PART-A

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.php

```
<!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>
```

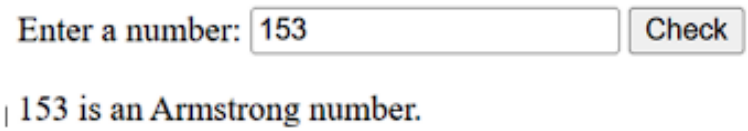
OUTPUT:


Contact Us
Name:
kiran
Email:
lol@gmail.com
Message:
lol
Submit

2. Write a PHP program that collects input from the user through a textbox and checks whether the given number is an Armstrong number or not.

```
<!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:



3. Write a PHP program demonstrating the usage of sessions to store and retrieve user credentials like username and password. Combine the login, welcome and logout functionalities into two files – 'index.php' and 'welcome.php'.

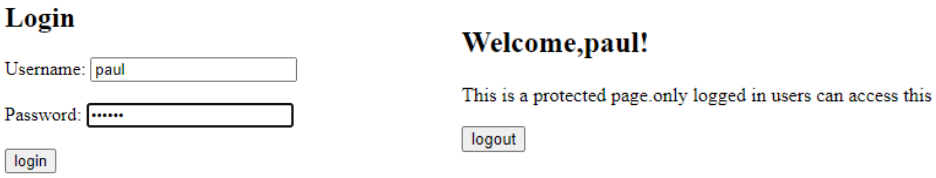
Index.php

```
<?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:

7. Write a PHP program that includes a user form with a text field and submit buttons for various string manipulations. It will display the result accordingly. (for replace, 'a' with 'x').

```
<!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:

String manipulation

Enter String lol

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

Length of the string:3

String manipulation

Enter String

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

Uppercase:LOL

String manipulation

Enter String

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

lowercase:lol

String manipulation

Enter String

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

after replacing a with b:bbc

String Manipulation

Enter a string:

Get Length | Upper Case | Lower Case | Replace | Check Palindrome | Shuffle | Word Count

The string is a palindrome.

String manipulation

Enter String

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

suffled string:bac

String manipulation

Enter String

GetLength | Uppercase | Lowercase | Replace | CheckPalindrome | shuffle | word count

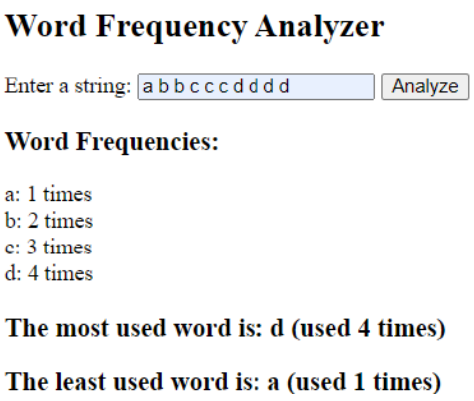
Word count: 1

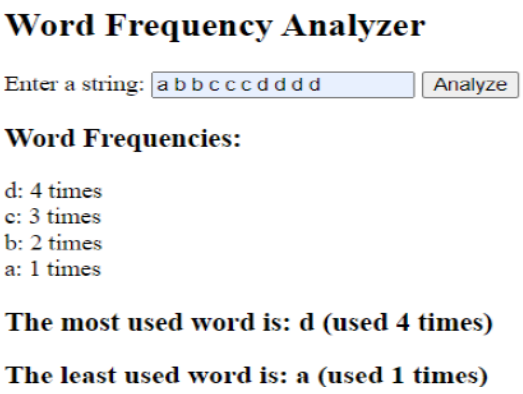
8. Write a PHP user interface program with an HTML form to input a string. Upon submission, it will display the number of times each word occurs, ignoring the distinction between capital and lowercase letters. It should also print the most and least used words

```
<!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:

Student Registration Form

First Name:	Paul
Last Name:	Walker
Address:	Sullia
E-Mail:	lol@gmail.com
Mobile:	8956412387
City:	Sullia
State:	Karnataka
Gender:	(•) Male ( ) Female
Hobbies:	[x] Reading [ ] Sports [ ] Music
Blood Group:	O+

