Well Formed".
- An XML document validated against a DTD is both "Well Formed" and "Valid".
- A "Valid" XML document is a "Well-Formed" XML document, which also conforms to the rules of a DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Note.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

- The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Web Services: -

Introduction to SAX Web Services: -

- The full form of SAX is Simple API for XML.

- SAX (Simple API for XML) is an application program interface (API) that allows a programmer to interpret a Web file that uses the Extensible Markup Language (XML) - that is, a Web file that describes a collection of data.
- SAX is an alternative to using the Document Object Model (DOM) to interpret the XML file.
- As its name suggests, it's a simpler interface than DOM and is appropriate where many or very large files are to be processed, but it contains fewer capabilities for manipulating the data content.
- SAX is an event-driven interface. The programmer specifies an event that may happen and, if it does, SAX gets control and handles the situation. SAX works directly with an XML parser.
- The benefits of the event driven parsing method include:
  - Easy readability of RSS feeds (or any other XML documents that you wish to see in a particular format).
  - Selective parsing of the XML document.
  - Extremely light on memory usage, especially compared to the DOM model.

Web Services Platform Elements:-

Web Services have three basic platform elements: SOAP, WSDL and UDDI.

- **SOAP:-**
  1. SOAP is a simple XML-based protocol to let applications exchange information over HTTP.
  2. SOAP stands for Simple Object Access Protocol.
  3. SOAP is a communication protocol.
  4. SOAP is a format for sending messages.
  5. SOAP is designed to communicate via Internet.
  6. SOAP is platform independent.
  7. SOAP is language independent.
  8. SOAP is based on XML.
  9. SOAP is simple and extensible.
  10. SOAP allows you to get around firewalls.

- **WSDL:-**
  1. WSDL is an XML-based language for describing Web services and how to access them.
  2. WSDL stands for Web Services Description Language.
  3. WSDL is based on XML.
  4. WSDL is used to describe Web services.
  5. WSDL is also used to locate Web services.

- **UDDI:-**
  1. UDDI is a directory service where businesses can register and search for Web services.
  2. UDDI stands for Universal Description, Discovery and Integration.
  3. UDDI is a directory for storing information about web services.
  4. UDDI is a directory of web service interfaces described by WSDL.
  5. UDDI communicates via SOAP.
  6. UDDI is built into the Microsoft .NET platform.

DTD:- Document Type Definition

1) mysqli affected_rows():- Return the number of affected rows in the previous mysql operation.
2) mysqli close():- close a previously open database connection
3) mysqli connect():- Opens a new connection to the mysql server
4) mysqli query():- Performs the query against the database.
5) mysqli connect_errno():- ex (mysqli_connect_errno())

query mysqli_query($con, $query)

connect mysqli_connect("example.com", "Anv", "abc123");