

1) **AIM:** write an XML file which will display the book information which includes the following:

- 1 Title of the book
- 2 Author name
- 3 Publisher Name
- 4 Editions
- 5 Price

Write a Document Type Definition (DTD) to validate the above XML file.

PROGRAM:

Ex1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<?DOCTYPE catalogue SYSTEM "book.dtd"?>

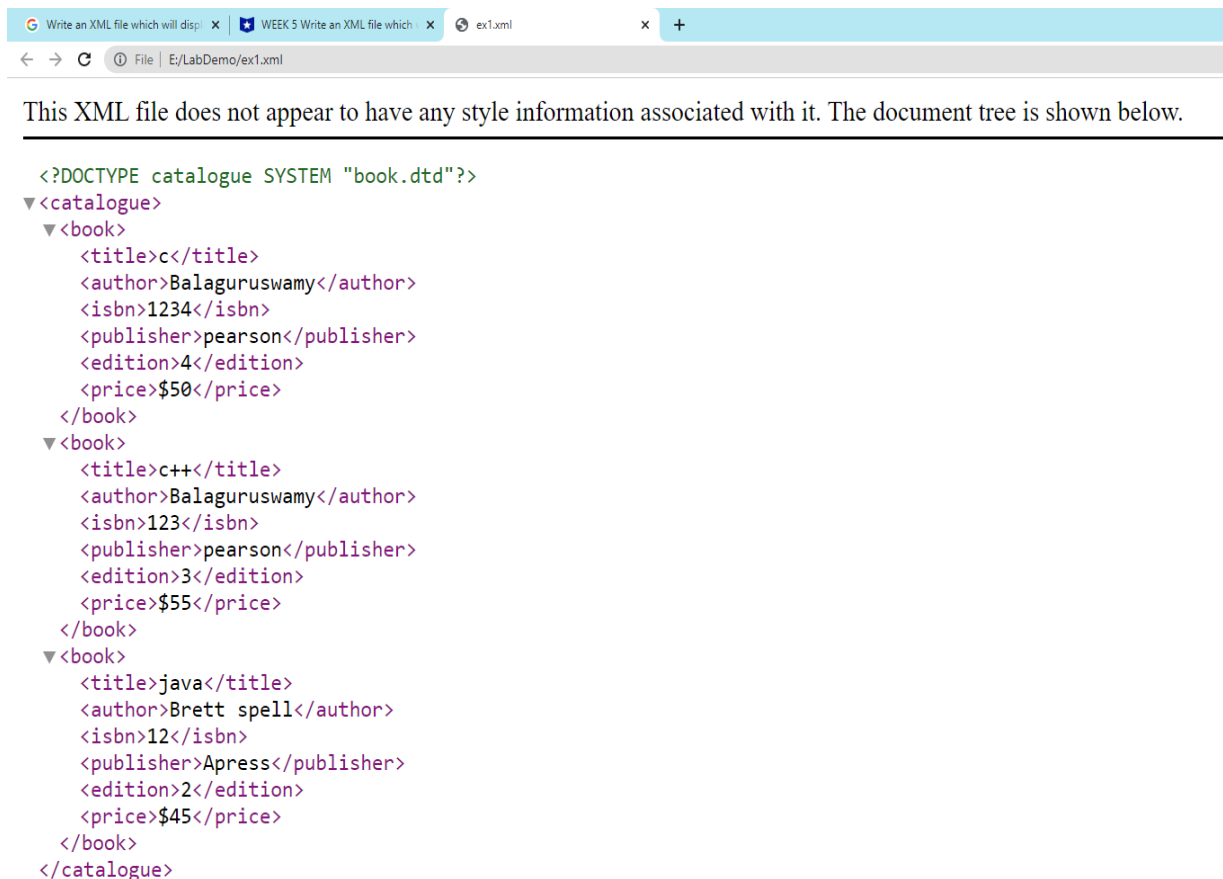
<catalogue>
  <book>
    <title>c</title>
    <author>Balaguruswamy</author>
    <isbn>1234</isbn>
    <publisher>pearson</publisher>
    <edition>4</edition>
    <price>$50</price>
  </book>
  <book>
    <title>c++</title>
    <author>Balaguruswamy</author>
    <isbn>123</isbn>
    <publisher>pearson</publisher>
    <edition>3</edition>
    <price>$55</price>
  </book>
  <book>
    <title>java</title>
    <author>Brett spell</author>
    <isbn>12</isbn>
    <publisher>Apress</publisher>
    <edition>2</edition>
    <price>$45</price>
  </book>
</catalogue>
```

Write a Document Type Definition (DTD) to validate the above XML file

Dtd file: book.dtd

```
<!ELEMENT catalogue (book)*>
<!ELEMENT book (title,author,isbn,publisher,edition,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

Output:



2. Write a PHP programs that uses arrays and functions in PHP.

Array in PHP

- An array stores multiple values in one single variable
- In PHP, there are three kinds of arrays:
 - Numeric array
 - Associative array
 - Multidimensional array

Numeric Array in PHP

Numeric array is an array with a numeric index

Numeric Array Example

```
<html>
  <body>
    <?php
      $flower_shop = array ("rose", "daisy","orchid");
      echo "Flowers: ".$flower_shop[0].",
        ".$flower_shop[1].", ".$flower_shop[2]."";
    ?>
  </body>
</html>
```

OUTPUT:

Flowers: rose, daisy, orchid

Associative array in PHP

Associative array is an array where each ID key is associated with a value

Associative array Example

```
<html>
  <body>
    <?php
      $flower_shop = array ( "rose" => "5.00", "daisy" => "4.00", "orchid"
        => "2.00" );
      // Display the array values
      echo "rose costs ".$flower_shop['rose'].",daisy costs
        ".$flower_shop['daisy'].",and orchild
        costs ".$flower_shop['orchild']."";
    ?>
  </body>
</html>
```

OUTPUT:

rose costs 5.00,daisy costs 4.00,and orchild costs

Loop through an Associative Array

```
<html>
  <body>
    <?php
      $flower_shop=array("rose"=>"5.00",
        "daisy"=>"4.00","orchid"=>"2.00");
      foreach($flower_shop as $x=>$x_value) {
        echo "Flower=" . $x . ", Value=" . $x_value;
        echo "<br>";
      }
    ?>
  </body>
</html>
```

OUTPUT:

```
Flower=rose, Value=5.00
Flower=daisy, Value=4.00
Flower=orchid, Value=2.00
```

Multidimensional array in PHP

Multidimensional array is an array containing one or more arrays

Multidimensional array Example

```
<html>
  <body>
    <?php
      $flower_shop = array(
        "rose" => array( "5.00", "7 items", "red" ),
        "daisy" => array( "4.00", "3 items", "blue" ),
        "orchid" => array( "2.00", "1 item", "white" ),
      );
      echo "rose costs ".$flower_shop['rose'][0].
        ", and you get ".$flower_shop['rose'][1]."<br>";
      echo "daisy costs ".$flower_shop['daisy'][0].
        ", and you get ".$flower_shop['daisy'][1]."<br>";
      echo "orchid costs ".$flower_shop['orchid'][0].
        ", and you get ".$flower_shop['orchid'][1]."<br>";
    ?>
  </body>
</html>
```

OUTPUT:

```
rose costs 5.00, and you get 7 items.
daisy costs 4.00, and you get 3 items.
orchid costs 2.00, and you get 1 item.
```

User Defined Function in PHP

Functions are group of statements that can perform a task

Syntax:

```
function functionName()  
{  
    code to be executed;  
}
```

User Defined Function Example

```
<html>  
    <body>  
        <?php  
            // Function definition  
            function myFunction()  
            {  
                echo "Hello world";  
            }  
            // Function call  
            myFunction();  
        ?>  
    </body>  
</html>
```

OUTPUT:

Hello world

Swap Numbers PHP Example

```
<html>
  <body>
    <?php
      $num1=10;
      $num2=20;
      echo "Numbers before swapping:<br/>";
      echo "Num1=".$num1;
      echo "<br/>Num2=".$num2;
      // Function call
      swap($num1,$num2);
      // Function definition
      function swap($n1,$n2)
      {
        $temp=$n1;
        $n1=$n2;
        $n2=$temp;
        echo "<br/><br/>Numbers after
          swapping:<br/>";
        echo "Num1=".$n1;
        echo "<br/>Num2=".$n2;
      }
    ?>
  </body>
</html>
```

OUTPUT:

Numbers before swapping:

Num1=10

Num2=20

Numbers after swapping:

Num1=20

Num2=10

PHP Functions - Adding parameters

```
<html>
  <body>
    <?php
      // Function definition
      function writeName($fname)
      {
        echo $fname . " Refsnes.<br />";
      }
      echo "My name is ";
      writeName("Kai Jim"); //Function call
      echo "My sister's name is ";
      writeName("Hege"); // Function call
      echo "My brother's name is ";
      writeName("Stale"); // Function call
    ?>
  </body>
</html>
```

OUTPUT:

My name is Kai Jim Refsnes.
My sister's name is Hege Refsnes.
My brother's name is Stale Refsnes.

PHP Functions - Return values

```
<html>
  <body>
    <?php
      // Function definition
      function add($x,$y)
      {
        $total=$x+$y;
        return $total;
      }
      // Function call
      echo "1 + 16 = " . add(1,16);
    ?>
  </body>
</html>
```

OUTPUT:

1 + 16 = 17

3) Write a PHP program for creating login form and validate users.

