SVKM’s NMIMS

MPSTME

Computer Engineering Department (MCA SEM I)

Web Programming

Lab Manual

**PART A**

**Experiment No. 09**

**PART-A**

**A.1 Aim:** Form processing in PHP

**A.2 Prerequisite:**

HTML, CSS, basic programming

**A.3 Outcome:**

After successful completion of this experiment students will be able to

1. Understand server side working of application
2. Install XAMPP
3. Implements form applications in PHP.

**A.4 Theory:**

* PHP scripts are executed on the server, and the result is returned as plain HTML to the browser.
* A PHP script starts with <?php and ends with ?>.
* PHP statements end with a semicolon ;

<?php

echo "Hello, World!";

?>

* Variables in PHP are declared with the **$** symbol.
* PHP automatically converts the variable to the correct data type based on its value (dynamic typing).
* Conditions in PHP:

Example:

Code for form:

<?php

if(isset($\_POST['submit']))

{

//$\_POST is used to retrieve html form values

echo "hii ". trim($\_POST['name']);

//trim is used to remove extra spaces

echo "<br>";

echo "You are ". (int)$\_POST['age']." years old";

$p=$\_POST['pass'];

//preg\_match - function in php to match regular expression

// The following regex is same as we have written in JS. atleast 1 capital, 1 small, 1 secial, 1 digit.

$valid = preg\_match('/^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*[0-9])(?=.\*[$%^&]).{8,12}$/', $p);

if(!$valid)

echo "<br>"."Incorrect password";

}

?>

A.5 **Tasks to be completed:**

1. Write html form to store student data with fields: Student ID, name, password, confirmpassword, email, checkbox (With coursename as options), radiobutton (gender), Date of birth, submit button.
2. Write php script to retrieve the form data, do following validations and then display all the form data
3. All fields are mandatory
4. Username can have only letters and length must be only 6
5. Password must have atleast 1 capital, 1 small, 1 special symbol with length between 6 and 12.
6. Password and confirm password fields must match.
7. **Date of birth year should not be less than 1980 and not more than 2010.**

**B.1 Software Code written by student:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Student Registration Form</title>

</head>

<body>

    <?php

    $errors = [];

    if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

        $studentID = trim($\_POST['studentID']);

        $name = trim($\_POST['name']);

        $password = $\_POST['password'];

        $confirmPassword = $\_POST['confirmPassword'];

        $email = filter\_var($\_POST['email'], FILTER\_SANITIZE\_EMAIL);

        $course = isset($\_POST['course']) ? $\_POST['course'] : [];

        $gender = isset($\_POST['gender']) ? $\_POST['gender'] : '';

        $dob = $\_POST['dob'];

        if (empty($studentID) || empty($name) || empty($password) || empty($confirmPassword) || empty($email) || empty($gender) || empty($dob)) {

            $errors[] = "All fields are mandatory.";

        }

        if (!preg\_match("/^[a-zA-Z]{6}$/", $name)) {

            $errors[] = "Name must contain only letters and be exactly 6 characters long.";

        }

        if ($password !== $confirmPassword) {

            $errors[] = "Passwords do not match.";

        }

        if (!preg\_match('/^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*[0-9])(?=.\*[@$%^&]).{6,12}$/', $password)) {

            $errors[] = "Password must contain at least one uppercase letter, one lowercase letter, one digit, and one special character, with a length between 6 and 12 characters.";

        }

        $dobYear = (int)date("Y", strtotime($dob));

        if ($dobYear < 1980 || $dobYear > 2010) {

            $errors[] = "Year of birth should be between 1980 and 2010.";

        }

        if (empty($errors)) {

            echo "<h3>Submitted Data:</h3>";

            echo "Student ID: " . htmlspecialchars($studentID) . "<br>";

            echo "Name: " . htmlspecialchars($name) . "<br>";

            echo "Email: " . htmlspecialchars($email) . "<br>";

            echo "Courses: " . implode(", ", $course) . "<br>";

            echo "Gender: " . htmlspecialchars($gender) . "<br>";

            echo "Date of Birth: " . htmlspecialchars($dob) . "<br>";

