## INDEX

<table>
<thead>
<tr>
<th>S. No</th>
<th>Contents</th>
<th>Page. No</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lab Objective</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Introduction About Lab</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Guidelines to Students</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>List of Lab Exercises</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Syllabus Programs (JNTU)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Solutions for JNTU Programs</td>
<td>9 - 74</td>
</tr>
</tbody>
</table>
LAB OBJECTIVE

- Overview Object Web Technologies
- Basic Concept of Web Technologies
- Advance Programming

1) Introduction to object oriented programming concepts- java as an object oriented programming language. Introduction to java application and applets-control structures-methods-arrays.

2) Object based and object oriented programming creating packages-using overloaded constructors-static class variables-data abstraction and information hiding-relation between super class objects and subclass objects composition verses inheritance-polymorphism- dynamic method binding abstract super classes and concrete super classes-inheriting interface-use of inner classes and wrapper classes-String to kenizer and String Suffer classes.


4) Exception handling and multithreading in object oriented programming- When exception handling should be used-java exception handling – exceptions and inheritance-multithreading in java-thread synchronization-daemon threads Runnable interface- Files and streams in java

5) Network and Database handling through object oriented programming –using JOSC –processing queries-overview of servlet –introduction to networking establishing a simple server and a client – introduction to RMI – implementing the remote interface.
INTRODUCTION ABOUT LAB

There are 66 systems (Compaq Presario) installed in this Lab. Their configurations are as follows:

Processor : DUAL CORE 1.67 GHz
RAM : 256 MB
Hard Disk : 40 GB
Mouse : Optical Mouse
Network Interface card : Present

Software

➢ All systems are configured in DUAL BOOT mode i.e., Students can boot from Windows XP or Linux as per their lab requirement.

This is very useful for students because they are familiar with different Operating Systems so that they can execute their programs in different programming environments.

➢ Each student has a separate login for database access

Oracle 9i client version is installed in all systems. On the server, account for each student has been created.

This is very useful because students can save their work (scenarios’, pl/sql programs, data related projects, etc) in their own accounts. Each student work is safe and secure from other students.

➢ Latest Technologies like DOTNET and J2EE are installed in some systems. Before submitting their final project, they can start doing mini project from 2nd year onwards.

➢ MASM (Macro Assembler) is installed in all the systems

Students can execute their assembly language programs using MASM. MASM is very useful students because when they execute their programs they can see contents of Processor Registers and how each instruction is being executed in the CPU.
- Rational Rose Software is installed in some systems. Using this software, students can depict UML diagrams of their projects.

- Softwares installed: C, C++, JDK1.5, MASM, OFFICE-XP, J2EE and DOTNET, Rational Rose.

- Systems are provided for students in the 1:1 ratio.

- Systems are assigned numbers and same system is allotted for students when they do the lab.
Guidelines to Students

- Equipment in the lab for the use of student community. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.

- Students are required to carry their observation / programs book with completed exercises while entering the lab.

- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab notice board.

- Lab can be used in free time / lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.

- Lab records need to be submitted on or before date of submission.

- Students are not supposed to use floppy disks

- Use of computer network is encouraged.
### Web Technologies Syllabus Programs (JNTU)

<table>
<thead>
<tr>
<th>S.No</th>
<th>Programs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2</td>
<td>Develop static pages (using only HTML) of an online Book store. The pages should resemble: <a href="http://www.amazon.com">www.amazon.com</a>. The website should consist the following pages: Home page, Registration and user Login, User profile page, Books catalog, Shopping cart, Payment By credit card, order confirmation.</td>
</tr>
<tr>
<td>3, 4</td>
<td>Validate the registration, user login, user profile and payment by credit card pages using JavaScript.</td>
</tr>
<tr>
<td>5</td>
<td>Write an XML file which will display the Book information which includes the following: 1) Title of the book 2) Author Name 3) ISBN number 4) Publisher name 5) Edition 6) Price. Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows. The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.</td>
</tr>
<tr>
<td>6</td>
<td>VISUAL BEANS: Create a simple visual bean with an area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click.</td>
</tr>
<tr>
<td>7</td>
<td>1) Install TOMCAT web server. While installation assign port number 8080. Make sure that these ports are available i.e., no other process is using this port. 2) Access the above developed static web pages for books web site, using these servers by Putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the urls: <a href="http://localhost:8080/rama/books.html">http://localhost:8080/rama/books.html</a></td>
</tr>
</tbody>
</table>
| **8** | **User Authentication:**  
Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 And, pwd4 respectively. Write a servlet for doing the following.  
1. Create a Cookie and add these four user id’s and passwords to this Cookie.  
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords ) available in the cookies. If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display “ You are not an authenticated user “. |
| **9** | Install a database(Mysql or Oracle).  
Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).  
Practice 'JDBC' connectivity. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2). |
| **10** | Write a JSP which does the following job:  
Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database. |
WEEK- 1, 2:

AIM:

- Develop static pages (using only HTML) of an online Book store. The pages should resemble: www.amazon.com. The website should consist the following pages.

  - Home page
  - Registration and user Login
  - User profile page
  - Books catalog
  - Shopping cart
  - Payment by credit card Order Confirmation

PROCEDURE:

- **Home page**

  **Main.html:**

  ```html
  <html>
  <head>
  <title>Amazon</title>
  </head>
  <body bgcolor="cyan">
  <center>
  <strong><h1>Welcome to AMAZON</h1></strong>
  <form method="post" action="login.html" target=_blank >
  <h4>for books</h4><input type="submit" value="click here">
  </form>
  </center>
  </body>
  </html>
  ```
• **Registration and user Login**

*Login.html:*

```html
<html>
<head>
<title>login page</title>
</head>
<body bgcolor="cyan">
<center>
<h1> AMAZON </h1>
<br>
<table align="right">
<tr>
<td><strong>user name</strong></td>
<td><input type="text"></td>
<td></td>
</tr>
<tr>
<td><strong>password</strong></td>
<td><input type="password"></td>
<td></td>
</tr>
<tr>
<td>
<form method="post" action="catalog.html"  >
<input type="submit" value="submit" >
</form>
</td>
<td>
<form method="post" action="userpro.html"  >
<input type="submit" value="register" >
<input type="reset" value="reset"></form></td>
</tr>
</table>
</center>
</body>
</html>
```
- **User profile page**

  **Userpro.html:**

  ```html
  <html>
  <head>
  <title>login page</title>
  </head>
  <body bgcolor="cyan">
  <center><strong><h1> AMAZON </h1></strong></center>
  <form method="post" action="catalog.html">
  <table align="left">
  <tr>
  <td><h4>user name</td>
  <td><input type="text"></td>
  </tr>
  <tr>
  <td><h4>password</td>
  <td><input type="password"></td>
  </tr>
  <tr>
  <td><h4>confirm password</td>
  <td><input type="password"></td>
  </tr>
  <tr>
  <td><h4>male &nbsp;&nbsp;</h4><option>
  <input type="radio" name="sex" id="male"></td>
  <td><h4>female &nbsp; &nbsp;</h4><select>
  <input type="radio" name="sex" id="female"></option>
  </tr>
  <tr>
  <td><h4>Address</td>
  <td><textarea name="address" rows=5 cols=19></textarea></td>
  </tr>
  </table>
  <tr>
  <td><input type="submit" value="submit"></td>
  <td><input type="reset" value="reset"></td>
  </tr>
  </form>
  </body>
  </html>
  ```
• **Books catalog**

**Catalog.html:**

```html
<html>
<head>
<title>books catalog</title>
</head>
<body bgcolor="cyan">
<center><h1>AMAZON</h1></center>
<form method="post" action="shopping.html">
<table>
<tr><td><b><h3>frontend books</h3></td><td></td></tr>
<tr><td></td><td><h4>C&Ds</h4></td></tr>
<tr><td></td><td><h4>Ads</h4></td></tr>
<tr><td></td><td><h4>JAVA</h4></td></tr>
<tr><td><b><h3>backend books</h3></td><td></td></tr>
<tr><td></td><td><h4>Oracle</h4></td></tr>
<tr><td></td><td><h4>Ms SQL Server</h4></td></tr>
</table>
</form>
</body>
</html>
```
Shopping cart

Shopping.html:

```html
<html>
<head><title>shopping cart</title></head>
<body bgcolor="cyan">
<center><h1>Shopping Cart</h1></center>
<br><br><br><br><br>
<table align="center">
<tr><td>Text Books</td><td>
<select>
<optgroup label="select the book">
<option value="C&Ds">C&Ds
<option value="Ads">Ads
<option value="Java">Java
<option value="Oracle">Oracle
<option value="Ms SQL Server">Ms SQL Server
<option value="MySql">MySql
</optgroup>
</select>
</td></tr>
<tr><td>Quantity</td><td>
<input type="text" id="q">
</td></tr>
</table>
</body>
</html>
```
Payment by credit card

Payment.html:

```html
<html>
<head><title>payment</title></head>
<body bgcolor="cyan">
<center><h1>Payment By Credit Card</h1></center>
<form method=post action="ordrconform.html">
<br><br><br><br><br>
<table align="center">
<tr>
<td><h4>Total Amount</h4></td>
<td><input type="text">
</td>
</tr>
<tr>
<td><h4>Credit Card Number</td>
<td><input type="text"></td>
</tr>
<tr>
<td><input type="submit" value=OK>
</td>
</tr>
</table>
</form>
</body>
</html>
```
- **Order Conformation**

  **Ordrconform:**
  
  ```html
  <html>
  <head><title>order conformation</title><M/head>
  <body bgcolor="cyan">
  <center>
  <h1><b>BOOK SHOPPING</h1>
  <pre><strong>
  <b>Your order Is Conformed
  </strong></pre>
  <h2><b>THANK YOU</h2>
  </center>
  </body></html>
  ```
OUTPUT:

Main.html:

Login.html:
Catalog.html:

Userpro.htm
Shopping.html:

Payment.html:
Ordrconform.html:
WEEK- 3, 4 :

AIM

➢ Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

PROCEDURE:

• **Home page:**

  **Main.html:**

  ```html
  <html>
  <frameset rows="25%,*">
  <frame src="top.html" name="top" scrolling="no" frameborder="0">
  <frameset cols="25%,75%">
  <frame src="left.html" name="left" scrolling="no" frameborder="0">
  <frame src="right.html" name="right" scrolling="auto" frameborder="0">
  </frameset>
  </frameset>
  </html>
  ```

  **Top.html:**

  ```html
  <html>
  <body bgcolor="pink">
  <br><br>
  <marquee><h1 align="center"><b><u>ONLINE BOOK STORAGE</u></b></h1></marquee>
  <br>
  <marquee><h1 align="center"><b><u>ONLINE BOOK STORAGE</u></b></h1></marquee>
  </body>
  </html>
  ```
**Right.html:**

```html
<html>
<body>
<br><br><br><br>
<h2 align="center">
<p> welcome to online book storage. Press login if you are having id otherwise press registration. </p>
</h2>
</body>   </html>
```

**Left.html:**

```html
<html>
<body bgcolor="pink">
<h3>
<ul>
<li><a href="login.html" target="right"><font color="black">LOGIN</font></a></li>
<br><br>
<li><a href="profile.html" target="right"><font color="black">USER PROFILE</font></a></li>
<br><br>
<li><a href="catalog.html" target="right"><font color="black">BOOKS CATALOG</font></a></li>
<br><br>
<li><a href="scart.html" target="right"><font color="black">SHOPPINGCART</font></a></li>
<br><br>
<li><a href="payment.html" target="right"><font color="black">PAYMENT</font></a></li>
<br><br>
</ul>
</body>   </html>
```
Registration and user Login

Login.html:

```html
<html>
<body bgcolor="pink"><br><br><br>
<script language="javascript">
function validate()
{
    var flag=1;
    if(document.myform.id.value==""||
document.myform.pwd.value=="")
    {
        alert("LoginId and Password must be filled")
        flag=0;
    }
    if(flag==1)
    {
        alert("VALID INPUT");
        window.open("catalog.html","right");
    }
    else
    {
        alert("INVALID INPUT");
        //document.myform.focus();
    }
}
</script>
<form name="myform">
    LOGIN ID:<input type="text" name="id"><br>
    PASSWORD:<input type="password" name="pwd"><br><br>
    <input type="button" value="ok" onClick="validate()">&nbsp;&nbsp;&nbsp;&nbsp;
    <input type="reset" value="clear" >
</form>
</body>
</html>
```
- **User profile page**

**Profile.html:**

```html
<html>
<body bgcolor="pink"><br><br>
<script type="text/javascript">
function validate()
{
var flag=1;
if(document.myform.name.value==""||
document.myform.addr.value==""||
document.myform.phno.value==""||
document.myform.id.value==""||
document.myform.pwd.value=="")
{
    alert("Enter all the details");
    flag=0;
}
var str=document.myform.phno.value;
var x=new RegExp("\d","g");
if(!((str.match(x)))
{
    if(!str.length==10))
    flag=0;
}
var str1=document.myform.id.value;
var x1=new RegExp("^[A-Z][a-zA-Z]+S","g");
if(!((str1.match(x1)))
{
    flag=0;
    alert("Invalid UserID");
}
var str1=document.myform.pwd.value;
var x1=new RegExp("^[A-Z][a-zA-Z]+S","g");
if(!((str1.match(x1)))
{
    flag=0;
    alert("Invalid password");
}
if(flag==1)
{
    alert("VALID INPUT");
    window.self.location.href="login.html";
}
</script>
</body>
</html>
```
else
{
    alert("INVALID INPUT");
    document.myform.focus();
}
</script>
<form name="myform">
<div align="center"><pre>
NAME :<input type="text" name="name"><br>
ADDRESS :<input type="text" name="addr"><br>
CONTACT NUMBER:<input type="text" name="phno"><br>
LOGINID :<input type="text" name="id"><br>
PASSWORD :<input type="password" name="pwd"></pre><br><br>
</div>
<br><br>
<div align="center">
<input type="button" value="ok" onClick="validate()">&nbsp;&nbsp;
<input type="reset" value="clear">
</form>

- **Books catalog:**

Scart.html:

```html
<html>
<body bgcolor="pink"><br><br>
<script language="javascript">
    function validate()
    {
        var flag=1;
        if(document.myform.title.value=="")
        {
            flag=0;
        }

        str=document.myform.title.value;

        if(str=="c"||str=="C")
        {
            document.myform.t1.value="C";
            document.myform.t2.value=444;
        }
        else if(str=="jsp"||str=="JSP")
```
```javascript
if(flag==1)
{
    alert("VALID INPUT");
}
else
{
    alert("INVALID INPUT");
    document.myform.focus();
}
</script>
<form name="myform" action="payment.html" target="right">
    <div align="center">
        BOOK TITLE :<input type="text" name="title">
    </div>
    Book Title:  <input type="text" name="t1" disabled>
    Book Cost:  <input type="text" name="t2" disabled>
    <input type="submit" value="ok" onClick="validate()"><br>
    <input type="reset" value="clear">
    <input type="submit" value="Purchase">
</form>
</body>
</html>
```
• **Shopping cart:**

**Catalog.html:**

```html
<html>
<body bgcolor="pink">
<script language="javascript">
function validate()
{
    var flag=1;
    if(document.myform.id.value=="" ||
       document.myform.title.value=="" ||
       document.myform.no.value=="" ||
       document.myform.cost.value=="")
    {
        flag=0;
    }

    str=document.myform.title.value;
    var str1=document.myform.cost.value;
    if(!(str=="c" && str1==444) || (str=="jsp" && str1==555))
    {
        flag=0;
    }

    if(flag==1)
    {
        alert("VALID INPUT");
    }
    else
    {
        alert("INVALID INPUT");
        document.myform.focus();
    }
}
</script>
<form name="myform" action="scart.html" target="right">
<div align="center">
<pre>
LOGIN ID           :<input type="text" name="id"><br>
TITLE              :<input type="text" name="title"><br>
NO.OF BOOKS       :<input type="text" name="no"><br>
COST OF BOOK      :<input type="text" name="cost"><br>
</pre>
</div>
</form>
</body>
</html>
```
• Payment by credit card

Payment.html:

```html
<html>
<body bgcolor="pink">

<script language="javascript">
function validate()
{
  var flag=1;
  if(document.myform.id.value==""|| document.myform.pwd.value==""||
    document.myform.amount.value==""|| document.myform.num.value=="")
  {
    flag=0;
  }
  var str=document.myform.amount.value;
  var x=new RegExp("\d","g");
  if(!(str.match(x)))
  {
    flag=0;
  }
  var str1=document.myform.num.value;
  var x1=new RegExp("\d","g");
  if(!(str1.match(x1)))
  {
    flag=0;
  }
  if(flag==1)
  {
    alert("VALID INPUT");
    window.self.location.href="order.html";
  }
}
</script>
</body>
</html>
```
alert("INVALID INPUT");
document.myform.focus();
}
</script>
<form name="myform">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id"><br>
PASSWORD  :<input type="password" name="pwd"><br>
AMOUNT   :<input type="text" name="amount"><br>
CREDITCARDNUMBER :<input type="PASSWORD" name="num"><br></div>
<br><br><br>
<div align="center">
<input type="button" value="ok" onClick="validate()">&nbsp;&nbsp;
<input type="reset" value="clear" >
</form>
</div>
OUTPUT:

Main.html

welcome to online book storage. Press login if you are having id otherwise press registration.

Login.html:

Welcome to online book storage. Enter your login ID and password.
Catalog.html:

ONLINE BOOK STORAGE

- LOGIN
- USER PROFILE
- BOOKS CATALOG
- SHOPPINGCART
- PAYMENT

Scart.html:

ONLINE BOOK STORAGE

- LOGIN
- USER PROFILE
- BOOKS CATALOG
- SHOPPINGCART
- PAYMENT

Prepared By:
S. PHANI KUMAR, IT Dept.
Payment.html:

Order.html
WEEK-5.

AIM: Write an XML file which will display the Book information which includes the following:
1) Title of the book
2) Author Name
3) ISBN number
4) Publisher name
5) Edition
6) Price

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

Display the XML file as follows.

The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.

1. Books.DTD:

```xml
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT ISBN_Number (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT edition (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

2. Th.CSS

```css
.thb
{
background-color:gray;
}
.bg
{
background-color:red;
}
```
3. **Books.XML:**

```xml
<?xml version="1.0"?>
<!DOCTYPE book SYSTEM "books.dtd">
<book>
  <details>
    <title> C</title>
    <author> BalaGuru Swami</author>
    <ISBN_Number>2536</ISBN_Number>
    <publisher>pearson</publisher>
    <edition>2</edition>
    <price>255/-</price>
  </details>
  <details>
    <title> C++</title>
    <author> BalaGuru Swami</author>
    <ISBN_Number>5236</ISBN_Number>
    <publisher>pearson</publisher>
    <edition>2</edition>
    <price>315/-</price>
  </details>
  <details>
    <title> E-Commerce</title>
    <author> Kalakata</author>
    <ISBN_Number>8562</ISBN_Number>
    <publisher>pearson</publisher>
    <edition>5</edition>
    <price>300/-</price>
  </details>
  <details>
    <title> CO</title>
    <author> Marris </author>
    <ISBN_Number>4578</ISBN_Number>
    <publisher>Dream Tech</publisher>
    <edition>5</edition>
    <price>270/-</price>
  </details>
  <details>
    <title> Web Technologies</title>
    <author> Kumar </author>
    <ISBN_Number>5423</ISBN_Number>
    <publisher>Willay</publisher>
    <edition>6</edition>
  </details>
</book>
```
4. **Books.HTML:**

```html
<html>
<head>
<link rel="stylesheet" type="text/css" href="th.css">
</head>
<body>

<script type="text/javascript">
if (window.XMLHttpRequest)
{ // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
}
else
{ // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.open("GET","books.xml",false);
xmlhttp.send();
xmlDoc=xmlhttp.responseXML;

document.write("<table border='1'>");
var x=xmlDoc.getElementsByTagName("details")
document.write("<tr><th class='thb'>TITLE</th><th class='thb'>AUTHOR</th><th class='thb'>ISBN_Number</th><th class='thb'>PUBLISHER</th><th class='thb'>EDITION</th><th class='thb'>PRICE</th></tr>");
for (i=0;i<x.length;i++)
{
  document.write("<tr><td>");
  document.write(x[i].getElementsByTagName("title")[0].childNodes[0].nodeValue);
  document.write("</td><th class='bg'>")
```
document.write(x[i].getElementsByTagName("author")[0].childNodes[0].nodeValue.toUpperCase());
    document.write("<th><td>");

document.write(x[i].getElementsByTagName("ISBN_Number")[0].childNodes[0].nodeValue);
    document.write("</td><td>");

document.write(x[i].getElementsByTagName("publisher")[0].childNodes[0].nodeValue);
    document.write("</td><td>");

document.write(x[i].getElementsByTagName("edition")[0].childNodes[0].nodeValue);
    document.write("</td><td>");

document.write(x[i].getElementsByTagName("price")[0].childNodes[0].nodeValue);
    document.write("</td></tr>");
}
document.write("</table>"¨);</script>
</body>
</html>
## OUTPUT:

<table>
<thead>
<tr>
<th>TITLE</th>
<th>AUTHOR</th>
<th>ISBN_Number</th>
<th>PUBLISHER</th>
<th>EDITION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>BALAGURU SWAMI</td>
<td>2536</td>
<td>Pearson</td>
<td>2</td>
<td>255/-</td>
</tr>
<tr>
<td>C++</td>
<td>BALAGURU SWAMI</td>
<td>5236</td>
<td>Pearson</td>
<td>2</td>
<td>315/-</td>
</tr>
<tr>
<td>E-Commerce</td>
<td>KALAKATA</td>
<td>8562</td>
<td>Pearson</td>
<td>5</td>
<td>300/-</td>
</tr>
<tr>
<td>CO</td>
<td>MARRIS</td>
<td>4578</td>
<td>Dream Tech</td>
<td>5</td>
<td>270/-</td>
</tr>
<tr>
<td>Web Technologies</td>
<td>KUMAR</td>
<td>5423</td>
<td>Willay</td>
<td>6</td>
<td>500/-</td>
</tr>
<tr>
<td>Web Programming</td>
<td>KUMAR</td>
<td>1258</td>
<td>Willay</td>
<td>6</td>
<td>500/-</td>
</tr>
</tbody>
</table>
Week-6:

VISUAL BEANS:
Create a simple visual bean with an area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click.

Process:

Create a New Bean

Here are the steps that you must follow to create a new Bean:
1. Create a directory for the new Bean.
2. Create the Java source file(s).
3. Compile the source file(s).
4. Create a manifest file.
5. Generate a JAR file.
6. Start the BDK.
7. Test.
The following sections discuss each of these steps in detail.

Create a Directory for the New Bean

You need to make a directory for the Bean. To follow along with this example, create colors directory. Then change to that directory.

Create the Source File for the New Bean

The source code for the Colors component is shown in the following listing. It is located in the file Colors.java.

The color of the component is determined by the private Color variable color, and its shape is determined by the private boolean variable rectangular. The constructor defines an anonymous inner class that extends MouseAdapter and overrides its mousePressed( ) method. The change( ) method is invoked in response to mouse presses. The component is initialized to a rectangular shape of 200 by 100 pixels.

The change( ) method is invoked to select a random color and repaint the component. The getRectangular( ) and setRectangular() methods provide access to the one property of this Bean. The change( ) method calls randomColor( ) to choose a color and then calls repaint( ) to make the change visible. Notice that the paint( ) method uses the rectangular and color variables to determine how to present the Bean.
SOURCE CODE:- Colors.java:

```java
import java.awt.*;
import java.awt.event.*;
public class Colors extends Canvas
{
    transient private Color color;
    private boolean rectangular;
    public Colors()
    {
        addMouseListener(new MouseAdapter() {
            public void mousePressed(MouseEvent me) {
                change();
            }
        });
        rectangular = false;
        setSize(200, 100);
        change();
    }
    public boolean getRectangular()
    {
        return rectangular;
    }
    public void setRectangular(boolean flag)
    {
        this.rectangular = flag;
        repaint();
    }
    public void change()
    {
        color = randomColor();
        repaint();
    }
    private Color randomColor()
    {
        int r = (int)(255*Math.random());
        int g = (int)(255*Math.random());
        int b = (int)(255*Math.random());
        return new Color(r, g, b);
    }
    public void paint(Graphics g)
    {
        Dimension d = getSize();
        int h = d.height;
    }
```
int w = d.width;
g.setColor(color);
if(rectangular)
{
g.fillRect(0, 0, w-1, h-1);
}
else
{
g.fillOval(0, 0, w-1, h-1);
}

Compile the Source Code for the New Bean

Compile the source code to create a class file. Type the following:
javac Colors.java.

Create a Manifest File

You must now create a manifest file. Put the source code for your manifest file in the file colors.mft.
It is shown here:

Manifest-Version: 1.0
Name: Colors.class
Java-Bean: True

This file indicates that there is one .class file in the JAR file and that it is a Java Bean. Notice that the Colors.class file in the current directory.

Generate a JAR File

Beans are included in the ToolBox window of the BDK only if they are in JAR files in the directory c:\bdk\jars. These files are generated with the jar utility.
Enter the following:

jar cvfm colors.jar colors.mft *.class

This command creates the file colors.jar.

Start the BDK

Change to the directory c:\bdk\beanbox and type run. This causes the BDK to start. You should see three windows, titled ToolBox, BeanBox, and Properties. The ToolBox window should include an entry labeled “Colors” for your new Bean.
Create an Instance of the Colors Bean

After you complete the preceding steps, create an instance of the Colors Bean in the BeanBox window. Test your new component by pressing the mouse anywhere within its borders. Its color immediately changes. Use the Properties window to change the rectangular property from false to true. Its shape immediately changes.

Screen shot for circle:
Screen shot for triangle:
Week-7:
1) Install TOMCAT web server. While installation assign port number 8080. Make sure that these ports are available i.e., no other process is using this port.
2) Access the above developed static web pages for books web site, using these servers by Putting the web pages developed in week-1 and week-2 in the document root.
Access the pages by using the urls: http://localhost:8080/rama/books.html

1. Install the TOMCAT web server:

Step 1:

Installation of JDK:
Before beginning the process of installing Tomcat on your system, ensure first the availability of JDK on your system program directory. Install it on your system if not already installed (because any version of tomcat requires the Java 1.6 or higher versions) and then set the class path (environment variable) of JDK. To set the JAVA_HOME Variable: you need to specify the location of the java run time environment to support the Tomcat else Tomcat server can not run.

This variable contains the path of JDK installation directory.

set JAVA_HOME=C:\Program Files\Java\jdk1.6

Note: it should not contain the path up to bin folder. Here, we have taken the URL path according to our installation convention.

For Windows OS, go through the following steps:

First, right click on the

My Computer->properties->advance->Environment Variables->New->set the Variable name = JAVA_HOME and variable value = C:\Program Files\Java\jdk1.6

Now click on all the subsequent ok buttons one by one. It will set the JDK path.

Step 2:

For setting the class path variable for JDK, do like this:

Prepared By:
S. PHANI KUMAR, IT Dept.
First, right click on the

**My Computer->properties->advance->Environment Variables->path.**

Now, set bin directory path of JDK in the path variable

**Step 3:**

The process of installing Tomcat 6.0 begins here from now. It takes various steps for installing and configuring the Tomcat 6.0.

For Windows OS, Tomcat comes in two forms: .zip file and .exe file (the Windows installer file). Here we are exploring the installation process by using the .exe file. First unpack the zipped file and simply execute the '.exe' file.

![Apache Tomcat Setup](image)

A Welcome screen shot appears that shows the beginning of installation process. Just click on the 'Next' button to proceed the installation process.

**Steps 4:**

A screen of 'License Agreement' displays.

---

**Prepared By:**

**S. PHANI KUMAR, IT Dept.**
Click on the 'I Agree' button.

**Step 5:**

A screen shot appears asking for the 'installing location'.

Choose the folder in which to install Apache Tomcat.
Choose the default components and click on the 'Next' button.

**Step 6:**

A screen shot of 'Configuration Options' displays on the screen. Choose the location for the Tomcat files as per your convenience. You can also opt the default Location

The port number will be your choice on which you want to run the tomcat server. The port number 8080 is the default port value for tomcat server to proceed the HTTP requests. The user can also change the 'port number' after completing the process of installation; for this, users have to follow the following tips.

Go to the specified location as "**Tomcat 6.0\conf\server.xml**". Within the server.xml file choose "Connector" tag and change the port number.

Now, click on the 'Next' button to further proceed the installation process.

**Step 7:**

A Window of Java Virtual Machine displays on the screen
This window asks for the location of the installed Java Virtual Machine. Browse the location of the JRE folder and click on the Install button. This will install the Apache tomcat at the specified location.

**Step 8:**

A processing window of installing displays on the screen.

To get the information about installer click on the "Show details" button
Step 9:

A screen shot of 'Tomcat Completion' displays on the screen.

![Apache Tomcat Setup]

Click on the 'Finish' button.

Step 10:

A window of Apache Service Manager appears with displaying the running process.

![Apache Service Manager]

Let the running process goes on.
Step 11:

After completing the installation process, the Apache Tomcat Manager appears on the toolbar panel like shown in the below picture.

![Apache Tomcat Manager](image)

Start the Tomcat Server:

1. Start the tomcat server from the bin folder of Tomcat 6.0 directory by double clicking the "tomcat6.exe" file.
2. OR create a shortcut of this .exe file at your desktop.
3. Now Open web browser and type URL http://localhost:8080 in the address bar to test the server
4. To Stop the Tomcat Server: Stop the server by pressing the "Ctrl + c" keys.

The screen of Apache Tomcat software looks like this:
PROCEDURE:

1. First install the tomcat into the system.
2. Then make a sub directory(eg., books) in the `tomcat\webapps`.
3. Under books create WEB-INF directory and also place week1 programs in this books directory only.
4. After this start tomcat by giving the following command at the instll_dir>tomcat>bin
   Catalina.bat run
5. At the I.E(web browser) give the url as http://localhost:8080/ books /main.html
6. Port no 8080 is assigned for the tomcat.

Screen shot:

![Screen shot of online book storage](attachment:image.png)

ONLINE BOOK STORAGE

- **LOGIN**

- **USER PROFILE**

- **BOOKS CATALOG**

- **SHOPPINGCART**

- **PAYMENT**

Welcome to online book storage. Press login if you are having id otherwise press registration.

Prepared By:
S. PHANI KUMAR, IT Dept.
Week-8:
User Authentication:

Assume four users user1, user2, user3 and user4 having the passwords pwd1, pwd2, pwd3 and pwd4 respectively. Write a servelet for doing the following.

1. Create a Cookie and add these four user id’s and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords) available in the cookies.
   If he is a valid user (i.e., user-name and password match) you should welcome him by name(user-name) else you should display “You are not an authenticated user”.

PROCEDURE:
1. First install the tomcat into the system.
2. Then make a subdirectory (e.g., tr) in the tomcat webapps.
3. Under tr create WEB-INF directory and also place the html files in this tr directory only.
4. Next under WEB-INF create two subclasses lib, classes and web.xml.
5. Next place all the class files under the classes and jar files (servlet-api.jar, classes12.jar etc…) under lib subdirectories.
6. After this start tomcat by giving the following command at the install_dir>tomcat>bin
7. Catalina.bat run
8. At the I.E (web browser) give the url as http://localhost:8080//tr/htmlfile or servlet url pattern
9. Portno 8080 is assigned for the tomcat.

Login.html:
<html>
<body bgcolor="pink">
<form action="show" method="get">
<center>User Name: <input type="test" name="name"><br>
Password:<input type="password" name="pass"><br>
<input type="submit" name="b1">
<input type="Button" name="b2" value="Reset">
</center>
</form>
</body>
</html>
Login.java:

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
/** Example using servlet initialization. Here, the message
  * to print and the number of times the message should be
  * repeated is taken from the init parameters.
  */
public class login extends HttpServlet
{
    public void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html");
        String na=request.getParameter("name");
        String pa=request.getParameter("pass");
        PrintWriter out = response.getWriter();
        Cookie nam1=new Cookie("user1","pace");
        Cookie nam2=new Cookie("user2","phani");
        Cookie nam3=new Cookie("user3","cse");
        Cookie nam4=new Cookie("user4","ece");
        Cookie pas1=new Cookie("pwd1","college");
        Cookie pas2=new Cookie("pwd2","kumar");
        Cookie pas3=new Cookie("pwd3","it");
        Cookie pas4=new Cookie("pwd4","eee");
        int flag=0;
        String nam[]=\{nam1.getValue(),nam2.getValue(),nam3.getValue(),nam4.getValue()\};
        String pas[]=\{pas1.getValue(),pas2.getValue(),pas3.getValue(),pas4.getValue()\};
for(int i=0;i<4;i++)
{
    if(nam[i].equals(na)&amp;&amp;pas[i].equals(pa))
    {
        flag=1;
    }
}
if(flag==1)
{
    out.println("<title>The ShowMessage Servlet</title>\n" +"<H1 ALIGN=CENTER>
WELCOME \n" +na.toUpperCase() + "</H1>\n");
    out.println("</BODY></HTML>\n");
}
else
{
    out.println("<title>The ShowMessage Servlet</title>\n");
    out.println("<BODY BGCOLOR="#FDF5E6"\n" +"<H1 ALIGN=CENTER>
User is invalid </H1>\n");
    out.println("</BODY></HTML>\n");
}
Web.xml:

```xml
<web-app>
    <servlet>
        <servlet-name>log</servlet-name>
        <servlet-class>login</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>log</servlet-name>
        <url-pattern>/show</url-pattern>
    </servlet-mapping>
</web-app>
```

Create a directory:

Create a directory “cookies”, in that directory copy login.html file and create a directory a "WEB-INF”. In that WEB-INF directory again create directory “classes” and copy web.xml file.

Compile the servlet:

Compile the servlet file then copy the class file of that servlet into the directory C:\Program Files\Apache Software Foundation\Tomcat 6.0\cookies\WEB-INF\classes.

Open the server:

1. Start tomcat by giving the following command at the instll_dir>tomcat>bin Catalina.bat run
2. At the I.E(web browser) give the url as http://localhost:8080/ cookies /login.html
OUTPUT:

**Login.html:**

![Login.html Image]

**Login.java:**

![Login.java Image]

WELCOME TO CSE
Week-9:

Install a database (Mysql or Oracle).
Create a table which should contain at least the following fields: name, password, email-id, phone number (these should hold the data from the registration form).

Practice 'JDBC' connectivity.
Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

Main.html:
<html>
<body>
<br /><br /><br /><br /><br />
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1>
<br/><br /><br />
<h2 align="center"><pre>
Welcome to online book storage.
Press LOGIN if you are having id otherwise press REGISTRATION
</pre></h2>
<br /><br /><pre>
<div align="center"><a href="login.html">LOGIN</a> <a href="reg.html">REGISTRATION</a></div>
</pre>
</body>
</html>

login.html:
<html>
<body>
<br /><br /><br />
<form name="myform" method="post" action="login">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
PASSWORD :<input type="password" name="pwd" /></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok"/>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
<input type="reset" value="clear" />
</div>
</form>
</body>
</html>
Profile.html
<html>
<body>
<form name="myform" method="post" action="profile">
<div align="center">
LOGIN ID:<input type="text" name="id" />
</div>
<br />
<br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
<input type="reset" value="clear" />
</div>
</form>
</body>
</html>

Userinfo.html
<html>
<head>
<title>User Info Entry Form</title>
</head>
<body bgcolor="white">
<form action="userinfo1.jsp" method="post">
<table>
<tr>
<td>Name:</td>
<td><input type="text" name="userName" >
</td>
</tr>
<tr>
<td>Sex:</td>
<td><input type="text" name="sex" >
(Male or female)
</td>
</tr>
<tr colspan=2>
<input type="submit">
</tr>
</table>
</form>
</body>
</html>
Reg.html
<html>
<body><br /><br />
<form name="myform" method="post" action="reg">
<table align="center">
<tr>
<td>NAME</td>
<td><input type="text" name="name" /></td>
</tr>
<tr>
<td>ADDRESS</td>
<td><input type="text" name="addr" /></td>
</tr>
<tr>
<td>CONTACT NUMBER</td>
<td><input type="text" name="phno" /></td>
</tr>
<tr>
<td>LOGINID</td>
<td><input type="text" name="id" /></td>
</tr>
<tr>
<td>PASSWORD</td>
<td><input type="password" name="pwd" /></td>
</tr>
</table>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
</div>
</form>
</body>
</html>

login.java
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class login extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse resp)
        throws ServletException,IOException
    {
        PrintWriter pw=req.getWriter();
        pw.println("<html><body>");
    
"}
```java
String id = req.getParameter("id");
String pwd = req.getParameter("pwd");
String s1 = "", s2 = "";
try {
    Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con =
        DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:XE","system",
            "tiger");
    Statement stmt = con.createStatement();
    String sqlstmt = "select * from login";
    ResultSet rs = stmt.executeQuery(sqlstmt);
    int flag = 0;
    while (rs.next())
    {
        s1 = rs.getString(4);
        s2 = rs.getString(5);
    }
    if (id.equals(s1) && pwd.equals(s2))
    {
        flag = 1;
    }
    if (flag == 0)
    {
        pw.println("<br><br>SORRY INVALID ID TRY AGAIN ID<br><br>");
        pw.println("<a href="login.html">press LOGIN to RETRY</a>");
    }
    else
    {
        pw.println("<br><br>WELCOME TO " + id.toUpperCase() + "<br><br>");
        pw.println("<h3><ul>
            <li><a href="profile.html"><fontcolor="black"> USER PROFILE</font></a></li>
            <li><a href="catalog.html"><fontcolor="black"> BOOKS CATALOG</font></a></li>
            <li><a href="order.html"><fontcolor="black"> ORDER CONFIRMATION</font></a></li>
        </ul><br><br>");
    }
} catch (Exception e)
{
    resp.sendError(500, e.toString());
}
```
reg.java:
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class reg extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
    {
        PrintWriter pw = resp.getWriter();
        resp.setContentType("text/html");
        pw.println("<html><body>");
        String name = req.getParameter("name");
        String addr = req.getParameter("addr");
        String phno = req.getParameter("phno");
        String id1 = req.getParameter("id");
        String pwd1 = req.getParameter("pwd");
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:XE","system","tiger");
            Statement stmt = con.createStatement();
            String sqlstmt = "select * from login";
            ResultSet rs = stmt.executeQuery(sqlstmt);
            int flag = 0;
            while (rs.next())
            {
                if (id1.equals(rs.getString(4)) && pwd1.equals(rs.getString(5)))
                {
                    flag = 1;
                }
            }
            if (flag == 1)
            {
                pw.println("<br><br>SORRY INVALID ID ALREADY EXISTS
                    TRY AGAIN WITH NEW ID<br><br>");
                pw.println("<a href="reg.html">press REGISTER to RETRY</a>");
            }
            else
            {
                Statement stmt1 = con.createStatement();
                stmt1.executeUpdate("insert into login values 
                    ("+name+","+addr+","+phno+","+id1+","+pwd1+")");
            }
        }
    }
}
Catalog.java:

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

public class catalog extends HttpServlet {
    public void service(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
        PrintWriter pw = resp.getWriter();
        String title = req.getParameter("title");
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection(
                    "jdbc:oracle:thin:@localhost:1522:XE","system","tiger");
            Statement stmt = con.createStatement();
            String sqlstmt = "select * from book where title='"+title+"'";
            ResultSet rs = stmt.executeQuery(sqlstmt);
            int flag = 0;
            while (rs.next()) {
                pw.println("<div align="center">" +
                        "TITLE :" + rs.getString(1) + ":" + <br>
                        "AUTHOR :" + rs.getString(2) + "<br>
                        "VERSION :" + rs.getString(3) + "<br>
                        "PUBLISHER :" + rs.getString(4) + "<br>
                        "COST :" + rs.getString(5) + "<br>";
                flag = 1;
            }
            if (flag == 0) {
                pw.println("<br><br>SORRY INVALID TITLE TRY AGAIN <br><br>" +
                        "YOUR DETAILS ARE ENTERED<br><br>
                        "<a href="login.html">press LOGIN to login</a>" +
                        "</body></html>" +
                        "catch(Exception e) {
                        resp.sendRedirect(500, e.toString());
                        }
                    } }
```