Aim: To write a PHP program for creating login form and validate users.

```
<!DOCTYPE HTML>

<html>

<head>

<style>

.error {color: #FF0000;}

</style>

</head>

<body>

<?php

// define variables and set to empty values

$nameErr = $passErr = "";

$name = $password = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    if (empty($_POST["name"])) {

        $nameErr = "Name is required";

    } else {

        $name = test_input($_POST["name"]);

    }

    if (empty($_POST["password"])) {

        $passErr = "Password is required";

    } else {

        $pass = test_input($_POST["password"]);

    }

}

function test_input($data) {

    $data = trim($data);

    $data = stripslashes($data);

    $data = htmlspecialchars($data);

    return $data;

}

?>

<h2>PHP Form Validation Example</h2>

<p><span class="error">* required field</span></p>
```

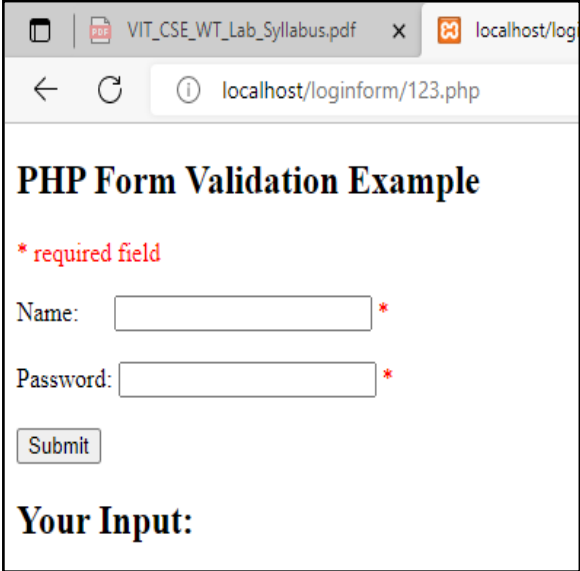


```

<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
    Name: <input type="text" name="name">
    <span class="error">* <?php echo $nameErr;?></span>
    <br><br>
    Password: <input type="text" name="password">
    <span class="error">* <?php echo $passErr;?></span>
    <br><br>
    <input type="submit" name="submit" value="Submit">
</form>
<?php
echo "<h2>Your Input:</h2>";
echo $name;
echo "<br>";
echo $password;
echo "<br>";
?>
</body>
</html>

```

OUTPUT:



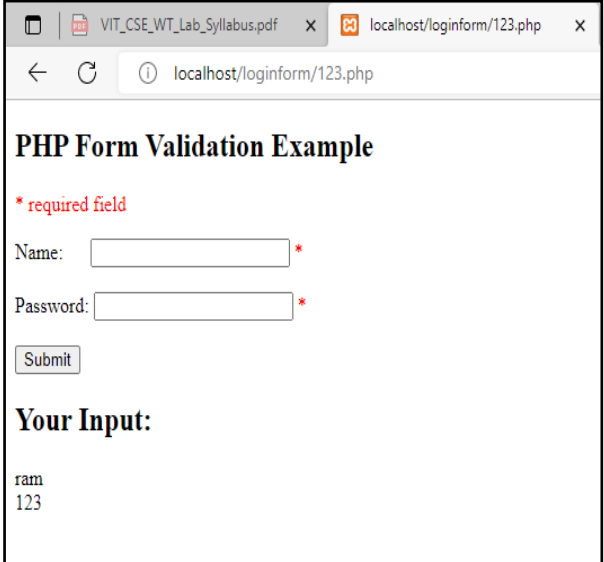
PHP Form Validation Example

* required field

Name: *

Password: *

Your Input:



PHP Form Validation Example

* required field

Name: *

Password: *

Your Input:

ram
123

4) Write a PHP program for to display all students in CSE using mysql student table.

Readphp.php

```
<!DOCTYPE html>

<html>

<head>

<title>Read Student Data From Database Using PHP </title>

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

</head>

<body>

<div class="maindiv">

<div class="divA">

<div class="title">

<h2>Read Data Using PHP</h2>

</div>

<div class="divB">

<div class="divD">

<p>Click On Menu</p>

<?php

$connection = mysql_connect("localhost", "root", "");

$db = mysql_select_db("demo", $connection); // Selecting Database

//MySQL Query to read data

$query = mysql_query("select * from student", $connection);

while ($row = mysql_fetch_array($query)) {

echo "<b>

<a href='readphp.php?id={$row['student_id']}'>{$row['student_name']}</a></b>";

echo "<br />";

}

?>

</div>

<?php

if (isset($_GET['id'])) {

$id = $_GET['id'];

$query1 = mysql_query("select * from student where student_id=$id", $connection);

while ($row1 = mysql_fetch_array($query1)) {

?>
```

```

<div class="form">
<h2>---Details---</h2>
<!-- Displaying Data Read From Database -->
<span>Name:</span> <?php echo $row1['student_name']; ?>
<span>E-mail:</span> <?php echo $row1['student_email']; ?>
<span>Contact No:</span> <?php echo $row1['student_contact']; ?>
<span>Address:</span> <?php echo $row1['student_address']; ?>
</div>
<?php
}
}
?>
<div class="clear"></div>
</div>
<div class="clear"></div>
</div>
</div>
<?php
mysql_close($connection); // Closing Connection with Server
?>
</body>
</html>

```

Database and Table Creation:

```

CREATE DATABASE demo;
CREATE TABLE student( student_id int(10) NOT NULL AUTO_INCREMENT,
student_name varchar(255) NOT NULL, student_email varchar(255) NOT NULL,
student_contact varchar(255) NOT NULL, student_address varchar(255) NOT NULL, student_dept
varchar(255) NOT NULL,
PRIMARY KEY (employee_id) )

```

style.css

```
@import "http://fonts.googleapis.com/css?family=Droid+Serif";

.maindiv {
margin:0 auto; width:980px;
height:500px;
background:#fff;
padding-top:20px;
font-size:14px;
font-family:'Droid Serif',serif
}

.title {
width:100%;
height:70px;
text-shadow:2px 2px 2px #cfcfcf;
font-size:16px;
text-align:center;
font-family:'Droid Serif',serif
}

.divA {
width:70%;
float:left;
margin-top:30px
}

.form {
width:400px;
float:left;
background-color:#fff;
font-family:'Droid Serif',serif;
padding-left:30px
}

.divB {
width:100%;
height:100%;
background-color:#fff;
border:dashed 1px #999
```

```
}  
.divD {  
width:200px; height:480px;  
padding:0 20px;  
float:left;  
background-color:#f0f8ff;  
border-right:dashed 1px #999  
}  
p {  
text-align:center;  
font-weight:700; color:#5678C0;  
font-size:18px;  
text-shadow:2px 2px 2px #cfcfcf  
}  
.form h2 {  
text-align:center;  
text-shadow:2px 2px 2px #cfcfcf  
}  
a {  
text-decoration:none;  
font-size:16px;  
margin:2px 0 0 30px;  
padding:3px;  
color:#1F8DD6  
}  
a:hover {  
text-shadow:2px 2px 2px #cfcfcf;  
font-size:18px  
}  
.clear {  
clear:both  
}  
span {  
font-weight:700  
}
```

Output:

READ DATA USING PHP	
Click On Menu	—Details—
Albert Peter	Name: Albert E-mail: alb123@gamil.com Contact No: 9999999999 Address: formget.com,USA

5) Create a PHP page for login system using session.

Aim: To create a PHP page for login system using session.

```
<?php
    ob_start();
    session_start();
?>
<html lang = "en">
    <head>
<link rel="stylesheet"
href=https://cdn.jsdelivr.net/npm/bootstrap@4.4.1/dist/css/bootstrap.min.css >
    <style>
    body {
        padding-top: 40px;
        padding-bottom: 40px;
        background-color: #ADABAB;
    }

    .form-signin {
        max-width: 330px;
        padding: 15px;
        margin: 0 auto;
        color: #017572;
    }

    .form-signin .form-signin-heading,
    .form-signin .checkbox {
        margin-bottom: 10px;
    }

    .form-signin .checkbox {
        font-weight: normal;
    }

    .form-signin .form-control {
        position: relative;
```

```
height: auto;
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;
box-sizing: border-box;
padding: 10px;
font-size: 16px;
}
```

```
.form-signin .form-control:focus {
  z-index: 2;
}
```

```
.form-signin input[type="email"] {
  margin-bottom: -1px;
  border-bottom-right-radius: 0;
  border-bottom-left-radius: 0;
  border-color:#017572;
}
```

```
.form-signin input[type="password"] {
  margin-bottom: 10px;
  border-top-left-radius: 0;
  border-top-right-radius: 0;
  border-color:#017572;
}
```

```
h2{
  text-align: center;
  color: #017572;
}
```

</style>

</head>

<body>

<h2>Enter Username and Password</h2>

<div class = "container form-signin">


```

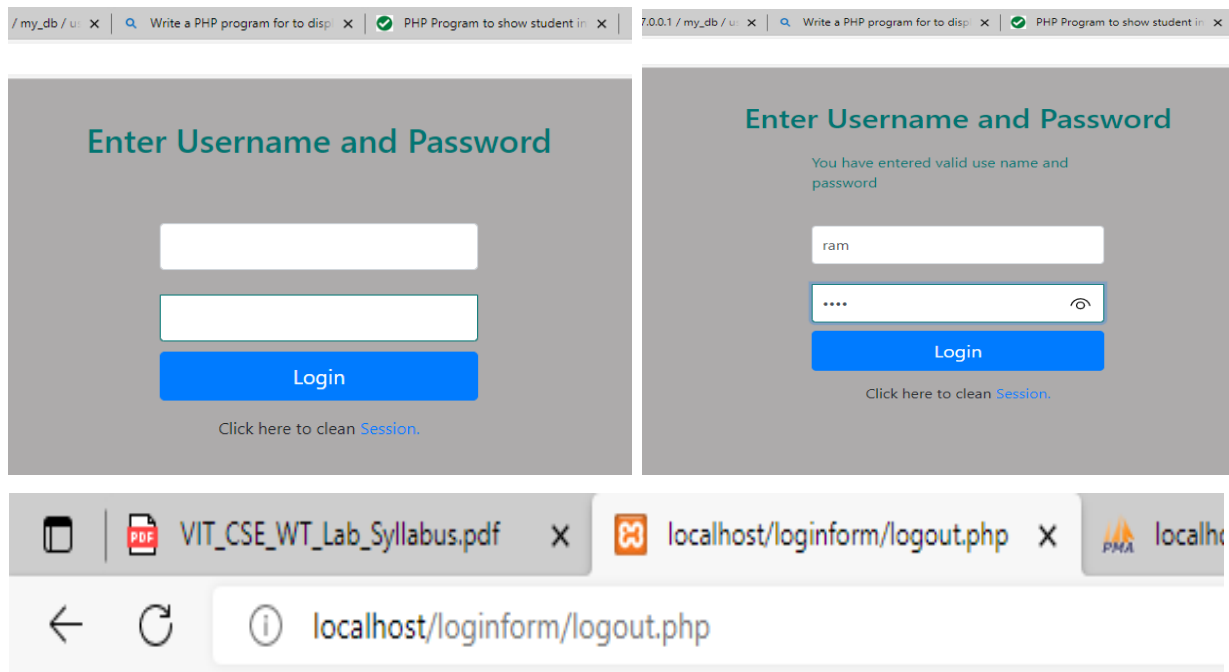
<?php
    $msg = "";
    if (isset($_POST['login']) && !empty($_POST['username'])
        && !empty($_POST['password'])) {
        if ($_POST['username'] == 'ram' &&
            $_POST['password'] == 1234) {
            $_SESSION['valid'] = true;
            $_SESSION['timeout'] = time();
            $_SESSION['username'] = 'ram';

            echo 'You have entered valid use name and password';
        }else {
            $msg = 'Wrong username or password';
        }
    }
?>
</div> <!-- /container -->
<div class = "container">
    <form class = "form-signin" role = "form"
    action = "<?php echo htmlspecialchars($_SERVER['PHP_SELF']);
    ?>" method = "post">
    <h4 class = "form-signin-heading"><?php echo $msg; ?></h4>
    <input type = "text" class = "form-control"
        name = "username" required autofocus></br>
    <input type = "password" class = "form-control"
        name = "password" required>
    <button class = "btn btn-lg btn-primary btn-block" type = "submit"
        name = "login">Login</button>
</form>

    <P style=text-align:center>
    Click here to clean <a href = "logout.php" tite = "Logout">Session.
</p>
</div>
</body>
</html>

```

OUTPUT:



You have cleaned session

6) Write a PHP program to connect MySQL

Aim: To make simple CRUD Application in PHP using MySQL and Bootstrap.