Submit

Student Information

First Name: Paul

Last Name: Walker

Address: Sullia

E-Mail: lol@gmail.com

Mobile: 8956412387

City: Sullia

State: Karnataka

Gender: Male

Hobbies: Reading

Blood Group: O+

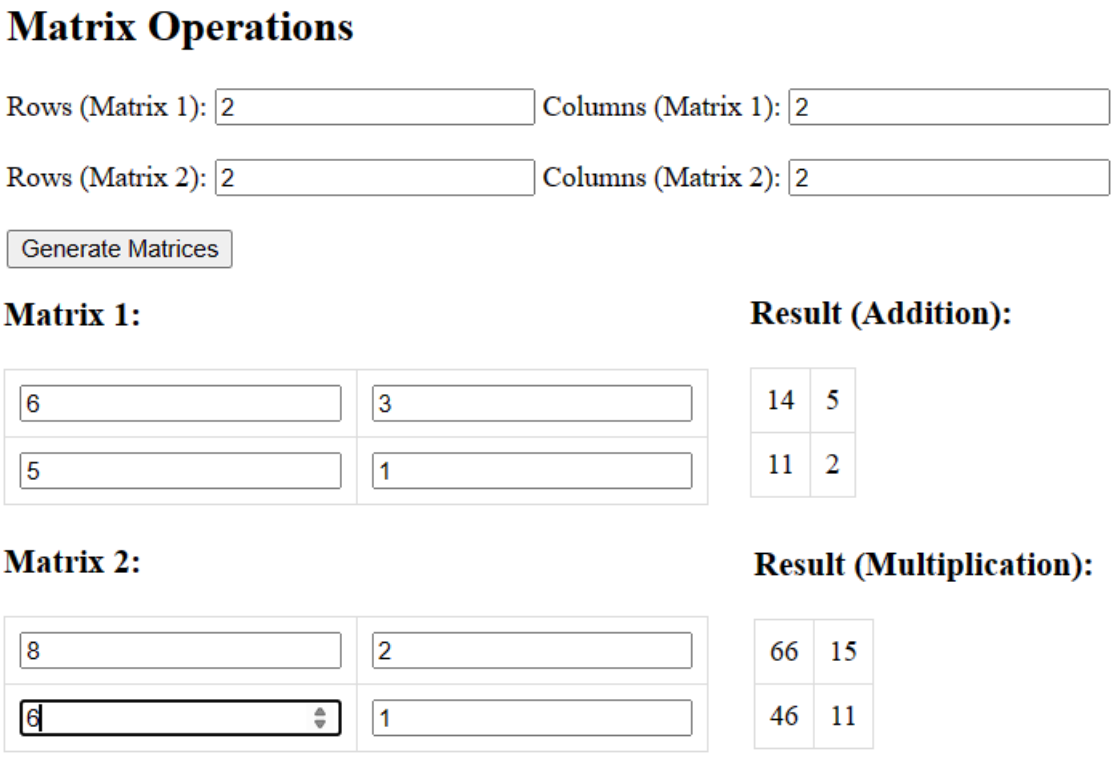
2. Develop a PHP program that facilitates the addition, multiplication of two matrices. Utilize HTML for the user interface and PHP for the backend logic. Dynamically generate the required number of textboxes based on the specified number of rows and columns.

```
<!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

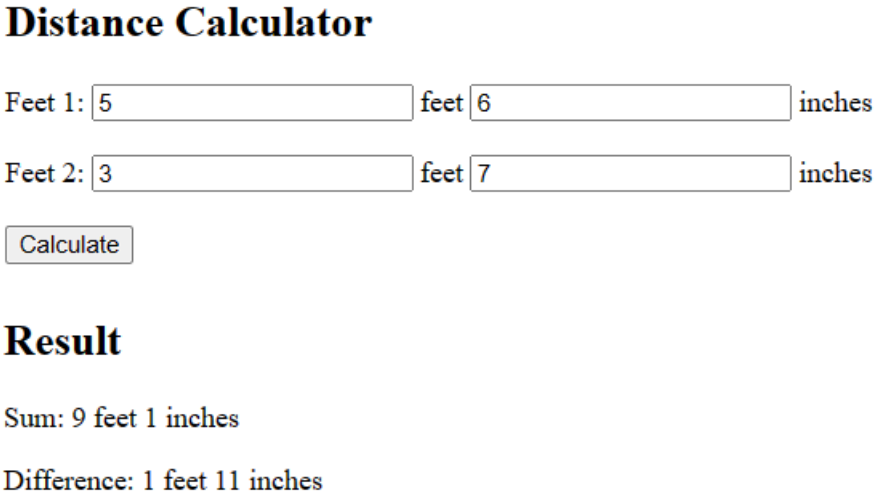
```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>Distance Calculator</title>
</head>
<body>
   <h2>Distance Calculator</h2>
   <form action="PB3.php" method="post">
      <label for="feet1">Feet 1:</label>
      <input type="number" id="feet1" name="feet1" required> feet
      <input type="number" 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:



4. Program to implement a login form where users enter their username and password. Validate the credentials against data stored in a MySQL database and grant access if they are correct.

