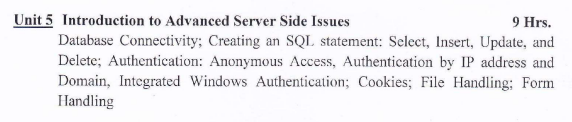
**Unit 5**

**Introduction to Advance Server-Side Issue**

****

# **Database Connectivity:**

PHP is the scripting language that runs in the application server which serves to implement business logic. MySQL is the database server/DBMS that implements the actual database. In order to make a dynamic web application, we need to implement different database applications using PHP. To perform database operations by using PHP, we need to establish a database connection between PHP and MySQL.

Following code demonstrates the steps to connect to the MySQL database server using PHP.

**Example**: Connect to MySQL database server.

<?php

// Connection parameters

$servername = "localhost";

$username = "username";//default is root

$password = "password";// default is blank

// Establish Connection

$conn= mysqli\_connect($servername,$username,$password);

// Check for connection

if (!$conn) {

die("Connection failed: "//display error message

}

// After connection logic...

echo "Connected to server";

// Close the connection when you are done

mysqli\_close($conn);

?>

**Crate a database:**

Let us create a database “college” using PHP.

<?php

// make a connection to the database server first then go ahead

// prepare query

$query = “CREATE DATABASE IF NOT EXISTS college”;

// run the query

mysqli\_query($conn,$query);

// close connection if you are done

mysqli\_close($conn);

?>

The above query creates a database named “college” if it doesn’t exist.

**Create Table in Selected Database:**

Let us create a table having schema: students(sid, name, gender,age,address,contact).

<?php

// Prepare connection parameters

$servername = "localhost";

$username = "root";//default is root

$password = "";// default is blank

// Establish Connection

$conn= mysqli\_connect($servername,$username,$password);

// Check if connection is successful or not

if ($conn->connect\_error) {

die("Connection failed: ");//display error message

}

// select a database in the server

mysqli\_select\_db($conn,"college");// replace db\_name with your database.

// prepare a query to execute

$query = "CREATE TABLE IF NOT EXISTS students(sid INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(30) NOT NULL, gender char, age INT NOT NULL, address VARCHAR(30) NOT NULL, contact VARCHAR(30))";

// run the query

$result=mysqli\_query($conn,$query);

if($result==true){

echo "Query executed successfully";

}

// close connection if you are done

mysqli\_close($conn);

?>

# **Create and Run SQL Statements (Select, Insert, Update, Delete)**

To create an SQL statement in PHP for MySQL, we can use the MySQLi functions. SQL statements are used for various database operations, such as selecting data, inserting new records, updating existing records, and deleting data. Here, we discuss about all these different types of database operations.

1. **Select Operation:**

SELECT operation is the SQL statement that is used to fetch/retrieve data from a database table. Performing a SELECT operation in PHP with MySQL involves fetching data from the database based on specific criteria.

The SELECT statement retrieves data from table row wise. Following code selects/fetches data from mysql table.

<?php

//setup parameters

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "college";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// prepare SQL Statement

$sql = "SELECT id, name FROM students";

// Get result in an array

$result = $conn->query($sql);

// fetch data and display as long as a row of data are fetched.

if ($result->num\_rows > 0) {

echo "<table border='1px'><tr><th>ID</th><th>Name</th></tr>";

// output data of each row

while($row = $result->fetch\_assoc()) {

echo "<tr><td>".$row['id']."</td><td>".$row['name']."</td></tr>";

}

echo "</table>";

} else {

echo "0 results";

}

$conn->close();

?>

1. **Insert Operation:**

The INSERT operation in SQL is used to add records in a table. The table schema/structure must be defined before executing the INSERT operation. We can use form to insert user record into the database using INSERT query. The insert operation can simply be executed by using mysqli\_query() function in PHP.

Following code uses two pages, form.html and insert.php. The form.html page is used to pass data from user browser to the server and insert.php contains the PHP script that actually receives the data from user and actually inserts it to the database server.

// insert.php

<?php

//setup parameters

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "college";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

// prepare SQL Statement

$sql = "insert into students values(5, 'Hari','Kathmandu','f',32)";

// Run the insert query and check if data inserted or not

if ($conn->query($sql) === TRUE) {

echo "New record inserted successfully";

} else {

echo "Error in inserting data";

}

$conn->close();

?>

1. **Update Operation: J**ust replace $sql=“Some update query”; in above code. For example “Update students set name= ‘Ram’ where id=5”;.
2. **Delete Operation:** Just replace $sql=“Some delete query”; in above code. For example “Delete from students where id=5”;.

# **Authentication**

Text

# **File Handling**

PHP provides various functions and techniques for handling files, which include creating, reading, writing, and manipulating files on the server or local file system. Here are some common file handling operations in PHP:

1. Opening and Closing File:
2. Reading Files:

We use fopen() and fclose() functions respectively to open and close a file.

Example: $filename = 'myfile.txt';

$handle = fopen($filename, 'r');

if ($handle) {

while (!feof($handle)) {

$line = fgets($handle);

echo $line; // Or process the $line as needed

}

fclose($handle);

} else {

echo "Unable to open the file.";

}

1. Writing to Files:
2. Appending to Files:
3. Checking File Existence:
4. Deleting Files:
5. Copying and Moving Files:
6. File related Functions: