Web Programming

Unit 5: Advanced PHP & MySQL

Compiled by: Ketaki Ghawali

Ketaki.ghawali@vsit.edu.in

VSIT | Vidyalankar School of Information Technology
NAAC Accredited College

Vidyalankar School of Information Technology
Wadala (E),
Mumbai
www.vsit.edu.in

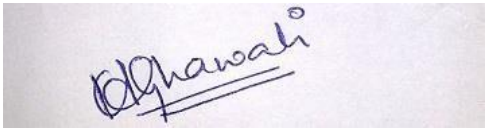
VSIT | Vidyalankar School of Information Technology

NAAC ACCREDITED COLLEGE

Certificate

This is to certify that the e-book titled "Web programming" comprises all elementary learning tools for a better understating of the relevant concepts. This e-book is comprehensively compiled as per the predefined eight parameters and guidelines.

Date: 10-12-2019



Ms. Ketaki Ghawali
Department of BSc IT

⚠ *DISCLAIMER: The information contained in this e-book is compiled and distributed for educational purposes only. This e-book has been designed to help learners understand relevant concepts with a more dynamic interface. The compilers of this e-book and Vidyalankar School of Information technology give full and due credit to the authors of the contents, developers and all websites from wherever information has been sourced. We acknowledge our gratitude towards the websites YouTube, Wikipedia, and Google search engine. No commercial benefits are being drawn from this project.*

Unit V :Advanced PHP and MySQL

Contents

- Php MySQL functions
- Integrating Web Forms and Databases
- Displaying Queries in Tables
- Strings and Regular Expressions
- Sessions
- Cookies
- HTTP
- E-Mail

Recommended Books

PHP 5.1 for Beginners by Ivan Bayross, Sharnam Shah, SPD

Prerequisites and Linking

Unit III	Pre-requisites	Linking				
Java Script	Sem-I	Sem. II	Sem. III	Sem. IV	Sem. V	Sem. VI
	C Programming	-	-	-	Enterprise Java	Project

PHP & MYSQL FUNCTIONS

- MySQL is a database system used on the web
- MySQL is a database system that runs on a server
- MySQL is ideal for both small and large applications
- MySQL is very fast, reliable, and easy to use
- MySQL uses standard SQL
- MySQL compiles on a number of platforms
- MySQL is free to download and use
- MySQL is developed, distributed, and supported by Oracle Corporation

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

DATABASE QUERIES

- A query is a question or a request.
- We can query a database for specific information and have a recordset returned.

MYSQL FUNCTIONS

1. mysql_connect: Open a connection to a MySQL server.

Parameters:
Server: the mysql server. It can also include a port number.
Username: the username assigned at the time of installation.
Password: the password.
Example:

```
<?php
$con=mysql_connect(“localhost”,”root”);
if(!$con)
{
die(“Error”);
}
?>
```

2. mysql_close: Close the mysql connection

Parameter:
Connection String
Example:

```
<?php
$con=mysql_connect(“localhost”,”root”);
if(!$con)
{
die(“Error”);
}

echo “Welcome!”;
mysql_close($con);
?>
```

3. mysql_create_db()

Attempts to create a database on the server associated with the specified link identifier.

Parameters:

Database_name: Name of the database to be created.

Link_identifier: the mysql connection.

Example:

```
<?php
$con=mysql_connect(“localhost”,”root”);
if(!$con)
{
die(“Error”);
}
if(mysql_create_db(“Test”,$con))
{
        echo “Database created successfully”;
}
else
echo “Error:”.mysql_error();
mysql_close($con);
?>
```

4. mysql_error()

Returns the text of error message from previous MySQL operation.

Example:

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Error connecting to mysql:".mysql_error());
}
?>
```

5. mysql_errno()

Returns the error number from the last MySQL function or 0 if no error occurred.

Example:

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Error connecting to mysql:".mysql_errno());
}
?>
```

6. mysql_fetch_array()

Returns an array that corresponds to the fetched row and moves the internal data pointer ahead.

Parameters:

Result: the result resource that is being evaluated. This result comes from a call to mysql_query().

