❖ phpMyAdmin

- phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web.
- phpMyAdmin supports a wide range of operations on MySQL and MariaDB.
- Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

❖ MySQL

- MySQL is one of the most popular relational database system being used on the Web today.
- It is freely available and easy to install, however if you have installed Wampserver it already there on your machine.
- MySQL database server offers several advantages:
    - MySQL is easy to use, yet extremely powerful, fast, secure, and scalable.
    - MySQL runs on a wide range of operating systems, including UNIX or Linux, Microsoft Windows, Apple Mac OS X, and others.
    - MySQL supports standard SQL (Structured Query Language).
    - MySQL is ideal database solution for both small and large applications.
    - MySQL is developed, and distributed by Oracle Corporation.
    - MySQL includes data security layers that protect sensitive data from intruders.
    - MySQL database stores data into tables like other relational database. A table is a collection of related data, and it is divided into rows and columns.

❖ PHP Connect to MySQL Server

- In PHP you can easily do this using the mysqli_connect() function.
- All communication between PHP and the MySQL database server takes place through this connection.
- Example :

```php
<?php

$mysql_host = 'localhost';
$mysql_user = 'root';
$mysql_password = '';

$mysql_database = 'training';

$mysql_err = 'sorry, could not connect.';
```

```php
// procedural style
$link = mysqli_connect($mysql_host, $mysql_user, $mysql_password, $m
ysql_database);

if($link === false){
    die("ERROR: Could not connect. " . mysqli_connect_error());
}

// Print host information
echo "Connect Successfully. Host info: " . mysqli_get_host_info($lin
k);

?>
```

❖ **Selecting Data From Database Tables**

- Let's make a SQL query using the SELECT statement, after that we will execute this SQL query through passing it to the PHP mysqli_query() function to retrieve the table data.
- Example :

```php
<?php

include '01_connecting_to_server_and_database.php';

$sql = "SELECT * FROM users";
if($result = mysqli_query($link, $sql)){
    if(mysqli_num_rows($result) > 0){
        echo "<table>";
            echo "<tr>";
                echo "<th>id</th>";
                echo "<th>Name</th>";
                echo "<th>Email</th>";
                echo "<th>Password</th>";
            echo "</tr>";
        while($row = mysqli_fetch_array($result)){
            echo "<tr>";
                echo "<td>" . $row['id'] . "</td>";
                echo "<td>" . $row['name'] . "</td>";
                echo "<td>" . $row['email'] . "</td>";
                echo "<td>" . $row['password'] . "</td>";
            echo "</tr>";
        }
        echo "</table>";
        // Free result set
```

```
        mysqli_free_result($result);
    } else{
        echo "No records matching your query were found.";
    }
} else{
    echo "ERROR: Could not able to execute $sql. " . mysqli_error($link
);
}

?>

<!DOCTYPE html>
<html lang="en">
<head>
    <link rel="stylesheet" href="style.css">
    <title>Document</title>
</head>
<body>

</body>
</html>
```