        }

    }

    ?>

    <h2>Student Registration Form</h2>

    <?php

    if (!empty($errors)) {

        echo "<h3>Errors:</h3><ul>";

        foreach ($errors as $error) {

            echo "<li>$error</li>";

        }

        echo "</ul>";

    }

    ?>

    <form action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>" method="POST">

        <label for="studentID">Student ID:</label>

        <input type="text" id="studentID" name="studentID" ><br><br>

        <label for="name">Name:</label>

        <input type="text" id="name" name="name" ><br><br>

        <label for="password">Password:</label>

        <input type="password" id="password" name="password" ><br><br>

        <label for="confirmPassword">Confirm Password:</label>

        <input type="password" id="confirmPassword" name="confirmPassword" ><br><br>

        <label for="email">Email:</label>

        <input type="email" id="email" name="email" ><br><br>

        <label>Course:</label><br>

        <input type="checkbox" name="course[]" value="PHP">PHP<br>

        <input type="checkbox" name="course[]" value="JavaScript">JavaScript<br>

        <input type="checkbox" name="course[]" value="Python">Python<br><br>

        <label>Gender:</label><br>

        <input type="radio" name="gender" value="Male" >Male<br>

        <input type="radio" name="gender" value="Female" >Female<br><br>

        <label for="dob">Date of Birth:</label>

        <input type="date" id="dob" name="dob" ><br><br>

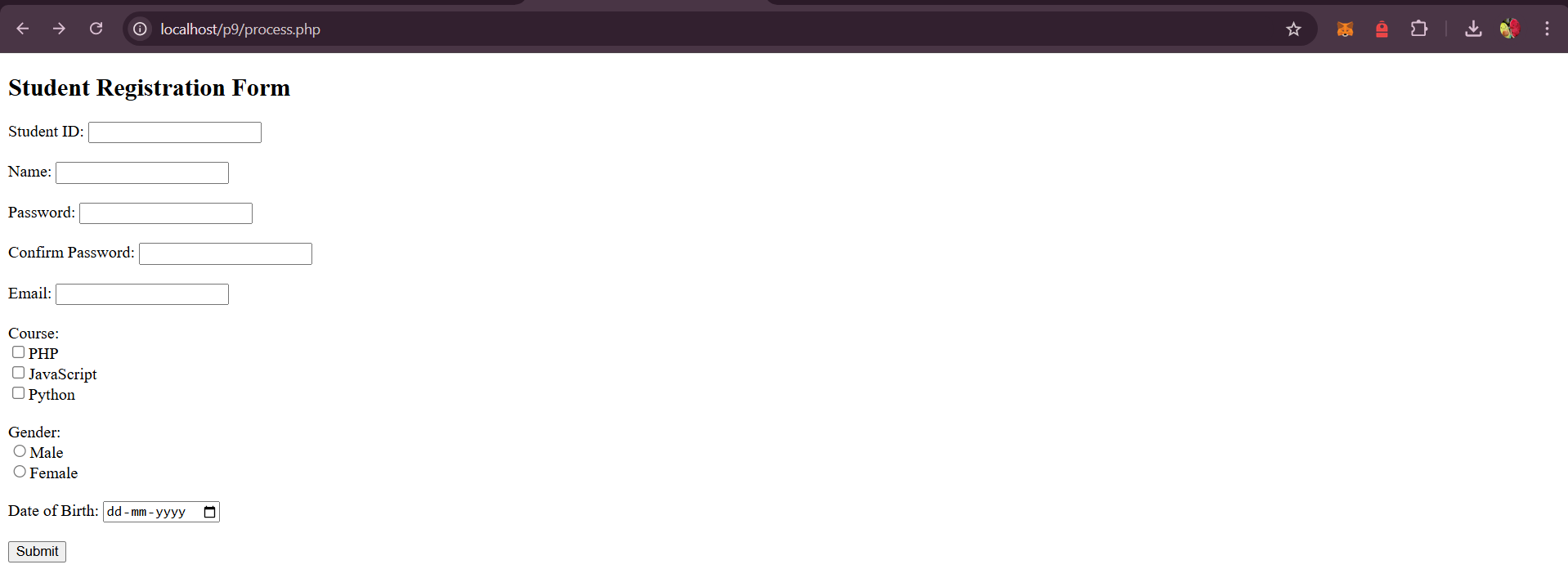
        <input type="submit" name="submit" value="Submit">

