WEEK-1:

HTML:

```
<!DOCTYPE html>
<html>
    <head>
        <title>ABC</title>
        <style>
            * {
                box-sizing: border-box;
            }
            body{
                padding: 20px;
                background:black;
            }

            .header{

                padding: 10px;
                font-size: 20px;
                text-align: center;
               background: skyblue;
               font-family:Helvetica;
               margin-top: 30px;
            }
            .main{
                padding:10px;
                font-size: 10px;
                background-color: whitesmoke;
            }
            .fakeimg {

            width: 100%;
            padding: 20px;
            }
            .card{
                display: block;
                margin-left: auto;
                margin-right: auto;
                width: 10%;
            }
            .footer{
                padding: 10px;
                background:darkslateblue;
                margin-top: 20px;
            }

        </style>
```

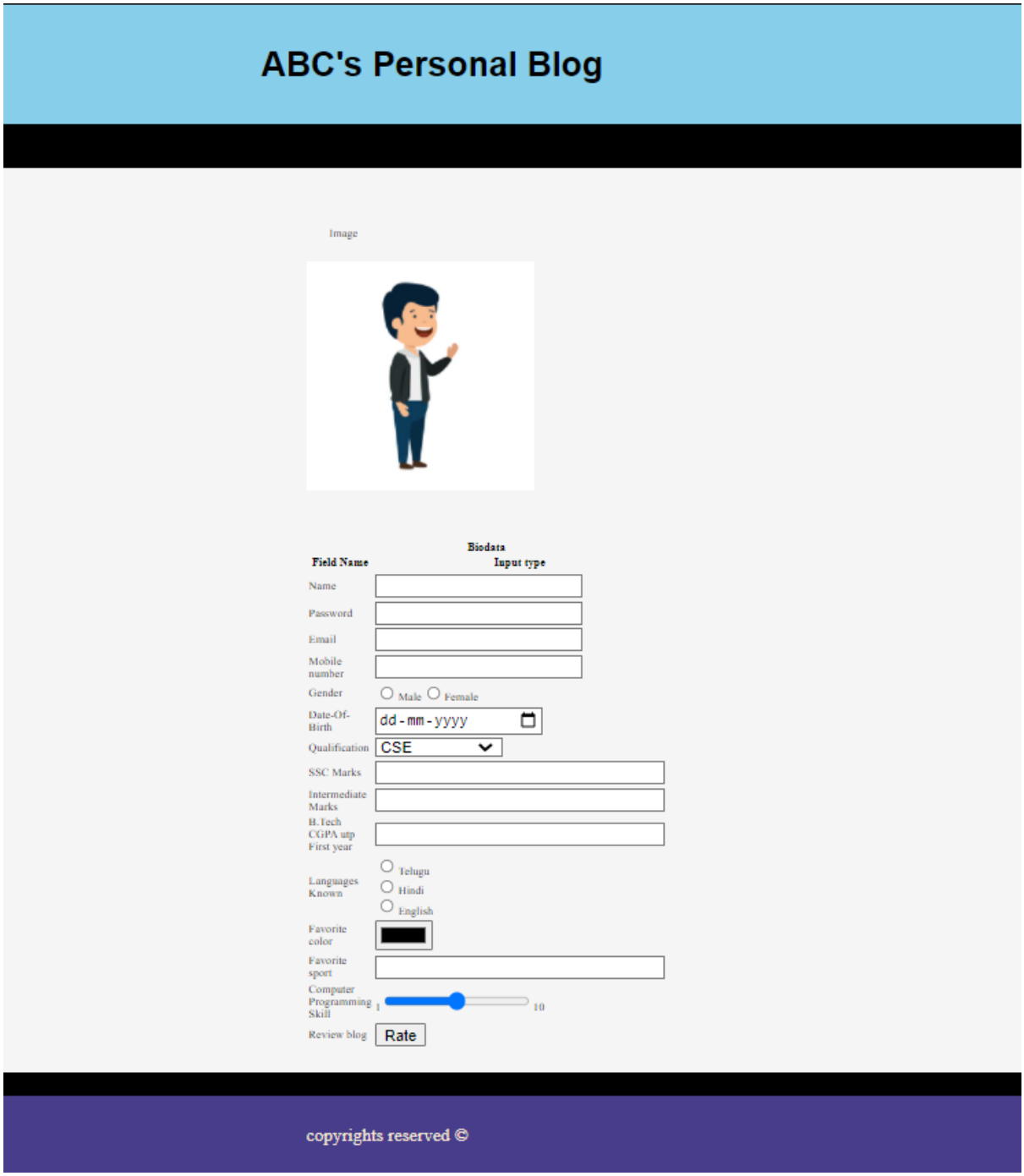
```
    </head>
    <script>

        function validateform(){
        var name=document.myform.name.value;
        var password=document.myform.password.value;
        var email = document.myform.email.value;
        var mobile=document.myform.mobilenumber.value;
        var atposition=email.indexOf("@");
        var dotposition=email.lastIndexOf(".");

        if (name.length<2|| name.length>15){
          alert("Name must be between 2 to 15 length");

          return false;
        }else if(password.length<4 || password.length>8){
          alert("Password must be between 4 to 8 length.");
          return false;
          }
          else if (atposition<1 || dotposition<atposition+2 ||
dotposition+2>=email.length){
            alert("Please enter a valid e-mail address \n
atpostion:"+atposition+"\n dotposition:"+dotposition);
            return false;
        }
        else if (isNaN(mobile)){
            document.getElementById("numloc").innerHTML="Enter Numeric value
only";
            return false;
        }
        else{
            return true;
        }
        }
        </script>
    <body style = "background-color: black;">

        <div class = "header" >
          <h2 style="color:black;">ABC's Personal Blog</h2>
        </div>
        <br>
        <br>

        <div class = "main">
            <br>
            <br>
            <div class = "card">
                <div class = "fakeimg" style = "height: 50px;">Image</div>
```

```
                <img src="blog.webp" width="200" height="200">
                <p>
                    <form name = "myform" method = "post" onsubmit = "return
validateform()">

                        <table>
                            <br>
                            <br>
                            <tr>
                            <th>Field Name</th>
                            <th>Input type</th>
                            </tr>
                            <tr>
                            <caption><b>Biodata</b></caption>
                            <tr>
                            <td>Name</td>
                            <td><input type="text" name ="name" required></td>
                            </tr>
                            <tr>
                            <td>Password</td>
                            <td><input type="password"  name =
"password"  required></td>
                            </tr>
                            <tr>
                            <td>Email</td>
                            <td><input type="text"  name = "email"
required></td>
                            </tr>
                            <tr>
                            <td>Mobile number</td>
                            <td><input type="text"  name = "mobilenumber"
required><span id="numloc"></span></td>
                            </tr>
                            <tr>
                            <td>Gender</td>
                            <td><input type="radio">Male<input
                            type="radio">Female</td>
                            </tr>
                            <tr>
                            <td>Date-Of-Birth</td>
                            <td><input type="date"></td>
                            </tr>
                            <tr>
                            <td>Qualification</td>
                            <td><select>
                            <optgroup label="UG">
                            <option value="CSE">CSE</option>
                            <option value="AI & DS">AI & DS</option>
```

```
                        <option value="MECH">MECH</option>
                        </optgroup>
                        <optgroup label="PG">
                        <option value="MBA">MBA</option>
                        <option value="MASTERS">MASTERS</option>
                        </optgroup>
                        </select></td>
                        </tr>
                        <tr>
                        <td>SSC Marks</td>
                        <td><input type="text" size="30" required></td>
                        </tr>
                        <tr>
                        <td>Intermediate Marks</td>
                        <td><input type="text" size="30" required></td>
                        </tr>
                        <tr>
                        <td>B.Tech CGPA utp First year</td>
                        <td><input type="text" size="30" required></td>
                        </tr>
                        <tr>
                        <td>Languages Known</td>
                        <br>
                        <td><input type="radio" >Telugu<br><input
type="radio">Hindi<br><input type="radio">English</td>
                        </tr>
                        <tr>
                        <td>Favorite color</td>
                        <td><input type="color"></td>
                        </tr>
                        <tr>
                        <td>Favorite sport</td>
                        <td><input type="text" size="30"></td>
                        </tr>
                        <tr>
                        <td>Computer Programming Skill</td>
                        <td>1<input type="range">10</td>
                        </tr>
                        </tr>
                        <tr>
                        <td>Review blog</td>
                        <td><a href="week2.html">
                            <button type="button" >Rate</button>
                        </a></td>

                        </table>
                    </form>
                </p>
```

```
            </div>

        </div>
        <div class="footer">
            <div class="card">
                <p style="color: antiquewhite;">     copyrights reserved ©</p>
            </div>
        </div>
    </body>
</html>
```

Output:



WEEK2:

HTML Forms:

```
<!DOCTYPE html>
<html>
<head>
    <title>Review page</title>
</head>
```

```
<body>

    <form>
    <b>NAME <input type="text"> </b>
    <br>
    <br>
    <p>DATE OF BIRTH</p>
    <label for="cars">MONTH</label>
    <select name="cars" id="cars">
    <option value="Jan">Jan</option>
    <option value="Feb">Feb</option>
    <option value="Mar">Mar</option>
    <option value="Apr">Apr</option>
    <option value="May">May</option>
    <option value="Jun">Jun</option>
    <option value="Jul">Jul</option>
    <option value="Aug">Aug</option>
    <option value="Sep">Sep</option>
    <option value="Oct">Oct</option>
    <option value="Nov">Nov</option>
    <option value="Dec">Dec</option>
    </select>
    <label for="DATE">DATE</label>
    <select name="cars" id="cars">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
    <option value="6">6</option>
    <option value="7">7</option>
    <option value="8">8</option>
    <option value="9">9</option>
    <option value="10">10</option>
    <option value="11">11</option>
    <option value="12">12</option>
    <option value="13">13</option>
    <option value="14">14</option>
    <option value="15">15</option>
    <option value="16">16</option>
    <option value="17">17</option>
    <option value="18">18</option>
    <option value="19">19</option>
    <option value="20">20</option>
    <option value="21">21</option>
    <option value="22">22</option>
    <option value="23">23</option>
    <option value="24">24</option>
```

```
    <option value="25">25</option>
    <option value="26">26</option>
    <option value="27">27</option>
    <option value="28">28</option>
    <option value="29">29</option>
    <option value="30">30</option>
    <option value="31">31</option>
    </select>
    <label for="YEAR">YEAR</label>
    <select name="YEAR" id="YEAR">
    <option value="1971">1971</option>
    <option value="1972">1972</option>
    <option value="1973">1973</option>
    <option value="1974">1974</option>
    <option value="1975">1975</option>
    <option value="1976">1976</option>
    <option value="1977">1977</option>
    <option value="1978">1978</option>
    <option value="1979">1979</option>
    <option value="1980">1980</option>
    <option value="1981">1981</option>
    <option value="1982">1982</option>
    <option value="1983">1983</option>
    <option value="1984">1984</option>
    <option value="1985">1985</option>
    <option value="1986">1986</option>
    <option value="1987">1987</option>
    <option value="1988">1988</option>
    <option value="1989">1989</option>
    <option value="1990">1990</option>
    <option value="1991">1991</option>
    <option value="1992">1992</option>
    <option value="1993">1993</option>
    <option value="1994">1994</option>
    <option value="1995">1995</option>
    <option value="1996">1996</option>
    <option value="1997">1997</option>
    <option value="1998">1998</option>
    <option value="1999">1999</option>
    <option value="2000">2000</option>
    <option value="2001">2001</option>
    <option value="2002">2002</option>
    <option value="2003">2003</option>
    <option value="2004">2004</option>
    <option value="2005">2006</option>
    <option value="2006">2006</option>
    <option value="2007">2007</option>
    <option value="2008">2008</option>
```

```
    </select>
    <br>
    <p>GENDER </p>
    <p><input type="radio" name = "gender" value="Male">Male</p>
    <p><input type="radio" name = "gender" value="Female">Female</p>
    <p>Rate the Blog</p>
    <p><input type="radio" name = "radiobutton" value="good">good</p>
    <p><input type="radio" name="radiobutton" value="not bad">not bad</p>
    <p><input type="radio"  name = "radiobutton" value="Fcan improve">can
improve</p>
    <button id="myButton" name = "radiobutton" class="float-left submit-
button"
    >Submit</button>
</form>

</body>
```

NAME

DATE OF BIRTH

MONTH Jan DATE 1 YEAR 1971

GENDER :

- Male
- Female

Rate the Blog

- good
- not bad
- can improve

Submit

WEEK-3:

JDBC:

a) Dany went to the kings landing supermarket and bought some groceries. As a vendor you have to ask the numbers of items she bought and store the itemid, itemname, cost of the items and store them in the „sales‟ table of database and also execute an SQL query to find the total cost and list out the item she bought.

A)

```
CREATE TABLE  "ITEMS"
  (  "ITEMID" VARCHAR2(4000),
 "ITEMNAME" VARCHAR2(4000),
 "COST" VARCHAR2(4000),
  )
```

```
insert into student(student_id, student_name, email, date_of_birth)
values('1', 'suresh', 'suresh12@gmail.com', '09/12/2003');
```

```
import java.sql.*;
import java.util.*;
public class store {
public static void main(String args[])throws Exception {
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","Bharath","9093");
System.out.println("Enter 2 to get information or 1 to enter information");
System.out.print("Choose option here:");
Scanner sc=new Scanner(System. (http://system.in/)in (http://system.in/));
int choose=sc.nextInt();

if(choose == 1)
```

```
{
 System.out.println("Enter the No of Items Bought.");
 int n=sc.nextInt();
 for(int i=0;i<n;i++) {
 System.out.print("ID:");
 int iteam_id=sc.nextInt();
 System.out.print("Item Name:");
 String iteam_name =sc.next();
 System.out.print("Item cost:");
 float iteam_cost=sc.nextFloat();
 System.out.print("Enter your name:");
 String cus_name=sc.next();
 PreparedStatement ps = con.prepareStatement("insert into sales values(?,?,?,?)");
 ps.setInt(1,iteam_id);
 ps.setString(2,iteam_name);
 ps.setFloat(3,iteam_cost);
 ps.setString(4, cus_name);
 ps.executeUpdate();
 }
 Statement stmt=con.createStatement();
 ResultSet rs=stmt.executeQuery("select * from sales");
 while(rs.next()) {
 System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3)+" "+rs.getString(4));
 }
 System.out.print("TOTAL Amount: ");
 ResultSet rs1=stmt.executeQuery("select sum(iteam_cost) from sales");
 while(rs1.next()) {
 System.out.println(rs1.getFloat(1));
 }
}
else
```

```
{
 System.out.print("Enter customer name: ");
 String name=sc.next();
 String vsql = "select * from sales where cus_name=?";
    PreparedStatement ps = con.prepareStatement(vsql);
    ps.setString(1,name);
    ResultSet rs =  ps.executeQuery();
 while(rs.next()) {
  System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getFloat(3)+" "+rs.getString(4));
  }

}

//stmt.executeQuery("truncate table sales");
}
}
```

output:

```
Enter 2 to get information or 1 to enter information 1
Enter the No of Items Bought.
1
ID:201
Item Name:Cakes
Item cost:1000
Enter your name:Bharath Krishna
201 Cakes 1000.0 Bharath
TOTAL:
1000.0

Enter 2 to get information or 1 to enter information 2
1          vivek     12345   delhi
```

B)

A workshop based on web development is being conducted in the university. Now James wants to know how many members of his class are interested in it and their details.

a. He created a new table namely Student in the Oracle Database with columns: Student_ID, Student_Name, Email, Date_of_Birth.

(a)

```
CREATE TABLE  "STUDENT"
   (  "STUDENT_ID" VARCHAR2(4000),
  "STUDENT_NAME" VARCHAR2(4000),
  "EMAIL" VARCHAR2(4000),
  "DATE_OF_BIRTH" VARCHAR2(4000)
   )
/
```

output:-

Table created

---

b. He stored some records into the Student table.

(b)

```
insert into student(student_id, student_name, email, date_of_birth)
values('1', 'suresh', 'suresh12@gmail.com', '09/12/2003');
insert into student(student_id, student_name, email, date_of_birth)
values('2', 'vinod', 'vinod123@gmail.com', '10/05/2002');
```

output:-

inserted successfully

inserted successfully

(c)

c. He needs to collect ID numbers and contact numbers of the interested students in a new table called "Workshop".

```
import java.sql.*;
import java.util.*;
class Workshop{
public static void main ( String args[] )throws Exception{
Driver dr = new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(dr);
Connection c =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manger");
Statement st = c.createStatement();
Scanner sc = new Scanner (System.in);
char op = 'y';
System.out.print("Do you want to insert your details?(Yes/No) : ");
String opt = sc.nextLine();
if(opt.equals("No") || opt.equals("no") || opt.equals("NO"))
op = 'n';
while(op == 'y'){
System.out.print("Enter the ID number : ");
int id = sc.nextInt();
System.out.print("Enter the contact number : ");
long pn = sc.nextLong();
try{
st.executeUpdate("Insert into workshop values("+id+","+pn+")");
}
catch(Exception e){
System.out.println(e);
```

```
System.out.println("Data not inserted");
}
System.out.print("Do you want to insert your details?(Yes/No) : ");
opt = sc.next();
if(opt.equals("No") || opt.equals("no") || opt.equals("NO"))
op = 'n';
}
ResultSet rs = st.executeQuery("Select * from workshop");
while(rs.next()){
System.out.println(rs.getInt(1)+"\t"+rs.getLong(2));
}
}
}
```

output:-

```
Do you want to insert your details?(Yes/No): Yes
Enter the ID number: 1
Enter the contact number: 123456789
Do you want to insert your details?(Yes/No): Yes
Enter the ID number: 2
Enter the contact number: 234567890
```

id	contactnumber
1	123456789
2	234567890

(d)

d. He wants to retrieve name and email address from the student table and update the Workshop table by creating 2 new columns and inserting the respective details.

```
import java.sql.*;
import java.util.*;
class Workshop_b{
public static void main ( String args[] )throws Exception{
Driver dr = new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(dr);
Connection c =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manger");
Statement st1 = c.createStatement();
Statement st2 = c.createStatement();
try{
st1.executeUpdate("alter table Workshop add (name varchar2(30), email
varchar2(30))");
}catch(Exception e){
System.out.println(e);
}
ResultSet rs = st1.executeQuery("select student_id, student_name, email from student");
while(rs.next()){
//System.out.println(rs.getLong(1) +"\t"+ rs.getString(2));
st2.executeUpdate("Update workshop set email='"+rs.getString(3)+"', name
='"+rs.getString(2)+"' where id="+rs.getLong(1));
}
}
}
```

output:-

id	contactnumber	name	email
1	123456789	suresh	suresh12@gmail.com
2	234567890	vinod	vinod123@gmail.com

(e)

e. If a student, suddenly wants to drop from attending the workshop, James is supposed to delete the student's data from workshop table.

```
import java.sql.*;
import java.util.*;
class Workshop{
public static void main ( String args[] )throws Exception{
Driver dr = new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(dr);
Connection c =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manger");
Statement st = c.createStatement();
Scanner sc = new Scanner (System.in);
char op = 'y';
System.out.print("Do you want to delete your details?(Yes/No) : ");
String opt = sc.nextLine();
if(opt.equals("No") || opt.equals("no") || opt.equals("NO"))
op = 'n';
while(op == 'y'){
System.out.print("Enter the ID number : ");
int id = sc.nextInt();
try{
st.executeUpdate("delete from workshop where id = "+id);
}
catch(Exception e){
System.out.println(e);
}
System.out.print("Do you want to delete your details?(Yes/No) : ");
opt = sc.next();
if(opt.equals("No") || opt.equals("no") || opt.equals("NO"))
```

```
op = 'n';
}
ResultSet rs = st.executeQuery("Select * from workshop");
while(rs.next()){
System.out.println(rs.getInt(1)+"\t"+rs.getLong(2));
}
}
}
```

output:-

Do you want to delete your details?(Yes/No): Yes

Enter ID number: 1

Do you want to delte your details>(Yes/No): No

id	contactnumber	name	email
2	234567890	vinod	vinod123@gmail.com

C)

c) Write the Java programs for the above situations using JDBC API.