Step 1 – Create Database

Step 2 – Create a New Table

Step 3 – Database Connection File

Step 4 – Create a js and CSS file

Step 5 – Insert form data into database

Step 6 – Update form data into database

Step 7 – Retrieve and Display List

Step 8 – Delete data into database

Step 1 – Create Database

First of all, We need to create a database. So go to PHPMyAdmin and create a new database name my_database.

Step 2 – Create a New Table

Now we need to create a table named users. So go to PHPMyAdmin and run the below SQL query for creating a table in database:

```
CREATE TABLE `users` (  
    `id` bigint(20) NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    `name` varchar(255) DEFAULT NULL,  
    `email` varchar(255) DEFAULT NULL,  
    `mobile` varchar(255) DEFAULT NULL  
    ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Step 3 – Database Connection File

Connection.php

```
<?php  
$servername='localhost';  
$username='root';  
$password='root';  
$dbname = "my_db";  
$conn=mysqli_connect($servername,$username,$password,"$dbname");  
if(!$conn){  
    die('Could not Connect MySql Server:'.mysql_error());  
}  
?>
```

Step 4 – Create a js and CSS file

Head.php

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.js"></script>
```

Step 5 – Insert form data into database

Create.php

```
<?php
require_once "connection.php";
if(isset($_POST['save']))
{
$name = $_POST['name'];
$mobile = $_POST['mobile'];
$email = $_POST['email'];
$sql = "INSERT INTO users (name,mobile,email)
VALUES ('$name','$mobile','$email')";
if (mysqli_query($conn, $sql)) {
header("location: index.php");
exit();
} else {
echo "Error: " . $sql . "
" . mysqli_error($conn);
}
mysqli_close($conn);
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Create Record</title>
<?php include "head.php"; ?>
</head>
<body>
```

```

<div class="container">
<div class="row">
<div class="col-lg-12">
<div class="page-header">
<h2>Create Record</h2>
</div>
<p>Please fill this form and submit to add employee record to the database.</p>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
<div class="form-group">
<label>Name</label>
<input type="text" name="name" class="form-control" value="" maxlength="50" required="">
</div>
<div class="form-group ">
<label>Email</label>
<input type="email" name="email" class="form-control" value="" maxlength="30"
required="">
</div>
<div class="form-group">
<label>Mobile</label>
<input type="mobile" name="mobile" class="form-control" value="" maxlength="12"
required="">
</div>
<input type="submit" class="btn btn-primary" name="save" value="submit">
<a href="index.php" class="btn btn-default">Cancel</a>
</form>
</div>
</div>
</div>
</div>
</body>
</html>

```

Step 6 – Update form data into database

Update.php

```

<?php
// Include database connection file
require_once "connection.php";

```

```

if(count($_POST)>0) {
    mysqli_query($conn,"UPDATE users set name=" . $_POST['name'] . ", mobile=" .
    $_POST['mobile'] . ",email=" . $_POST['email'] . " WHERE id=" . $_POST['id'] . "");
    header("location: index.php");
    exit();
}
$result = mysqli_query($conn,"SELECT * FROM users WHERE id=" . $_GET['id'] . "");
$row= mysqli_fetch_array($result);
?>
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Update Record</title>
<?php include "head.php"; ?>
</head>
<body>
<div class="container">
<div class="row">
<div class="col-lg-12">
<div class="page-header">
<h2>Update Record</h2>
</div>
<p>Please edit the input values and submit to update the record.</p>
<form action="<?php echo htmlspecialchars(basename($_SERVER['REQUEST_URI'])); ?>"
method="post">
<div class="form-group">
<label>Name</label>
<input type="text" name="name" class="form-control" value="<?php echo $row["name"]; ?>"
maxlength="50" required="">
</div>
<div class="form-group">
<label>Email</label>
<input type="email" name="email" class="form-control" value="<?php echo $row["email"];
?>" maxlength="30" required="">

```

```

</div>
<div class="form-group">
<label>Mobile</label>
<input type="mobile" name="mobile" class="form-control" value="<?php echo
$row["mobile"]; ?>" maxlength="12"required="">
</div>
<input type="hidden" name="id" value="<?php echo $row["id"]; ?>" />
<input type="submit" class="btn btn-primary" value="Submit">
<a href="index.php" class="btn btn-default">Cancel</a>
</form>
</div>
</div>
</div>
</body>
</html>

```

Step 7 – Retrieve and Display List

Index.php

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Retrieve Or Fetch Data From MySQL Database Using PHP With Bootstrap</title>
<?php include "head.php"; ?>
<script type="text/javascript">
$(document).ready(function(){
$("[data-toggle="tooltip"]').tooltip();
});
</script>
</head>
<body>
<div class="container">
<div class="row">
<div class="col-lg-12 mx-auto">
<div class="page-header clearfix">
<h2 class="pull-left">Users List</h2>

```

```

<a href="create.php" class="btn btn-success pull-right">Add New User</a>
</div>
<?php
include_once 'connection.php';
$result = mysqli_query($conn,"SELECT * FROM users");
?>
<?php
if (mysqli_num_rows($result) > 0) {
?>
<table class='table table-bordered table-striped'>
<tr>
<td>Name</td>
<td>Email id</td>
<td>Mobile</td>
<td>Action</td>
</tr>
<?php
$i=0;
while($row = mysqli_fetch_array($result)) {
?>
<tr>
<td><?php echo $row["name"]; ?></td>
<td><?php echo $row["email"]; ?></td>
<td><?php echo ($row["mobile"])?($row["mobile"]):('N/A'); ?></td>
<td><a href="update.php?id=<?php echo $row["id"]; ?>" title='Update Record'><span
class='glyphicon glyphicon-pencil'></span></a>
<a href="delete.php?id=<?php echo $row["id"]; ?>" title='Delete Record'><i class='material-
icons'><span class='glyphicon glyphicon-trash'></span></a>
</td>
</tr>
<?php
$i++;
}
?>
</table>

```



```

<?php
}
else{
echo "No result found";
}
?>
</div>
</div>
</div>
</body>
</html>

```

Step 8 – Delete data into database

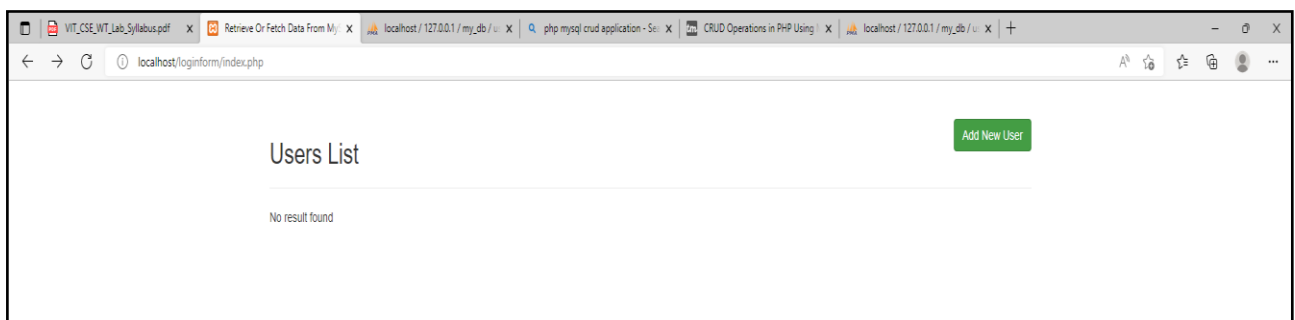
Delete.php

```

<?php
include_once 'connection.php';
$sql = "DELETE FROM users WHERE id=" . $_GET["id"] . """;
if (mysqli_query($conn, $sql)) {
header("location: index.php");
exit();
} else {
echo "Error deleting record: " . mysqli_error($conn);
}
mysqli_close($conn);
?>

```

OUTPUT:



localhost/loginform/create.php

Create Record

Please fill this form and submit to add employee record to the database.

Name

Email

Mobile

localhost/loginform/index.php

Users List

[Add New User](#)

Name	Email id	Mobile	Action
ram	ram@gmail.com	14123512353	Edit Delete

localhost/loginform/update.php?id=2

Update Record

Please edit the input values and submit to update the record.

Name

Email

Mobile

localhost/loginform/index.php

Users List

[Add New User](#)

Name	Email id	Mobile	Action
ram	ram@gmail.com	1634	Edit Delete

localhost/loginform/index.php

Users List

[Add New User](#)

No result found

7) Write a Node JS program to read and write file system

Aim: To write a Node JS program to read and write file system operation

File System Module

fs module is a build-in module in node js used to access file system. fs module has methods to read, write, append, rename, and delete data from a file stored in File System.

Example

```
const http=require('fs');
```

Read File

To read file in node js, we have both synchronous and asynchronous functions.

Read File Sync

fs.readFileSync method read a file synchronously.

Filename: Index.js

```
const fs=require('fs');  
var data=readFileSync('data.txt');  
console.log(data);
```

/*data.txt*/

hello node

Output:

<Buffer 68 65 6c 6c 6f 20 6e 6f 64 65>

Filename: Index.js

```
const fs=require('fs');  
var data=readFileSync('data.txt').toString();  
console.log(data);
```

/*data.txt*/

hello node

Output:

hello node

Read File

fs.readFile read any file asynchronously using a callback function as parameter.

Filename: Index.js

```
const fs=require('fs');  
fs.readFile("data.txt",(err,data)=>{  
  if(err){  
    console.log("Error : ", err);  
  }  
  else{  
    console.log(data.toString());  
  }  
})
```

*/*data.txt*/*

hello node

Output:

hello node

Read File with encoding

fs.readFile can have a option {encoding:'utf8'} to encode binary file. Without encoding, NodeJS will not decode the file into a string.

Filename: Index.js

```
const fs=require('fs');  
fs.readFile("data.txt",{encoding:'utf8'},(err,data)=>{  
  if(err){  
    console.log("Error : ", err);  
  }  
  else{  
    console.log(data); // .toString() not required  
  }  
})
```

*/*data.txt*/*

hello node

Output:

hello node

Check File Stats

To check file properties in NodeJS, we use fs.stat method. Node JS provides two APIs for both synchronous and asynchronous operations.

stat method

Filename: Index.js

```
const fs=require('fs');
fs.stat('src/data.txt', (err, stats) => {
  if (err) {
    console.error(err)
  }
  else{
    console.log(stats.isFile()); // true
    console.log(stats.isDirectory()); // false
    console.log(stats.size); // 1024
  }
});
```

statSync method

Filename: Index.js

```
const fs=require('fs');
try{
  const stats = fs.statSync('/Users/joe/test.txt');
}
catch(err){
  console.error(err);
}
```

Write File

To write in a file, node js use writeFile / writeFileSync methods.

```
const fs=require('fs');
fs.writeFileSync('data.txt','Hello Node');
```

If data.txt file is missing in current directory, node js will create a new directory with same name and then write. There is no need to create a file first and then write.

writeFileSync

To write in a file synchronously, node js used fs.writeFileSync method. The first parameter is file name and second is text data.

Filename: Index.js

```
const fs=require("fs");  
fs.writeFileSync('data.txt','Hello Node JS');
```

writeFile

To write in a file asynchronously, node js used fs.writeFile method. The first parameter is file name and second is text data and third is callback to handle errors..

Filename: Index.js

```
const fs=require("fs");  
fs.writeFile('data.txt','hello Node',(err)=>{  
  if(err){  
    console.log(err)  
  }  
})
```

Write File with utf-8

Filename: Index.js

```
const fs=require("fs");  
fs.writeFile('data.txt','hello Node','utf8',(err)=>{  
  if(err){  
    console.log(err)  
  }  
})
```

Append in file

To append in a file, use `appendFile` or `appendFileSync` methods of `fs`. This will not overwrite in file like `writeFile` and `writeFileSync`.

appendFileSync

`appendFileSync` method of `fs` append file asynchronously.

Filename: Index.js

```
const fs=require('fs');
fs.appendFileSync('src/data.txt','hello Node 1','utf8',(err)=>{
  if(err){
    console.log(err)
  }
});
fs.appendFileSync('src/data.txt','hello Node 2','utf8',(err)=>{
  if(err){
    console.log(err)
  }
});
```

Output:

Hello Node 1
Hello Node 2

appendFile

`appendFile` method of `fs` append file synchronously.

Filename: Index.js

```
const fs=require('fs');
fs.appendFile('src/data.txt','hello Node 1','utf8',(err)=>{
  if(err){
    console.log(err)
  }
});
fs.appendFile('src/data.txt','hello Node 2','utf8',(err)=>{
  if(err){
    console.log(err)
  }
});
```

Output:

Hello Node 1
Hello Node 2

delete file

To delete files, node js use fs.unlink or fs.unlinkSync methods.

fs.unlinkSync

```
const fs=require('fs');  
fs.unlinkSync('data.txt');
```

By using fs.unlinkSync without exception handling can create runtime errors. To handle this, use fs.unlinkSync with exception handling

Filename: Index.js

```
const fs=require('fs');  
try{  
  fs.unlinkSync('data.txt');  
  console.log('file deleted successfully');  
}  
catch(err){  
  console.log('Error',err);  
}
```

fs.unlink

fs.unlink is a asynchronously method to delete file with two arguments. First is file name and second is callback function.

Filename: Index.js

```
const fs=require('fs');  
fs.unlink('data.txt',(err)=>{  
  if(err){  
    console.log('Error:', err);  
  }  
  else{  
    console.log('file deleted successfully');  
  }  
})
```

Output:

file deleted successfully

8) Write a Node JS program to connect the MongoDB

Aim:

To perform CRUD Operation in MongoDB

Step 1

Connect your system with the internet and open the command prompt and then run

Command **install mongodb --save**

Step 2

Create a database in MongoDB using Node.js and VS Code. first, open VS Code and create a folder where you want to make database program. and then open this folder in VS Code

Step 3

For performing any crud operation in MongoDB, you need a database and a collection. First. create a database and then create a collection.

Step 4 - Create a database

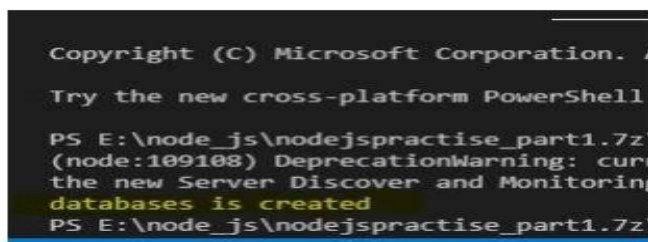
Create a .js page(createdatabase.js) and now write the code:

createdatabase.js

```
var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url='mongodb://localhost:27017/';
MongoClient.connect(url,function(error, databases){// use for to connect to the datab
ases
if(error){
throw error;
}
var dbobject=databases.db('navigcollection');//use for create database
console.log("databases is created")
databases.close();
})
```

Compile

node createdb.js



Step 5 - Create collection in database

Create a .js file ("createcollection.js") and write code.

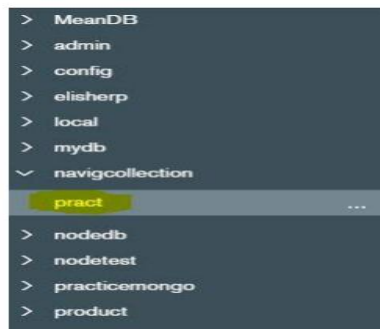
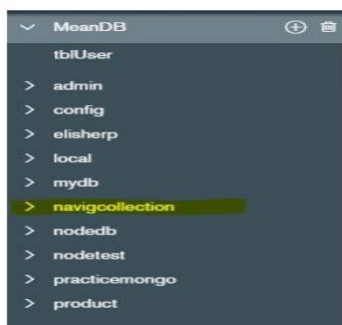
createcollection.js

```
var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url="mongodb://localhost:27017/"
MongoClient.connect(url,function(error,databases){
if(error){
throw error;
}
var dbase=databases.db("navigcollection");
dbase.createCollection("pract",function(error,response){
if(error){
throw error;
}
console.log("collection is created.....")
databases.close();
});
});
```

Compile

```
node createcollection.js
```

```
PS E:\node_js\nodejspractise_part1.7z\mongo_db\mongodbmy> node createcollection.js
(node:111080) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
collection is created.....
PS E:\node_js\nodejspractise_part1.7z\mongo_db\mongodbmy> 
```



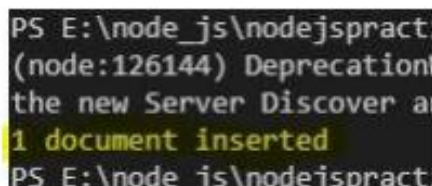
Step 6 - Insert record in database

insert.js"

```
var mongodb = require('mongodb');
var MongoClient = mongodb.MongoClient;
var url = "mongodb://localhost:27017/";
mongoClient.connect(url, function(err, databases) {
  if (err)
  {
    throw err;
  }
  var nodetestDB = databases.db("navigcollection"); //here
  var customersCollection = nodetestDB.collection("pract");
  var customer = {_id:111, name:"Santosh Kumar" , address: "B-222, Sector19, NOIDA",
  orderdata:"Arrow Shirt"};
  customersCollection.insertOne(customer, function(error, response) {
    if (error) {
      throw error;
    }
    console.log("1 document inserted");
    databases.close();
  });
});
```

Compile

```
node insert.js
```



```
PS E:\node_js\nodejspract
(node:126144) Deprecation
the new Server Discover a
1 document inserted
PS E:\node_js\nodejspract
```

Now insert record into the collection, create again a js file ("insertmanydocu.js") and write the code:

insertmanydocu.js

```
var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url='mongodb://localhost:27017/';
```

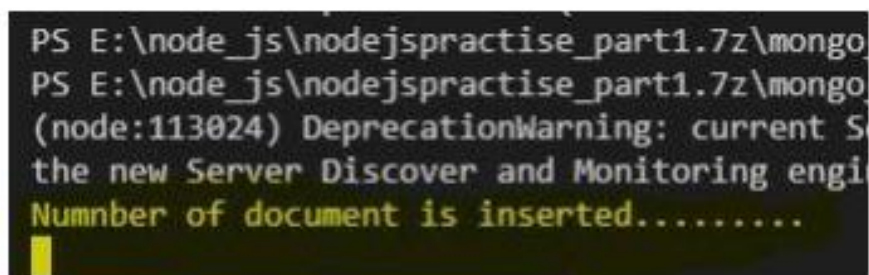
```

MongoClient.connect(url,function(error,databases){
  if(error){
    throw error;
  }
  var nodtst=databases.db("navigcollection");
  var pract=[
    {_id:11,name:"Chaman Gautam" , address: "Harvansh nagar Ghaziabad", orderdata:"Jeans"},
    {_id:12,name:"Shivani" , address: "Harvansh nagar Ghaziabad", orderdata:"Jeans"},
    {_id:13,name:"Menu" , address: "Harvansh nagar Ghaziabad", orderdata:"Top"},
    {_id:14,name:"Brajbala" , address: "Harvansh nagar Ghaziabad", orderdata:"Dining table"},
    {_id:15,name:"Ramsaran" , address: "Harvansh nagar Ghaziabad", orderdata:"Washing machine"},
    {_id:16,name:"Dheeraj" , address: "Harvansh nagar Ghaziabad", orderdata:"Jeans"}
  ]
  nodtst.collection('pract').insertMany(pract , function(error,response){
    if(error){
      throw error;
    }
    console.log("Number of document is inserted.....");
  })
. })

```

Compile

node insertmanydocu.js



```

PS E:\node_js\nodejspractise_part1.7z\mongo
PS E:\node_js\nodejspractise_part1.7z\mongo
(node:113024) DeprecationWarning: current S
the new Server Discover and Monitoring engi
Number of document is inserted.....

```

Step 7 - Find record from database

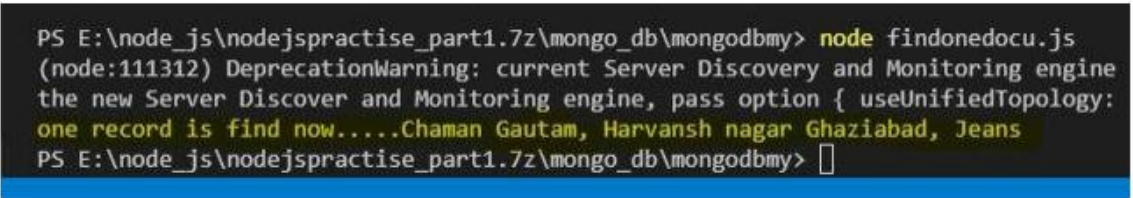
Find 1 record from collection

Now create a .js page("find1docu.js") create a page and write the code:

```
var mongodb=require("mongodb");
var MongoClient=mongodb.MongoClient;
var url='mongodb://localhost:27017/';
MongoClient.connect(url, function(error, databases){
if(error){
throw error;
}
var nodtst = databases.db("navigcollection");
nodtst.collection("pract").findOne({name:'Shivani'}, function(err, result) {
if (err) throw err;
console.log("one record is find now....."+result.name + ", " + result.address + ", " +
result.orderdata);
databases.close();
})
})
```

Compile

node findonedocu.js



```
PS E:\node_js\nodejspractise_part1.7z\mongo_db\mongodbmy> node findonedocu.js
(node:111312) DeprecationWarning: current Server Discovery and Monitoring engine
the new Server Discover and Monitoring engine, pass option { useUnifiedTopology:
one record is find now.....Chaman Gautam, Harvansh nagar Ghaziabad, Jeans
PS E:\node_js\nodejspractise_part1.7z\mongo_db\mongodbmy> █
```

find many record from collection

Now, create a new .js page("findmanudocu.js") and write the following code:

findmanudocu.js

```
var mongodb=require("mongodb");
var MongoClient=mongodb.MongoClient;
var url='mongodb://localhost:27017/';
MongoClient.connect(url, function(error, databases){
if(error){
throw error;
}
var nodtst = databases.db("navigcollection");
```

```

nodb.collection("pract").find({}).toArray(function(err, totalpract) {
if (err) throw err;
for(i = 0; i < totalpract.length; i++) {
let pract = totalpract[i];
console.log(pract.name + ", " + pract.address + ", " + pract.orderdata);
}
//console.log(result);
databases.close();
});
});

```

Compile

```
node findmanydocu.js
```

Step 8 - update record in collection

Update one record from collection

updateone.js

```

var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url="mongodb://localhost:27017/"
MongoClient.connect(url,function(error,databases){
if(error){
throw error;
}
var nodbst=databases.db("navigcollection");
var whereClause = { name:/Chaman Gautam/};
var newvalues = { $set: { name:"Lucky Gautam"} };
nodb.collection("pract").updateOne(whereClause,newvalues,function(err,res){
if(error){
throw error;
}
console.log(res.result.n + "document updated");
});
});

```

Compile

```
node updateone.js
```

Now update many records from collection

updatemany.js

```
var mongodb = require('mongodb');
var MongoClient = mongodb.MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, databases) {
  if (err)
  {
    throw err;
  }
  var nodeDB = databases.db("practicemongo"); //here
  var myquery = { address: /Harvansh nagar/ };
  var newvalues = { $set: { name: "Shivani" } };
  nodeDB.collection("pract").updateMany(myquery, newvalues, function(err, res) {
    if (err) throw err;
    console.log(res.result.nModified + " document(s) updated");
    databases.close();
  });
});
```

Compile

```
node updatemany.js
```

Step 9 - now delete operation

delete one record from collection

deleteone.js

```
var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url ='mongodb://localhost:27017/';
MongoClient.connect(url,function(error,databases){
  if(error)
  {
    throw error;
  }
  var noddst=databases.db('navigcollection');
  var deleteQuery={ name:'Menu'};
```

```

nodb.collection("pract").deleteOne(deleteQuery,function(error,response){
if(error){
throw error;
}
console.log(response.result.n+" 1 document deleted.....");
databases.close();
})
});

```

Compile

```
node deleteone.js
```

deletemany.js

```

var mongodb=require('mongodb');
var MongoClient=mongodb.MongoClient;
var url='mongodb://localhost:27017/';
MongoClient.connect(url,function(error,databases){
if(error)
{
throw error;
}
var nodbst=databases.db('navigcollection');
var deleteQuery={ };
nodb.collection('pract').deleteMany(deleteQuery,function(error,response){
if(error){
throw error;
}
console.log(response.result.n + "document(s) deleted successfully .....");
databases.close();
})
})

```


9) Write a servlet program which receives data from HTML forms and respond it. Create one Servlet to retrieve “ServletContext Initialization Parameters “which you have given in the web.xml file.

Login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login</title>
</head>
<body>
    <form action="ValidServ" method="post">
        Username: <input type="text" name="txtuser" /><br/>
        Password: <input type="password" name="txtpass" /><br/>
        <input type="submit" value="Submit" />
        <input type="reset" value="clear" />
    </form>
</body>
</html>
```

ValidServ.java

```
import java.io.IOException;
import java.util.Enumeration;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ValidServ extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ServletConfig cfg;

    public ValidServ() {
        super();
    }
```

```

// TODO Auto-generated constructor stub
}

/**
 * @see Servlet#init(ServletConfig)
 */
public void init(ServletConfig config) throws ServletException {
    cfg = config;
}

public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
    String un = request.getParameter("txtuser");
    String pw = request.getParameter("txtpass");
    boolean flag = false;
    Enumeration<String> initparams = cfg.getInitParameterNames();
    while(initparams.hasMoreElements())
    {
        String name = initparams.nextElement();
        String pass = cfg.getInitParameter(name);
        if(un.equals(name) && pw.equals(pass))
        {
            flag = true;
        }
    }
    if(flag)
    {
        response.getWriter().print("Valid user!");
    }
    else
    {
        response.getWriter().print("Invalid user!");
    }
}
}

```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
  <servlet>
    <servlet-name>ValidServ</servlet-name>
    <servlet-class>ValidServ</servlet-class>
    <init-param>
      <param-name>user1</param-name>
      <param-value>pass1</param-value>
    </init-param>
    <init-param>
      <param-name>user2</param-name>
      <param-value>pass2</param-value>
    </init-param>
    <init-param>
      <param-name>user3</param-name>
      <param-value>pass3</param-value>
    </init-param>
    <init-param>
      <param-name>user4</param-name>
      <param-value>pass4</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>ValidServ</servlet-name>
    <url-pattern>/ValidServ</url-pattern>
  </servlet-mapping>
</web-app>
```

10) Write a servlet program to authenticate four users using cookies.

Aim: To write a servlet program to authenticate four users using cookies.

cologin.html:

```
<html>
<head>
<title> login Page </title>
<p style= "background:yellow; top:100px; left:250px; position:absolute; ">
</head>
<body>
<form ACTION="clogin">
<label> Login </label>
<input type="text" name="usr" size="20"> <br> <br>
<label> Password </label>
<input type="password" name="pwd" size="20"> <br> <br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

cologin1.html

```
<html>
<head>
<title> login Page </title>
<p style= "background:yellow; top:100px; left:250px; position:absolute; ">
</head>
<body>
<form ACTION="clogin1">
<label> Login </label>
<input type="text" name="usr" size="20"> <br> <br>
<label> Password </label>
<input type="password" name="pwd" size="20"> <br> <br>
<input type="submit" value="submit">
</form>
</body>
</html>
```

Addcook.java:

```
import javax.servlet.* ;
import javax.servlet.http.*;
import java.io.*;

public class Addcook extends HttpServlet
{
    String user,pas;

    public void service(HttpServletRequest req,HttpServletResponse res) throws
    ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter out=res.getWriter();
        Cookie c1=new Cookie("usr1","suni");
        Cookie p1=new Cookie("pwd1","ani");
        Cookie c2=new Cookie("usr2","abc");
        Cookie p2=new Cookie("pwd2","123");
        Cookie c3=new Cookie("usr3","def");
        Cookie p3=new Cookie("pwd3","456");
        Cookie c4=new Cookie("usr4","mno");
        Cookie p4=new Cookie("pwd4","789");
        res.addCookie(c1);
        res.addCookie(p1);
        res.addCookie(c2);
        res.addCookie(p2);
        res.addCookie(c3);
        res.addCookie(p3);
        res.addCookie(c4);
        res.addCookie(p4);
        out.println("COOKIE ADDED");
    }
}
```