Example:

```
<?php
$con= mysql_connect("localhost", "root")
if(!$con)
    die("Could not connect: " . mysql_error());

mysql_select_db("mydb");

$result = mysql_query("SELECT * FROM mytable");

while ($row = mysql_fetch_array($result)) {
    echo " $row[0]. $row[1]";
}
mysql_close($con);
?>
```

7. mysql_fetch_assoc()

Returns an associative array that corresponds to the fetched row and moves the internal data pointer ahead. It only returns an associative array.

Parameter:

Result: the result resource that is being evaluated.

```
<?php
$con= mysql_connect("localhost", "root")
if(!$con)
    die("Could not connect: " . mysql_error());

mysql_select_db("mydb");

$result = mysql_query("SELECT * FROM mytable");

while ($row = mysql_fetch_assoc($result)) {
    echo  $row[“userid”]. $row[“fullname”];
}
mysql_close($con);
?>
```

8. mysql_query()

mysql_query() sends a unique query to the currently active database on the server that is associated with the specified link identifier.

Parameters:

Query:

An sql query.

Link identifier:

The mysql connection.

Example:

```
<?php
$con = mysql_connect("localhost","root");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
```

```
 }

mysql_select_db("college11", $con);
$result = mysql_query("SELECT * FROM mytable");
?>
```

CREATING A DATABASE

To create and delete a database you should have admin privilege. Its very easy to create a new MySQL database. PHP uses **mysql_query** function to create a MySQL database. This function takes two parameters and returns TRUE on success or FALSE on failure.

SYNTAX:

```
mysql_query(SQL,Connection);
```

Sr.No	Parameter & Description
1	**sql** Required - SQL query to create a database
2	**connection** Optional - if not specified then last opend connection by mysql_connect will be used.

EXAMPLE:

```
<?php
$con = mysql_connect("localhost","root");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
$sql = "CREATE DATABASE college2";
if (mysql_query($sql,$con))
  {
  echo "Database college1 created";
  }
else
  {
  echo "Error creating database: " . mysql_error();
  }
mysql_close($con);
?>
```

SELECTING A DATABASE

- Once you establish a connection with a database server then it is required to select a particular database where your all the tables are associated.
- This is required because there may be multiple databases residing on a single server and you can do work with a single database at a time.
- PHP provides function **mysql_select_db** to select a database.It returns TRUE on success or FALSE on failure.

- SYNTAX:

mysql_select_db(dbname,connection);

Sr.No	Parameter & Description
1	**db_name** Required - Database name to be selected
2	**connection** Optional - if not specified then last opend connection by mysql_connect will be used.

EXAMPLE:

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Error connecting to mysql:".mysql_error());
}
mysql_select_db("college11",$con);
mysql_close($con);
?>
```

https://www.youtube.com/watch?v=5QHBj4brHNM

CREATING TABLES

- To create tables in the new database you need to do the same thing as creating the database.
- First create the SQL query to create the tables then execute the query using mysql_query() function.

EXAMPLE:

```
<? php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("could not connect".mysql_error());
}
mysql_select_db("college", $con);

$sql="CREATE TABLE student
(
sno int;
sname varchar(15);
percentage float;
)";
if(mysql_query($sql,$con))
{ echo("table created");}
else
echo "Error creating table:".mysql_error();

mysql_close($con);
?>
```

INSERTING RECORDS/DATA IN TABLE

```
<?php
//INSERTING RECORDS IN DATABASE
$con = mysql_connect("localhost","root","");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("my_dbase", $con);
$sql="INSERT INTO Persons VALUES ('KETAKI', 'GHAWALI', '20')";
if(mysql_query($sql,$con))
{
echo "Record inserted successfully";
}
mysql_query("INSERT INTO Persons VALUES ('PALLAVI', 'TAWDE', '21')");
mysql_close($con);
?>
```

https://www.youtube.com/watch?v=78RUaIe4X0A&t=2s

SELECTING DATA FROM TABLE

- Data can be fetched from MySQL tables by executing SQL SELECT statement through PHP function mysql_query.
- You have several options to fetch data from MySQL.
- The most frequently used option is to use function **mysql_fetch_array()**.
- This function returns row as an associative array, a numeric array, or both. This function returns FALSE if there are no more rows.