c) Tony is a wildlife photographer. He went to Amazon rainforest to explore the beauty of the flora and fauna. He captured pictures of different trees and animals. He also described each animal and plant in a separate notepad. Now he wants to save the name, the category (animal/plant), its image and the description file of the species into a database. Help Tony by providing him with an executable java program.

a. Create a table with columns Name, Category, Image and Description_File before executing the file.

b. The name, category, path of the image(.jpg), path of the description file (.txt) should be read as input from the console.

```
import java.sql.*;
import java.io.*;
import java.util.*;
```

```
public class Amazon{
public static void main(String[]args)throws Exception {
Scanner sc= new Scanner(System.in);
Driver d= new oracle.jdbc.driver.OracleDriver();
DriverManager.registerDriver(d);
Connection
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","manger");
int cont=1;
while(cont==1){
PreparedStatement st =con.prepareStatement("insert into forest values(?,?,?,?)");
System.out.println("Enter the name of the plant/animal ?");
String name =sc.next();
System.out.println("Enter the category ?");
String category=sc.next();
System.out.println("Enter the path of the image");
String ipath=sc.next();
System.out.println("Enter the path of the text file");
String fpath=sc.next();
st.setString(1,name);
st.setString(2,category);
FileInputStream f =new FileInputStream(ipath);
st.setBinaryStream(3,f,f.available());
File fs = new File(fpath);
FileReader fr=new FileReader(fs);
st.setCharacterStream(4,fr,(int)fs.length());
st.executeUpdate();
System.out.println("Do you want to Continue(yes/no) :");
String flag = sc.next();
if(flag.equals("yes")){
cont=1;
}
```

```
else{
cont=0;
}
}
con.close();
}
}
```

output:-

```
Enter the name of the plant/animal : Tiger
Enter the category : big cats
Enter path of the image : c:\tiger.jpg
Enter the path of the text file: c:\tiger.txt
Do you want to Continue(yes/no): no
```

WEEK-4:

Basic Servlet Program to validate Login data

1. Design a Login Application using Servlet and JDBC validate them by fetching data from database.

Log.java

```
import java.io.*;

import javax.servlet.*;

import javax.servlet.http.*;

import java.sql.*;


public class log extends HttpServlet

{

 public void service(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException{

  System.out.println("service called...");

  //to send the response to the browser

  PrintWriter out = res.getWriter();

  res.setContentType("text/html");

  String uname = req.getParameter("uname");

  String pwd = req.getParameter("pwd");

  try{

   Class.forName("oracle.jdbc.driver.OracleDriver");

   Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","28052003");

   String vsql = "select * from userdetails where uname=? and pwd=?";

   PreparedStatement pstmt = con.prepareStatement(vsql);

   pstmt.setString(1,uname);

   pstmt.setString(2,pwd);

   ResultSet rs = pstmt.executeQuery();

   if( rs.next() ){

    req.setAttribute("uname",uname);
```

```
    RequestDispatcher rd = req.getRequestDispatcher("/Welcome");
    rd.forward(req,res);
   }else{
    out.println("Invalid username/password<br>");
    RequestDispatcher rd = req.getRequestDispatcher("index.html");
    rd.include(req,res);
   }
   out.println("</body>");
   out.println("<html>");
   con.close();
  }catch(Exception e){
   res.sendError(500,"Our application is failed due to:" + e);
  }
 }
}
```

Welcome.java:

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class Welcome extends HttpServlet
{
 public void service(HttpServletRequest req,HttpServletResponse res)throws
ServletException,IOException{
  System.out.println("service called...");
```

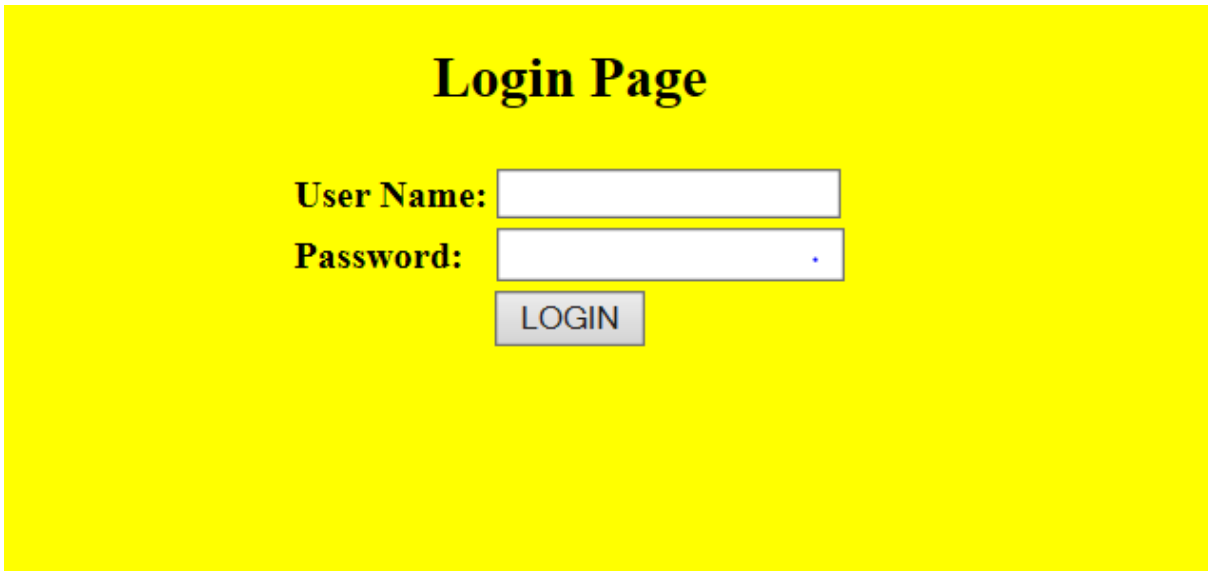
```
 PrintWriter out = res.getWriter();



 res.setContentType("text/html");



 String uname = (String)req.getParameter("uname");


 out.println("Welcome to:" + uname);
}
}
```

index.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Login Application</title>
</head>
<body bgcolor="yellow" >
  <form action="log">
  <table align="center">
   <tr>
    <th colspan="2"><h2 align=center>Login Page</h2></th>
   </tr>
   <tr>
    <td><b>User Name:</b></td>
    <td><input type="text" name="uname"></td>
   </tr>
   <tr>
    <td><b>Password:</b></td>
    <td><input type="password" name="pwd"></td>
   </tr>
   <tr>
    <td colspan=2 align="center"><input type="submit" value="LOGIN"/>
   </tr>
 </table>
 </form>
</body>
</html>
```

Web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://Java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" id="WebApp_ID" version="2.5">

	<servlet>
		<servlet-name>login</servlet-name>
		<servlet-class>log</servlet-class>
	</servlet>
	<servlet>
		<servlet-name>welcome</servlet-name>
		<servlet-class>Welcomep</servlet-class>
	</servlet>
	<servlet-mapping>
		<servlet-name>login</servlet-name>
		<url-pattern>/log</url-pattern>
	</servlet-mapping>
	<servlet-mapping>
		<servlet-name>welcome</servlet-name>
		<url-pattern>/welcome</url-pattern>
	</servlet-mapping>
	<welcome-file-list>
		<welcome-file>index.html</welcome-file>
	</welcome-file-list>
</web-app>
```



WEEK-5:

Servlets Seesions, Cookies

1. Servlet program to read initialization parameters using ServletConfig and ServletContext object.

1)

CODE

Lab51.java

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

public class lab51 extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)  throws
ServletException,IOException
{
res.setContentType("text/html");
PrintWriter pw=res.getWriter();

//creating ServletContext object
ServletContext context = getServletContext();

//Getting the value of the initialization parameter and printing it
String driverName=context.getInitParameter("dname");
pw.println("driver name is="+driverName);

pw.close();

}

}
```

Web.xml

```
<web-app>

<servlet>
<servlet-name>ESDLAB5</servlet-name>
<servlet-class>lab51</servlet-class>
</servlet>

<context-param>
<param-name>dname</param-name>
<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
</context-param>

<servlet-mapping>
<servlet-name>ESDLAB5</servlet-name>
<url-pattern>/context</url-pattern>
</servlet-mapping>

</web-app>
```

2)

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form method="post" action="PwdCheck">
name:<input type="text" name="uname"/>
password<input type="text" name="pwd"/>
<input type="submit" value="login"/>
</form>
</body>
</html>
```

PwdCheck.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
* Servlet implementation class PwdCheck
*/
public class PwdCheck extends HttpServlet {
private static final long serialVersionUID = 1L;

 /**
 * @see HttpServlet#HttpServlet()
 */
 public PwdCheck() {
 super();
 // TODO Auto-generated constructor stub
 }
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//response.getWriter().append("Served at: ").append(request.getContextPath());
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//doGet(request, response);
```

```
PrintWriter out=response.getWriter();
try {
response.setContentType("text/html");
String uname=request.getParameter("uname");
String pwd=request.getParameter("pwd");
if(pwd.equals("Admin"))
{out.println("Welcome "+uname);}
else {
out.println("Wrong Password");
out.println("please try again");
out.println("<form method=\"post\" action=\"PwdCheck\">\r\n" +
"name:<input type=\"text\" name=\"uname\"/>\r\n" +
"password<input type=\"text\" name=\"pwd\"/>\r\n" +
"<input type=\"submit\" value=\"submit\"/>\r\n" +
"</form>");
out.println("<a href='PwdChange?uname="+uname+"'>Forgot Password</a>");
out.close();}
}
catch(Exception e) {out.println(e);}
}
}
```

PwdChange.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
* Servlet implementation class PwdChange
*/
public class PwdChange extends HttpServlet {
private static final long serialVersionUID = 1L;

