**Practical 7 – Creating EJB application using Singleton Session**

a) **Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans.**

**Index.html**

```
<html>
 <head>
 <title>TODO supply a title</title>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <meta http-equiv="Refresh" content="0; URL=ServletClient">
 </head>
 <body>
 <div>TODO write content</div>
 </body>
</html>
```

**ServletClient.java**

```
package servlet;
import java.io.IOException;
import java.io.PrintWriter;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import ejb.CountServletHitsBean;
import jakarta.ejb.EJB;
@WebServlet(name = "ServletClient", urlPatterns = {"/ServletClient"})
public class ServletClient extends HttpServlet {
@EJB CountServletHitsBean obj;
@Override
protected void service (HttpServletRequest req, HttpServletResponse res) throws ServletException,
IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
out.print("<b>Number of times this Servlet is accessed </b>: "+obj.getCount());
}
}
```

**CountServletHitsBean.java**

```
Package ejb;
import jakarta.ejb.Singleton;
import Jakarta.ejb.LocalBean;
@Singleton
@LocalBean
public class CountServletHitsBean {
privateinthitCount;
public synchronized int getCount()
{
return hitCount++;
}
}
```

**b) Develop simple Marks Entry Application to demonstrate accessing Database using EJB.**

create table marks (id int primary key auto_increment, sname varchar(35), marks1 int, marks2 int, marks3 int);

**index.jsp**

```
<%@page import="ejb.MarksEntryBean"%>
<%@page import="javax.naming.InitialContext"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
private static MarksEntryBean obj;
public void jspInit()
{
try
{
InitialContext ic=new InitialContext();
obj=(MarksEntryBean)ic.lookup("java:global/Demo7C-1.0-SNAPSHOT/MarksEntryBean");
}
catch(Exception e)
{
System.out.println(e);
}
}
%>
<%
if(request.getParameter("InsertMarks")!=null)
{
String sname;
int marks1, marks2, marks3;
sname = request.getParameter("sname");
marks1=Integer.parseInt(request.getParameter("m1"));
marks2=Integer.parseInt(request.getParameter("m2"));
marks3=Integer.parseInt(request.getParameter("m3"));
obj.addMarks(sname,marks1,marks2,marks3);
out.print("Marks entered successfully..!!!!");
}
%>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<h2>Enter Details</h2>
<form name="result" method="post">
Enter student's name: <input type='text' name="sname" /><br>
Enter subject 1 marks: <input type='text' name="m1" /><br>
Enter subject 2 marks: <input type='text' name="m2" /><br>
Enter subject 3 marks: <input type='text' name="m3" /><br>
<input type='submit' name="InsertMarks" /><br>
</form>
</body>
</html>
```

**MarksEntryBean.java**

```
package ejb;
import java.sql.*;
import jakarta.ejb.Stateful;
@Stateful
public class MarksEntryBean {
String sname;
int m1,m2,m3;
Connection con=null;
Statement st=null;
String query="";
public void addMarks(String sname,int m1,int m2,int m3)
{
try
{
Class.forName("com.mysql.cj.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb", "root","tiger");
st=con.createStatement();
query="insert into marks (sname,marks1,marks2,marks3) values
('"+sname+"','"+m1+"','"+m2+"','"+m3+"')";
st.executeUpdate(query);
System.out.print("Marks entered sucessfully!!");
}
catch(Exception e){System.out.println(e);}
}
}
```

**c) Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean].**

Step 1:

Web-> web application -> <Your Rollno>_7C -> select dedicated folders for storing libraries -> finish.

