HTML File

<html>

<head>

<title>PHP User Registration Form</title>

<link href="./css/style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<form name="frmRegistration" method="post" action="">

<div class="demo-table">

<div class="form-head">Sign Up</div>

<?php

if (! empty($errorMessage) && is\_array($errorMessage)) {

?>

<div class="error-message">

<?php

foreach($errorMessage as $message) {

echo $message . "<br/>";

}

?>

</div>

<?php

}

?>

<div class="field-column">

<label>Username</label>

<div>

<input type="text" class="demo-input-box"

name="userName"

value="<?php if(isset($\_POST['userName'])) echo $\_POST['userName']; ?>">

</div>

</div>

<div class="field-column">

<label>Password</label>

<div><input type="password" class="demo-input-box"

name="password" value=""></div>

</div>

<div class="field-column">

<label>Confirm Password</label>

<div>

<input type="password" class="demo-input-box"

name="confirm\_password" value="">

</div>

</div>

<div class="field-column">

<label>Display Name</label>

<div>

<input type="text" class="demo-input-box"

name="firstName"

value="<?php if(isset($\_POST['firstName'])) echo $\_POST['firstName']; ?>">

</div>

</div>

<div class="field-column">

<label>Email</label>

<div>

<input type="text" class="demo-input-box"

name="userEmail"

value="<?php if(isset($\_POST['userEmail'])) echo $\_POST['userEmail']; ?>">

</div>

</div>

<div class="field-column">

<div class="terms">

<input type="checkbox" name="terms"> I accept terms

and conditions

</div>

<div>

<input type="submit"

name="register-user" value="Register"

class="btnRegister">

</div>

</div>

</div>

</form>

</body>

</html>

Style Sheet

body {

font-family: Arial;

color: #333;

font-size: 0.95em;

}

.form-head {

color: #191919;

font-weight: normal;

font-weight: 400;

margin: 0;

text-align: center;

font-size: 1.8em;

}

.error-message {

padding: 7px 10px;

background: #fff1f2;

border: #ffd5da 1px solid;

color: #d6001c;

border-radius: 4px;

margin: 30px 0px 10px 0px;

}

.success-message {

padding: 7px 10px;

background: #cae0c4;

border: #c3d0b5 1px solid;

color: #027506;

border-radius: 4px;

margin: 30px 0px 10px 0px;

}

.demo-table {

background: #ffffff;

border-spacing: initial;

margin: 15px auto;

word-break: break-word;

table-layout: auto;

line-height: 1.8em;

color: #333;

border-radius: 4px;

padding: 20px 40px;

width: 380px;

border: 1px solid;

border-color: #e5e6e9 #dfe0e4 #d0d1d5;

}

.demo-table .label {

color: #888888;

}

.demo-table .field-column {

padding: 15px 0px;

}

.demo-input-box {

padding: 13px;

border: #CCC 1px solid;

border-radius: 4px;

width: 100%;

}

.btnRegister {

padding: 13px;

background-color: #5d9cec;

color: #f5f7fa;

cursor: pointer;

border-radius: 4px;

width: 100%;

border: #5791da 1px solid;

font-size: 1.1em;

}

.response-text {

max-width: 380px;

font-size: 1.5em;

text-align: center;

background: #fff3de;

padding: 42px;

border-radius: 3px;

border: #f5e9d4 1px solid;

font-family: arial;

line-height: 34px;

margin: 15px auto;

}

.terms {

margin-bottom: 5px;

}

Validate User Information

function validateMember()

{

$valid = true;

$errorMessage = array();

foreach ($\_POST as $key => $value) {

if (empty($\_POST[$key])) {

$valid = false;

}

}

if($valid == true) {

if ($\_POST['password'] != $\_POST['confirm\_password']) {

$errorMessage[] = 'Passwords should be same.';

$valid = false;

}

if (! isset($error\_message)) {

if (! filter\_var($\_POST["userEmail"], FILTER\_VALIDATE\_EMAIL)) {

$errorMessage[] = "Invalid email address.";

$valid = false;

}

}

if (! isset($error\_message)) {

if (! isset($\_POST["terms"])) {

$errorMessage[] = "Accept terms and conditions.";

$valid = false;

}

}

}

else {

$errorMessage[] = "All fields are required.";

}

if ($valid == false) {

return $errorMessage;