 /**
 * @see HttpServlet#HttpServlet()
 */
 public PwdChange() {
 super();
 // TODO Auto-generated constructor stub
 }
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//response.getWriter().append("Served at: ").append(request.getContextPath());
PrintWriter out=response.getWriter();
try {
response.setContentType("text/html");
String uname=request.getParameter("uname");
out.println("Hello "+uname);
out.println("Please enter new password");
```

```
out.println("<form method=\"post\" action=\"PwdCheck\">\r\n" +
"new password<input type=\"text\" name=\"pwd\"/>\r\n" +
"<a href=\'index.html\'>submit</a>\r\n" +
"</form>");
}
catch(Exception e) {out.println(e);}
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//doGet(request, response);
}
}
```

Web.xml - notebook

```
<webapp>

 <servlet>



 <servlet-name>ESDLA5</servlet-name>

 <servlet-class>PwdCheck</servlet-class>

 </servlet>


 <servlet>


 <servlet-name>ESDLB5</servlet-name>

 <servlet-class>PwdChange</servlet-class>

 </servlet>


<servlet-mapping>

 <servlet-name>ESDLA5</servlet-name>

 <url-pattern>/PwdCheck</url-pattern>

 </servlet-mapping>


 <servlet-mapping>
```

<servlet-name>ESDLB5</servlet-name>

<url-pattern>/PwdChange</url-pattern>

</servlet-mapping>

</web-app>

3)

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href='Noofclicks'>Click me</a>
</body>
</html>
```

Noofclicks.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
/**
* Servlet implementation class HitCounter
*/
public class Noofclicks extends HttpServlet {
private static final long serialVersionUID = 1L;
 private int y;
 /**
 * @see HttpServlet#HttpServlet()
 */
 public Noofclicks() {
 super();
 // TODO Auto-generated constructor stub
 }
/**
* @see Servlet#init(ServletConfig)
*/
@SuppressWarnings("unused")
public void init(ServletConfig config) throws ServletException {
// TODO Auto-generated method stub
```

```
int y;
y=0;
//x=Integer.toString(y);
//Cookie ck=new Cookie("num",x);
}
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//response.getWriter().append("Served at: ").append(request.getContextPath());
y++;
PrintWriter out=response.getWriter();
response.setContentType("text/html");
out.println("No of clicks = "+y);
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
doGet(request, response);
}
}
```

Web.xml

```
<webapp>
<servlet>

 <servlet-name>ESDL5</servlet-name>
 <servlet-class>Noofclicks</servlet-class>
 </servlet>
  <servlet-mapping>
 <servlet-name>ESDL5</servlet-name>
 <url-pattern>/Noofclicks</url-pattern>
 </servlet-mapping>
</web-app>
```

4)

Index.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
```

```
</head>
<body>
<form action="FeeStructure" method="post">
name:<input type="text" name="uname"/>
password<input type="password" name="pwd"/>
<input type="submit" value="login"/>
</form>
</body>
</html>
```

FeeStructure.java

```
import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

/**

* Servlet implementation class FeeStruc

*/

public class FeeStructure extends HttpServlet {

private static final long serialVersionUID = 1L;


 /**

 * @see HttpServlet#HttpServlet()

 */

 public FeeStructure() {

 super();

 // TODO Auto-generated constructor stub

 }

/**

* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

*/

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws

ServletException, IOException {

// TODO Auto-generated method stub
```

```
//response.getWriter().append("Served at: ").append(request.getContextPath());

}

/**

* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)

*/

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws

ServletException, IOException {

// TODO Auto-generated method stub

//doGet(request, response);

PrintWriter out=response.getWriter();

response.setContentType("text/html");

try {

String name=request.getParameter("uname");

String pwd=request.getParameter("pwd");

if(name.equals("suchi")) {

if(pwd.equals("MyPassword")) {

out.println("Welcome "+name+"\nPlease fill the details");

out.println("<form action=\"Feesession\" method=\"post\">\r\n" +

"name<input type=\"text\" name=\"uname\"/>\r\n" +

"college<input type=\"text\" name=\"clg\"/>\r\n" +

"amount<input type=\"text\" name=\"amt\"/>\r\n" +

"<input type=\"submit\" value=\"pay\"/>\r\n" +

"</form>");

}

}

else {out.println("Wrong Password..Try again\n");

out.println("<form action=\"FeeStructure\" method=\"post\">\r\n" +

"name:<input type=\"text\" name=\"uname\"/>\r\n" +

"password<input type=\"password\" name=\"pwd\"/>\r\n" +

"<input type=\"submit\" value=\"login\"/>");

}

}catch(Exception e) {out.println(e);}

}
```

```
}
```

Feesession.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
/**
* Servlet implementation class FeeSession
*/
public class Feesession extends HttpServlet {
private static final long serialVersionUID = 1L;

 /**
 * @see HttpServlet#HttpServlet()
 */
 public Feesession() {
 super();
 // TODO Auto-generated constructor stub
 }
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//response.getWriter().append("Served at: ").append(request.getContextPath());
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
```

```
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//doGet(request, response);
PrintWriter out=response.getWriter();
response.setContentType("text/html");
try {
String name=request.getParameter("uname");
String clg=request.getParameter("clg");
String amt=request.getParameter("amt");
HttpSession session=request.getSession();
session.setAttribute("name",name);
session.setAttribute("clg",clg);
session.setAttribute("amt",amt);
out.println("You are going to pay to "+clg+" an amount of "+amt);
out.println("\nDo you want to continue?");
out.println("<form action=\"bill\" method=\"post\">\r\n" +
"<input type=\"submit\" value=\"yes\"/>\r\n" +
"</form>\r\n" +
"<form action=\"FeeStructure\" method=\"post\">\r\n" +
"<input type=\"submit\" value=\"no\"/>\r\n" +
"</form>");
}catch(Exception e) {out.println(e);}
}}
```

Bill.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```

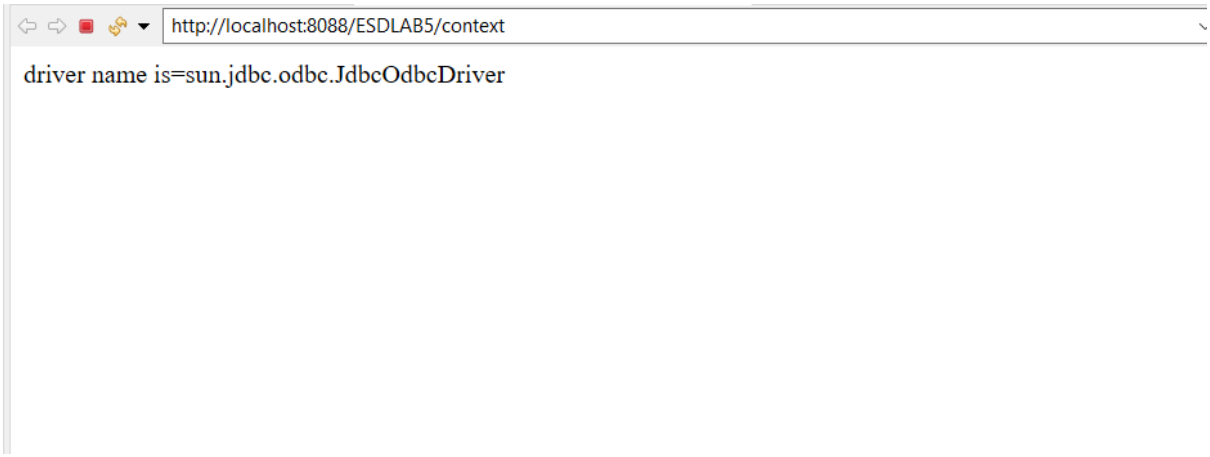
```
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
/**
* Servlet implementation class Bill
*/
public class bill extends HttpServlet {
private static final long serialVersionUID = 1L;

 /**
 * @see HttpServlet#HttpServlet()
 */
 public bill() {
 super();
 // TODO Auto-generated constructor stub
 }
/**
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
*/
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//response.getWriter().append("Served at: ").append(request.getContextPath());
}
/**
* @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
response)
*/
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws
ServletException, IOException {
// TODO Auto-generated method stub
//doGet(request, response);
PrintWriter out=response.getWriter();
response.setContentType("text/html");
```

```
try {
HttpSession session=request.getSession();
out.println("Bill");
String name=(String)session.getAttribute("name");
String clg=(String)session.getAttribute("clg");
String amt=(String)session.getAttribute("amt");
out.println("</br>PAYMENT ID:"+session.getId());
out.println("</br>NAME:"+name);
out.println("</br>COLLEGE:"+clg);
out.println("</br>AMOUNT:"+amt);
session.invalidate();
}catch(Exception e) {out.println(e);}
}
}
```

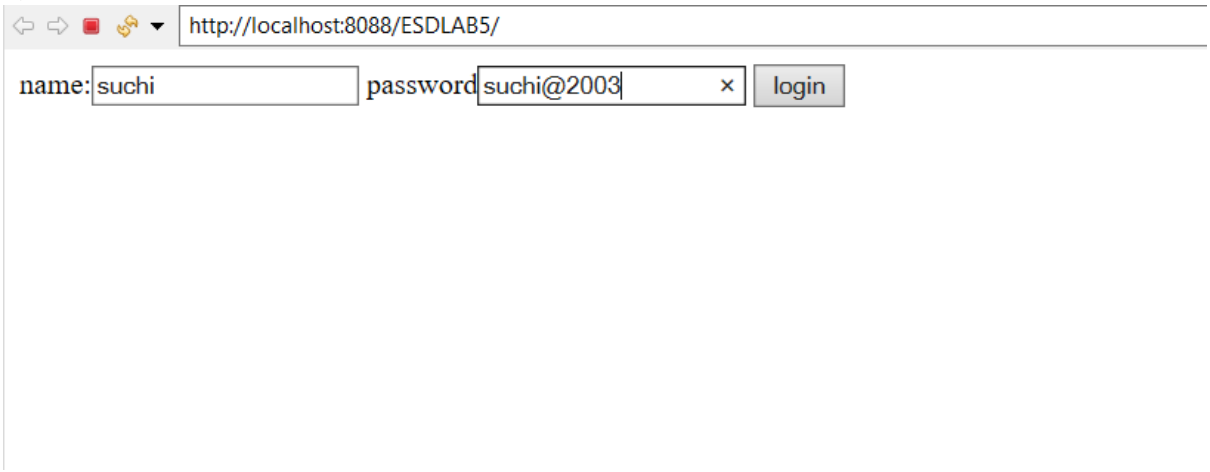
Web.xml

