1. Create an HTML form with fields for user's name, email, and message. Write a PHP script to handle form data submission and display the submitted information.

**PA1.html**

<!DOCTYPE html>

<html lang="en">

<body>

    <h2>Contact Us</h2>

    <form action="submit.php" method="post">

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

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

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

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

        <label for="message">Message:</label><br>

        <textarea id="message" name="message" rows="4" required></textarea><br>

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

    </form>

</body>

</html>

Submit.html

<!DOCTYPE html>

<html lang="en">

<body>

    <h2>Form Submission Result</h2>

    <?php

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

        $name = $\_POST["name"];

        $email = $\_POST["email"];

        $message = $\_POST["message"];

        echo "<p><strong>Name:</strong> $name</p>";

        echo "<p><strong>Email:</strong> $email</p>";

        echo "<p><strong>Message:</strong> $message</p>";

    } else {

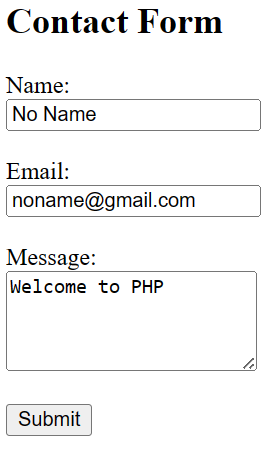
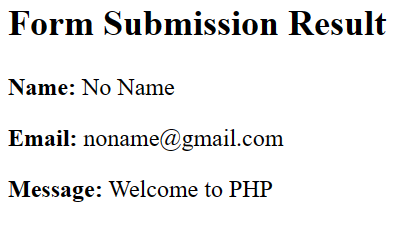
        echo "<p>No form data submitted</p>";

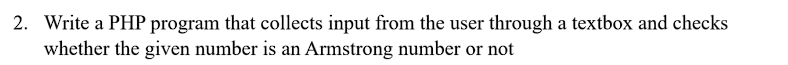
    }

    ?>

</body></html>

**Output:**





<!DOCTYPE html>

<html>

<head>

    <title>Armstrong Number Checker</title>

</head>

<body>

    <h2>Armstrong Number Checker</h2>

    <form method="post">

        Enter a number: <input type="text" name="number">

        <input type="submit" value="Check">

    </form>

    <?php

    // Function to check if a number is an Armstrong number

    function isArmstrong($number) {

        $sum = 0;

        $temp = $number;

        $digits = strlen($number);

        while ($temp != 0) {

            $remainder = $temp % 10;

            $sum += $remainder \*\* $digits;

            $temp = (int)($temp / 10);

        }

        if ($sum == $number) {

            return true;

        } else {

            return false;

        }

    }

    // Check if form is submitted

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

        $inputNumber = $\_POST["number"];

          if (is\_numeric($inputNumber)) {

            if (isArmstrong($inputNumber)) {

                echo "<p>$inputNumber is an Armstrong number.</p>";

            } else {

                echo "<p>$inputNumber is not an Armstrong number.</p>";

            }

        } else {

            echo "<p>Please enter a valid number.</p>";

        }

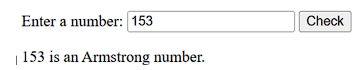
    }

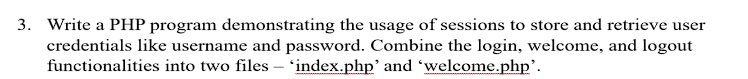
    ?>

</body>

</html>

**Output:**

****



**Index.php**

<?php

session\_start();

// Check if the user is already logged in, if yes, redirect to welcome.php

if(isset($\_SESSION['username'])) {

    header("Location: welcome.php");

    exit;

}

// Check if the form is submitted

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

    // Dummy username and password for demonstration purposes

    $uname = "prakash";

    $pwd = "password";

    $username = $\_POST["username"];

    $password = $\_POST["password"];

    // Check if username and password are correct

    if ($username === $uname && $password === $pwd) {

        // Set session variables

        $\_SESSION["username"] = $uname;

        $\_SESSION["password"] = $pwd;

        // Redirect to welcome page

        header("Location: welcome.php");

        exit;

    } else {

              echo "<p>Invalid username or password!</p>";

    }

}

?>

<!DOCTYPE html>

<html>

<head>

    <title>Login</title>

</head>

<body>

    <h2>Login</h2>

    <form method="post">

        <label for="username">Username:</label>

        <input type="text" name="username" required><br><br>

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

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

        <input type="submit" value="Login">

    </form>

</body>

</html>

**Welcome.php**

<?php

session\_start();

// Display welcome message

$username = $\_SESSION['username'];

// Check if logout is clicked

if(isset($\_POST['logout'])) {

    // Unset all session variables

    session\_unset();

    // Destroy the session

    session\_destroy();

    // Redirect to login page

    header("Location: index.php");

    exit;

}

?>

<!DOCTYPE html>

<html>

<head>

    <title>Welcome</title>

</head>

<body>

    <h2>Welcome, <?php echo $username; ?>!</h2>

    <p>This is a protected page. Only logged in users can access this.</p>

    <form method="post">

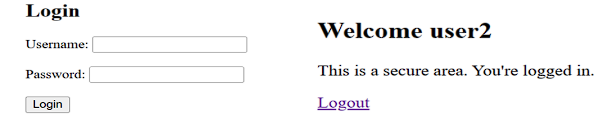
        <input type="submit" name="logout" value="Logout">

    </form>

</body>

</html>

**Output:**

****

**4.**Write a simple PHP program with a user interface for a mathematical calculator using HTML forms.

<!DOCTYPE html>

<html>

<head>

    <title>Simple PHP Calculator</title>

</head>

<body>

<h2>Simple PHP Calculator</h2>

<form method="post">

    <input type="text" name="n1" placeholder="Enter First number">

    <select name="op">

        <option value="+">+</option>

        <option value="-">-</option>

        <option value="\*">\*</option>

        <option value="/">/</option>

    </select>

     <input type="text" name="n2" placeholder="Enter Second number">

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

</form>

<?php

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

    $n1 = $\_POST['n1'];

    $n2 = $\_POST['n2'];

    $op = $\_POST['op'];