}

return;

}

PHP MYSQL Code

<?php

namespace Phppot;

use \Phppot\Member;

if (! empty($\_POST["register-user"])) {

$username = filter\_var($\_POST["userName"], FILTER\_SANITIZE\_STRING);

$displayName = filter\_var($\_POST["firstName"], FILTER\_SANITIZE\_STRING);

$password = filter\_var($\_POST["password"], FILTER\_SANITIZE\_STRING);

$email = filter\_var($\_POST["userEmail"], FILTER\_SANITIZE\_STRING);

require\_once ("Member.php");

/\* Form Required Field Validation \*/

$member = new Member();

$errorMessage = $member->validateMember($username, $displayName, $password, $email);

if (empty($errorMessage)) {

$memberCount = $member->isMemberExists($username, $email);

if ($memberCount == 0) {

$insertId = $member->insertMemberRecord($username, $displayName, $password, $email);

if (! empty($insertId)) {

header("Location: thankyou.php");

}

} else {

$errorMessage[] = "User already exists.";

}

}

}

?>

Check if user already exists

function isMemberExists($username, $email)

{

$query = "select \* FROM registered\_users WHERE user\_name = ? OR email = ?";

$paramType = "ss";

$paramArray = array($username, $email);

$memberCount = $this->ds->numRows($query, $paramType, $paramArray);

return $memberCount;

}

Insert member data to the database

function insertMemberRecord($username, $displayName, $password, $email)

{

$passwordHash = md5($password);

$query = "INSERT INTO registered\_users (user\_name, display\_name, password, email) VALUES (?, ?, ?, ?)";

$paramType = "ssss";

$paramArray = array(

$username,

$displayName,

$passwordHash,

$email

);

$insertId = $this->ds->insert($query, $paramType, $paramArray);

return $insertId;

}

### DataSource.php

$stmt->execute();

$result = $stmt->get\_result();

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

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

$resultset[] = $row;

}

}

if (! empty($resultset)) {

return $resultset;

}

}

/\*\*

\* To insert

\* @param string $query

\* @param string $paramType

\* @param array $paramArray

\* @return int

\*/

public function insert($query, $paramType, $paramArray)

{

print $query;

$stmt = $this->conn->prepare($query);

$this->bindQueryParams($stmt, $paramType, $paramArray);

$stmt->execute();

$insertId = $stmt->insert\_id;

return $insertId;

}

/\*\*

\* To execute query

\* @param string $query

\* @param string $paramType

\* @param array $paramArray

\*/

public function execute($query, $paramType="", $paramArray=array())

{

$stmt = $this->conn->prepare($query);

if(!empty($paramType) && !empty($paramArray)) {

$this->bindQueryParams($stmt, $paramType="", $paramArray=array());

}

$stmt->execute();

}

/\*\*

\* 1. Prepares parameter binding

\* 2. Bind prameters to the sql statement

\* @param string $stmt

\* @param string $paramType

\* @param array $paramArray

\*/

public function bindQueryParams($stmt, $paramType, $paramArray=array())

{

$paramValueReference[] = & $paramType;

for ($i = 0; $i < count($paramArray); $i ++) {

$paramValueReference[] = & $paramArray[$i];

}

call\_user\_func\_array(array(

$stmt,

'bind\_param'

), $paramValueReference);

}

/\*\*

\* To get database results

\* @param string $query

\* @param string $paramType

\* @param array $paramArray

\* @return array

\*/

public function numRows($query, $paramType="", $paramArray=array())

{

$stmt = $this->conn->prepare($query);

if(!empty($paramType) && !empty($paramArray)) {

$this->bindQueryParams($stmt, $paramType, $paramArray);

}

$stmt->execute();

$stmt->store\_result();

$recordCount = $stmt->num\_rows;

return $recordCount;

}

}

## Database script

--

-- Table structure for table `registered\_users`

--

CREATE TABLE IF NOT EXISTS `registered\_users` (

`id` int(8) NOT NULL AUTO\_INCREMENT,

`user\_name` varchar(255) NOT NULL,

`first\_name` varchar(255) NOT NULL,

`last\_name` varchar(255) NOT NULL,

`password` varchar(25) NOT NULL,

`email` varchar(55) NOT NULL,

`gender` varchar(20) NOT NULL,

PRIMARY KEY (`id`)

);