```
<?php
$con = mysql_connect("localhost","root");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("college1", $con);
$result = mysql_query("SELECT * FROM student",$con);
while($row = mysql_fetch_array($result))
  {
  echo $row[0] . " " . $row[1]." ".$row[2];
  echo "<br />";
```

```
 }
mysql_close($con);
?>
```

- PHP provides another function called **mysql_fetch_assoc()** which also returns the row as an associative array.

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Error connecting to mysql:".mysql_error());
}
mysql_select_db("college11",$con);
$sql="select * from student";
$result=mysql_query($sql,$con);
while($row=mysql_fetch_assoc($result))
{
echo $row['sno']." ".$row['sname']." ".$row['percentage']."<br>";
}
mysql_close($con);
?>
```

UPDATING DATA FROM TABLE

- Data can be updated into MySQL tables by executing SQL UPDATE statement through PHP function **mysql_query**.
- To update a record in any table it is required to locate that record by using a conditional clause.

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Could not connect:".mysql_error());
}
mysql_select_db("college11",$con);
```

```
$q="UPDATE student SET sname='Hrishikesh' WHERE sno='101'";
if(!mysql_query($q,$con))
{
echo "Error:".mysql_error();
}
else
echo "Table Updated";
mysql_close($con);
?>
```

DELETING DATA FROM TABLE

- Data can be deleted from MySQL tables by executing SQL DELETE statement through PHP function **mysql_query**.
- To delete a record in any table it is required to locate that record by using a conditional clause.

```
<?php
$con=mysql_connect("localhost","root");
if(!$con)
{
die("Could not connect:".mysql_error());
}
mysql_select_db("college11",$con);
$q="DELETE FROM student WHERE sno='101'";
if(!mysql_query($q,$con))
{
echo "Error:".mysql_error();
}
else
echo "Record Deleted";
mysql_close($con);
?>
```

INTEGRATING WEB FORMS AND DATABASES

- The PHP superglobals $_GET and $_POST are used to collect form-data.
- The example below displays a simple HTML form with two input fields and a submit button:

```
<html>
<body>

<form action="welcome.php" method="POST">
Name: <input type="text" name="name"><br>
Roll No: <input type="text" name="rno"><br>
Percentage: <input type="text" name="percentage"><br>
<input type="submit">
</form>
</body>
</html>
```

- When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named "welcome.php". The form data is sent with the HTTP POST method.

The "welcome.php" looks like this:

```
<?php
$con = mysql_connect("localhost","root");
if (!$con)
 {
 die('Could not connect: ' . mysql_error());
 }
mysql_select_db("college11", $con);
$sql="INSERT INTO student VALUES
($_POST[Sname],'$_POST[rno]',$_POST[percentage])";
if (!mysql_query($sql,$con))
 {
 die('Error: ' . mysql_error());
 }
echo "1 record added";
mysql_close($con);
?>
```

DISPLAYING QUERIES IN TABLES

- Very often you will need to use a MySQL table to store data inside it and then output that data by using a PHP script.
- To display the table data it is best to use HTML, which upon filling in some data on the page invokes a PHP script which will update the MySQL table.
- To populate a new database table with data you will first need an HTML page which will collect that data from the user.
- The next step will be to run a select query through the database table to display the required records.
- While displaying this content, we will include the html table code.

```
<?php
$con = mysql_connect("localhost","root");
if (!$con)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("college1", $con);
$result = mysql_query("SELECT * FROM student");
echo "<table border='1'>
<tr>
<th>Roll No</th>
<th>Student Name</th>
<th>Course</th>
</tr>";

while($row = mysql_fetch_array($result))
  {
  echo "<tr>";
  echo "<td>" . $row['Rno'] . "</td>";
  echo "<td>" . $row['Name'] . "</td>";
  echo "<td>".$row['Course']."</td>";
  echo "</tr>";
  }
echo "</table>";
```

```
mysql_close($con);

?>
```

SESSIONS

- Although you can store data using cookies but it has some security issues.
- Since cookies are stored on user's computer it is possible for an attacker to easily modify a cookie content to insert potentially harmful data in your application that might break your application.
- Also every time the browser requests a URL to the server, all the cookie data for a website is automatically sent to the server within the request.
- It means if you have stored 5 cookies on user's system, each having 4KB in size, the browser needs to upload 20KB of data each time the user views a page, which can affect your site's performance.
- You can solve both of these issues by using the PHP session.
- A PHP session stores data on the server rather than user's computer.
- In a session based environment, every user is identified through a unique number called session identifier or SID.
- This unique session ID is used to link each user with their own information on the server like emails, posts, etc.

Starting a PHP Session

- Before you can store any information in session variables, you must first start up the session.
- To begin a new session, simply call the PHP session_start() function.
- It will create a new session and generate a unique session ID for the user.
- The PHP code in the example below simply starts a new session.

