UNIT - 4
PHP AND MYSQL

Prepared By: Prof. Disha H. Parekh

MYSQL CONNECTIVITY

- **`mysql_connect()`**
  - The **`mysql_connect()`** function opens a non-persistent MySQL connection.
  - This function returns the connection on success, or FALSE and an error on failure. You can hide the error output by adding an '@' in front of the function name.
- Syntax
  - **`mysql_connect(server,user,pwd,newlink,clientflag)`**

MYSQL CONNECTIVITY

Parameter	Description
server	Specifies the server to connect to
user	Specifies the username to log in with.
pwd	Specifies the password to log in with.
newlink	If a second call is made to mysql_connect() with the same arguments, no new connection will be established; instead, the identifier of the already opened connection will be returned
clientflag	•MYSQL_CLIENT_SSL - Use SSL encryption •MYSQL_CLIENT_COMPRESS - Use compression protocol •MYSQL_CLIENT_IGNORE_SPACE - Allow space after function names •MYSQL_CLIENT_INTERACTIVE - Allow interactive timeout seconds of inactivity before closing the connection

MYSQL CONNECTIVITY

```
<?php
  $con =
  mysql_connect("localhost","mysql_user","mysql_p
  wd");
  if (!$con){
    die('Could not connect: '.mysql_error());
  }
  echo 'Connected successfully';
  mysql_close($con);
?>
```

MYSQL CONNECTIVITY

- **`mysql_close()`**
  - The **`mysql_close()`** function closes a non-persistent MySQL connection.
  - This function returns TRUE on success, or FALSE on failure.
- Syntax:
  - **`mysql_close(connection)`**

Parameter	Description
connection	Specifies the MySQL connection to close. If not specified, the last connection opened by **`mysql_connect()`** is used.

MYSQL CONNECTIVITY

- **`mysqli_select_db()`**
  - The mysql_select_db() function sets the active MySQL database.
  - This function returns TRUE on success, or FALSE on failure.
- Syntax:
  - **`Mysqli_select_db(connection,database)`**

Parameter	Description
database	Required. Specifies the database to select.
connection	Optional. Specifies the MySQL connection. If not specified, the last connection opened by mysql_connect() or mysql_pconnect() is used.

MYSQL CONNECTIVITY

```
<?php
  $servername = "localhost";
  $username = "Disha"; // for lab uname = "root";
  $password = "xyz"; // for lab pwd = " ";
  $conn = new mysqli($servername, $username,
  $password);
   if($conn->connect_error) {
        die("Connection failed" .$conn->
  connect_error); }
$db_select = mysqli_select_db($conn,"dhp");
   If(!db_select) {
        die("Cannot connect to the database");
   }
Echo"Connection established";
Echo"Database connected";
?>
```

MYSQL CREATING A TABLE (PROCEDURAL)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "CREATE TABLE student (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP)";
if (mysqli_query($conn, $sql)) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . mysqli_error($conn);
}
mysqli_close($conn);
?>
```

MYSQL CREATING A TABLE (OBJECT ORIENTED)

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error); }
$sql = "CREATE TABLE Student (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP )";
if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";}

else {
        echo "Error creating table: " . $conn->error; }
$conn->close();
?>
```

REPORTING MYSQL ERRORS

- Reasons for not connecting to a database server include:
  - The database server is not running
  - Insufficient privileges to access the data source
  - Invalid username and/or password

REPORTING MYSQL ERRORS

- The mysqli_errno() function returns the error code from the last attempted MySQL function call or 0 if no error occurred

- The mysqli_error() — Returns the text of the error message from previous MySQL operation

- The mysqli_errno() and mysqli_error() functions return the results of the previous mysqli*() function

SUPPRESSING ERRORS WITH THE ERROR CONTROL OPERATOR

- By default, functions in the mysql package display errors and warnings as they occur
- Use the error control operator (@) to suppress error messages
- The error control operator can be prepended to any expression although it is commonly used with expressions

SELECT MULTIPLE RECORDS AS ASSOCIATIVE ARRAY

- *mysqli_fetch_assoc()* : The functino is used to fetch multiple records as an associative array.
- The returned array holds the strings fetched from database, where the column names will be the key used to access the internal data.
- Eg:
  - Z:\IMCA6\PHP\Unit-4\program to select data from table.docx

SELECT MULTIPLE RECORDS AS ARRAY

- fetch_array() : Function returns an array of both mysqli_fetch_row and mysqli_fetch assoc merged together, it is an extended version of the mysqli_fetch_row() function and both numeric and string can be used as keys to access the data.
- Eg:
  - Z:\IMCA6\PHP\Unit-4\program to select multiple records as array.docx
- *fetch_object()* : To fetch database result set as an objects, just use MySqli fetch_object(). The attributes of the object represent the names of the fields found within the result set.

SELECT COUNT TOTAL RECORDS OF A TABLE

- For pagination, counting total no. of records in a table becomes utmost necessary.
- To do that we write a select count query and put the result in a function namely, fetch_rows().
- Eg:
  - Z:\IMCA6\PHP\Unit-4\program to count total no. of records in a table.docx

MYSQL CRUD OPERATIONS

- Basically crud operations are create, read, update and delete on the same page, ie to create a table we use create query , as we enter the data in the table, that data will be displayed in tabular form on the same page showing the available options such as edit ,delete in front of them.
- If we click on the delete button, the entries can be deleted to change any field we click on the edit button

PREPARED STATEMENT

- Another important feature of MySqli is the Prepared Statements, it allows us to write query just once and then it can be executed repeatedly with different parameters.
- Prepared Statements significantly improves performance on larger table and more complex queries.
- The queries are parsed separately by the server, making it resilient to malicious code injection.
- The code below uses Prepared statement to fetch records from the database.
- ? placeholder in the SQL query acts like marker and will be replaced by a parameter, which could be string, integer, double or blob.
- Eg:
  - Z:\IMCA6\PHP\Unit-4\program that shows prepared statement.docx

MYSQLI – ORDER BY QUERY

- We have seen SQL **SELECT** command to fetch data from MySQLi table.
- When you select rows, the MySQLi server is free to return them in any order, unless you instruct it otherwise by saying how to sort the result.
- But you sort a result set by adding an ORDER BY clause that names the column or columns you want to sort by.
- Here is generic SQL syntax of SELECT command along with ORDER BY clause to sort data from MySQL table −
  - SELECT field1, field2,...fieldN table_name1, table_name2... ORDER BY field1, [field2...] [ASC [DESC]]
  - You can sort returned result on any field provided that filed is being listed out.

MYSQLI – ORDER BY QUERY

- You can sort result on more than one field.
- You can use keyword ASC or DESC to get result in ascending or descending order. By default, it's ascending order.
- You can use WHERE...LIKE clause in usual way to put condition.
- Eg:
  - Z:\IMCA6\PHP\Unit-4\Php script showing usage of ORDER BY Clause for mysqli db.docx