INT322

Topics To Cover

1. Data Validation

For Data Validation is recommended to use $_POST and a variable that holds every validation from passing:

```
if($_POST){
            if($_POST['username'] == ""){
                $usernameError = '<- Enter a valid username';
                $isValidInput = false;
 }
//If Validation passes
if($_POST && $isValidInput){
             $username = $_POST['username'];
}
```

//On the HTML in order to show the error and re-populate all fields

```
<label for="username">Username:</label>
<input type="text" name="username"
       value="<?php if ($_POST) echo $_POST['username']; ?>">
<span><?php if(!($isValidInput)) echo $usernameError ?> </span>
```

//In order to repopulate only when there is an error

```
<input type="text" name="lastName"
       value="<?php if ($_POST & !($isValidInput)) echo $_POST['lastName']; ?>">
```

// Repopulate radio buttons

```
<input type="radio" name="sex" value="m"
<?php if ($_POST & !($isValidForm)){
          if (isset($_POST['sex'])){
                 if(($_POST['sex']) == "m") echo "checked";
                                                              }}?>>
```

// Repopulate selects option buttons

```
<option name="small" value="small"
       <?php if($_POST &!($isValidForm)){
         if(isset($_POST['t-shirt'])){
              if($_POST['t-shirt'] == 'small')
                     echo "selected";}} ?>>Small</option>
```

2. Validation with Regex

```
$courseRegex = '/^\s*([A-Z]{3}\d{3}[A-Z]{1,3})\s*$/';
$courseCode = $_POST['courseCode'];

// If you need to find more than one match you can use the variables $matches which
is an array that holds every match from the preg match function

if(preg_match($courseRegex,$courseCode,$matches)){
        $isValidCourse = true;
        $courseCodeError = "";
         echo = “Match found”
                       }
Else{
         echo = “Match not found”
        $isValidCourse = false;
        $courseCodeError = '<- Must be XXX999X or XXX999XX or XXX999XXX';
}
```

3. Connect to a Database

```
      $servername = 'localhost';
      $dbusername = 'root';
      $dbpassword = '';
      $dbname = 'lab4';


//Procedural Way (procedural sucks gorilla dong)
$conn = mysql_connect( $servername, $dbusername,$dbpassword)
mysql_select_db($dbname, $conn)

//OOP Way
$conn = new mysqli ($servername,$dbusername,$dbpassword,$dbname);

//To show the error just add die to any expression like this
$result = mysqli_query($conn, $sql_query) or die('Query failed: '.
mysql_error($conn));

//mysql_errno gives the number of the error too

**To close the database**

mysql_close($conn);
Or OOP way
$conn->close();
```

4. Create Database

```
$servername = "localhost";
$username = "username";
$password = "password";
$conn = new mysqli($servername, $username, $password);
$sql = "CREATE DATABASE newDatabase";
$query = mysqli_query($conn, $sql);
```

5. Create Table

```
$sql_query="CREATE TABLE product(id int zerofill not null AUTO_INCREMENT,itemName
VARCHAR(40) not null, model VARCHAR(30) not null, os VARCHAR(12) not null, price
decimal(10,2) not null,  PRIMARY KEY(id))";

$query = mysqli_query($conn, $sql_query);
```

6. Remove Table

```
$sql_query="DROP TABLE product";
$query = mysqli_query($conn, $sql_query);
```

7. Insert Data on a Database (Register Function)

```
$sql = "SELECT * FROM users";
if($query){
   if(mysqli_num_rows($query)>0){
       while($row == mysqli_fetch_assoc($query)){
              if($row[‘First_Name’] == $firstName &&
                 $row[‘Last_Name’] == $lastName){
                     Echo “The user already exists”
              Else{
                     $sql = "INSERT INTO users(First_Name, Last_Name, Sex) VALUES
                     ('$firstName','$lastName','$sex')";
                     $query = mysqli_query($conn, $sql);
              }
```

8. Check Data on a Database (Sign In Function)

```
$email = $_POST['email'];
$sql = "SELECT * FROM users";
$query = mysqli_query($conn, $sql) or die(mysqli_error($mysqli));

if($query){
       if(mysqli_num_rows($query)>0){
              while($row == mysqli_fetch_assoc($query)){
                     if($row['username'] == $email){
                            //Relocate to another page
                            header('Location: login.php');
```

9. Select Data on a Database and displaying it (tabular)

```
$result = mysqli_query($conn, $sql_query)
//Check if successful
if($result){
	//Check that the return is bigger than 0
	if (mysqli_num_rows($result) > 0) {
	//diplay the table header
	while($row = mysqli_fetch_assoc($result)){
		"<td>" . $row["ID"] . "</td> .
		"<td>" . $row["First_Name"] . "</td>" .
		"<td>" . $row["Last_Name"] . "</td>" .
		//Using Ternary for booleans fields
		"<td>" . ($row["Multiple_Yes"] == true ? "yes" : "no") . "</td>";
	}
```