Clogin.java:

```
import javax.servlet.* ;
import javax.servlet.http.*;
import java.io.*;

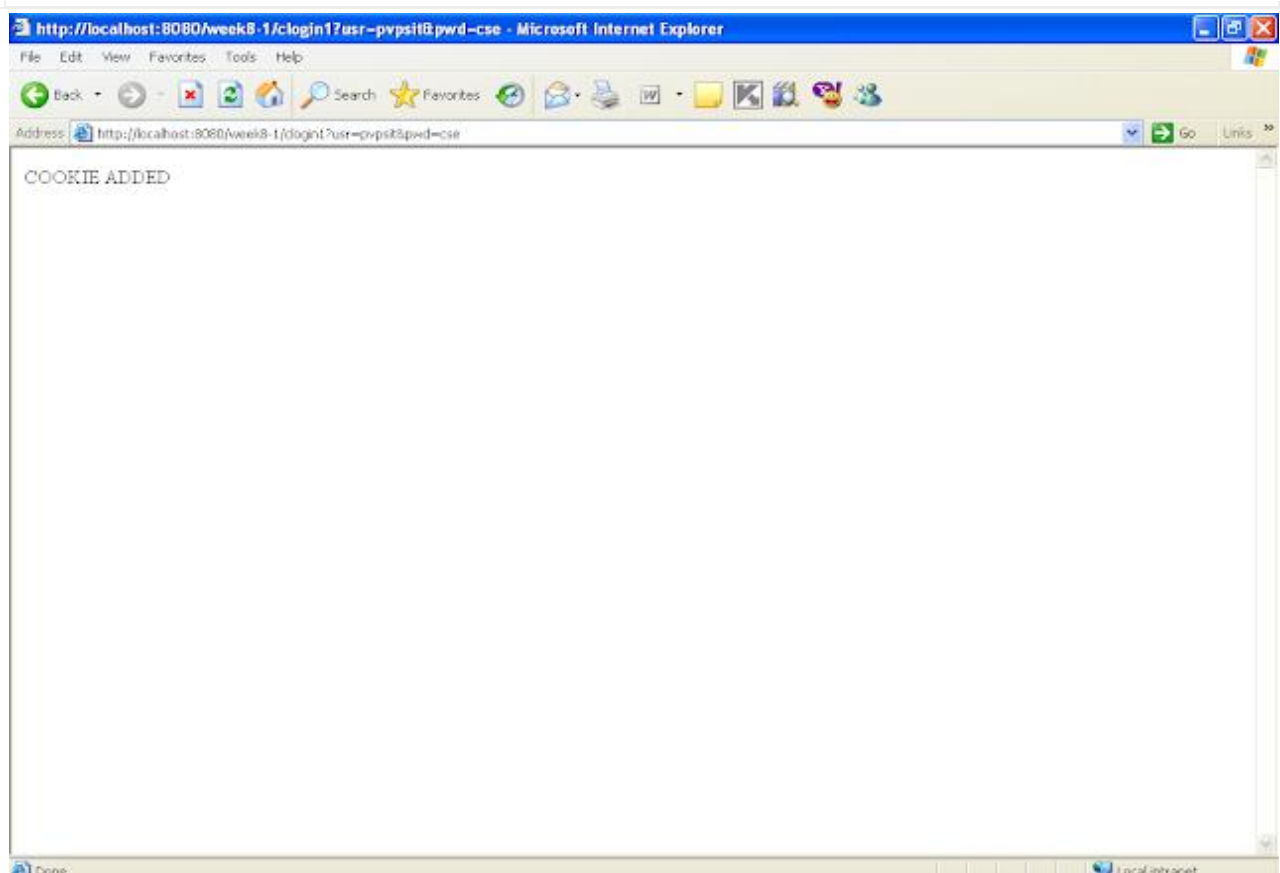
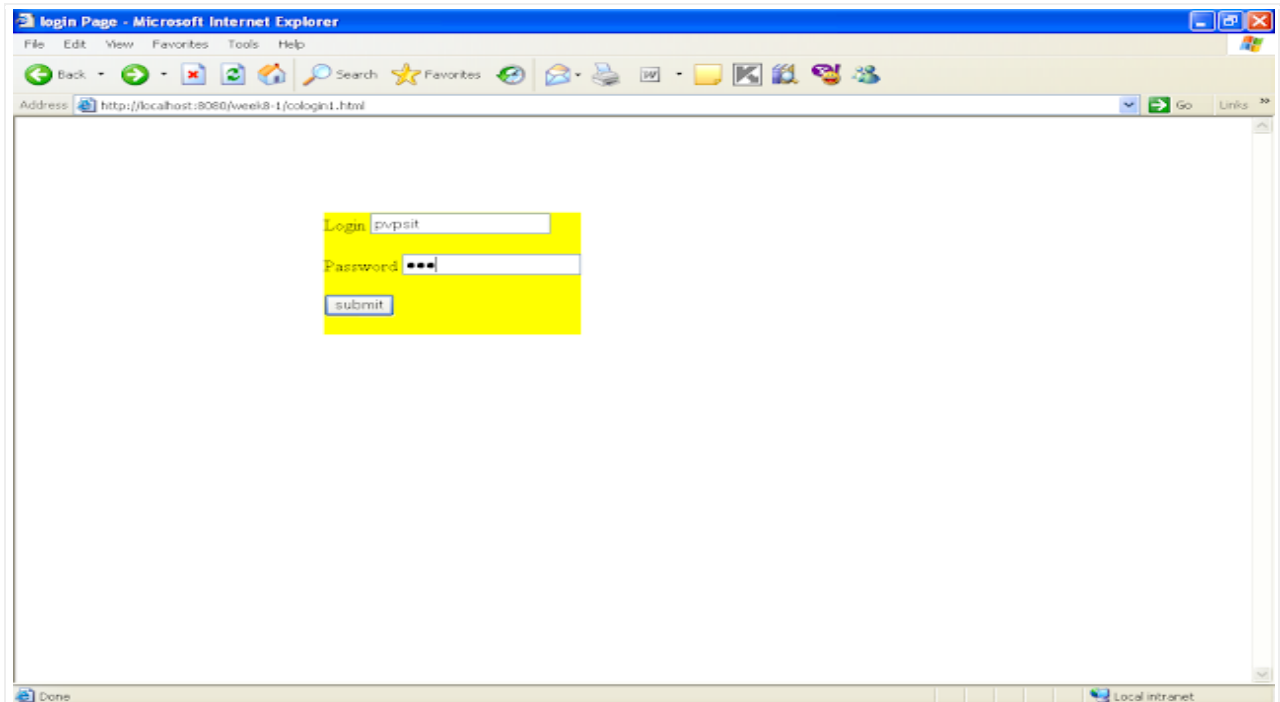
public class Clogin extends HttpServlet
{
    String user,pas;

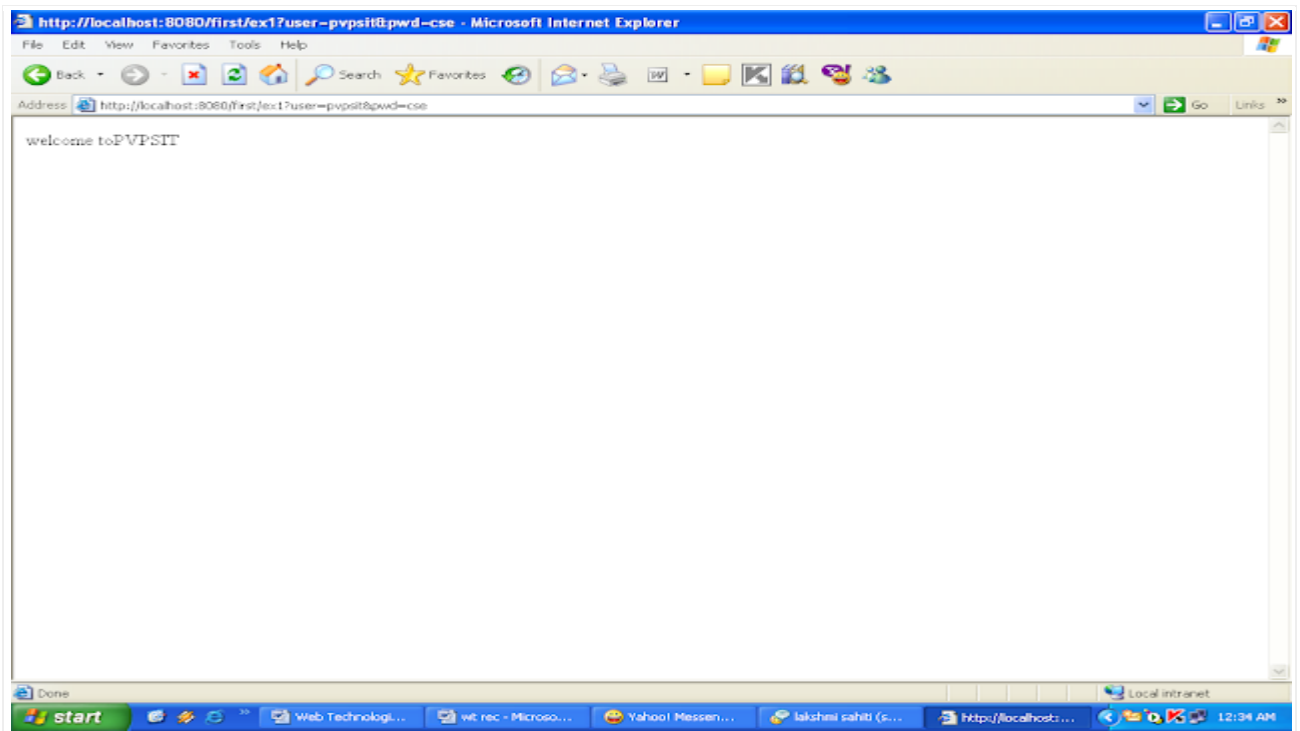
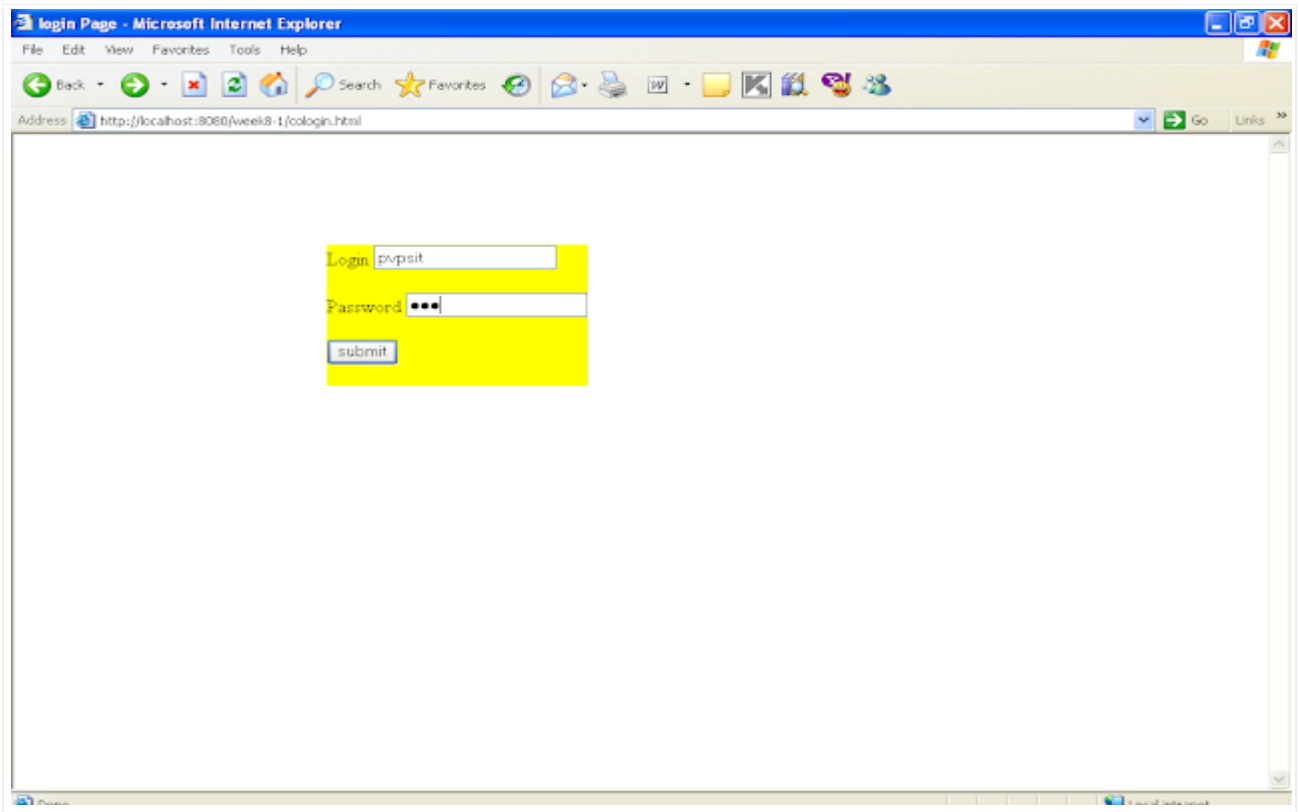
    public void service(HttpServletRequest req,HttpServletResponse res) throws
    ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter out=res.getWriter();
        user=req.getParameter("usr");
        pas=req.getParameter("pwd");
        Cookie[] c=req.getCookies();
        for(int i=0;i<c.length;i++)
        {
            if((c[i].getName().equals("usr1")&&c[i+1].getName().equals("pwd1"))||
            c[i].getName().equals("usr2")
            &&c[i+1].getName().equals("pwd2"))||(c[i].getName().equals("usr3")&&
            c[i+1].getName().equals("pwd3"))||(c[i].getName().equals("usr4")&&
            c[i+1].getName().equals("pwd4")) )
            {
                if((user.equals(c[i].getValue()) && pas.equals(c[i+1].getValue())) )
                {
                    //RequestDispatcher rd=req.getRequestDispatcher("/cart.html");
                    rd.forward(req,res);
                }
                else
                {
                    out.println("YOU ARE NOT AUTHORISED USER ");
                    //res.sendRedirect("/cookdemo/cologin.html");
                }
            }
        }
    }
}
```

