**Tables required:-**

-**Members**

-**Emp\_LogTime**

**Fields required:-**

-**id**: primary key of the table

-**emp\_id**: id of the employee

-**SID**: **Session.SessionID** (for session being opened on the site or portal when an employee logs in)

-**Login\_Time**: time of login

-**Logout\_Time**: time of logout

-**online**: boolean repesentation to tell whether employee is online or offline

**Procedure:**

When the employee logs in, a query will be generated and sent to employee **id** value from the **Members** table.

This value will be stored in a session variable.

**Session(“member”)=rs(“Id”)**

Now we have the employee **id** value in the **Session(“member”)** variable.

Now we have to change all the records in the **User\_LogTime** table for this user that have the **online** field set as True (which will initialy be false).

Now we will insert a new record in the **Emp\_LoginTime** table with the **id** of the employee, the **SessionID** of this employee and with the time of the login:

query=”UPDATE Emp\_LogTime SET online=True WHERE online=False AND emp\_id=” &\_Session(“member”)conn.Execute(query)

Now, insert a new record in the **Emp\_LogTime** table with the **emp\_id**, **SessionID** & the **Login\_Time**:

query=”INSERT INTO Emp\_LogTime (emp\_id, SID, Login\_Time)”

query=query &\_”VALUES(“& Session(“member”) & “,”,&\_Session,SessionID & “,#”

& now () & “#)” conn.Execute(query)

The code above is called when the session began and you got the login time and wrote the same in the database.

When the employee clicks logout the **Session.Abandon** will be called.

If the employee quits the page without logging out, then after the **Session.Timeout** expires **Session\_OnEnd** will be executed.

In this procedure **Logout\_Time** is updated and **online** field is set to False:

query=”UPDATE Emp\_LogTime SET Logout\_Time=#” & now() & “#, online=False