    // Checking if both operands are numeric

    if (!is\_numeric($n1) || !is\_numeric($n2)) {

        echo "Please enter valid numeric operands.";

    } else {

        // Performing calculation based on the selected operator

        switch ($op) {

            case '+':

                $result = $n1 + $n2;

                echo "Result: $n1 $op $n2 = $result";

                break;

            case '-':

                $result = $n1 - $n2;

                echo "Result: $n1 $op $n2 = $result";

                break;

            case '\*':

                $result = $n1 \* $n2;

                echo "Result: $n1 $op $n2 = $result";

                break;

            case '/':

                // Checking if the second operand is not zero for division

                if ($n2 == 0) {

                    echo "Cannot divide by zero.";

                } else {

                    $result = $n1/ $n2;

                    echo "Result: $n1 $op $n2 = $result";

                }

                break;

            default:

                echo "Invalid operator selected.";

        }

    }

}

?>

</body>

</html>

**Output:**



5. Write a PHP program with user interface for calculating age based on the user’s birth date.

Display the age in years, month, and days

<!DOCTYPE html>

<html>

<head>

    <title>Age Calculator</title>

</head>

<body>

<h2>Age Calculator</h2>

<form method="post">

    Enter your birth date: <input type="date" name="birthdate">

    <input type="submit" name="calculate" value="Calculate">

</form>

<?php

if(isset($\_POST['calculate'])) {

    // Get the birth date from the form

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

    // Calculate age

    $bdate\_tstamp = strtotime($birthdate);

    $cdate\_tstamp = time();

    $age\_in\_seconds = $cdate\_tstamp - $bdate\_tstamp;

    // Convert seconds to years, months, and days

    $age\_years = floor($age\_in\_seconds / (365.25 \* 24 \* 60 \* 60));

    $age\_months = floor(($age\_in\_seconds - ($age\_years \* 365.25 \* 24 \* 60 \* 60)) / (30.44 \* 24 \* 60 \* 60));

    $age\_days = floor(($age\_in\_seconds - ($age\_years \* 365.25 \* 24 \* 60 \* 60) - ($age\_months \* 30.44 \* 24 \* 60 \* 60)) / (24 \* 60 \* 60));

    // Display the age

    echo "<p>Your age is $age\_years years, $age\_months months, and $age\_days days.</p>";

}

?>

</body>

</html>

**Output:**

****

**6.**Write a PHP program to create an associative array representing a dictionary with words as keys and their meanings as values. Allow users to input a word and display its meaning if found, otherwise, display a "Word not found" message. (Please keep 10 pairs of keys and their meanings)

<?php

// Define the dictionary as an associative array

$dictionary = array(

    "apple" => "round fruit",

    "banana" => "curved fruit",

    "carrot" => "orange root vegetable",

    "dog" => "domesticated mammal",

    "elephant" => "large mammal with tusks",

    "flower" => "reproductive plant part",

);

// Check if the form has been submitted

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

    // Get the word entered by the user

    $searchWord = strtolower($\_POST["word"]);

    // Check if the word exists in the dictionary

    if (array\_key\_exists($searchWord, $dictionary)) {

        // If found, display its meaning

        $meaning = $dictionary[$searchWord];

        echo "<p>Meaning of \"$searchWord\": $meaning</p>";

    } else {

        // If not found, display a message

        echo "<p>Word not found</p>";

    }

}

?>

<!DOCTYPE html>

<html lang="en">

 <body>

    <h1>Dictionary</h1>

    <form method="post" >

        <label for="word">Enter a word:</label>

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

        <button type="submit">Search</button>

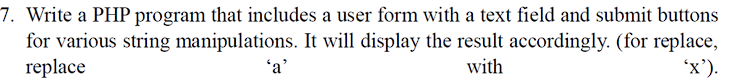
    </form>

</body>

</html>

**Output:**

****



<!DOCTYPE html>

<html>

<head>

    <title>String Manipulation</title>

</head>

<body>

    <h2>String Manipulation</h2>

    <form method="post">

        <label for="inputString">Enter a string:</label><br>

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

        <input type="submit" name="operation" value="Get Length">

        <input type="submit" name="operation" value="Upper Case">

        <input type="submit" name="operation" value="Lower Case">

        <input type="submit" name="operation" value="Replace">

        <input type="submit" name="operation" value="Check Palindrome">

        <input type="submit" name="operation" value="Shuffle">

        <input type="submit" name="operation" value="Word Count">

    </form>

    <?php

    if(isset($\_POST['inputString'])) {

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

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

        switch ($operation) {

            case 'Get Length':

                echo "<p>Length of the string: " . strlen($inputString) . "</p>";

                break;

            case 'Upper Case':

                echo "<p>Uppercase: " . strtoupper($inputString) . "</p>";

                break;

            case 'Lower Case':

                echo "<p>Lowercase: " . strtolower($inputString) . "</p>";

                break;

            case 'Replace':

                $resultString = str\_replace('a', 'b', $inputString);

                echo "<p>After replacing 'a' with 'b': $resultString</p>";

                break;

            case 'Check Palindrome':

                if(strtolower($inputString) == strrev(strtolower($inputString))) {

                    echo "<p>The string is a palindrome.</p>";

                } else {

                    echo "<p>The string is not a palindrome.</p>";

                }

                break;

       case 'Shuffle':

                echo "<p>Shuffled string: " . str\_shuffle($inputString) . "</p>";

                break;

        case 'Word Count':

                $words = str\_word\_count($inputString);

                echo "<p>Word count: $words</p>";

                break;

            default:

                echo "<p>Please select a valid operation.</p>";

                break;

        }

    }

    ?>

</body>

</html>

**Output:**

****

8. Write a PHP user interface program with an HTML form to input a string. Upon submission, it will display the number of times each word occurs, ignoring the distinction between capital and lowercase letters. It should also print the most and least used words

<!DOCTYPE html>

<html>

<head>

    <title>Character Occurrence Counter</title>

</head>

<body>

<h2>Character Counter</h2>

<form method="post">

    <label for="input\_string">Enter a string:</label><br>

    <input type="text"  name="input\_string"><br><br>

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

</form>

<?php

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

    // Retrieve input string from form

    $input\_string = $\_POST['input\_string'];