```
<?php

session_start();

?>
```

Storing and Accessing Session Data

- You can store all your session data as key-value pairs in the $_SESSION[] superglobal array.
- The stored data can be accessed during lifetime of a session. Consider the following script, which creates a new session and registers two session variables.

```
<?php

// Start the session

session_start();

// Set session variables

$_SESSION["favcolor"] = "green";
```

```
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>
```

- To access the session data we set on our previous example from any other page on the same web domain — simply recreate the session by calling session_start() and then pass the corresponding key to the $_SESSION associative array.

```
<?php
// Echo session variables that were set on previous page
session_start();
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>
```

Destroying a Session

- If you want to remove certain session data, simply unset the corresponding key of the $_SESSION associative array, as shown in the following example:

```
<?php
session_start();
if(isset($_SESSION['favcolor']))
unset($_SESSION['favcolor']);
echo "Session variable destroyed!";
?>
```

COOKIES

- A cookie is a small text file that lets you store a small amount of data (nearly 4KB) on the user's computer.
- They are typically used to keeping track of information such as username that the site can retrieve to personalize the page when user visit the website next time.
- **Setting a Cookie**
- The setcookie() function is used to set a cookie in PHP.
- Make sure you call the setcookie() function before any output generated by your script otherwise cookie will not set.
- The basic syntax of this function can be given with:

```
setcookie(name,value,expiry,path,domain,secure);
```

Parameters:

Parameter	Description
name	The name of the cookie.
value	The value of the cookie. Do not store sensitive information since this value is stored on the user's computer.
expires	The expiry date in UNIX timestamp format. After this time cookie will become inaccessible. The default value is 0.
path	Specify the path on the server for which the cookie will be available. If set to /, the cookie will be available within the entire domain.
domain	Specify the domain for which the cookie is available to e.g www.example.com.
secure	This field, if present, indicates that the cookie should be sent only if a secure HTTPS connection exists.

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
```

- **Accessing Cookie Value**
- The PHP $_COOKIE superglobal variable is used to retrieve a cookie value.
- It typically an associative array that contains a list of all the cookies values sent by the browser in the current request, keyed by cookie name.
- The individual cookie value can be accessed using standard array notation, for example to display the username cookie set in the previous example, you could use the following code.

```
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
```

?>

- **Removing a Cookie**
- You can delete a cookie by calling the same setcookie() function with the cookie name and any value (such as an empty string) however this time you need the set the expiration date in the past, as shown in the example below:

```
<?php
setcookie($cookie_name, $cookie_value, time() - (86400 * 30), "/");
?>
```

https://www.youtube.com/watch?v=UZcVUfaby9U&t=2s

EMAIL

- Sending email messages are very common for a web application, for example, sending welcome email when a user create an account on your website, sending newsletters to your registered users, or getting user feedback or comment through website's contact form, and so on.

- You can use the PHP built-in mail() function for creating and sending email messages to one or more recipients dynamically from your PHP application either in a plain-text form or formatted HTML.
- The basic syntax of this function can be given with:

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

Parameters:

Parameter	Description
Required — The following parameters are required	
to	The recipient's email address.
subject	Subject of the email to be sent. This parameter i.e. the subject line cannot contain any newline character (\n).
message	Defines the message to be sent. Each line should be separated with a line feed-LF (\n). Lines should not exceed 70 characters.
Optional — The following parameters are optional	
headers	This is typically used to add extra headers such as "From", "Cc", "Bcc". The additional headers should be separated with a carriage return plus a line feed-CRLF (\r\n).
parameters	Used to pass additional parameters.