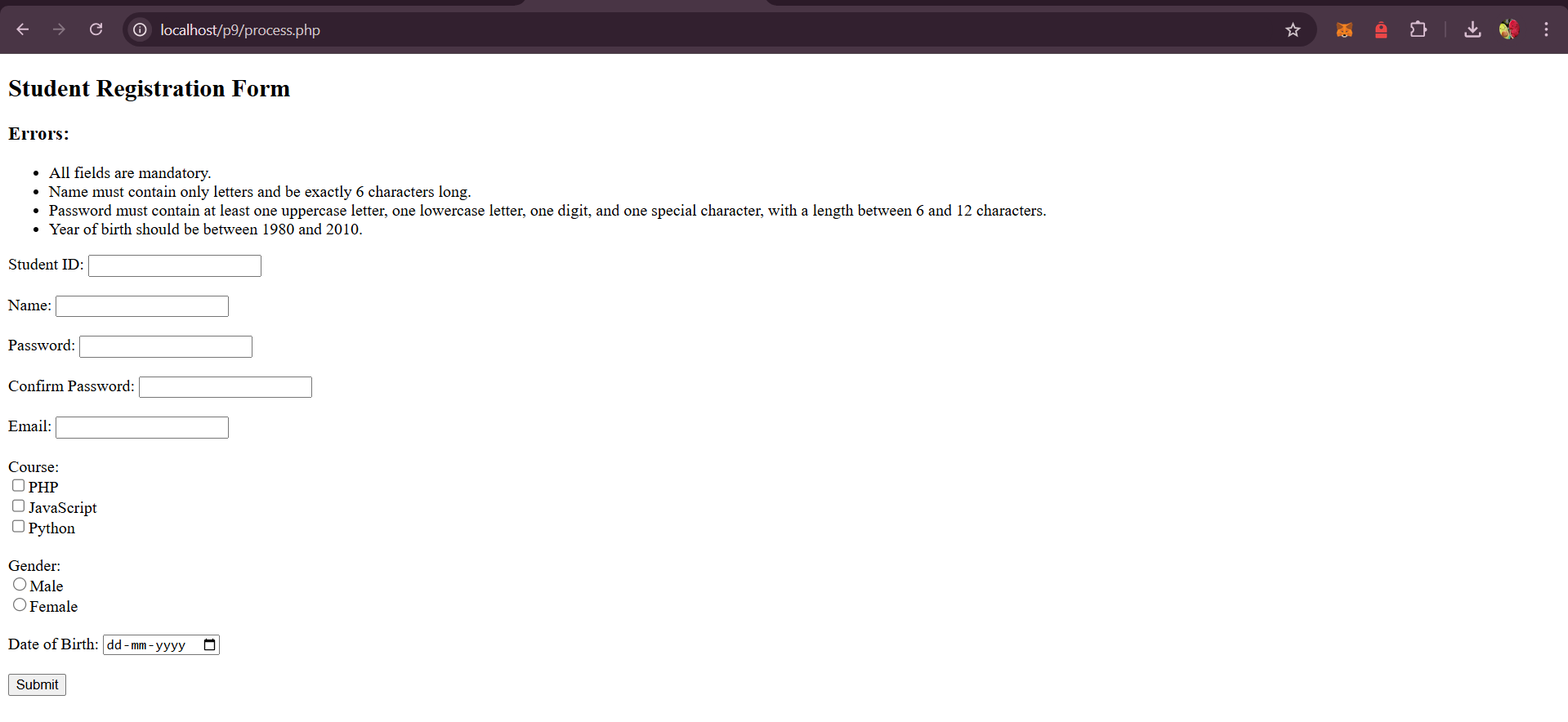
    </form>

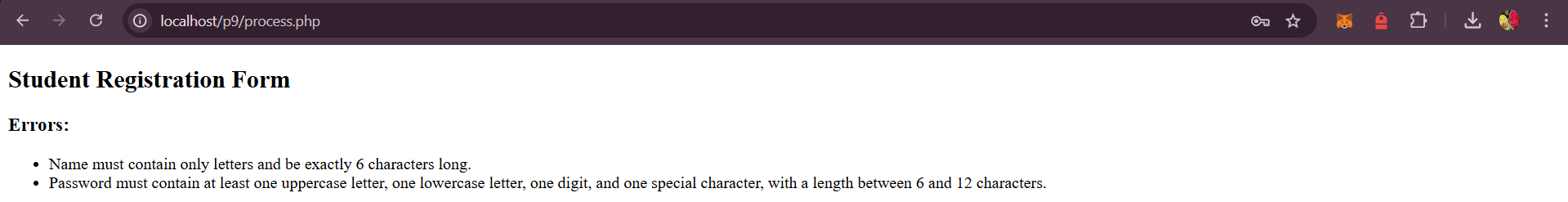
</body>

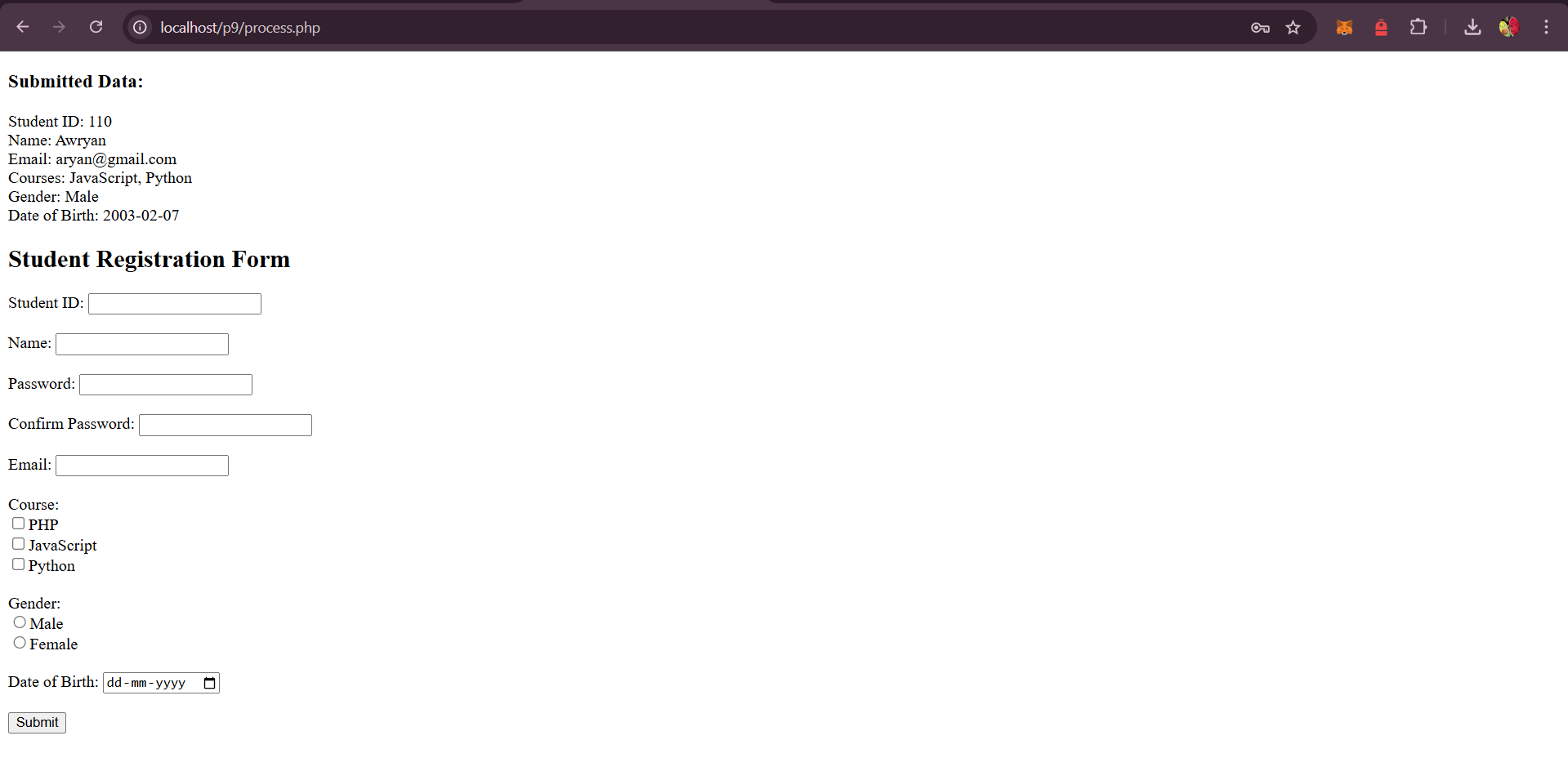
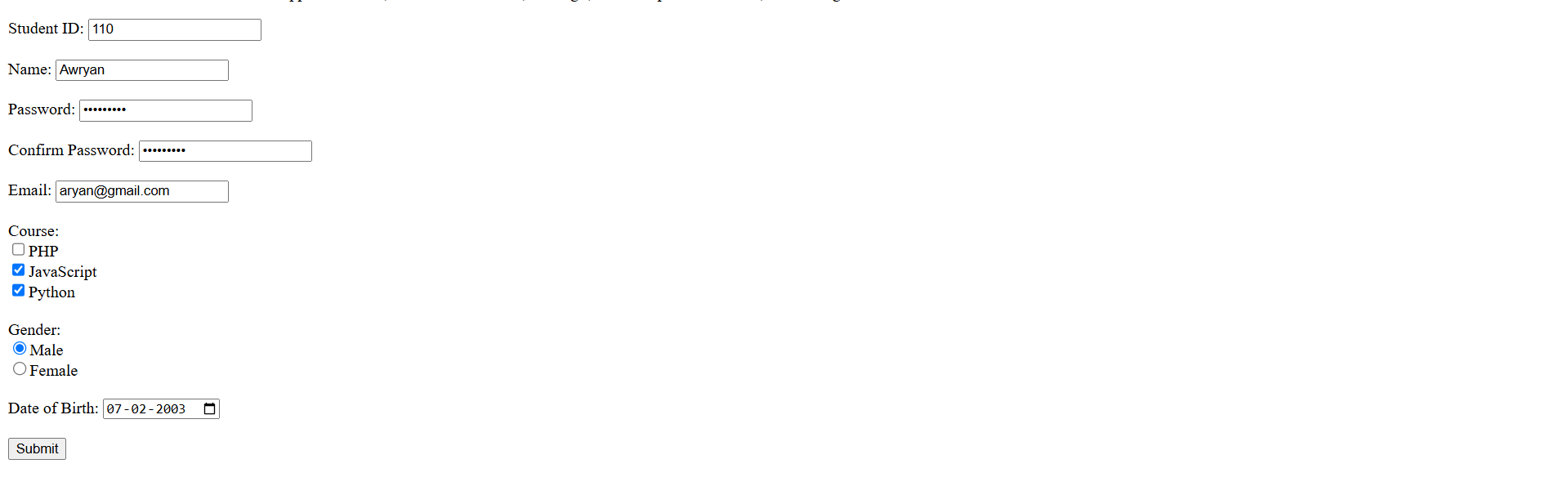
</html>

**B.2 Output**

****

****

****

****

**B.3 Conclusion:**

**Successfully implemented and completed Form processing in PHP**

**B.4 Question of Curiosity**

**1. What is $\_POST and $\_GET in PHP?**

* **$\_POST** is used to retrieve data sent via HTTP POST (e.g., form data not visible in the URL).
* **$\_GET** is used for data sent via HTTP GET, visible in the URL (e.g., search queries).

**2. What are some security concerns when processing forms in PHP?**

* **SQL Injection**: Prevent with prepared statements.
* **XSS**: Escape output with htmlspecialchars().
* **CSRF**: Use CSRF tokens to protect forms.
* **File Uploads**: Validate file types and sanitize file names.

**3. How do you handle file uploads through a form in PHP?**

* <form enctype="multipart/form-data">