    // Convert input string to lowercase to ignore case distinction

    $input\_string = strtolower($input\_string);

    // Count occurrences of each character

    $char\_counts = array\_count\_values(str\_split($input\_string));

    // Sort character counts in descending order

    arsort($char\_counts);

    // Get the maximum occurrence count

    $max\_count = max($char\_counts);

    // Find all characters with the maximum occurrence count

    $most\_used\_chars = array\_keys($char\_counts, $max\_count);

    // Get the minimum occurrence count

    $min\_count = min($char\_counts);

    // Find all characters with the minimum occurrence count

    $least\_used\_chars = array\_keys($char\_counts, $min\_count);

        // Display character frequencies

    echo "<h3>Character Frequencies:</h3>";

    foreach ($char\_counts as $char => $count) {

        echo "'$char' : $count times<br>";

    }

        // Display most used characters

    echo "<p>Most used character(s) is/are: ";

    foreach ($most\_used\_chars as $char) {

        echo "'$char' ";

    }

    echo "</p>";

    // Display least used characters

    echo "<p>Least used character(s) is/are: ";

    foreach ($least\_used\_chars as $char) {

        echo "'$char' ";

    }

    echo "</p>";

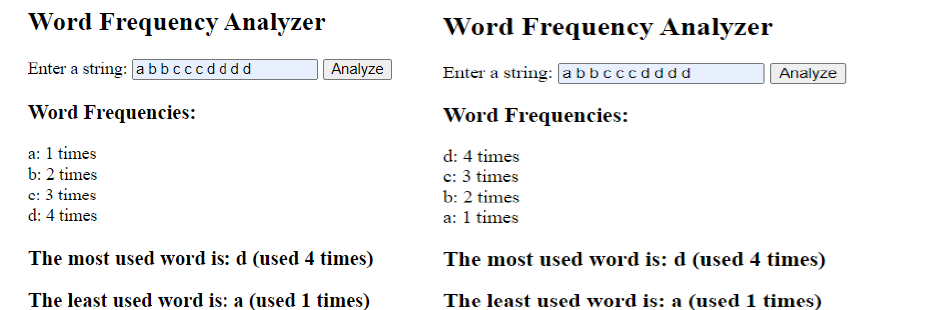
}

?>

</body>

</html>

**Output:**



PART-B

1. PHP program to implement student registration form using Labels, Text Boxes, Text Area, Checkbox, Radio Buttons, Select and Submit button. (First Name, Last Name, Address, E-Mail, Mobile, City, State, Gender, Hobbies, Blood Group). Display user inserted value in a new PHP page in a neat format.

**PB1.html**

<!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>

<style>

table {

border-collapse: collapse;

width: 50%;

}

th, td {

padding: 8px;

text-align: left;

border-bottom: 1px solid #ddd;

}

</style>

</head>

<body>

<h2>Student Registration Form</h2>

<form action="PB1.php" method="post">

<table>

<tr>

<td><label for="firstName">First Name:</label></td>

<td><input type="text" id="firstName" name="firstName" required></td>

</tr>

<tr>

<td><label for="lastName">Last Name:</label></td>

<td><input type="text" id="lastName" name="lastName" required></td>

</tr>

<tr>

<td><label for="address">Address:</label></td>

<td><textarea id="address" name="address" required></textarea></td>

</tr>

<tr>

<td><label for="email">E-Mail:</label></td>

<td><input type="email" id="email" name="email" required></td>

</tr>

<tr>

<td><label for="mobile">Mobile:</label></td>

<td><input type="text" id="mobile" name="mobile" required></td>

</tr>

<tr>

<td><label for="city">City:</label></td>

<td><input type="text" id="city" name="city" required></td>

</tr>

<tr>

<td><label for="state">State:</label></td>

<td><input type="text" id="state" name="state" required></td>

</tr>

<tr>

<td><label for="gender">Gender:</label></td>

<td>

<input type="radio" id="male" name="gender" value="Male" required>

<label for="male">Male</label>

<input type="radio" id="female" name="gender" value="Female" required>

<label for="female">Female</label>

</td>

</tr>

<tr>

<td><label for="hobbies">Hobbies:</label></td>

<td>

<input type="checkbox" id="hobby1" name="hobbies[]" value="Reading">

<label for="hobby1">Reading</label>

<input type="checkbox" id="hobby2" name="hobbies[]" value="Sports">

<label for="hobby2">Sports</label>

<input type="checkbox" id="hobby3" name="hobbies[]" value="Music">

<label for="hobby3">Music</label>

</td>

</tr>

<tr>

<td><label for="bloodGroup">Blood Group:</label></td>

<td>

<select id="bloodGroup" name="bloodGroup" required>

<option value="">Select</option>

<option value="A+">A+</option>

<option value="A-">A-</option>

<option value="B+">B+</option>

<option value="B-">B-</option>

<option value="AB+">AB+</option>

<option value="AB-">AB-</option>

<option value="O+">O+</option>

<option value="O-">O-</option>

</select>

</td>

</tr>

</table>

<br>

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

</form>

</body>

</html>

**PB1.PHP**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Student Information</title>

</head>

<body>

<h2>Student Information</h2>

<?php

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

echo "<p><strong>First Name:</strong> " . $\_POST["firstName"] . "</p>";

echo "<p><strong>Last Name:</strong> " . $\_POST["lastName"] . "</p>";

echo "<p><strong>Address:</strong> " . $\_POST["address"] . "</p>";

echo "<p><strong>E-Mail:</strong> " . $\_POST["email"] . "</p>";

echo "<p><strong>Mobile:</strong> " . $\_POST["mobile"] . "</p>";

echo "<p><strong>City:</strong> " . $\_POST["city"] . "</p>";

echo "<p><strong>State:</strong> " . $\_POST["state"] . "</p>";

echo "<p><strong>Gender:</strong> " . $\_POST["gender"] . "</p>";

if (!empty($\_POST["hobbies"])) {

echo "<p><strong>Hobbies:</strong> " . implode(", ", $\_POST["hobbies"]) . "</p>";