Prepared By:
S. PHANI KUMAR, IT Dept.
pw.println("<a href="/catalog.html">press HERE to RETRY</a>" );
}
pw.println("</body></html>");
}
catch(Exception e)
{
    resp.sendError(500,e.toString());
}
}

profile.java:
import java.sql.*;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class profile extends HttpServlet
{
    public void service(HttpServletRequest req,HttpServletResponse resp)
        throws ServletException,IOException
    {
        PrintWriter pw=resp.getWriter();
        pw.println("<html><body>");
        String id=req.getParameter("id");
        try
        {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:XE","system","tiger");
            Statement stmt=con.createStatement();
            String sqlstmt="select * from login where id='"+id+"';
            ResultSet rs=stmt.executeQuery(sqlstmt);
            int flag=0;
            pw.println("<br><br><br>");
            while(rs.next())
            {
                pw.println("<div align=center>");
                pw.println("NAME :"+rs.getString(1)+"<br>");
                pw.println("ADDRESS :"+rs.getString(2)+"<br>");
                pw.println("PHONE NO :"+rs.getString(3)+"<br>");
                flag=1;
            }
            if(flag==0)
            {
                pw.println("<br><br>SORRY INVALID ID TRY AGAIN ID<br><br>");
                pw.println("<a href="/profile.html">press HERE to RETRY</a>");
            }
            pw.println("</body></html>");
        }
    }
}
catch(Exception e)  
{  
    resp.sendError(500,e.toString());  
}

Order.java:

import java.sql.*;  
import java.io.*;  
import javax.servlet.*;  
import javax.servlet.http.*;  
public class order extends HttpServlet
{
    public void service(HttpServletRequest req, HttpServletResponse resp)
        throws ServletException, IOException
    {
        int count;
        PrintWriter pw=resp.getWriter();
        pw.println("<html><body>);
        String id=req.getParameter("id");
        String pwd=req.getParameter("pwd");
        String title=req.getParameter("title");
        String count1=req.getParameter("no");
        String date=req.getParameter("date");
        String cno=req.getParameter("cno");
        try
        {
            count=Integer.parseInt(count1);
            Class.forName("oracle.jdbc.driver.OracleDriver");  
            Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:XE","system","tiger");
            Statement stmt=con.createStatement();
            String sqlstmt="select * from login";
            ResultSet rs=stmt.executeQuery(sqlstmt);
            int flag =0,amount,x;
            while(rs.next())
            {
                if(id.equals(rs.getString(4))&&pwd.equals(rs.getString(5)))
                {
                    flag=1;
                }
            }
        }
        if(flag==0)
        {
            pw.println("<br><br>SORRY INVALID ID TRY AGAIN ID<br><br>");
            pw.println("<a href="/order.html" press HERE to RETRY</a>";)
        }
        else
        {

