//Q.1 java A) index.html

<html>

<body>

<form action="slip19.jsp" method="post">

<input type="text" name="username">

<input type="submit">

</form>

</body>

</html>

//jsp file

<html>

<body>

<%

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

java.util.Date d=new java.util.Date();

int hr=d.getHours();

if(hr<12)

{

out.println("Good Morning:"+name);

}

if(hr>12 && hr<16)

{

out.println("Good Afternoon:"+name);

}

if(hr>16)

{

out.println("Good Evening:"+name);

}

%>

</body>

</html>

//Q.1 B)

import java.sql.\*;

public class type {

public static void main(String args[]) throws Exception {

Class.forName("com.mysql.cj.jdbc.Driver");

Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/student", "root", "8623989585");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery("select \* from student");

while (rs.next()) {

String rno = rs.getString("rno");

String name = rs.getString("name");

int per = rs.getInt("per");

System.out.println(rno + " " + name + " " + per);

}

con.close();

}

}

//Q.2 A)

Q2 .net palindrom

Private Sub CheckPalindrome(ByVal strString As String)

Dim str As String

str = StrReverse(strString)

If str.Equals(strString) Then

MessageBox.show("This string is Palindrome.")

Else

MessageBox.show("This string is not Palindrome.")

End If

End Sub

B)

using System;

namespace feedback

{

public partial class feedbackselect : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void btnvote\_Click(object sender, EventArgs e)

{

if (rdogood.Checked == true)

{

int goodCount;

if (ViewState["gcount"] != null)

goodCount = Convert.ToInt32(ViewState["gcount"]) + 1;

else

goodCount = 1;

ViewState["gcount"] = goodCount;

}

if (rdosatisfactory.Checked == true)

{

int satisfactoryCount;

if (ViewState["scount"] != null)

satisfactoryCount = Convert.ToInt32(ViewState["scount"]) + 1;

else

satisfactoryCount = 1;

ViewState["scount"] = satisfactoryCount;

}

if (rdobad.Checked == true)

{

int badCount;

if (ViewState["bcount"] != null)

badCount = Convert.ToInt32(ViewState["bcount"]) +

1; else

badCount = 1;

ViewState["bcount"] = badCount;

}

int totalCount;

if (ViewState["count"] != null)

totalCount = Convert.ToInt32(ViewState["count"]) +

1; else

totalCount = 1;

ViewState["count"] = totalCount;

double gper = (Convert.ToDouble(ViewState["gcount"]) /

Convert.ToDouble(ViewState["count"])) \* 100.0f;

lblgood.Text = gper.ToString() + "%";

double sper = (Convert.ToDouble(ViewState["scount"])

/ Convert.ToDouble(ViewState["count"])) \* 100.0f;

lblsatisfactory.Text = sper.ToString() + "%";

double bper = (Convert.ToDouble(ViewState["bcount"]) /

Convert.ToDouble(ViewState["count"])) \* 100.0f;

lblbad.Text = bper.ToString()+"%";

} } }