}

echo "<p><strong>Blood Group:</strong> " . $\_POST["bloodGroup"] . "</p>";

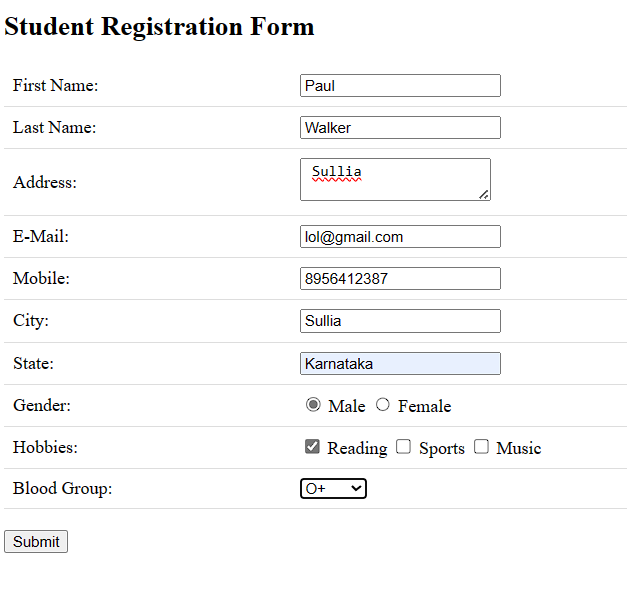
}

?>

</body>

</html>

**Output:**

2. Develop a PHP program that facilitates the addition, multiplication of two matrices. Utilize HTML for the user interface and PHP for the backend logic. Dynamically generate the required number of textboxes based on the specified number of rows and columns.

<!DOCTYPE html>

<html lang="en">

<head>

     <title>Matrix Operations</title>

    <style>

        table {

            border-collapse: collapse;

            margin-bottom: 20px;

        }

        th, td {

            padding: 8px;

            text-align: center;

            border: 1px solid #ddd;

        }

    </style>

</head>

<body>

    <h2>Matrix Operations</h2>

    <form action="" method="post">

        <label for="rows1">Rows (Matrix 1):</label>

        <input type="number" id="rows1" name="rows1" min="1" required>

        <label for="cols1">Columns (Matrix 1):</label>

        <input type="number" id="cols1" name="cols1" min="1" required><br><br>

        <label for="rows2">Rows (Matrix 2):</label>

        <input type="number" id="rows2" name="rows2" min="1" required>

        <label for="cols2">Columns (Matrix 2):</label>

        <input type="number" id="cols2" name="cols2" min="1" required><br><br>

        <input type="submit" name="generate" value="Generate Matrices">

    </form>

    <?php

    if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["generate"])) {

        $rows1 = $\_POST["rows1"];

        $cols1 = $\_POST["cols1"];

        $rows2 = $\_POST["rows2"];

        $cols2 = $\_POST["cols2"];

        echo '<form action="" method="post">';

        echo "<h3>Matrix 1:</h3>";

        generateMatrixInput("matrix1", $rows1, $cols1);

        echo "<h3>Matrix 2:</h3>";

        generateMatrixInput("matrix2", $rows2, $cols2);

        echo '<br><button type="submit" name="addition">Add</button>';

        if ($cols1 == $rows2) { // Check compatibility for multiplication

            echo '<button type="submit" name="multiplication">Multiply</button>';

        } else {

            echo '<button type="button" disabled>Multiply</button>';

            echo "<p>Matrices are not compatible for multiplication. Number of columns in Matrix 1 must match number of rows in Matrix 2.</p>";

        }

        echo '</form>';

    }

    function generateMatrixInput($matrixName, $rows, $cols) {

        echo "<table>";

        for ($i = 0; $i < $rows; $i++) {

            echo "<tr>";

            for ($j = 0; $j < $cols; $j++) {

                echo '<td><input type="number" name="' . $matrixName . '[' . $i . '][' . $j . ']" required></td>';

            }

            echo "</tr>";

        }

        echo "</table>";

    }

    if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["addition"])) {

        $matrix1 = $\_POST["matrix1"];

        $matrix2 = $\_POST["matrix2"];

        $result = matadd($matrix1, $matrix2);

        echo "<h3>Result (Addition):</h3>";

        displayMatrix($result);

    }

    if ($\_SERVER["REQUEST\_METHOD"] == "POST" && isset($\_POST["multiplication"])) {

        $matrix1 = $\_POST["matrix1"];

        $matrix2 = $\_POST["matrix2"];

        $result = matmul($matrix1, $matrix2);

        echo "<h3>Result (Multiplication):</h3>";

        displayMatrix($result);

    }

    function matadd($matrix1, $matrix2) {

        $result = array();

        for ($i = 0; $i < count($matrix1); $i++) {

            for ($j = 0; $j < count($matrix1[0]); $j++) {

                $result[$i][$j] = $matrix1[$i][$j] + $matrix2[$i][$j];

            }

        }

        return $result;

    }

    function matmul($matrix1, $matrix2) {

        $result = array();

        $cols1 = count($matrix1[0]);

        $rows2 = count($matrix2);

        for ($i = 0; $i < count($matrix1); $i++) {

            for ($j = 0; $j < count($matrix2[0]); $j++) {

                $result[$i][$j] = 0;

                for ($k = 0; $k < count($matrix2); $k++) {

                    $result[$i][$j] += $matrix1[$i][$k] \* $matrix2[$k][$j];

                }

            }

        }

        return $result;

    }

    function displayMatrix($matrix) {

        echo "<table>";

        foreach ($matrix as $row) {

            echo "<tr>";

            foreach ($row as $value) {

                echo "<td>$value</td>";

            }

            echo "</tr>";

        }

        echo "</table>";

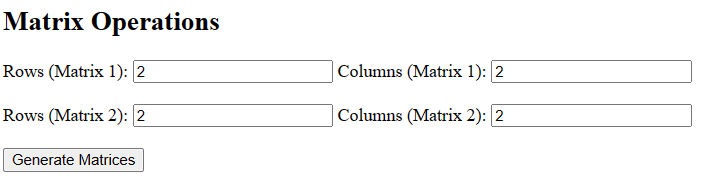
    }

    ?>

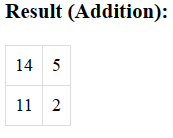
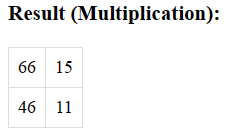
</body>

</html>

**Output:**





3. Write a PHP program that implements a class to add and find the difference of two distance values given in feet and inches. The user inputs are collected through an HTML form.