```
<webapp>
<servlet>
  <servlet-name>ESD5</servlet-name>
 <servlet-class>FeeStructure</servlet-class>
 </servlet>
  <servlet-mapping>
 <servlet-name>ESD5</servlet-name>
 <url-pattern>/FeeStructure</url-pattern>
 </servlet-mapping>

 <servlet>
 <servlet-name>ESL5</servlet-name>
 <servlet-class>Feesession</servlet-class>
 </servlet>
  <servlet-mapping>
 <servlet-name>ESL5</servlet-name>
 <url-pattern>/Feesession</url-pattern>
 </servlet-mapping>

 <servlet>

 <servlet-name>EDL5</servlet-name>
 <servlet-class>bill</servlet-class>
 </servlet>
  <servlet-mapping>
 <servlet-name>EDL5</servlet-name>
 <url-pattern>/bill</url-pattern>
 </servlet-mapping>
      </web-app>
```

1)
Outputs:


http://localhost:8088/ESDLAB5/context
driver name is=sun.jdbc.odbc.JdbcOdbcDriver

2)

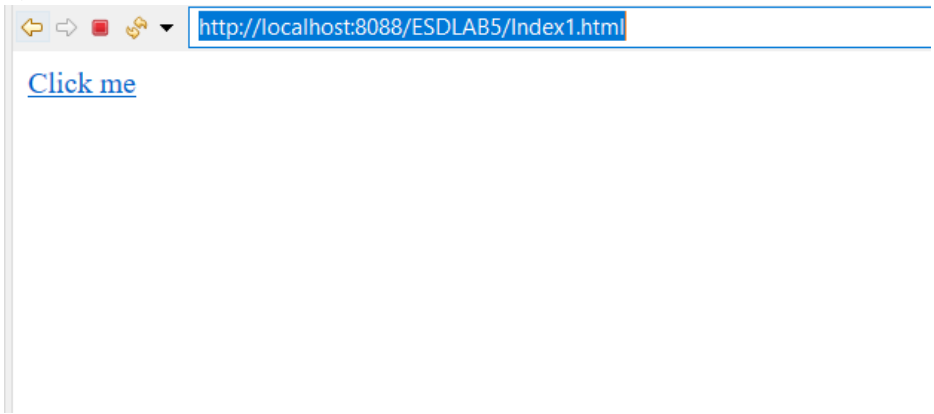

http://localhost:8088/ESDLAB5/
name:
suchi
password
suchi@2003
login

Wrong Password please try again
name: password submit

Forgot Password

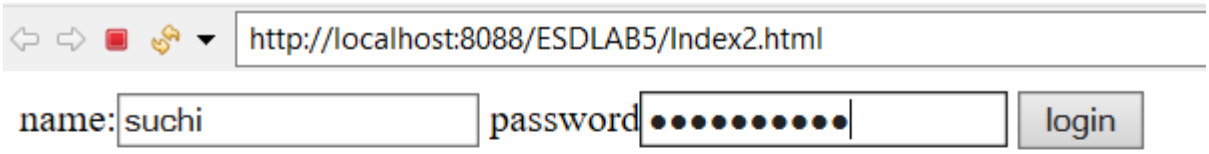
Hello suchi Please enter new password
new password submit

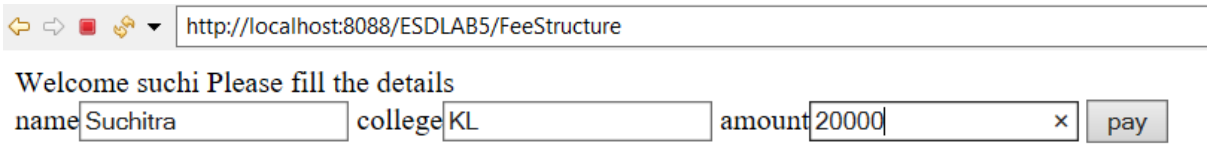
3)



No of clicks = 3

4)

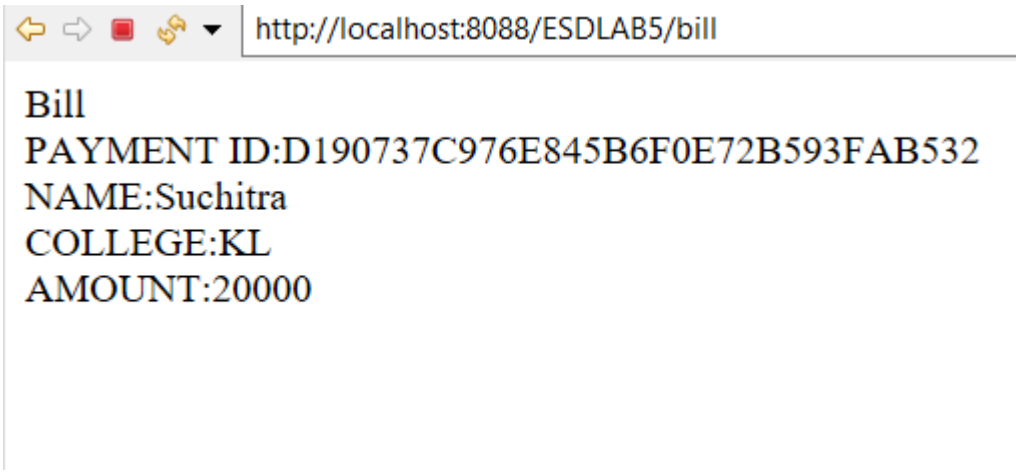




You are going to pay to KL an amount of 20000 Do you want to continue?

yes

no



WEEK-6:

JSP:

1)

login.jsp:-

```
<%@ page import ="java.sql.*" %>
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
</head>
<body>
<form action="LoginRegister">
<table background-color: Lightgreen; margin-left: 20px; margin-left: 20px;>
<tr>
<td><h3 style="color: red;">Login page !!!</h3></td>
<td></td>
</tr>
<tr>
<td>UserName :</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Password :</td>
<td><input type="password" name="password"></td>
</tr>
<tr>
<td><input type="submit" name="submit" value="Login"></td>
<td><a href="register.jsp">Registration</a>
```

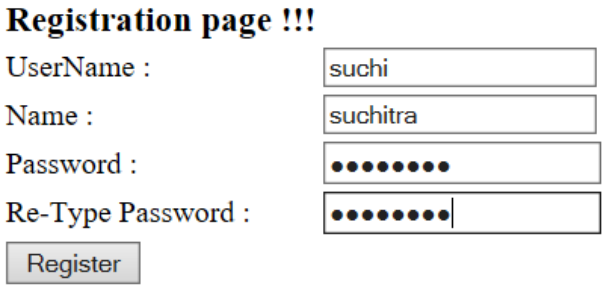
```
</tr>
</table>
</form>
</body>
</html>
```

---

register.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" >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>registration</title>
</head>
<body>
<form action="LoginRegister">
<table background-color:skyblue; margin-left: 20px; margin-left: 20px;>
<tr>
<td><h3 style="color: black;">Registration page !!!</h3></td>
<td></td>
</tr>
<tr>
<td>UserName :</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Name :</td>
<td><input type="text" name="name"></td>
</tr>
```

```
<tr>
<td>Password :</td>
<td><input type="password" name="password1"></td>
</tr>
<tr>
<td>Re-Type Password :</td>
<td><input type="password" name="password2"></td>
</tr>
<tr>
<td><input type="submit" name="submit" value="Register"></td>
<td></td>
</tr>
</table>
</form>
</body>
</html>
```



2)

login.jsp:-

```
<%@ page import ="java.sql.*" %>
```

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
 pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
</head>
<body>
<form action="LoginRegister">
<table background-color: Lightgreen; margin-left: 20px; margin-left: 20px;>
<tr>
<td><h3 style="color: red;">Login page !!!</h3></td>
<td></td>
</tr>
<tr>
<td>UserName :</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Password :</td>
<td><input type="password" name="password"></td>
</tr>
<tr>
<td><input type="submit" name="submit" value="Login"></td>
<td><a href="register.jsp">Registration</a>
</tr>
</table>
</form>
</body>
</html>
```

---

register.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" >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title>registration</title>
</head>
<body>
<form action="LoginRegister">
<table background-color:skyblue; margin-left: 20px; margin-left: 20px;>
<tr>
<td><h3 style="color: black;">Registration page !!!</h3></td>
<td></td>
</tr>
<tr>
<td>UserName :</td>
<td><input type="text" name="username"></td>
</tr>
<tr>
<td>Name :</td>
<td><input type="text" name="name"></td>
</tr>
<tr>
<td>Password :</td>
<td><input type="password" name="password1"></td>
</tr>
<tr>
<td>Re-Type Password :</td>
```

```
<td><input type="password" name="password2"></td>
</tr>
<tr>
<td><input type="submit" name="submit" value="Register"></td>
<td></td>
</tr>
</table>
</form>
</body>
</html>
```

WEEK-7:

Hibernate:

1)

Goodouwn.hbm.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<hibernate-mapping>
 <class name="Goodouwn" table="Goodouwn">
 <id name="id" column="id"></id>
 <property name="amount" column="amount"></property>
 <property name="cost" column="cost"> </property>
 <property name="type" column="type"></property>
 </class>
</hibernate-mapping>
```

Goodouwn.java

```
public class Goodouwn {
 private int id;

 private int cost;
 private String type;
 private int amount;
 public Goodouwn(int id, String type, int amount,int cost) {
  super();
  this.id = id;
  this.type = type;
  this.amount = amount;
  this.cost=cost;
```