10. Remove Data from a Database

```
//This will remove the whole table
$query = 'DELETE from product';
mysqli_query($conn, $query);
```

11. Send Email

```
$message = "Username: " . $row['username'] . "\r\n" . "Passowrd hint: " .
$row['passwordHint'];
$email = “admin@mail.com”;

mail($email, 'Your password recovery', $message);
```

-------------------- Another Example ----------------------

```
$to = "somebody@example.com";
$subject = "My subject";
$txt = "Hello world!";
$headers = "From: webmaster@example.com" . "\r\n" .
"CC: somebodyelse@example.com";

mail($to,$subject,$txt,$headers);
```

12. Create and Remove Directories

```
if (!is_dir('examples')) {
    mkdir('examples');
}

rmdir('examples');
```

13. Create and Remove Files

```
//Create a file
// a is for append, w is for write
$newfile = fopen('test.html', 'a');
fwrite($newfile, 'Hello world!\r\nHello wolrd on another line');
fclose($newfile);

unlink('test.html');
```

14. Write to a file

```
$myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
$txt = "John Doe\n";
fwrite($myfile, $txt);
$txt = "Jane Doe\n";
fwrite($myfile, $txt);
fclose($myfile);
//if using w and you reopen the file you will overwrite whatever is there
```

15. Read from a file

```
// Read from a file and start connection
// File secret.txt
rocalhost
root

assign1

---------------------

$data = file('secret.txt');
$servername = trim($data[0]);
$dbusername = trim($data[1]);
$dbpassword = trim($data[2]);
$dbname     = trim($data[3]);
$conn       = new mysqli ($servername,$dbusername,$dbpassword,$dbname);

//Read from a file and loop through each section and add to a database

$phone = file('phone.txt');
$os    = file('os.txt');
$price = file ('price.txt');
$model = file ('model.txt');

//Inserting data
for($i = 0; $i < sizeof($phone); $i++){
       $query = 'INSERT INTO product set itemName = "'
              .trim($phone[$i]).'", model = "'
              .trim($model[$i]).'", os = "'
              .trim($os[$i]).'", price = "'
              .trim($price[$i]).'"'.';';
//Running query and printing error in case
      mysqli_query($conn,$query) or die ('Error -> ' . mysqli_error($conn));
}
```

16. Hashed passwords

```
$password = 'mypassword';
$hash = crypt($password);

//Pass the text and the salt
$hashConversion = crypt($_POST['pass'], $_POST['salt']);

//In order to check from the database using the salt
if($row['password'] == crypt($password,$salt)){}

// Trick to use unique salts , using each passwords
$salt = substr($user_password, -4);

//Nesting crypts is fine
$hashed_password1 = crypt( crypt( $password ) );

//There are two other types of encryption md5 and sha1
$hash = sha1(md5(sha1(md5(sha1(md5(sha1($value)))))));
```

17. Include and Require

```
include 'filename';
require 'filename';

----------------------------

Example:
<?php include ‘header.php’;?>
<?php
//Actual body php code
?>
<?php include footer.php’;?>

** The difference is that
** Include will keep running the code regardless of errors
** Require will stop the code from running (required by the application)

----------------------------

include_once("myClasses.php");
//Include the OOP Classes
```

18. Redirect

```
header('Location: login.php');
```

19. Cookies

```
//Setting Cookie
setcookie("name","value",time() + (86400 * 30));
time() + (86400 * 30) = day

//Getting Cookie
echo $_COOKIE["name"];

//Updating Cookie (redeclare the cookie)
setcookie("color","red");
echo $_COOKIE["color"];
setcookie("color","blue");
echo $_COOKIE["color"];

//Deleting Cookie
unset($_COOKIE["yourcookie"]);
Or
setcookie("yourcookie","yourvalue",time()-1);
```

20. Sessions

```
//Start a session after logging
//Let's say that the user and password we're fine

if($row['username'] == $email && $row['password'] == $password){

       session_start();
       $_SESSION['isLogged'] = true;
       $_SESSION['username'] = $user;
       header('Location: protectedstuff.php?user=' . $user);
       Break;

// On the PROTECTEDSTUFF.PHP
session_start();
// check if user is logged in, if not redirect to login page
if(isset($_SESSION['isLogged']) && $_SESSION['isLogged']){
       echo "<h5>You are logged in!</h5>";
}else{
       header('Location: login.php');
}
```

```
//In order to remove a session
session_start();

// remove all session variables
session_unset();

// destroy the session
session_destroy();
```