**PB3.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Distance Calculator</title>

</head>

<body>

    <h2>Distance Calculator</h2>

    <form action="PB3.php" method="post">

        <label for="feet1">Feet 1:</label>

        <input type="number" id="feet1" name="feet1" required> feet

        <input type="number" id="inches1" name="inches1" required> inches<br><br>

        <label for="feet2">Feet 2:</label>

        <input type="number" id="feet2" name="feet2" required> feet

        <input type="number" id="inches2" name="inches2" required> inches<br><br>

        <input type="submit" value="Calculate">

    </form>

</body>

</html>

**PB3.php**

<?php

class Distance {

    public $feet;

    public $inches;

    public function \_\_construct($feet, $inches) {

        $this->feet = $feet;

        $this->inches = $inches;

    }

    public function add($feet2, $inches2) {

        $ft = $this->feet + $feet2;

        $in = $this->inches + $inches2;

        if ($in >= 12) {

            $ft += floor($in / 12);

            $in = $in % 12;

        }

        return new Distance($ft, $in);

    }

    public function subtract($feet2, $inches2) {

        $ft = $this->feet - $feet2;

        $in = $this->inches - $inches2;

        if ($in < 0) {

            $ft--; // Reduce a foot

            $in += 12; // Add 12 inches

        }

        return new Distance($ft, $in);

    }

    public function getDistance() {

        return $this->feet . " feet " . $this->inches . " inches";

    }

}

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

    $feet1 = $\_POST["feet1"];

    $inches1 = $\_POST["inches1"];

    $feet2 = $\_POST["feet2"];

    $inches2 = $\_POST["inches2"];

    $distance1 = new Distance($feet1, $inches1);

    $distance2 = new Distance($feet2, $inches2);

    // Adding distances

    $sum = $distance1->add($feet2, $inches2);

    // Finding difference

    $difference = $distance1->subtract($feet2, $inches2);

    echo "<h2>Result</h2>";

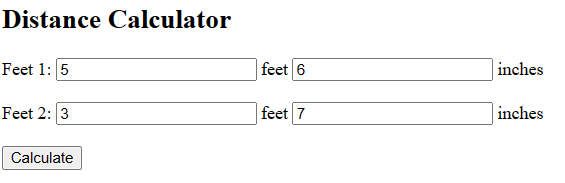
    echo "<p>Sum: " . $sum->getDistance() . "</p>";

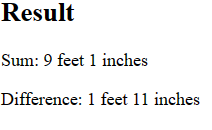
    echo "<p>Difference: " . $difference->getDistance() . "</p>";

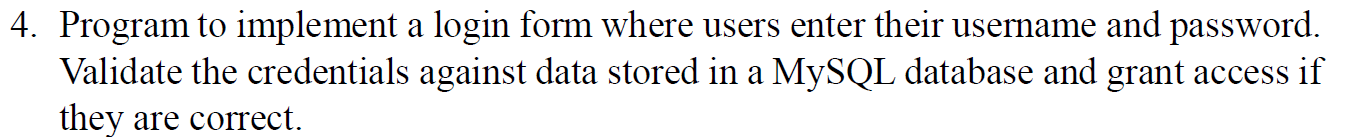
}

?>

**Output:**







CREATE TABLE login(

    uname VARCHAR(15) PRIMARY KEY,

    passwd VARCHAR(10)) ;

<?php

// MySQL database connection settings

$host = "localhost";

$username = "root";

$password = "";

$database = "BCA";

// Establishing a connection to the MySQL database

$conn = new mysqli($host, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

// Function to validate user login

function login($username, $password, $conn) {

    // SQL query to retrieve user from database

    $sql = "SELECT \* FROM login WHERE uname = '$username' AND passwd = '$password'";

    $result = $conn->query($sql);

    // If a matching user is found, grant access

    if ($result->num\_rows == 1) {

        echo "Login successful!";

    } else {

        echo "Invalid username or password";

    }

}

// Check if the login form has been submitted

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

    // Retrieve form data

    $username = $\_POST["username"];

    $password = $\_POST["password"];

    // Call the login function

    login($username, $password, $conn);

}

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Login Form</title>

</head>

<body>

    <h2>Login</h2>

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

        <label for="username">Username:</label><br>

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

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

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

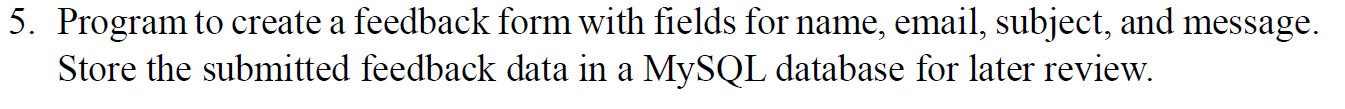
        <input type="submit" value="Login">

    </form>

</body>

</html>

**Output:**



CREATE TABLE feedback (

  name varchar(20) ,

  subject varchar(20) ,

  email varchar(20) ,

  message varchar(100)

);

<?php

// MySQL database connection settings

$host = "localhost";

$username = "root";

$password = "";

$database = "BCA";

// Establishing a connection to the MySQL database

$conn = new mysqli($host, $username, $password, $database);

// Check connection

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

// Check if the feedback form has been submitted

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

    // Retrieve form data

    $name = $\_POST["name"];

    $email = $\_POST["email"];

    $subject = $\_POST["subject"];

    $message = $\_POST["message"];

    // SQL query to insert feedback into database

    $sql = "INSERT INTO feedback (name, email, subject, message) VALUES ('$name', '$email', '$subject', '$message')";

    if ($conn->query($sql) === TRUE) {

        echo "Feedback submitted successfully!";

    } else {

        echo "Error: " . $sql . "<br>" . $conn->error;

    }

}

// Close the database connection

$conn->close();