```
 }
 public int getId() {
  return id;
 }
 public void setId(int id) {
  this.id = id;
 }
 public int getCost() {
  return cost;
 }
 public void setCost(int cost) {
  this.cost = cost;
 }
 public String getType() {
  return type;
 }
 public void setType(String type) {
  this.type = type;
 }
 public int getAmount() {
  return amount;
 }
 public void setAmount(int amount) {
  this.amount = amount;
 }
}
```

---

hibernate.cfg.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
```

```
   "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
   "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>

   <session-factory>

      <property name="dialect">org.hibernate.dialect.Oracle10gDialect</property>
      <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
      <property name="connection.username">siva</property>
      <property name="connection.password">siva</property>
      <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
   <mapping resource="Goodouwn.hbm.xml"/>
   </session-factory>

</hibernate-configuration>
```

Storing.java

```
import org.hibernate.Session;
import java.util.Scanner;

import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
import org.hibernate.service.ServiceRegistry;
import org.hibernate.Transaction;
public class Storing {
 public static void main(String args[])
 {
  Scanner reader=new Scanner(System.in);
```

```
while(true)
{
 System.out.println("Enter yes if u want to enter details or no if u want to exit ");
 String s=reader.nextLine();
 if(s.equals("yes"))
 {
  Configuration cfg=new Configuration();
  cfg.configure();
  SessionFactory sf=cfg.buildSessionFactory();
  Session se=sf.openSession();
  System.out.println("Enter the type of rice(polished/unpolished)");
  String type=reader.nextLine();

  System.out.println("Enter id(integer) ");
  int id=reader.nextInt();
  System.out.println("Enter  amount(kg) ");
  int amount=reader.nextInt();
  System.out.println("Enter  cost ");
  int cost=reader.nextInt();

  Goodouwn go=new Goodouwn(id,type,amount,cost);
  se.save(go);
  Transaction tx=se.beginTransaction();
  tx.commit();
  se.close();
  System.out.println("SAVED");

 }
 else
 {
  break;
```

```
  }
 }
}

}
```

Output:

```
Enter the id of the rice bag
1
Enter the quantity of rice bag in kgs
5
Enter the cost of rice bag
20
Enter the type of rice bag(polished/nonpolished)
polished
Do you want to insert(yes/no) :
no
Data has been entered
```

2)

LOGIC file:-

```
import java.util.Scanner;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
public class SellBag {
public static void main(String[]args)
{
Scanner sc = new Scanner(System.in);
Configuration cfg = new Configuration();
cfg.configure("hibernate.cfg.xml");
SessionFactory sf=cfg.buildSessionFactory();
```

```
Session s= sf.openSession();
System.out.println("Enter the id of the rice bag");
int x = sc.nextInt();
Object o = s.load(Ricebag.class,new Integer(x));
Ricebag r = (Ricebag)o;
System.out.println("the cost of the rice bag is :"+r.getCost());
System.out.println("Want to purchase the bag(yes/no)");
String y = sc.next();
if(y.equals("yes")) {
Transaction tx = s.beginTransaction();
s.delete(r);
tx.commit();
System.out.println("the bag is sold");
}
else
{
System.out.println("Not sold");
}
s.close();
}
}
```

Output:

```
Enter the id of the rice bag
10
the cost of the rice bag is :20.0
Want to purchase the bag(yes/no)
yes
the bag is sold
```

3)

POJO file(GokulFabrics.java):-

```
public class GokulFabrics {
private int fid;
private String type;
private double availability;
private double costpermetre;
public int getFid() {
return fid;
}
public void setFid(int fid) {
this.fid = fid;
}
public String getType() {
return type;
}
public void setType(String type) {
this.type = type;
}
public double getAvailability() {
return availability;
}
public void setAvailability(double availability) {
this.availability = availability;
}
public double getCostpermetre() {
return costpermetre;
}
public void setCostpermetre(double costpermetre) {
this.costpermetre = costpermetre;
}
}
```

Mapping file(gokulfabrics.hbm.xml):-

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="lab10post.GokulFabrics" table="fabrics">
<id name = "fid" column="idno">
<generator class = "assigned"/>
</id>
<property name = "type" column="type"/>
<property name = "availability" column="length"/>
<property name = "costpermetre" column="cost"/>
</class>
</hibernate-mapping>
```

Configuration file(hibernate.cfg.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<!-- connection details -->
<property name = "connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
<property name = "connection.url">jdbc:oracle:thin:@localhost:1522:xe</property>
<property name="connection.user">system</property>
<property name= "connection.password">system</property>
```

```
<!-- Hibernate details -->
<property name="show_sql">true</property>
<property name="hbm2ddl.auto">update</property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<!-- mapping resources -->
<mapping resource = "gokulfabrics.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

LOGIC file:-

```
import java.util.Scanner;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import lab10in1.Ricebag;
public class InsertFabric {
public static void main(String[]args) {
Scanner sc= new Scanner(System.in);
Boolean continues=true;
Configuration cfg= new Configuration();
cfg.configure("hibernate.cfg.xml");
SessionFactory sf= cfg.buildSessionFactory();
Session s = sf.openSession();
while(continues)
{
GokulFabrics g= new GokulFabrics();
System.out.println("Enter the id of the fabric");
int id =sc.nextInt();
System.out.println("Enter the type of fabric");
```

```
String type=sc.next();
System.out.println("Enter the availability of fabric in metres");
double avail=sc.nextDouble();
System.out.println("Enter the cost per metre ");
double c=sc.nextDouble();
g.setFid(id);
g.setType(type);
g.setAvailability(avail);
g.setCostpermetre(c);
s.save(g);
Transaction tx= s.beginTransaction();
tx.commit();
System.out.println("Do you want to insert(yes/no) :" );
String proceed=sc.next();
if(proceed.equals("yes"))
{
continues=true;
}
else
{
continues=false;
}
}
System.out.println("Data has been entered");
s.close();
}
}
```

LOGIC file:-

```
import java.util.Scanner;
```

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
public class UpdateFabric {
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter the id of fabric");
int x= sc.nextInt();
System.out.println("Enter the length of fabric sold out");
double y = sc.nextDouble();
Configuration cfg = new Configuration();
cfg.configure("hibernate.cfg.xml");
SessionFactory sf = cfg.buildSessionFactory();
Session s = sf.openSession();
Object o=s.load(GokulFabrics.class,new Integer(x));
GokulFabrics g=(GokulFabrics)o;
Transaction tx= s.beginTransaction();
double sellingcost = g.getCostpermetre()*y;
System.out.println("The selling cost of fabric is :"+sellingcost);
double rem=g.getAvailability()-y;
g.setAvailability(rem);
tx.commit();
System.out.println("Data is updated");
s.close();
}
}
```

```
Enter the id of the fabric
1
Enter the type of fabric
nylon
Enter the availability of fabric in metres
25.9
Enter the cost per metre
5
Do you want to insert(yes/no) :
no
Data has been entered
```

Enter the id of the fabric

1

Enter the length of fabric sold out

10

The selling cost of the fabric is

50

WEEK-8:

HQL, HCQL:

1. Stoins is the manager of Minimal Cube company. He maintains the records of employees working in his company. Create a java class where he gets and sets the values of EmpID, EmpName, EmpSalary, EmpAddress. Use Hibernate Frame-work to reduce manual work. When employees are terminated he is likely to delete the record of employee in his database and update the data when it is required. After performing all the operations he is likely to know the employees working in his company at the end of every month, so retrieve the data those who are working in his company. Create separate java class for retrieving, updating and deleting so that the Mr. Stoins can easily work when an employee data needs to update, delete or retrieve. Use Concept of HCQL (Hibernate Criteria Query Language).

hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<!-- related to connection start -->
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
<property name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
<property name="connection.user"></property>
<property name="connection.password"></property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<property name="hbm2ddl.auto">update</property>
<property name="show_sql">true</property>
<mapping resource="MinimalcubeEmp.hbm.xml"></mapping>
</session-factory>
</hibernate-configuration>
```

MinimalcubeEmp.hbm.xml

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
   "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
   "http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
 <class name="com.mypackage.MinimalcubeEmp" table="MinimalcubeEmp">
  <id name="id" type="int" column="ID">
          <generator class="native"></generator>
  </id>
  <property name="name" column="NAME"/>
  <property name="salary" column="SALARY"/>
  <property name="address" column="ADDRESS"/>
 </class>
</hibernate-mapping>
```

MinimalcubeEmp.java

```
package com.mypackage;
public class MinimalcubeEmp {
                private int id;
                private String name;
                private String address;
                private double salary;
                public int getId() {
                        return id;
                }
                public void setId(int id) {
```

```
		this.id = id;
	}
	public String getName() {
		return name;
	}
	public void setName(String name) {
		this.name = name;
	}
	public String getAddress() {
		return address;
	}
	public void setAddress(String address) {
		this.address = address;
	}
	public double getSalary() {
		return salary;
	}
	public void setSalary(double salary) {
		this.salary = salary;
	}

}
```

ins.java

```
package com.mypackage;
import org.hibernate.Session;
```

```
import org.hibernate.Criteria;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
import java.util.*;
public class ins {
public static void main(String args[]) {
        Configuration cfg = new Configuration();
   cfg.configure("hibernate.cfg.xml");
   SessionFactory factory = cfg.buildSessionFactory();
   Session session = factory.openSession();
   session.beginTransaction();
   Criteria criteria = session.createCriteria(MinimalcubeEmp.class);
        List<MinimalcubeEmp> emplist = (List<MinimalcubeEmp>) criteria.list();
   for(MinimalcubeEmp emp : emplist){
        System.out.println("Id: " + emp.getId());
      System.out.println("Name: "  + emp.getName());
      System.out.println("Salary: " + emp.getSalary());
      System.out.println("Address: "  + emp.getAddress());
      System.out.println();
   }
   session.close();
   factory.close();
}
}
```