Sending Plain Text Emails

- The simplest way to send an email with PHP is to send a text email. In the example below we first declare the variables — recipient's email address, subject line and message body — then we pass these variables to the mail() function to send the email.

```
<?php
$to="ghawaliketaki@gmail.com";
$subject="test mail";
$message="Hello! This is simple email";
$from="ghawaliketaki@gmail.com"
$header="From:$from";
mail($to,$subject,$message,$header);
echo "Email sent";
?>
```

Sending HTML Formatted Emails

- When you send a text message using PHP, all the content will be treated as simple text. We're going to improve that output, and make the email into a HTML-formatted email.

- To send an HTML email, the process will be the same. However, this time we need to provide additional headers as well as an HTML formatted message.

```
<?php
$to = "ketaki.ghawali@vsit.edu.in";
$subject = "HTML email";
$message = "
<html>
<head>
<title>HTML email</title>
</head>
<body>
<p>This email contains HTML Tags!</p>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>John</td>
<td>Doe</td>
</tr>
</table>
</body>
</html>
";
// Always set content-type when sending HTML email
$headers = "MIME-Version: 1.0" . "\r\n";
$headers .= "Content-type:text/html;charset=UTF-8" . "\r\n";
// More headers
$headers .= 'From: <ghawaliketaki@gmail.com>' . "\r\n";
$headers .= 'Cc: ghawaliketaki@gmail.com' . "\r\n";
mail($to,$subject,$message,$headers);
?>
```

□ □ □ □ □

- **GQs**

1. What is a cookie? How to store and retrieve the values in cookie in PHP? Explain different MYSQL Functions.
2. Explain any five PHP/MySQL functions with example.
3. Write a PHP program to send email with attachment.
4. How to start and destroy session and how to store a session variable in PH? Explain.
5. Write a short note on regular expressions in PHP.
6. Write a PHP code to create a database “Company” and to create a table “Employee” (emp_id, emp_name, emp_dept, emp_salary)
7. Explain SQL queries to create a table and insert a row in the table.

a. Write a short note on cookies in PHP.

b. Write a PHP program to create a database named “College”. Create a table named “Student” with fields(sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 and 75 in a tabular format.
How mail is sent using PHP?

8. What is a character class in a regular expression? List and explain the pre -defined character classes.
9. Design a PHP page for authenticating a user.
10. Write a PHP program that uses regular expression functions to display all the uppercase characters from the string “This is a demonstration of PHP Regular Expressions”.
11. Write a short note on sessions in PHP.
12. Write a program to accept RollNo, Student Name, and Percentage from the user and save the values in the table Student(rno, sname, percentage) in database “College”.
13. Explain the mail function
14. Write a PHP program to create a cookie and retrieve its value.
15. Explain the following PHP/MySQL functions with example:
    i.mysql_query() ii. mysql_error()
16. List various fetching functions. Explain with example any two of them.
17. List and explain function used to create database programmatically.
18. Write a PHP program to display Boolean data from database on check button.
19. How to open and close a connection to MySQL Database Server?
20. Write a program to insert values into MySQL database.
21. How to build forms from queries?

- **Multiple Choice Questions:**

1.	How would you start a session?	
	a. session()	b. begin_session()
	c. session_start()	d. session(start)
2.	Mysql_connect() does not take following parameter:	
	a. database host	b. user ID
	c. password	**d. database name**
3.	How do you create a cookie in php?	
	a. setcookie()	b. makecookie()
	c. createcookie()	d. none

4.	Which function is used to close an earlier established connection?	
	a. mysql_close()	b. mysql_connect()
	c. mysql_exit()	d. None of the above
5.	…….. is a small text file that lets you store a small amount of data on the user's computer	
	a. http	b. session
	c. email	**d. cookie**
6.	To destroy a session the ……. function is used.	
	a. isset()	b. unset()
	c.session_destroy()	d. None of the above
7.	To create a session ……. is used.	
	a. isset()	**b.** session_destroy
	c.$_SESSION["variablename"]	d. All of the above
8.	A PHP ……. Stores data on the server side	
	a. cookie	**b. session**
	c. function	d. mysql
9.	There is a separate function to delete a cookie. True or False?	
	a. True	**b. False**
10.	Which function is used to send an email?	
	a. cookie	**b. email**
	c. session	d. None of the above