?>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Feedback Form</title>

</head>

<body>

    <h2>Feedback Form</h2>

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

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

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

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

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

        <label for="subject">Subject:</label><br>

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

        <label for="message">Message:</label><br>

        <textarea id="message" name="message" rows="4" required></textarea><br><br>

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

    </form>

</body>

</html>

**Output:**

6. Develop a dynamic PHP application to efficiently manage and store customer information, encompassing key fields like Customer Number, Customer Name, Item Purchased, and Mobile Number in Database. This application should provide a user-friendly interface with strategically placed buttons to trigger specific functionalities. These functionalities include:

1.**Add Customer Information: Clicking this button should dynamically reveal a form for entering new customer details. Include proper validation checks for mobile numbers (10 digits), and also for Customer id ensuring accuracy in data input.**

**2.Delete Customer Records: Triggering this button should prompt the appearance of a form, specifically requesting the Customer ID to identify and delete the corresponding customer record. And provide appropriate messages for incorrect inputs.**

**3.Search for Particular Entries: This function should unveil a search form when activated, allowing users to input Customer id to find specific customer records.**

**4.Sort Database Based on Customer Id: Clicking this button should facilitate the sorting of the entire database based on customer id.**

**5.Display Complete Set of Records: Activating this function should present a comprehensive display of all customer records.**

**6.Interface Design: Initially, the interface should only showcase functional buttons. Upon clicking a button, the respective form should dynamically appear, offering a tailored and focused user experience.**

**7.Give proper messages after every transaction.**

CREATE TABLE customer (

custid INT ,

cname VARCHAR(20),

itemname VARCHAR(50),

mobileno BIGINT(20));

<?php

// Database connection

$servername = "localhost";

$username = "root";

$password = "";

$database = "bca";

$conn = new mysqli($servername, $username, $password, $database);

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

// Function to add customer information

if (isset($\_POST['add\_customer'])) {

    $customer\_id = $\_POST['customer\_id'];

    $customer\_name = $\_POST['customer\_name'];

    $item\_purchased = $\_POST['item\_purchased'];

    $mobile\_number = $\_POST['mobile\_number'];

    // Validate mobile number

    if (strlen($mobile\_number) != 10 || !ctype\_digit($mobile\_number)) {

        echo "Invalid mobile number";

    } else {

        $sql = "INSERT INTO customer (custid, cname, itemname, mobileno) VALUES ('$customer\_id', '$customer\_name', '$item\_purchased', '$mobile\_number')";

        if ($conn->query($sql) === TRUE) {

            echo "Customer added successfully"."<br>";

        } else {

            echo "Error: " . $sql . "<br>" . $conn->error;

        }

    }

}

/// Function to delete customer record

if (isset($\_POST['delete\_customer'])) {

    $customer\_id = $\_POST['customer\_id'];

    $sql = "DELETE FROM customer WHERE custid='$customer\_id'";

    if ($conn->query($sql) === TRUE) {

        if ($conn->affected\_rows > 0) {

            echo "Customer deleted successfully";

        } else {

            echo "Error: Customer with ID " . $customer\_id . " not found";

        }

    } else {

        echo "Error deleting customer: " . $conn->error;

    }

}

// Function to search for particular entries

if (isset($\_POST['search\_customer'])) {

    $customer\_id = $\_POST['customer\_id'];

    $sql = "SELECT \* FROM customer WHERE custid='$customer\_id'";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

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

            echo "Customer ID: " . $row['custid'] . ", Name: " . $row['cname'] . ", Item Purchased: " . $row['itemname'] . ", Mobile Number: " . $row['mobileno'] . "<br>";

        }

    } else {

        echo "No results found";

    }

}

// Function to display all records

if (isset($\_POST['display\_all'])) {

    $sql = "SELECT \* FROM customer";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

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

            echo "Customer ID: " . $row['custid'] . ", Name: " . $row['cname'] . ", Item Purchased: " . $row['itemname'] . ", Mobile Number: " . $row['mobileno'] . "<br>";

        }

    } else {

        echo "No results found";

    }

}

// Function to sort database based on customer id and display all records

if (isset($\_POST['sort\_and\_display'])) {

    $sql = "SELECT \* FROM customer ORDER BY cname";

    $result = $conn->query($sql);

    if ($result->num\_rows > 0) {

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

            echo "Customer ID: " . $row['custid'] . ", Name: " . $row['cname'] . ", Item Purchased: " . $row['itemname'] . ", Mobile Number: " . $row['mobileno'] . "<br>";

        }

    } else {

        echo "No results found";

    }

}

$conn->close();

?>

<!DOCTYPE html>

<html>

<head>

    <title>Customer Management System</title>

</head>

<body>

    <button onclick="document.getElementById('addForm').style.display = 'block'">Add Customer Information</button>

    <button onclick="document.getElementById('deleteForm').style.display = 'block'">Delete Customer Records</button>

    <button onclick="document.getElementById('searchForm').style.display = 'block'">Search for Particular Entries</button>