update.java

```
package com.mypackage;
```

```
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.cfg.Configuration;
import java.util.*;
public class update {
        public static void main(String args[]) {
        Configuration cfg = new Configuration();
        Scanner sc=new Scanner(System.in);
   cfg.configure("hibernate.cfg.xml");
   SessionFactory factory = cfg.buildSessionFactory();
   Session session = factory.openSession();
   Transaction transaction = session.beginTransaction();
   System.out.println("Enter Employee Id to update information");
   int id=sc.nextInt();
   sc.nextLine();
   System.out.println("Enter attribute to update");
   String attr=sc.nextLine();
   if(attr.equals("salary")) {
        System.out.println("Enter Salary");
        double sa=sc.nextDouble();
   String updateq= "update MinimalcubeEmp set salary=? where Id=?";
   Query query2 = session.createQuery(updateq);
   query2.setParameter(1, id);
   query2.setParameter(0, sa);
   query2.executeUpdate();}
   if(attr.equals("name")) {
        System.out.println("Enter Name");
        sc.nextLine();
        String na=sc.nextLine();
```

```
    String updateq= "update MinimalcubeEmp set name=? where Id=?";
    Query query2 = session.createQuery(updateq);
    query2.setParameter(1, id);
    query2.setParameter(0, na);
    query2.executeUpdate();}
    if(attr.equals("address")) {
            System.out.println("Enter Address");
            String ad=sc.nextLine();
    String updateq= "update MinimalcubeEmp set address=? where Id=?";
    Query query2 = session.createQuery(updateq);
    query2.setParameter(1, id);
    query2.setParameter(0, ad);
    query2.executeUpdate();}
    transaction.commit();
    session.clear();
    session.close();
    System.out.println("Update Completed !");
}
}
```

delete.java

```
package com.mypackage;
import org.hibernate.Query;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
```

```
import org.hibernate.cfg.Configuration;
import java.util.*;
public class delete {
	public static void main(String args[]) {
		Configuration cfg = new Configuration();
		Scanner sc=new Scanner(System.in);
	  cfg.configure("hibernate.cfg.xml");
	  SessionFactory factory = cfg.buildSessionFactory();
	  Session session = factory.openSession();
	  Transaction transaction = session.beginTransaction();
	  System.out.println("Enter Employee Id to delete");
	  int id=sc.nextInt();
	  sc.nextLine();
	  String qryString2 = "delete from MinimalcubeEmp where id=?";
      Query query2 = session.createQuery(qryString2);
      query2.setParameter(0, id);
      int count2 = query2.executeUpdate();
		transaction.commit();
		session.clear();
		session.close();
		System.out.println("Delete Completed !");
	}
}
```

```
log4j:WARN No appenders could be found for logger (org.hibernate.cfg.Environment).
log4j:WARN Please initialize the log4j system properly.
Enter Employee Id to update information
123
Enter attribute to update
address
Enter Address
Vuyyuru
Hibernate: update MinimalcubeEmp set ADDRESS=? where Id=?


Update Completed !
```

```
log4j:WARN No appenders could be found for logger (org.hibernate.cfg.Environment).
log4j:WARN Please initialize the log4j system properly.
Enter Employee Id to delete
110
Hibernate: delete from MinimalcubeEmp where ID=?
Delete Completed !
```

```
log4j:WARN Please initialize the log4j system properly.
Hibernate: select this_.ID as ID0_0_, this_.NAME as NAME0_0_, this_.SALARY as SALARY0_0_, this_.ADDRESS as ADDRESS0_0_ from MinimalcubeEmp this_
Id: 18
Name: Vijaya
Salary: 833333.3333
Address: Vijayawada

Id: 20
Name: Chinmayi
Salary: 833333.3333
Address: Hyderabad

Id: 74
Name: Suchitra
Salary: 833333.3333
Address: Vijayawada

Id: 123
Name: Likhitha
Salary: 833333.3333
Address: Vuyyuru

Id: 169
Name: Manasa
Salary: 833333.3333
Address: Vijayawada

Id: 4
Name: Akanksha
Salary: 833333.3333
Address: Guntur
```

2)

hibernate.cfg.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
        "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
        "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<!-- related to connection start -->
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
```

```
<property name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
<property name="connection.user"></property>
<property name="connection.password"></property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<property name="hbm2ddl.auto">update</property>
<property name="show_sql">true</property>
<mapping resource="OnlineCust.hbm.xml"></mapping>
<mapping resource="OnlineOrder.hbm.xml"></mapping>
</session-factory>
</hibernate-configuration>
```

OnlineCust.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="OnlineCust" table="onlinecust">
<id name="id" column="id"/>
<property name="fname" column="fname"/>
<property name="lname" column="lname"/>
<property name="city" column="city"/>
<property name="country" column="country"/>
<property name="phn" column="phn"/>
<set name="children" cascade="all" lazy="false">
<key column="cid"/>
<one-to-many class="OnlineOrder"/>
</set>
```

```
</class>
</hibernate-mapping>
```

OnlineOrder.hbm.xml

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="OnlineOrder" table="onlineorder">
<id name="oid" column="oid"/>
<property name="orderdate" column="orderdate"/>
<property name="ordernumber" column="ordernumber"/>
<property name="amount" column="amount"/>
<property name="cid" column="cid"  insert="false" />
</class>
</hibernate-mapping>
```

OnlineCust.java

```
import java.util.Set;
public class OnlineCust {
        private int id,phn;
        private String fname,lname,city,country;
        private Set children;
        public int getId() {
```

```
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
	public int getPhn() {
		return phn;
	}
	public void setPhn(int phn) {
		this.phn = phn;
	}
	public String getFname() {
		return fname;
	}
	public void setFname(String fname) {
		this.fname = fname;
	}
	public String getLname() {
		return lname;
	}
	public void setLname(String lname) {
		this.lname = lname;
	}
	public String getCity() {
		return city;
	}
	public void setCity(String city) {
		this.city = city;
	}
	public String getCountry() {
		return country;
```

```
	}
	public void setCountry(String country) {
		this.country = country;
	}
	public Set getChildren() {
		return children;
	}
	public void setChildren(Set children) {
		this.children = children;
	}
}
```

OnlineOrder.java

```
public class OnlineOrder {
	private int oid,ordernumber,cid;
	private String orderdate;
	private double amount;
	public int getOid() {
		return oid;
	}
	public void setOid(int oid) {
		this.oid = oid;
	}
	public int getOrdernumber() {
		return ordernumber;
	}
	public void setOrdernumber(int ordernumber) {
```

```
		this.ordernumber = ordernumber;
	}
	public int getCid() {
		return cid;
	}
	public void setCid(int cid) {
		this.cid = cid;
	}
	public String getOrderdate() {
		return orderdate;
	}
	public void setOrderdate(String orderdate) {
		this.orderdate = orderdate;
	}
	public double getAmount() {
		return amount;
	}
	public void setAmount(double amount) {
		this.amount = amount;
	}
}
```

shopping.java

```
import java.util.Iterator;
import java.util.List;
import org.hibernate.Query;
import org.hibernate.Session;
```

```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class shopping {
	public static void main(String args[])
	{
		Configuration cfg = new Configuration();
		cfg.configure("hibernate.cfg.xml");
		SessionFactory factory = cfg.buildSessionFactory();
		Session session = factory.openSession();
		Query qry= session.createQuery("select c.amount,c.ordernumber,o.fname,
o.lname,o.city,o.country from OnlineCust o Left Join o.children c where o.id=c.cid");
		List l = qry.list();
		Iterator it=l.iterator();
		while(it.hasNext())
		{
			Object rows[] = (Object[])it.next();
			System.out.println(rows[0]+ "   " +rows[1]+ "   " +rows[2]+ "   " +rows[3]+ "
" +rows[4]+ "   " +rows[5]);
		}

	}
}
```

Week 9
Hibernate Inheritance
---

1)

Payment.java:

```
public class Payment{
private int paymentId;
private double amount;
public int getPaymentId() {
return paymentId;
}
public void setPaymentId(int paymentId) {
this.paymentId = paymentId;
}
public double getAmount() {
return amount;
}
public void setAmount(double amount) {
this.amount = amount;
}
}
```

CreditCard.java:-

```
public class CreditCard extends Payment{
private String CreditCardType;
public String getCreditCardType() {
return CreditCardType;
}
public void setCreditCardType(String creditCardType) {
CreditCardType = creditCardType;
}
}
```

Cheque.java:-

```
public class Cheque extends Payment{
private String ChequeType;
public String getChequeType() {
return ChequeType;
}
public void setChequeType(String chequeType) {
ChequeType = chequeType;
}
}
```

Customer.java:-

```
import org.hibernate.*;
import org.hibernate.cfg.*;
public class Customer {
 public static void main(String[] args)
 {
 Configuration cfg = new Configuration();
 cfg.configure("hibernate.cfg.xml");
 SessionFactory factory = cfg.buildSessionFactory();
 Session session = factory.openSession();
 CreditCard c=new CreditCard();
 c.setPaymentId(10);
 c.setAmount(2500);
 c.setCreditCardType("VisaCard");
 Cheque c1=new Cheque();
 c1.setPaymentId(11);
 c1.setAmount(2600);
 c1.setChequeType("ICICI Bank");
 Transaction tx = session.beginTransaction();
 session.save(c);
session.save(c1);
 tx.commit();
 session.close();
 factory.close();
 }
}
```

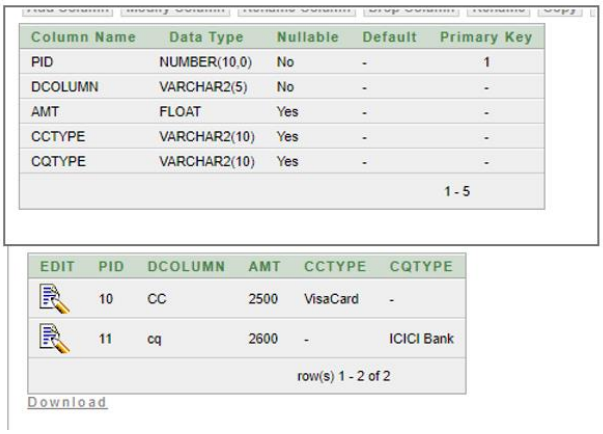
Payment.hbm.xml:-

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="str.Payment" table="PAYMENT">
<id name="paymentId" column="pid" />
<discriminator column="dcolumn" type="string" length="5"/>
<property name="amount" column="amt" />
<subclass name="str.CreditCard" discriminator-value="CC">
<property name="CreditCardType" column="cctype" length="10" />
</subclass>
<subclass name="str.Cheque" discriminator-value="cq">
<property name="ChequeType" column="cqtype" length="10" />
</subclass>
</class>
</hibernate-mapping>
```