Web.xml:

```
<web-app>
<servlet>
<servlet-name>him</servlet-name>
<servlet-class>Clogin</servlet-class>
</servlet>
<servlet>
<servlet-name>him1</servlet-name>
<servlet-class>Addcook</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>him</servlet-name>
<url-pattern>/clogin</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>him1</servlet-name>
<url-pattern>/clogin1</url-pattern>
</servlet-mapping>
</web-app>
```

OUTPUT:





11) Write a servlet that, on entry of a student roll no, displays the full details of that student details (Using student table with roll no, Name, Address, date of birth, course fields).

Database Creation

```
CREATE TABLE "STUDENT"  
    ("ROLLNO" NUMBER, "NAME" VARCHAR2 (40),  
    "ADDRESS" VARCHAR2 (60), "DOB" DATE,  
    "COURSE" VARCHAR2 (40),  
    CONSTRAINT PRIMARY KEY ("ROLLNO")  
    )
```

index.html

```
<html>  
<body>  
<form action="servlet/Search">  
    Enter your Rollno:<input type="text" name="roll"/><br/>  
<input type="submit" value="search"/>  
</form>  
</body>  
</html>
```

Search.java

```
import java.io.*;  
import java.sql.*;  
import javax.servlet.ServletException;  
import javax.servlet.http.*;  
  
public class Search extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
        response.setContentType("text/html");  
        PrintWriter out = response.getWriter();  
  
        String rollno=request.getParameter("roll");  
        int roll=Integer.valueOf(rollno);  
  
        try{  
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            Connection con=DriverManager.getConnection(  
                "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");  
  
            PreparedStatement ps=con.prepareStatement("select * from student where rollno=?");  
            ps.setInt(1,roll);
```

```

out.print("<table width=50% border=1>");
out.print("<caption>Student Details </caption>");
ResultSet rs=ps.executeQuery();

/* Printing column names */
ResultSetMetaData rsmd=rs.getMetaData();
int total=rsmd.getColumnCount();
out.print("<tr>");
for(int i=1;i<=total;i++)
{
out.print("<th>" +rsmd.getColumnName(i)+"</th>");
}
out.print("</tr>");

/* Printing result */
while(rs.next())
{
out.print("<tr><td>" +rs.getInt(1)+"</td><td>" +rs.getString(2)+"
</td><td>" +rs.getString(3)+"</td><td>" +rs.getDate(4)+"</td>
<td>" +rs.getString(5)+"</td></tr>");

}
out.print("</table>");

} catch (Exception e2) {e2.printStackTrace();}

finally{out.close();}
}
}

```

web.xml file

```

<web-app>

    <servlet>
        <servlet-name>Search</servlet-name>
        <servlet-class>Search</servlet-class>
    </servlet>

    <servlet-mapping>
        <servlet-name>Search</servlet-name>
        <url-pattern>/servlet/Search</url-pattern>
    </servlet-mapping>

</web-app>

```

Output:

12) Write JSP program to register a student using registration form using student table.

Registration.jsp

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

    <title>User Data</title>

</head>

<style>

    div.ex {

        text-align: right;

        width: 300px;

        padding: 10px;

        border: 5px solid grey;

        margin: 0px

    }

</style>

<body>

<h1>Registration Form</h1>

<div class="ex">

    <form action="RegistrationController" method="post">

        <table style="width: 50%">

            <tr>

                <td>Full Name</td>

                <td><input type="text" name="fullname"/></td>

            </tr>

            <tr>

                <td>Username</td>

                <td><input type="text" name="userName"/></td>

            </tr>

            <tr>

                <td>Password</td>
```

```

        <td><input type="password" name="pass"/></td>
    </tr>
    <tr>
        <td>Address</td>
        <td><input type="text" name="address"/></td>
    </tr>
    <tr>
        <td>Age</td>
        <td><input type="text" name="age"/></td>
    </tr>
    <tr>
        <td>Qualification</td>
        <td><input type="text" name="qual"/></td>
    </tr>
    <tr>
        <td>Percentage</td>
        <td><input type="text" name="percent"/></td>
    </tr>
    <tr>
        <td>Year Passed</td>
        <td><input type="text" name="yop"/></td>
    </tr>
</table>
<input type="submit" value="register"/>
</form>
<br>
create a student table in test database before registering this form
<br> Syntax : <br>
<i>create table student(name varchar(100), userName varchar(100), pass varchar(100), addr
varchar(100), age int,
        qual varchar(100), percent varchar(100), year varchar(100));</i>
</div>
</body>
</html>

```

RegistrationController.java

```
package com.candid;

import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/RegistrationController")

public class RegistrationController extends HttpServlet {

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {

        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("fullname");
        String userName = request.getParameter("userName");
        String pass = request.getParameter("pass");
        String addr = request.getParameter("address");
        String age = request.getParameter("age");
        String qual = request.getParameter("qual");
        String percent = request.getParameter("percent");
        String year = request.getParameter("yop");

        // validate given input
        if (name.isEmpty() || addr.isEmpty() || age.isEmpty() || qual.isEmpty() || percent.isEmpty() ||
year.isEmpty()) {

            RequestDispatcher rd = request.getRequestDispatcher("registration.jsp");
            out.println("<font color=red>Please fill all the fields</font>");
```

```

        rd.include(request, response);
    } else {
        // inserting data into mysql(mariadb) database
        // create a test database and student table before running this to create table
        //create table student(name varchar(100), userName varchar(100), pass varchar(100), addr
varchar(100), age int, qual varchar(100), percent varchar(100), year varchar(100));
        try {
            Class.forName("org.mariadb.jdbc.Driver");
            // loads mysql(mariadb) driver

            Connection con = DriverManager.getConnection("jdbc:mariadb://localhost:3306/test",
"root", "root");

            String query = "insert into student values(?,?,?,?,?,?,?,?)";
            PreparedStatement ps = con.prepareStatement(query); // generates sql query
            ps.setString(1, name);
            ps.setString(2, userName);
            ps.setString(3, pass);
            ps.setString(4, addr);
            ps.setInt(5, Integer.parseInt(age));
            ps.setString(6, qual);
            ps.setString(7, percent);
            ps.setString(8, year);
            ps.executeUpdate(); // execute it on test database
            System.out.println("successfully inserted");
            ps.close();
            con.close();
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
        }
        RequestDispatcher rd = request.getRequestDispatcher("home.jsp");
        rd.forward(request, response);
    }
}
}

```


Success page (home.jsp)

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-
8859-1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

    <title>Display</title>

    <style>

        table#nat {

            width: 50%;

            background-color: #c48ec5;

        }

    </style>

</head>

<body>

<%

    String name = request.getParameter("fullName");
    String userName = request.getParameter("userName");
    String pass = request.getParameter("pass");
    String addr = request.getParameter("address");
    String age = request.getParameter("age");
    String qual = request.getParameter("qual");
    String percent = request.getParameter("percent");
    String year = request.getParameter("yop");

%>

<table id="nat">

    <tr>

        <td>Full Name</td>

        <td><%= name %>

        </td>

    </tr>

    <tr>

        <td>User Name</td>
```

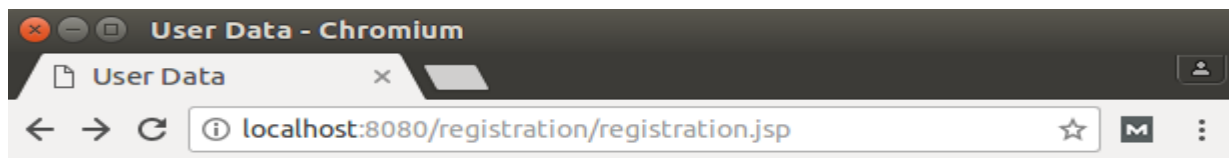
```

        <td><%= userName %>
    </td>
</tr>
<tr>
    <td>Address</td>
    <td><%= addr %>
    </td>
</tr>
<tr>
    <td>Age</td>
    <td><%= age %>
    </td>
</tr>
<tr>
    <td>Qualification</td>
    <td><%= qual %>
    </td>
</tr>
<tr>
    <td>Percentage</td>
    <td><%= percent %>
    </td>
</tr>
<tr>
    <td>Year of Passout</td>
    <td><%= year %>
    </td>
</tr>
</table>
<br>
use " <i> select * from student; </i> " in mysql(mariadb) client to verify it.
</body>
</html>