21. Uploading Files

```
$_FILES contains 5 pieces of information
1.$_FILES['FileName']['name'] => Contains the name of the file
2.$_FILES['FileName']['size'] => Contains the size of the file
3.$_FILES['FileName']['type'] => Contains the type of the file
4.$_FILES['FileName']['error'] => Contains the error code of the file
5.$_FILES['FileName']['tmp_Name'] => Contains temporary location of the file

<input type="file" name="userFile" />

if (move_uploaded_file($_FILES['FileName']['tmp_name'], "uploads/" .
$_FILES['FileName']['name']) === FALSE){
       echo "Could not move uploaded file to \"uploads/" .
htmlentities($_FILES['FileName']['name']) . "\"<br />\n";
}
else{
       echo "Successfully uploaded \"uploads/" .
htmlentities($_FILES['FileName']['name']) . "\"<br />\n";
}
```

22. File Permissions

```
//Change file permisions
chmod("uploads/" . $_FILES['FileName']['name'], 0644);

//Check file permissions
fileperms($fileName)

//Lists files on a directory
$Dir = "/var/html/uploads";
       $DirOpen = opendir($Dir);
while ($CurFile = readdir($DirOpen)) {
       echo $CurFile . "<br />\n";
              }
closedir($DirOpen);

//Get current directory
getcwd();
```

```
//Display files of current directory
$path = getcwd();
$items = scandir($path);
echo "<p>Contents of $path</p>";
echo '<ul>';
foreach ($items as $item) {
    echo '<li>' . $item . '</li>';
}
echo '</ul>';

//Information about the directory
filesize($EntryFullName)
filetype($EntryFullName)
```

23. Passing variables through GET

```
//Passing through header
header('Location: protectedstuff.php?user=' . $user);

//Passing on the link tag
<a href=”<?php echo ‘protectedstuff.php?user=' . $user ?>”>Go to main.html</a>


//On the protected page in order to catch the variable
//Take the stored variable
$user = $_GET[‘user’];
Echo “You’re logged in as $user”
```

24. Display Time

```
$timeDatabase = "SELECT CURDATE()";
$date = mysqli_query($conn, $timeDatabase);
      $printDate = mysqli_fetch_assoc($date);
      echo "Query Date: " . $printDate['CURDATE()'];
```

25. OOP Structure

```
class DBLink{
    // Variables
    private $conn;
    private $result = "";

    // Functions
    //Connecting to the database
    public function __construct(){
        $this->conn = mysqli_connect('localhost','root','','lab6');
    }
    //Starting the query
    public function query($sql_query){
        $this->result = mysqli_query($this->conn,$sql_query); //or die ("Query
failed =>" . $this->conn->connect_error);
        return $this->result;
    }




    //Custom function
    //Checking if the result is empty
    public function isEmptyResult(){
        $empty = mysqli_num_rows($this->result);
        return $empty;
    }
    //Closing Connection
    public function __destruct(){
        mysqli_close($this->conn);
    }

} // End of DBLink


//On the main.php

// Start the classes
$mysqli = new DBLink();
// This will trigger the __construct function and start the connection

// Is possible to initilialize any function from the class
$show   = "SELECT itemName, price FROM product;";
$result = $mysqli->query($show);

//Initializing the custom class
$remainingRows = $mysqli->isEmptyResult($result = $mysqli->query($show))

//The class will automatically call the __destruct() function
```

--------------------**Another Example**--------------------

```
class BankAccount {

             public $Balance = 958.20;
             public function withdrawal($Amount) {
                    $this->Balance -= $Amount;
             }
}


if (class_exists("BankAccount"))
             $Checking = new BankAccount();
else
```

```
              exit("<p>The BankAccount class is not available!</p>");

printf("<p>Your checking account balance is $%.2f.</p>",
              $Checking->Balance);
              $Cash = 200;
              $Checking->withdrawal(200);
              printf("<p>After withdrawing $%.2f, your checking account balance
                     is $%.2f.</p>", $Cash, $Checking->Balance);

// Using GETTER & SETTER

class BankAccount {
              private $Balance = 0;
       //Setter
              public function setBalance($NewValue) {
                     $this->Balance = $NewValue;
              }
       //Getter
              public function getBalance() {
                     return $this->Balance;
              }
}
if (class_exists("BankAccount"))
              $Checking = new BankAccount();
else
              exit("<p>The BankAccount class is not
                            available!</p>");
$Checking->setBalance(100);
echo "<p>Your checking account balance is "
              . $Checking->getBalance() . "</p>\n";



//Using serialize , have no idea what I need to use this garbage
//JS is so much better, INT322 = the most useless crap in the universe

$serialized_array=serialize($multidimentional_array);
$original_array=unserialize($serialized_array);
```