Hibernate.cfg.xml:-

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver
</property>
<property name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
<property name="connection.username">system</property>
<property name="connection.password">manager</property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<property name="show_sql">true</property>
<property name="hbm2ddl">auto</property>
<mapping resource="Payment.hbm.xml" />
</session-factory>
</hibernate-configuration>
```

OUTPUTS:



Column Name	Data Type	Nullable	Default	Primary Key
PID	NUMBER(10,0)	No	-	1
DCOLUMN	VARCHAR2(5)	No	-	-
AMT	FLOAT	Yes	-	-
CCTYPE	VARCHAR2(10)	Yes	-	-
CQTYPE	VARCHAR2(10)	Yes	-	-

1 - 5

EDIT	PID	DCOLUMN	AMT	CCTYPE	CQTYPE
	10	CC	2500	VisaCard	-
	11	cq	2600	-	ICICI Bank

row(s) 1 - 2 of 2

Download

---

2)

Employee.java:-

```
public class Employee {
private int id;
private String name;
public int getId() {
return id;
}
public void setId(int id) {
```

```
this.id = id;
}
public String getName() {
return name;
}
public void setName(String name) {
this.name = name;
}
}
```

Contract_employee.java:-

```
public class Contract_Employee extends Employee{
 private float pay_per_hour;
 private String contract_duration;
public float getPay_per_hour() {
return pay_per_hour;
}
public void setPay_per_hour(float pay_per_hour) {
this.pay_per_hour = pay_per_hour;
}
public String getContract_duration() {
return contract_duration;
}
public void setContract_duration(String contract_duration) {
this.contract_duration = contract_duration;
}
}
```

Regular_employee.java:-

```
public class Regular_Employee extends Employee{
private float salary;
private int bonus;
public float getSalary() {
return salary;
}
public void setSalary(float salary) {
this.salary = salary;
}
public int getBonus() {
return bonus;
}
public void setBonus(int bonus) {
this.bonus = bonus;
}
}
```

StoreData.java:-

```
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
public class StoreData {
public static void main(String[] args) {
StandardServiceRegistry ssr=new
StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();
Metadata meta=new MetadataSources(ssr).getMetadataBuilder().build();
SessionFactory factory=meta.getSessionFactoryBuilder().build();
Session session=factory.openSession();
Transaction t=session.beginTransaction();
Employee e1=new Employee();
 e1.setName("Gaurav Chawla");

 Regular_Employee e2=new Regular_Employee();
 e2.setName("Vivek Kumar");
 e2.setSalary(50000);
 e2.setBonus(5);

 Contract_Employee e3=new Contract_Employee();
 e3.setName("Arjun Kumar");
 e3.setPay_per_hour(1000);
 e3.setContract_duration("15 hours");

 session.persist(e1);
 session.persist(e2);
 session.persist(e3);

 t.commit();
 session.close();
 System.out.println("success");
}
}
```

Employee.hbm.xml:-

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-mapping PUBLIC
 "-//Hibernate/Hibernate Mapping DTD 5.3//EN"
 "http://hibernate.sourceforge.net/hibernate-mapping-5.3.dtd">
 <hibernate-mapping>
 <class name="com.minik.Employee" table="emp122">
```

```
<id name="id">
<generator class="increment"></generator>
</id>

<property name="name"></property>

<union-subclass name="com.minik.Regular_Employee" table="regemp122">
<property name="salary"></property>
<property name="bonus"></property>
</union-subclass>

<union-subclass name="com.minik.Contract_Employee" table="contemp122">
<property name="pay_per_hour"></property>
<property name="contract_duration"></property>
</union-subclass>
</class>
</hibernate-mapping>
```

Hibernate.cfg.xml:-

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
 "-//Hibernate/Hibernate Configuration DTD 5.3//EN"
 "http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">
<hibernate-configuration>
 <session-factory>
 <property name="hbm2ddl">auto </property>
 <property name="dialect">org.hibernate.dialect.Oracle9Dialect</property>
 <property name="connection.url">jdbc:oracle:thin:@localhost:1521:xe</property>
 <property name="connection.username">system</property>
 <property name="connection.password">manager</property>
 <property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
 <mapping resource="employee.hbm.xml"/>
 </session-factory>
</hibernate-configuration>
```

OUTPUTS:

Table name: Contemp122

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	1
NAME	VARCHAR2(255)	Yes	-	-
PAY_PER_HOUR	FLOAT	Yes	-	-
CONTRACT_DURATION	VARCHAR2(255)	Yes	-	-

1 - 4

EDIT	ID	NAME	PAY_PER_HOUR	CONTRACT_DURATION
	3	karthik	1000	15 hours

row(s) 1 - 1 of 1

Table name: Emp122

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	1
NAME	VARCHAR2(255)	Yes	-	-
				1 - 2

EDIT	ID	NAME
	1	baradwaz
row(s) 1 - 1 of 1		

Table name: Regemp122

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	1
NAME	VARCHAR2(255)	Yes	-	-
SALARY	FLOAT	Yes	-	-
BONUS	NUMBER(10,0)	Yes	-	-
				1 - 4

EDIT	ID	NAME	SALARY	BONUS
	2	bharath	50000	5
row(s) 1 - 1 of 1				

Table name: HT_contemp(no data)

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	-
				1 - 1

Table name: HT_ emp122(no data)

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	-
				1 - 1

Table name: HT_regemp122(no data)

Column Name	Data Type	Nullable	Default	Primary Key
ID	NUMBER(10,0)	No	-	-
				1 - 1

---

3)

Payment.java:-

```
public class Payment{
private int paymentId;
private double amount;
public int getPaymentId() {
return paymentId;
}
public void setPaymentId(int paymentId) {
this.paymentId = paymentId;
}
```

```
public double getAmount() {
return amount;
}
public void setAmount(double amount) {
this.amount = amount;
}
}
```

CreditCard.java:-

```
public class CreditCard extends Payment{
private String CreditCardType;
public String getCreditCardType() {
return CreditCardType;
}
public void setCreditCardType(String creditCardType) {
CreditCardType = creditCardType;
}
}
```

Cheque.java:-

```
public class Cheque extends Payment{
private String ChequeType;
public String getChequeType() {
return ChequeType;
}
public void setChequeType(String chequeType) {
ChequeType = chequeType;
}
}
```

Customer.java:-

```
import org.hibernate.*;
import org.hibernate.cfg.*;
public class Customer {
 public static void main(String[] args)
 {
 Configuration cfg = new Configuration();
 cfg.configure("hibernate.cfg.xml");
 SessionFactory factory = cfg.buildSessionFactory();
 Session session = factory.openSession();
 CreditCard c=new CreditCard();
 c.setPaymentId(10);
 c.setAmount(2500);
 c.setCreditCardType("VisaCard");
 Cheque c1=new Cheque();
```

```
 c1.setPaymentId(11);
 c1.setAmount(2600);
 c1.setChequeType("ICICI Bank");
 Transaction tx = session.beginTransaction();
 session.save(c);
 session.save(c1);
 tx.commit();
 session.close();
 factory.close();
 }
}
```

Payment.hbm.xml:-

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC
"-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
<class name="str.Payment" table="PAYMENT">
<id name="paymentId" column="pid" />
<property name="amount" column="amt" />
<joined-subclass name="str.CreditCard" table="CreditCard">
<key column="dummy1" />
<property name="CreditCardType" column="cctype" length="10" />
</joined-subclass>
<joined-subclass name="str.Cheque" table="Cheque">
<key column="dummy2" />
<property name="ChequeType" column="cqtype" length="10" />
</joined-subclass>
</class>
</hibernate-mapping>
```

Hibernate.cfg.xml:-

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration PUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory>
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver
</property>
<property name="connection.url">jdbc:oracle:thin:@localhost:1521:XE</property>
<property name="connection.username">system</property>
<property name="connection.password">manager</property>
<property name="dialect">org.hibernate.dialect.OracleDialect</property>
<property name="show_sql">true</property>
```

<property name="hbm2ddl">auto</property>
<mapping resource="Payment.hbm.xml" />
</session-factory>
</hibernate-configuration>

OUTPUTS:
Table name: cheque

Column Name	Data Type	Nullable	Default	Primary Key
DUMMY2	NUMBER(10,0)	No	-	1
CQTYPE	VARCHAR2(10)	Yes	-	-
				1 - 2

EDIT	DUMMY2	CQTYPE
	11	ICICI Bank
	row(s) 1 - 1 of 1	

Table name: credit card

Column Name	Data Type	Nullable	Default	Primary Key
DUMMY1	NUMBER(10,0)	No	-	1
CCTYPE	VARCHAR2(10)	Yes	-	-
				1 - 2

EDIT	DUMMY1	CCTYPE
	10	VisaCard
	row(s) 1 - 1 of 1	

Table name: payment

Column Name	Data Type	Nullable	Default	Primary Key
PID	NUMBER(10,0)	No	-	1
AMT	FLOAT	Yes	-	-
				1 - 2

EDIT	PID	AMT
	10	2500
	11	2600
	row(s) 1 - 2 of 2	

Table name:HT_payment(no data)

Column Name	Data Type	Nullable	Default	Primary Key
PID	NUMBER(10,0)	No	-	-
				1 - 1

Table name: HT_cheque(no data)

Column Name	Data Type	Nullable	Default	Primary Key
DUMMY2	NUMBER(10,0)	No	-	-
				1 - 1

Table name: HT_creditcard(no data)

Column Name	Data Type	Nullable	Default	Primary Key
DUMMY1	NUMBER(10,0)	No	-	-
				1 - 1