```

Output:

Screenshot 1

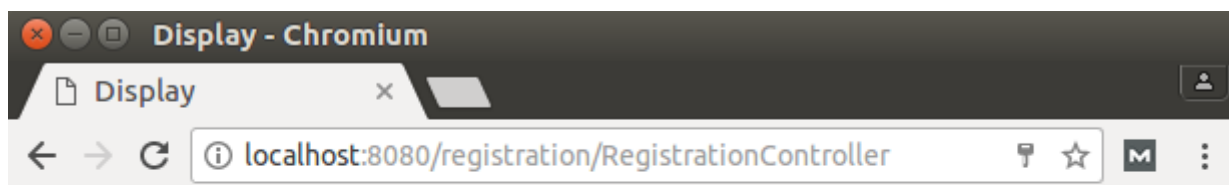


Registration Form

Full Name	Candidjava
Username	candidjava
Password	*****
Address	Chennai
Age	23
Qualification	ME
Percentage	70
Year Passed	2010

create a student table in test database before registering this form
Syntax :
*create table student(name varchar(100), userName varchar(100), pass varchar(100),
addr varchar(100), age int, qual varchar(100), percent varchar(100), year
varchar(100));*

Screenshot 2



Full Name	Candidjava
User Name	candidjava
Address	Chennai
Age	23
Qualification	ME
Percentage	70
Year of Passout	2010

use " *select * from student;* " in mysql client to verify it.

13) Write JSP program for authenticating user by his password using login form.

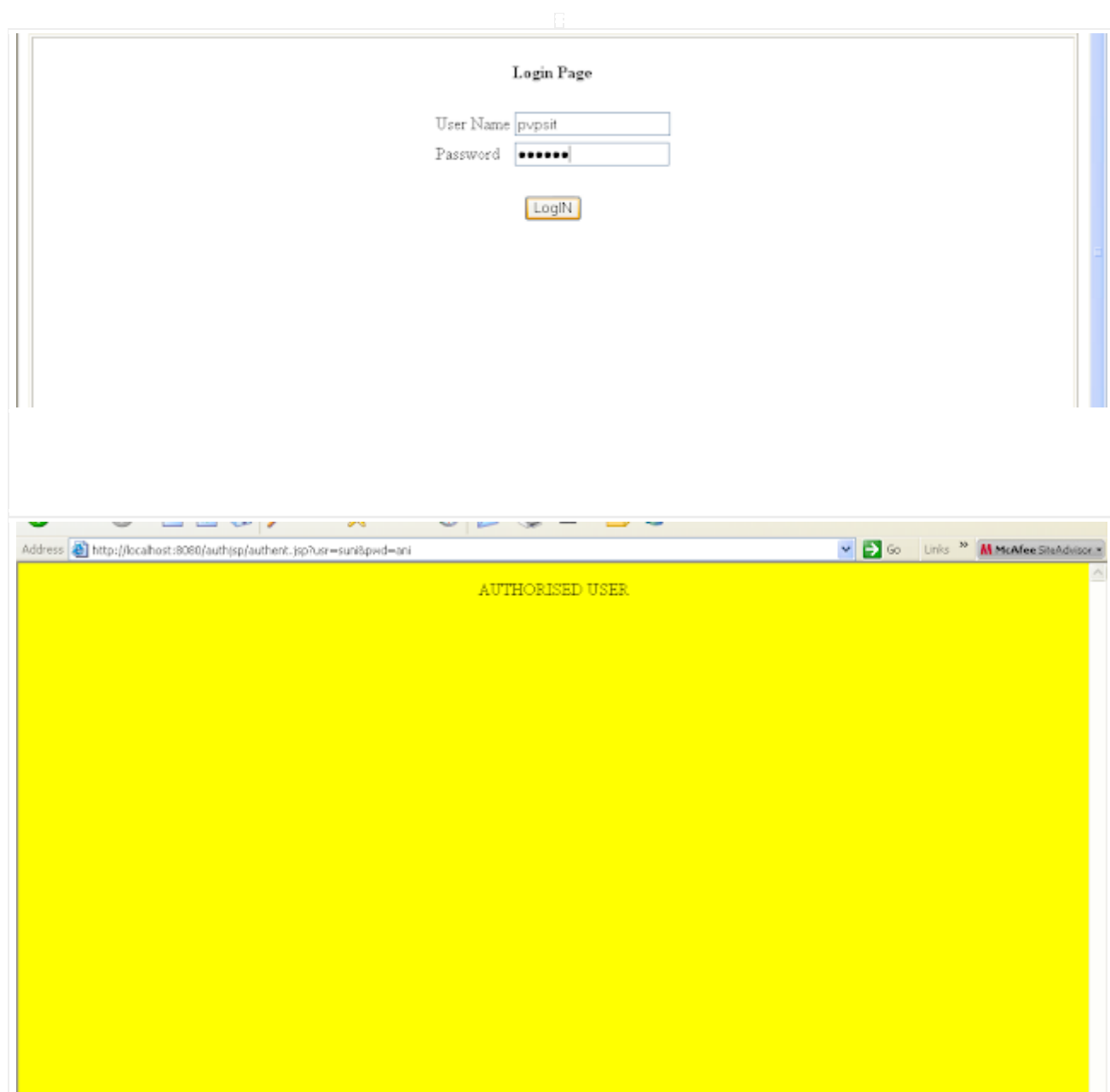
Login.html:

```
<html>
<body>
<table border="1" width="100%" height="100%">
<tr>
    <td valign="top" align="center"><br/>
        <form action="auth.jsp"><table>
            <tr>
                <td colspan="2" align="center"><b>Login Page</b></td>
            </tr>
            <tr>
                <td colspan="2" align="center"><b>&nbsp;</b></td>
            </tr>
            <tr>
                <td>User Name</td>
                <td><input type="text" name="user"/></td>
            </tr>
            <tr>
                <td>Password</td>
                <td><input type="password" name="pwd"/></td>
            </tr>
            <tr>
                <td>&nbsp;</td>
                <td>&nbsp;</td>
            </tr>
            <tr>
                <td colspan="2" align="center"><input type="submit" value="LogIN"/></td>
            </tr>
        </table>
    </form>
    </td>
</tr>
</table>
</body>
</html>
```

Auth.jsp:

```
<% @page import="java.sql.*;"%>
<html>  <head>
<title>
This is simple data base example in JSP</title>
</title>
</head>
<body bgcolor="yellow">
<% !String uname,pwd;%>
<%
uname=request.getParameter("user");
pwd=request.getParameter("pwd");
try
{
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con=DriverManager.getConnection("jdbc:oracle:thin","scott","tiger");
    Statement st=con.createStatement();
    ResultSet rs=st.executeQuery("select name,password from personal where
name='"+uname+"' and password='"+pwd+"'");
    if(rs.next())
    {
        out.println("Authorized person");
    }
    else
    {
        out.println("UnAuthorized person");
    }
    con.close();
}
catch(Exception e){out.println(""+e);}
%>
</body>
</html>
```

OUTPUT:



14) Create table to store the details of book(book name, price, quantity, amount) and extract data from table and display all books using JSP and JDBC.

PROGRAM:

Retrieve.java:

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.io.*;
import java.util.*;

public class Retrieve extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse res) throws
    ServletException,IOException
    {
        res.setContentType("text/html");
        PrintWriter out=res.getWriter();
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection
            con=DriverManager.getConnection("jdbc:oracle:thin:@195.100.101.158:1521:cclab","scott","tiger
            ");
            Statement s=con.createStatement();
            ResultSet r=s.executeQuery("select * from cart");
            out.println("<center> <table border=1>");
            out.println("<thead> <th> Book name </th> <th> Price </th> <th> Quantity </th> <th> Amount
            </th> </thead>");
            while(r.next())
            {
                out.println("<tr> <td> "+r.getString(1)+"</td> ");
                out.println("<td> "+r.getString(2)+"</td> ");
                out.println("<td> "+r.getInt(3)+"</td> ");
                out.println("<td> "+r.getString(4)+"</td> </tr>");
            }
            out.println("</table></center>");
            con.close();
        }
    }
}
```

```

}
catch(SQLException sq)
{
out.println("sql exception"+sq);
}

catch(ClassNotFoundException cl)
{
out.println("class not found"+cl);
}
}
}

```

web.xml:

```

<web-app>
<servlet>
<servlet-name>set</servlet-name>
<servlet-class>Cartenter</servlet-class>
</servlet>
<servlet>
<servlet-name>display</servlet-name>
<servlet-class>Retrieve</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>set</servlet-name>
<url-pattern>/enterdata</url-pattern>
</servlet-mapping>
<servlet-mapping>
<servlet-name>display</servlet-name>
<url-pattern>/display1</url-pattern>
</servlet-mapping>
</web-app>

```


CREATE THE TABLE AND INSERT VALUES INTO THE TABLE:

```
Oracle SQL*Plus
File Edit Search Options Help

SQL*Plus: Release 8.1.7.0.0 - Production on Mon Oct 13 14:12:17 2008

(c) Copyright 2008 Oracle Corporation. All rights reserved.

Connected to:
Oracle8i Enterprise Edition Release 8.1.7.0.0 - Production
With the Partitioning option
JServer Release 8.1.7.0.0 - Production

SQL> create table cart(name varchar2(20),price varchar2(4),quantity number,amount varchar2(4));
create table cart(name varchar2(20),price varchar2(4),quantity number,amount varchar2(4))
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

SQL> select * from cart;

no rows selected

SQL> create table cart_page(name varchar2(20),price varchar2(4),quantity number,amount varchar2(4));

Table created.

SQL> insert into cart_page values('xml book','$20',2,'$40');

1 row created.

SQL> commit;

Commit complete.

SQL> |
```

OUTPUT:

http://localhost:8080/week-11/display1 - Microsoft Internet Explorer

Address: http://localhost:8080/week-11/display1

Book name	Price	Quantity	Amount
Web Programming	\$35.5	2	\$70
XML Bible	\$30	1	\$30