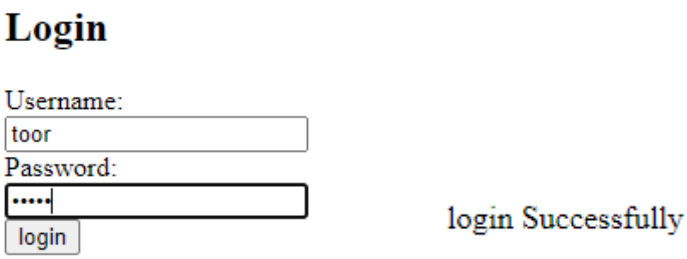
```
CREATE TABLE login(
   uname VARCHAR(15) PRIMARY KEY,
   passwd VARCHAR(10)) ;
```

```
<?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:



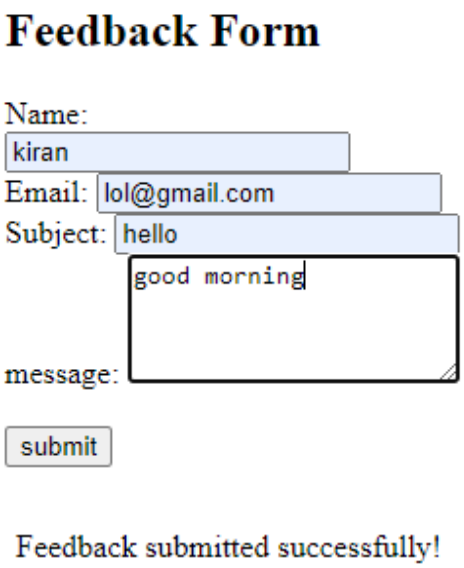
5. Program to create a feedback from with fields for name, email, subject and message. Store the submitted feedback data in a MySQL database for later review.

```
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:

Add Customer Information

Delete Customer Records

Search for Particular Entries

Sort Database and Display All Records

Display All Records

Customer ID: 1234
Customer Name: kiran
Item Purchased: mouse
Mobile Number: 8754962131
Add Customer

Customer ID: 1010, Name: kiran, Item Purchased: mouse, Mobile Number: 8754962131

Add Customer Information

Delete Customer Records

Search for Particular Entries

Sort Database and Display All Records

Display All Records

Customer ID: 1234, Name: kiran, Item Purchased: mouse, Mobile Number: 8754962131
Customer ID: 1010, Name: kiran, Item Purchased: mouse, Mobile Number: 8754962131
Customer ID: 1235, Name: lol, Item Purchased: key, Mobile Number: 4521689731

Add Customer Information

Delete Customer Records

Search for Particular Entries

Sort Database and Display All Records

Display All Records

7. **A PHP and MySQL program that features a book shopping form that takes in the book number, book title, price, quantity and a option to choose the book code. The bill with the discounted amount and net bill amount is then displayed. Additionally, bill data are stored in the table.**

    Code Discount rate
    101 15%
    102 20%
    103 25%
    Any other 5%
    Find the discount amount and Net bill amount. Display the bill.

```
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:

Book Shopping Form

Book Number:

Book Title:

Price:

Quantity:

Submit

Bill saved successfully.

Book Bill

Book Number	Book Title	Price	Quantity	Discount Rate (%)	Discount Amount	Net Bill Amount
101	hello	250	5	15	37.5	1212.5

8. **Develop a web application for proficiently managing hotels reservations, employing PHP for backend logic and MySQL for data storage. The application should feature a well-structured database table encompassing essential fields such as Room Number (Primary key), Room Type (e.g., single semi, single deluxe, double semi, double deluxe, dormitory), Capacity and Status (booked or available). Within the application's user interface, provide a textbox for entering the room number and two distinct buttons for check-in and check-out functionalities. Ensure that both check-in and check-out operations are executed based on the entered room number, facilitating a seamless and intuitive user experience**
    A. **Insert 5 records into the table through interface, reflecting both available and booked rooms.**
    B. **Lists all avalilable rooms and booked rooms on the webpage.**
    C. **Change the booking status to "booked" when a user checks in.**
    D. **Change the room status to "available" when a user checks out.**
    E. **Displays appropriate messages for successful booking, check-out or if the room number is not present or not in the expected status.**

```
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:

Hotel Reservation System

Insert Records

Room Number:

Room Type:

Capacity:

Status:

Insert Record

Available Rooms

Room Number	Room Type	Capacity
103	AC	4

Booked Rooms

Room Number	Room Type	Capacity
101	AC	4
102	AC	6

Check-in

Room Number:

Check-in

Check-out

Room Number:

Check-out

Room checked out successfully.