Unit-V

PHP & MYSQL

CS-504 (A) IWT

Syllabus (Unit-V)

UNIT 05

PHP and MySQL: Basic commands with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names, creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP myadmin and database bugs

BASIC COMMANDS WITH PHP EXAMPLES

Basics of PHP with commands are mentioned below:

echo

The “echo” command is used to display text or data on a web page or in the command line interface. It is often used to display dynamic content, such as the results of a form submission or the output of a PHP function.

```
echo "Hello";
```

BASIC COMMANDS WITH PHP EXAMPLES

print

In PHP, the “print” command is used to output text or data to the screen. It is similar to the “echo” command, but there are a few differences. For example, the “print” command can only take a single argument, whereas the “echo” command can take multiple arguments.

```
print "text to be displayed";
```

BASIC COMMANDS WITH PHP EXAMPLES

if/else

The “if” statement is a conditional statement that is used to execute code only if a certain condition is met. The basic syntax of the “if” statement is as follows:

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

BASIC COMMANDS WITH PHP EXAMPLES

The “else” statement is often used in conjunction with the “if” statement to execute code when the condition is false. The basic syntax of the “if-else” statement is as follows:

```
if (condition) {

 // code to be executed if condition is true

} else {

  // code to be executed if condition is false

}
```

BASIC COMMANDS WITH PHP EXAMPLES

switch

The “switch” statement is another type of conditional statement that can be used to execute different blocks of code based on the value of a variable or expression.

The basic syntax of the “switch” statement is as follows:

```
switch (expression) {
  case value1:
    // code to be executed if expression is equal to value1
    break;
  case value2:
    // code to be executed if expression is equal to value2
    break;
  // more cases can be added as needed
  default:
    // code to be executed if expression does not match any of the cases above
    break;
}
```

BASIC COMMANDS WITH PHP EXAMPLES

PHP Loops:

- **While Loop**: in PHP, while loop can be executed till the mention expression is considering as true.

```
WHILE [condition or expression] {
[CODE]
}
```

- **For Loop:** For loop is used to execute the same code for the mention number of times.

```
FOR (exp 1, exp 2, exp 3) {
[CODE]
}
```

- **Do While Loop**: Similar to the while loop, the code will be executed until the true value is in while expression. The main difference with while is, the code mention inside the do at least execute one whether the expression is true or not, but while not ensure the same.

```
DO {
[CODE]
} WHILE (condition)
```

- **FOREACH Loop**: This loop is accepting an array as a variable and considering of executing code till the last element of the array.

```
FOREACH ($arr_var as $val) {
[CODE]
}
```

Introduction to MySQL

Introduction to MySQL

- 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 several 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.

PHP Connect to MySQL

db_conn.php

```php
<?php
$servername = "localhost";
$username = “root";
$password = "";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

PHP Create a MySQL Database

A database consists of one or more tables.

You will need special CREATE privileges to create or to delete a MySQL database.

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
// Create database
$sql = "CREATE DATABASE student_info";
if ($conn->query($sql) === TRUE) {
  echo "Database created successfully";
} else {
  echo "Error creating database: " . $conn->error;
}
$conn->close();
?>
```

selecting a database

```php
<?php
$mysqli = new mysqli("localhost","root","","mysql");

if ($mysqli -> connect_errno) {
  echo "Failed to connect to MySQL: " . $mysqli -> connect_error;
  exit();
}
// Return name of current default database
if ($result = $mysqli -> query("SELECT DATABASE()")) {
  $row = $result -> fetch_row();
  echo "Default database is " . $row[0];
  $result -> close();
}
// Change db to "student_info" db
$mysqli -> select_db("student_info");

// Return name of current default database
if ($result = $mysqli -> query("SELECT DATABASE()")) {
  $row = $result -> fetch_row();
  echo "Database Changed, New Default database is " . $row[0];
  $result -> close();
}
$mysqli -> close();
?>
```

listing database

```php
<?php
$dbhost = 'localhost';
$dbuser = 'root';
$dbpass = '';
$mysqli = new mysqli($dbhost, $dbuser, $dbpass);

if($mysqli->connect_errno ) {
   printf("Connect failed: %s", $mysqli->connect_error);
   exit();
}
//printf('Connected successfully.');
if ($result = $mysqli->query("SHOW DATABASES")) {
   printf("Show Database executed successfully..!");
   echo "Database list are: ";
   while($row = mysqli_fetch_array($result)){
       print_r($row);
   }
}
if ($mysqli->errno) {
    printf("Could not find database: %s", $mysqli->error);
}
$mysqli->close();
?>
```

