COS221
L17 - Web Programming
(Chapter 14 in Edition 6 and Chapter 11 in Edition 7)

Linda Marshall

31 March 2023

Web programming

- Many applications provide a web interface for accessing one or more database systems. Web interfaces are rendered using HTML on the client side and generated using a scripting language on the server side.
- As information is extracted from the database in real-time, the web pages displaying this information needs to be dynamic.
- There are many techniques to that can be used to program dynamic features into web pages. Examples of scripting languages used to create web pages include ASP, JSP, PHP, Ruby, Python...
- The focus of this lecture is on the server side scripting using PHP.

Web programming in PHP

- A PHP interpreter accepts as input a text file (program) which includes PHP commands.
- These commands are processed by a Hyper-text Preprocessor, which will execute the PHP commands and create the desired HTML file.
- Provision for database access is made using a library of PHP functions that needs to be included in the PHP interpreter.
- PHP programs are executed on the Web server computer.

Three-tier architecture

In a three-tier architecture:

- Web pages are served to the client as HTML
- The application layer (middle tier) is defined and managed using PHP
- The database forms the bottom tier of the architecture

A Simple PHP Program - Hello World

```
<?php
if ($_POST['user_name']) {
    // Printing a welcome message if the user submitted their name through the HTML form
    print("Welcome, ") ;
    print($_POST['user_name']);
}
else {
    // Printing the form to enter the user name since no name has been entered yet
    print <<<_HTML_
    <FORM method="post" action="$_SERVER['PHP_SELF']">
        Enter your name: <input type="text" name="user_name">
        <BR/>
        <INPUT type="submit" value="SUBMIT NAME">
    </FORM>
    _HTML_;
}
?>
```

API's for Database Connectivity in PHP

- Libraries exist to link to a database using PHP. These libraries provide similar functionality, but are vendor specific. Examples of libraries include:
  - PEAR (PHP Extension and Application Repository)
  - PDO (PHP Data Objects)
  - MySQLi (MySQL improved) extension
- The first two libraries work on various database systems while MySQLi works only on MySQL and a variant works on MariaDB.
- The examples in the textbook relate to PEAR.
- All three examples are written in an object-oriented style. MySQLi supports a procedural API.
- When using PEAR and PDO, the PHP code does not need to change when the underlying database changes. With MySQLi, due to close coupling, the PHP code will need to change.

Connecting to and Disconnecting from a Database - PEAR

Data source name (dsn) as a string

```
<?php
require_once 'DB.php';

$dsn = 'pgsql://someuser:apasswd@localhost/thedb';
$options = array(
    'debug'       => 2,
    'portability' => DB_PORTABILITY_ALL,
);

$db =& DB::connect($dsn, $options);
if (PEAR::isError($db)) {
    die($db->getMessage());
}

// ...

$db->disconnect();
?>
```

Connecting to and Disconnecting from a Database - PDO

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

try {
    $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
    // set the PDO error mode to exception
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Connected successfully";
    }
catch(PDOException $e)
    {
    echo "Connection failed: " . $e->getMessage();
    }
?>

// ....

$conn = null;
```

Connecting to and Disconnecting from a Database - MySQLi

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

// Create connection - OO style
$conn = new mysqli($servername, $username, $password);
// procedural style: $conn = mysqli_connect($servername, $username, $password);


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

// ...

$conn->close();
```

Create a table - MySQLi (OO style)

```
// sql to create table
$sql = "CREATE TABLE MyGuests (
id 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 MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}
```

Collecting Data from Forms and Populating the Database - MySQLi (OO style)

```
<html>
<head>
<title>Insert data in database using mysqli</title>
<link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>

<div id="main">
<h1>Insert data into database using mysqli</h1>
<div id="login">
<h2>Student's Form</h2>
<hr/>
<form action="" method="post">
<label>Student Name :</label>
<input type="text" name="stu_name" id="name" required="required" placeholder="Please Enter Name"/><br /><b
<label>Student Email :</label>
<input type="email" name="stu_email" id="email" required="required" placeholder="john123@gmail.com"/><br/>
<label>Student City :</label>
<input type="text" name="stu_city" id="city" required="required" placeholder="Please Enter Your City"/><br
<input type="submit" value=" Submit " name="submit"/><br />
</form>
</div>
<!-- Right side div -->
<div id="formget">
<a href=https://www.formget.com/app><img src="formget.jpg" alt="Online Form Builder"/></a>
</div>

</div>
```

Collecting Data from Forms and Populating the Database - MySQLi (OO style) - *continued*

```
<?php
if(isset($_POST["submit"])){
$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);
}

$sql = "INSERT INTO students (student_name, student_email, student_city)
VALUES ('".$_POST["stu_name"]."','".$_POST["stu_email"]."','".$_POST["stu_city"]."')";

if ($conn->query($sql) === TRUE) {
echo "<script type= 'text/javascript'>alert('New record created successfully');</script>";
} else {
echo "<script type= 'text/javascript'>alert('Error: " . $sql . "<br>" . $conn->error."');</script>";
}

$conn->close();
}
?>
</body>
</html>
```

Retrieval from Database Tables - MySQLi (OO style)

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
    border: 1px solid black;
}
</style>
</head>
<body>

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

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

Retrieval from Database Tables - MySQLi (OO style) - *continued*

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table><tr><th>ID</th><th>Name</th></tr>";
    // output data of each row
    while($row = $result->fetch_assoc()) {
     echo "<tr><td>" . $row["id"]. "</td><td>" . $row["firstname"]. " " . $row["lastname"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

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

Acknowledgements

Examples taken from:

- The textbook
- w3schools - `http://w3schools.com`
- PEAR manual - `https://pear.php.net/manual/en/`
- Tuts Make -`https://www.tutsmake.com`
- IT SourceCode - `https://itsourcecode.com/free-projects/php-project/populate-drop-down-list-from-database-using-php-mysqli`

A tutorial for MariaDB can be found on TutorialsPoint - `https://www.tutorialspoint.com/mariadb/index.htm`