    <form id="sortForm" method="post" style="display: inline;">

        <input type="hidden" name="sort\_and\_display" value="true">

        <input type="submit" value="Sort Database and Display All Records">

    </form>

    <form id="displayForm" method="post" style="display: inline;">

        <input type="hidden" name="display\_all" value="true">

        <input type="submit" value="Display All Records">

    </form>

    <div id="addForm" style="display: none;">

        <form method="post">

            Customer ID: <input type="text" name="customer\_id" required><br>

            Customer Name: <input type="text" name="customer\_name" required><br>

            Item Purchased: <input type="text" name="item\_purchased" required><br>

            Mobile Number: <input type="text" name="mobile\_number" required><br>

            <input type="submit" name="add\_customer" value="Add Customer">

        </form>

    </div>

    <div id="deleteForm" style="display: none;">

        <form method="post">

            Customer ID: <input type="text" name="customer\_id" required><br>

            <input type="submit" name="delete\_customer" value="Delete Customer">

        </form>

    </div>

    <div id="searchForm" style="display: none;">

        <form method="post">

            Customer ID: <input type="text" name="customer\_id" required><br>

            <input type="submit" name="search\_customer" value="Search Customer">

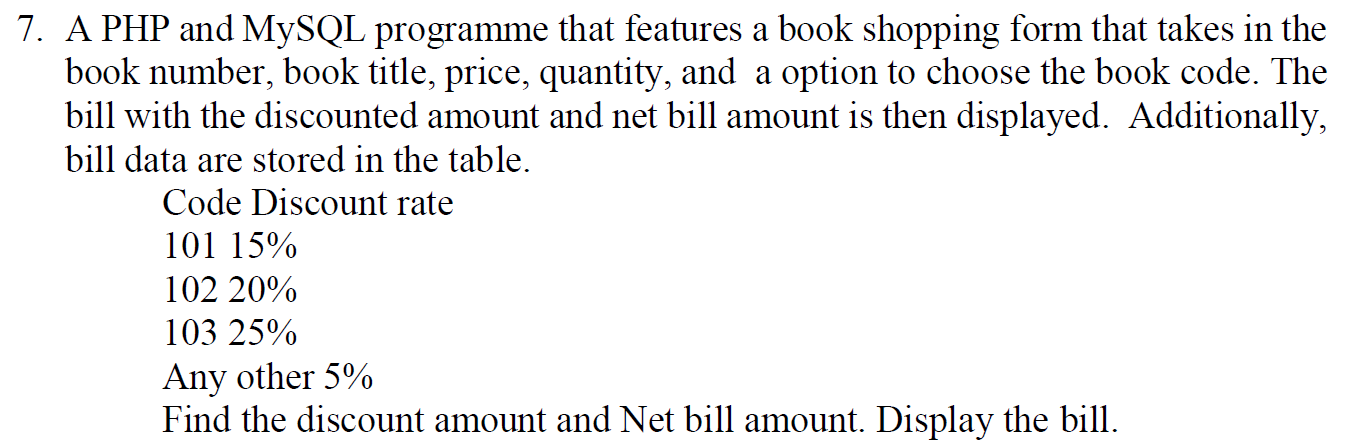
        </form>

    </div>

</body>

</html>

**Output:**



CREATE TABLE book (

  book\_number int ,

  book\_title varchar(255) ,

  price decimal(10,2) ,

  quantity int ,

  discount\_rate decimal(5,2) ,

  discount\_amount decimal(10,2),

  net\_bill\_amount decimal(10,2)

)

<!DOCTYPE html>

<html>

<head>

    <title>Book Shopping Form</title>

     <style>

        table {

            border-collapse: collapse;

            width: 100%;

        }

        th, td {

            border: 1px solid black;

            padding: 8px;

            text-align: left;

        }

        th {

            background-color: #f2f2f2;

        }

    </style>

</head>

<body>

    <h2>Book Shopping Form</h2>

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

        Book Number: <input type="number" name="book\_number" required><br><br>

        Book Title: <input type="text" name="book\_title" required><br><br>

        Price: <input type="number" name="price" step="0.01" required><br><br>

        Quantity: <input type="number" name="quantity" required><br><br>

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

    </form>

    <?php

    // MySQL connection

    $servername = "localhost";

    $username = "root";

    $password = "";

    $database = "bca";

    // Create connection

    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection

    if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

    }

    // Function to calculate discount amount based on discount rate

    function calDiscount($price, $discount\_rate) {

        return ($price \* $discount\_rate) / 100;

    }

    // Function to calculate net bill amount

    function calNetBill($price, $quantity, $discount\_amount) {

        return ($price \* $quantity) - $discount\_amount;

    }

    // Handling form submission

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

        $book\_number = $\_POST["book\_number"];

        $book\_title = $\_POST["book\_title"];

        $price = $\_POST["price"];

        $quantity = $\_POST["quantity"];

        // Calculate discount rate based on book number

        switch ($book\_number) {

            case 101:

                $discount\_rate = 15;

                break;

            case 102:

                $discount\_rate = 20;

                break;

            case 103:

                $discount\_rate = 25;

                break;

            default:

                $discount\_rate = 5;

        }

        // Calculate discount amount

        $discount\_amount = calDiscount($price, $discount\_rate);

        // Calculate net bill amount

        $net\_bill\_amount = calNetBill($price, $quantity, $discount\_amount);

        // Insert data into MySQL table

        $sql = "INSERT INTO BOOK (book\_number, book\_title, price, quantity, discount\_rate, discount\_amount, net\_bill\_amount)

                VALUES ('$book\_number', '$book\_title', '$price', '$quantity', '$discount\_rate', '$discount\_amount', '$net\_bill\_amount')";

        if ($conn->query($sql) === TRUE) {

            echo "Bill saved successfully.";

        } else {

            echo "Error: " . $sql . "<br>" . $conn->error;

        }

        // Display bill data for the current input data

        echo "<h2 style='text-align: center;'>Book Bill</h2>";

        echo "<table>";

        echo "<tr><th>Book Number</th><th>Book Title</th><th>Price</th><th>Quantity</th><th>Discount Rate (%)</th><th>Discount Amount</th><th>Net Bill Amount</th></tr>";

        echo "<tr>";

        echo "<td>".$book\_number."</td>";

        echo "<td>".$book\_title."</td>";

        echo "<td>".$price."</td>";

        echo "<td>".$quantity."</td>";

        echo "<td>".$discount\_rate."</td>";

        echo "<td>".$discount\_amount."</td>";

        echo "<td>".$net\_bill\_amount."</td>";

        echo "</tr>";

        echo "</table>";

    }

    // Close MySQL connection

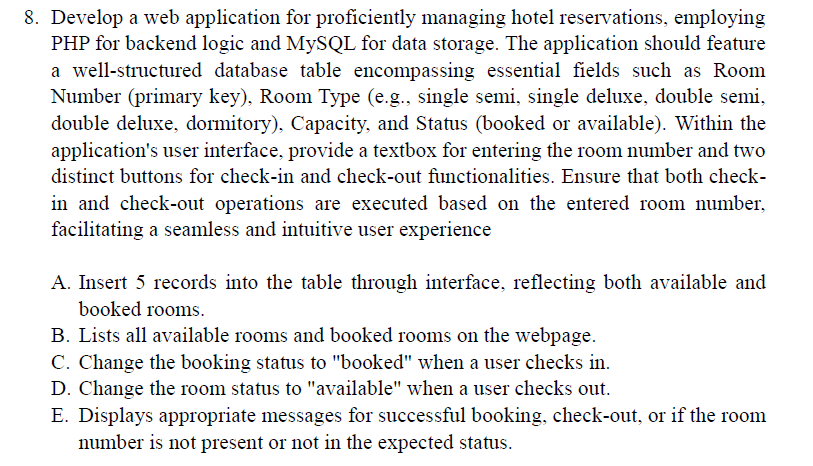
    $conn->close();