listing table names

```
<?php
//open database connection
$mysqli = new mysqli('localhost','root','','student_info');

//Display error message
if ($mysqli->connect_error) {
    die('Error : ('. $mysqli->connect_errno .') '. $mysqli-
>connect_error);
}

$sql="SELECT TABLE_NAME FROM information_schema.TABLES WHERE
TABLE_SCHEMA = 'student_info'";
$result=$mysqli->query($sql);
while ( $tables = $result->fetch_assoc())
{
echo "<br>".$tables['TABLE_NAME'];
}
// Free memory by clearing result
$result->free();

// close connection
$mysqli->close();
```

PHP MySQL Create Table

A database table has its own unique name and consists of columns and rows.

Create a MySQL Table Using MySQLi and PDO

The CREATE TABLE statement is used to create a table in MySQL.

We will create a table named “student", with five columns: “rollno", "firstname", "lastname", "email" and "reg_date":

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
// sql to create table
$sql = "CREATE TABLE student (
rollno INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
)";
if ($conn->query($sql) === TRUE) {
  echo "Table student created successfully";
} else {
  echo "Error creating table: " . $conn->error;
}
$conn->close();
?>
```

PHP MySQL Insert Data

Insert Data Into MySQL Using MySQLi and PDO

After a database and a table have been created, we can start adding data in them.

Here are some syntax rules to follow:

- The SQL query must be quoted in PHP
- String values inside the SQL query must be quoted
- Numeric values must not be quoted
- The word NULL must not be quoted

The INSERT INTO statement is used to add new records to a MySQL table:

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

PHP MySQL Insert Data

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO student (firstname, lastname, email)
VALUES ('Amit', 'Kumar', 'amit@gmail.com')";

if ($conn->query($sql) === TRUE) {
  echo "New record created successfully";
} else {
  echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

PHP MySQL Insert Multiple Records

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
$sql = "INSERT INTO student (firstname, lastname, email)
VALUES ('Anuj', 'Kumar', 'anuj@gmail.com');";
$sql .= "INSERT INTO student (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com');";
$sql .= "INSERT INTO student (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";

if ($conn->multi_query($sql) === TRUE) {
  echo "New records created successfully";
} else {
  echo "Error: " . $sql . "<br>" . $conn->error;
}
$conn->close();
?>
```

altering tables

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "ALTER TABLE student ADD branch VARCHAR(30) AFTER lastname";

if ($conn->query($sql) === TRUE) {
  echo "Table Altered successfully";
} else {
  echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

PHP MySQL Select Data

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT rollno, firstname, lastname FROM student";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
  // output data of each row
  while($row = $result->fetch_assoc()) {
    echo "rollno: " . $row["rollno"]. " - Name: " . $row["firstname"]. " " .
$row["lastname"]. "<br>";
  }
} else {
  echo "0 results";
}
$conn->close();
?>
```

PHP MySQL Delete Data

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

// sql to delete a record
$sql = "DELETE FROM student WHERE rollno=3";

if ($conn->query($sql) === TRUE) {
  echo "Record deleted successfully";
} else {
  echo "Error deleting record: " . $conn->error;
}

$conn->close();
?>
```

PHP MySQL Update Data

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "UPDATE student SET lastname='Singh' WHERE rollno=2";

if ($conn->query($sql) === TRUE) {
  echo "Record updated successfully";
} else {
  echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

deleting database

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "student_info";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}

$sql = "DROP DATABASE college_db";

if ($conn->query($sql) === TRUE) {
  echo "Databse Deleted successfully";
} else {
  echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

PHP myadmin and database bugs

phpMyAdmin is the most trusted and user-friendly database managers and mostly used for web-based applications or programs. In the following article, we will be learning about the importance of the phpMyAdmin tool in the web world.

phpMyAdmin | Pre-requisites: To install phpMyAdmin software, you need a server running platform like **Windows** or **Linux** supports operating systems.

- **Web Browser:** You need a web browser interface to run the tool.
- **PHP scripting language:** You need a server-side language.
- **Apache Web server:** You need a web server to store phpMyAdmin files.
- **MySQL or MariaDB Database:** You need a database to manage application data.

End of Unit – V

Thank You