        }
Statement stmt2=con.createStatement();
String s="select cost from book where title=\"+title+\";"
ResultSet rs1=stmt2.executeQuery(s);
int flag1=0;
while(rs1.next())
{
    flag1=1;
    x=Integer.parseInt(rs1.getString(1));
    amount=count*x;
    pw.println("<br>AMOUNT:"+amount+"<br><br><br><br>");
    Statement stmt1=con.createStatement();
    stmt1.executeUpdate("insert into details values('"+id+"','"+title+"','"+amount+"','"+cno+"')");
    pw.println("YOUR ORDER has taken<br>");
    if(flag1==0)
    {
        pw.println("SORRY INVALID ID TRY AGAIN ID<br><br>");
        pw.println("<a href="/order.html">press HERE to RETRY</a>");
    }
}
pw.println("</body></html>");
con.close();
catch(Exception e)
{
    resp.sendError(500,e.toString());
}
OUTPUT:

**Main.html:**

![Image of Main.html](image_url)

**ONLINE BOOK STORAGE**

Welcome to online book storage. Press LOGIN if you are having id otherwise press REGISTRATION

![Image of Login and Registration forms](image_url)

**Registration:**

![Image of Registration form](image_url)
Login Page:

![Login Page Screen Shot]

Login Servlet page:

![Login Servlet Page Screen Shot]

WELCOME TO CHINNA

- USER PROFILE
- BOOKS CATALOG
- ORDER CONFIRMATION
Profile page:

Profile Servlet page:
Catalog page:

![Catalog page screenshot](image1)

Catalog Servlet page:

![Catalog Servlet page screenshot](image2)
Order page:

Order Servlet page:

AMOUNT :1000

YOUR ORDER has taken
Week-10:

Write a JSP which does the following job:
Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database.

**Main.html:**

```html
<html>
<body>
<br /><br /><br /><br /><br />
<h1 align="center"><U>ONLINE BOOK STORAGE</U></h1><br /><br /><br />
<h2 align="center"><pre>
Welcome to online book storage.
Press LOGIN if you are having id otherwise press REGISTRATION
</h2></pre></h2>
<br /><br /><pre>
<div align="center"><a href="login.html">LOGIN</a> <a href="reg.html">REGISTRATION</a></div>
</pre>
</body>
</html>
```

**Login.html:**

```html
<html>
<body><br /><br />
<form name="myform" method="post" action="login.jsp">
<div align="center"><pre>
LOGIN ID :<input type="text" name="id" /><br />
PASSWORD :<input type="password" name="pwd" /></pre><br /><br />
</div>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
</div>
</form>
</body>
</html>
```
Reg.html:

<html>
<body><br /><br />
<form name="myform" method="post" action="reg.jsp">
<table align="center">
<tr>  
  <td>NAME</td>  
  <td>:<input type="text" name="name" /></td> 
</tr>
<tr>  
  <td>ADDRESS</td>  
  <td>:<input type="text" name="addr" /></td> 
</tr>
<tr>  
  <td>CONTACT NUMBER</td>  
  <td>:<input type="text" name="phno" /></td> 
</tr>
<tr>  
  <td>LOGINID</td>  
  <td>:<input type="text" name="id" /></td> 
</tr>
<tr>  
  <td>PASSWORD</td>  
  <td>:<input type="password" name="pwd" /></td> 
</tr>
</table>
<br /><br />
<div align="center">
<input type="submit" value="ok" onclick="validate()" />
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<input type="reset" value="clear" />
</div>
</form>
</body>
</html>
Reg.jsp:

```jsp
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<%  
response.setContentType("text/html");
out.println("<html><body>");
String name=request.getParameter("name");
String addr=request.getParameter("addr");
String phno=request.getParameter("phno");
String id1=request.getParameter("id");
String pwd1=request.getParameter("pwd");
int no=Integer.parseInt(phno);
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
  if(id1.equals(rs.getString(1))&&pwd1.equals(rs.getString(2)))
  {
    flag=1;
  }
}
if(flag==1)
{
  out.println("<br><br>SORRY INVALID ID ALREADY EXITS TRY AGAIN WITH NEW ID<br><br>"),
out.println("<a href="reg.html">press REGISTER to RETRY</a>"),
}
else
{
  Statement stmt1=con.createStatement();
  stmt1.executeUpdate("insert into login
values("+name+","+addr+","+phno+","+id1+","+pwd1+");")
out.println("<br><br>YOUR DETAILS ARE ENTERED<br><br>");
out.println("<a href="login.html">press LOGIN to login</a>");
}
out.println("</body></html>");
con.close();
%>
```
Login.jsp:
<%@page import="java.sql.*"%>
<%@page import="java.io.*"%>
<html>
<body>
<% String id=request.getParameter("id");
String pwd=request.getParameter("pwd");
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","tiger");
Statement stmt=con.createStatement();
String sqlstmt="select id,pwd from login";
ResultSet rs=stmt.executeQuery(sqlstmt);
int flag=0;
while(rs.next())
{
    if(id.equals(rs.getString(1))&pwd.equals(rs.getString(2)))
    {
        flag=1;
    }
}
if(flag==0)
{
    out.println("<br><br>SORRY INVALID ID TRY AGAIN ID<br><br>");
    out.println("<a href="login.html">press LOGIN to RETRY</a>";
}
else
{
    out.println("<br><br>VALID LOGIN ID<br><br>");
    out.println("WELCOME <br>To<br>"+id);
}
con.close();
%
</body>
</html>
OUTPUT:

Main.html:

ONLINE BOOK STORAGE

Welcome to online book storage.
Press LOGIN if you are having id otherwise press REGISTRATION

LOGIN
REGISTRATION

Registration page:

NAME
ADDRESS
CONTACT NUMBER
LOGINID
PASSWORD
ok clear
Login page:

VALID LOGIN ID

WELCOME
To

PACE