    ?>

</body>

</html>

**Output:**



CREATE TABLE rooms (

room\_number INT NOT NULL PRIMARY KEY,

room\_type VARCHAR(50),

capacity INT,

status ENUM('available', 'booked')

);

<!DOCTYPE html>

<html>

<head>

    <title>Hotel Reservation System</title>

</head>

<body>

    <h2>Hotel Reservation System</h2>

    <!-- Form for inserting records -->

    <h3>Insert Records</h3>

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

        <table>

            <tr>

                <td>Room Number:</td>

                <td><input type="number" name="room\_number" required></td>

            </tr>

            <tr>

                <td>Room Type:</td>

                <td><input type="text" name="room\_type" required></td>

            </tr>

            <tr>

                <td>Capacity:</td>

                <td><input type="number" name="capacity" required></td>

            </tr>

            <tr>

                <td>Status:</td>

                <td><input type="text" name="status" required></td>

            </tr>

        </table>

        <br>

        <input type="submit" name="insert" value="Insert Record">

    </form>

    <?php

    // MySQL connection

    $servername = "localhost";

    $username = "root";

    $password = "";

    $database = "BCA";

    // Create connection

    $conn = new mysqli($servername, $username, $password, $database);

    // Check connection

    if ($conn->connect\_error) {

        die("Connection failed: " . $conn->connect\_error);

    }

    // Insert records

    if (isset($\_POST['insert'])) {

        $room\_number = $\_POST["room\_number"];

        $room\_type = $\_POST["room\_type"];

        $capacity = $\_POST["capacity"];

        $status = $\_POST["status"];

        $sql\_insert = "INSERT INTO rooms (room\_number, room\_type, capacity, status)

                       VALUES ('$room\_number', '$room\_type', '$capacity', '$status')";

        if ($conn->query($sql\_insert) === TRUE) {

            echo "Record inserted successfully.";

        } else {

            echo "Error: " . $sql\_insert . "<br>" . $conn->error;

        }

    }

    ?>

    <!-- Displaying available and booked rooms in tabular form -->

    <h3>Available Rooms</h3>

    <?php

    $sql\_available = "SELECT \* FROM rooms WHERE status = 'available'";

    $result\_available = $conn->query($sql\_available);

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

        echo "<table border='1'>

                <tr>

                    <th>Room Number</th>

                    <th>Room Type</th>

                    <th>Capacity</th>

                </tr>";

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

            echo "<tr>

                    <td>" . $row["room\_number"] . "</td>

                    <td>" . $row["room\_type"] . "</td>

                    <td>" . $row["capacity"] . "</td>

                </tr>";

        }

        echo "</table>";

    } else {

        echo "No available rooms.";

    }

    ?>

    <h3>Booked Rooms</h3>

    <?php

    $sql\_booked = "SELECT \* FROM rooms WHERE status = 'booked'";

    $result\_booked = $conn->query($sql\_booked);

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

        echo "<table border='1'>

                <tr>

                    <th>Room Number</th>

                    <th>Room Type</th>

                    <th>Capacity</th>

                </tr>";

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

            echo "<tr>

                    <td>" . $row["room\_number"] . "</td>

                    <td>" . $row["room\_type"] . "</td>

                    <td>" . $row["capacity"] . "</td>

                </tr>";

        }

        echo "</table>";

    } else {

        echo "No booked rooms.";

    }

    // Close MySQL connection

    $conn->close();

    ?>

    <!-- Form for checking in -->

    <h3>Check-in</h3>

    <form method="post" >

        Room Number: <input type="number" name="room\_number" required><br><br>

        <input type="submit" name="check\_in" value="Check-in">

    </form>

    <!-- Form for checking out -->

    <h3>Check-out</h3>

    <form method="post" >

        Room Number: <input type="number" name="room\_number" required><br><br>

        <input type="submit" name="check\_out" value="Check-out">

    </form>

    <?php

    // MySQL connection (reopen for check-in and check-out)

    $conn = new mysqli($servername, $username, $password, $database);

    // Check-in functionality

    if (isset($\_POST['check\_in'])) {

        $room\_number = $\_POST["room\_number"];

        $sql\_check\_room = "SELECT \* FROM rooms WHERE room\_number = '$room\_number' AND status = 'available'";

        $result\_check\_room = $conn->query($sql\_check\_room);

        if ($result\_check\_room->num\_rows == 1) {

            $sql\_check\_in = "UPDATE rooms SET status = 'booked' WHERE room\_number = '$room\_number'";

            if ($conn->query($sql\_check\_in) === TRUE) {

                echo "Room checked in successfully.";

            } else {

                echo "Error: " . $sql\_check\_in . "<br>" . $conn->error;

            }

        } else {

            echo "Room is not available for check-in.";

        }

    }

    // Check-out functionality

    if (isset($\_POST['check\_out'])) {

        $room\_number = $\_POST["room\_number"];

        $sql\_check\_room = "SELECT \* FROM rooms WHERE room\_number = '$room\_number' AND status = 'booked'";

        $result\_check\_room = $conn->query($sql\_check\_room);

        if ($result\_check\_room->num\_rows == 1) {

            $sql\_check\_out = "UPDATE rooms SET status = 'available' WHERE room\_number = '$room\_number'";

            if ($conn->query($sql\_check\_out) === TRUE) {

                echo "Room checked out successfully.";

            } else {

                echo "Error: " . $sql\_check\_out . "<br>" . $conn->error;

            }

        } else {

            echo "Room is either not booked or does not exist.";

        }

    }

    // Close MySQL connection

    $conn->close();

    ?>

</body>

</html>

**Output:**