**Index.jsp**

```
<%@page import="javax.jms.JMSException"%>
<%@page import="javax.naming.InitialContext"%>
<%@page import="javax.jms.Connection"%>
<%@page import="javax.jms.TextMessage"%>
<%@page import="javax.jms.MessageProducer"%>
<%@page import="javax.jms.Session"%>
<%@page import="javax.jms.Queue"%>
<%@page import="javax.jms.ConnectionFactory"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%!
private static ConnectionFactory connectionFactory;
private static Queue queue;
Connection connection=null;
Session mySession=null;
MessageProducer messageProducer=null;
TextMessage message=null;
%>
<html>
 <head>
 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
 <title>JSP Page</title>
 </head>
 <body>
 Welcome to My Home Page
 <%
 try{
 InitialContext ic= new InitialContext();
 queue= (Queue)ic.lookup("jms/Queue");
 connectionFactory=(ConnectionFactory)ic.lookup("jms/QueueFactory");
  connection= connectionFactory.createConnection();
 mySession=connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
 messageProducer=mySession.createProducer(queue);
 message=mySession.createTextMessage();
 message.setText(request.getRemoteAddr());
 messageProducer.send(message);
 }
 catch(JMSException e)
 {
 System.out.println("Exception Occoured "+e.toString());
 }
```

```
%>
</body>
</html>
```

Step 2
Open your database logindb
Create table name -> userstat
column names are
Firstvisitdt – timestamp
Hostname – varchar 30 Primary Key
Visits – int

Step 3
Create a Session Bean named as VisitorStatBean -> Select Stateless -> package name as ejb, do not select Local / Remote

**VisitorStatBean.java**

```
package ejb;
import java.sql.*;
import javax.annotation.PostConstruct;
import javax.annotation.PreDestroy;
import javax.ejb.MessageDriven;
import javax.ejb.Stateless;
@Stateless
@MessageDriven
public class VisitorStatBean {
private Connection conn=null;
private ResultSet rs;
private Statement st=null;
private String query =null;
@PostConstruct
public void connect()
{
 try {
 Class.forName("com.mysql.cj.jdbc.Driver").newInstance();
 conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/logindb", "root", "tiger");
 }
 catch (Exception e) {
 System.err.println(e.getMessage());
 }
}
@PreDestroy
public void disconnect()
{
 try {
 conn.close();
 } catch (Exception e) {
```

```
 System.err.println(e.getMessage());
 }
}
public void addVisitor(String host)
{
 try {
 st= conn.createStatement();
query="insert into userstat (hostname,visits) values ('"+host+"','1')";
st.executeUpdate(query);
 }
 catch (SQLException e)
 {
 try {
 st=conn.createStatement();
query="update userstat set visits=visits+1 where hostname='"+host+"' ";
st.executeUpdate(query);

 }
 catch (SQLException ex) {
 System.err.println("Cannot Update"+e.getMessage());
 }
 }
}
}
```

Step 4

Right click on Source Packages -> Select New ->Other ->Enterprise Java Bean -> MessageDrivenBean ->EJB Name: BasicMessageBean -> Package: ejb -> Select Project Destination -> Click on Add Button -> Destination Name: jms/Queue -> Destination Type select the option Queue->click on OK-> Click on Next ->Activation Configuration Properties should be as it is. -> Click on Finish

**BasicMessageBean.java**

```
package ejb;
import javax.annotation.Resource;
import javax.ejb.ActivationConfigProperty;
import javax.ejb.EJB;
import javax.ejb.MessageDriven;
import javax.ejb.MessageDrivenContext;
import javax.jms.JMSException;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.TextMessage;
@MessageDriven(activationConfig = {
 @ActivationConfigProperty(propertyName = "destinationLookup", propertyValue = "jms/Queue"),
 @ActivationConfigProperty(propertyName = "destinationType", propertyValue =
"javax.jms.Queue")
})
public class BasicMessageBean implements MessageListener {
```

```
@EJB VisitorStatBean vs;
@Resource
private MessageDrivenContext mdc;
 public BasicMessageBean() {
 }

 @Override
 public void onMessage(Message message) {
 try {
 if(message instanceof TextMessage){
 TextMessage msg= (TextMessage) message;
 vs.addVisitor(msg.getText());
 }
 }
  catch (JMSException e) {
 mdc.setRollbackOnly();
 }
 }
}
```

Step 5: Before deploying and running the application, Glassfish Server setting is required.
Browse the path: Localhost:4848 on any browser.
Find Resources -> connectors -> Connector Resources
double click on Connector Resources -> click on 'New' Button -> write JNDI name as -> jms/QueryFactory. Find Admin Object Resources and double click on that -> click on 'New' Button -> write JNDI name as -> jms/Queue.

Step 6: run index.jsp file.