

**globsyn**   
*Taking People To The Next Level*

## APPRETIARE: AN INTRANET APPRAISAL MANAGEMENT SYSTEM

---

### ***Group Members:***

PRINCE KUMAR SINGH, JIS COLLEGE OF ENGINEERING, 111230110182

MD.ARSHAD FEEROZ, GURU NANAK INSTITUTE OF TECHNOLOGY, 111430110053

SURAJDEV KUMAR PANDEY, GURU NANAK INSTITUTE OF TECHNOLOGY, 111430110118

SUMAN KUMAR MAITY, GURU NANAK INSTITUTE OF TECHNOLOGY, 111430110114

## Table of Contents

Acknowledgement .....	4
Project Objective .....	8
Project Scope.....	9
Requirement Specification.....	10
Database Design.....	12
Application Work Flow and ER diagram .....	16-17
Screenshots .....	18-30
Future Scope of Improvements.....	31
Code .....	32-51
Certificate.....	52-57

# Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty ( Mr .*Suman Aich* ) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

(*Prince Kumar Singh*)



# Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty ( Mr. *Suman Aich* ) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

*(Md.Arshad Feeroz)*



# Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty (Mr. *Suman Aich* ) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

*(Surajdev kumar pandey)*



# Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty (Mr. *Suman Aich* ) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

*(Suman Kumar Maity)*



## **Project Objective:**

Apprentice Appraisal management system is an intranet based Employees Performance, Appraisal and Recognition system used by all employees in different sections of the company. Salary hike and promotion depends upon the employees' performance. This portal is a one stop shop for all the employees to provide details like tasks performed, performance measures improved etc to their superiors. It allows superiors to evaluate and analyze their subordinate employees' performance and work done by them and target achieved in a given period of time and to rate them.

It provides a very good interface between superiors and subordinates. Based on these ratings and overall performance shown by employees ranks will be allotted to them among the group of employees with same designation. This rank is called Team Lead (project) rank. This is the basic criteria for recognizing employee's performance and to provide salary hike/promotion to any employee.

This application maintains the entire data in a centralized (as well as decentralized) and secured database server to maintain consistency in report generation and allows users to access from any location. This is an online application that allows multi user access of system and to track or manage the data simultaneously. Various roles and authentications have been provided and access to various areas in the tool is restricted according to the role given to users.

The aim of this application is to reduce the manual effort needed to manage the details of tasks and performance measures of each and every employee which is very tedious. Also, maintaining historical data used by HR team in generating consolidated data is not possible. This portal helps them in generating consolidated rank or required reports with a single click. Also this application provides an interface to management and other users to manage the details of and to generate required reports. This helps to prevent unnecessary delays and human errors. This system helps in generating foolproof reports within no time by users which is very difficult in current system (manual system).

This system design is modularized into various categories. This system has enriched UI so that a novice user does not feel any operational difficulties. This system mainly concentrates in designing various reports requested by the users as well as higher with export to excel options.

### **Core Entities of this application:**

- General Employees.
- Senior Employees (Project Lead/Team Lead).
- Admin.

### **General Employees**

The General Employees can view their own feedback given to them by their managers and also view the final report. This module consists of login of the employee and access controls.

### **Senior Employees**

The Senior Employees will keep track and rate their subordinates on frequent intervals based on their technical performance, leadership and social behavior. The performance is measured on a scale. This module consists of Employee information (attendance, social behavior, etc) forms and Employee feedback forms from each project.

### **ADMIN**

The ADMIN will keep track of the qualities of the senior employees, register the new employee, view reports, etc. They further notice the leadership skills of the project/team leads and measure their performance on a scale.



## **Project Scope**

**Better quality data:** Getting meaningful performance data from paper-based appraisal forms is both difficult and time consuming. For example, compiling training needs across a company typically involves reading through every employee's personal development plan and identifying common needs, which can take days to complete. A good online system will produce data like this at the touch of a button.

**Streamlining the process :** A paper-based process requires physical forms to be passed around the organization between employees, managers and HR for completion and signature. Not only is this laborious for staff, trying to keep track of who is at what stage of the process is nigh-on impossible. An online system will offer automated workflows and approvals, making the process simpler and quicker for employees and managers to complete.

**Encouraging year-round performance management:** Paper-based appraisal processes are often a once or twice-a-year affair, with objectives locked in a draw and never looked at or updated between reviews. An online performance management system, if set up well, can encourage employees and managers to review performance progress regularly and ensure that objectives are updated to stay relevant throughout the year.

**Generation Y' expect everything to be online.** The 'internet generation', now making up an increasing percentage of the workforce, typically manages their lives online. Organizations wishing to present themselves as an employer of choice to these individuals will struggle if they are still using paper-based forms, or even Word documents, to manage their processes.

**Fully scalable**— handles from only 20 to thousands of employees, covering all staff and managerial levels.

**Accommodates the Performance** Cycle alternatives: 1, 2, 3, 4, 6 or 12-monthly performance appraisals, as well as employee anniversary-based and ad-hoc reviews.

**You better align individual objectives** with corporate objectives and make measuring performance simpler and more consistent.

**You get a better user experience ,** which leads to greater engagement and compliance with the program.

## Requirement Specification

- **Domain Description:** - Oracle form is a software product for creating screens that interact with an oracle database. It has an IDE including an object navigator, property sheet and code editor that uses PL/SQL. It was originally developed to run server-side in character mode terminal sessions. It was ported to other platforms, including windows, to function in a client-server environment. Later versions were ported to java where it runs in a Java EE container and can integrate with java and web services. The primary focus of forms is to create data entry systems that access an oracle database.

Oracle database (Oracle DB) is a relational database management system (RDMS) from the oracle corporation originally developed in 1977 by Lawrence Ellison and other developers. Oracle DB is one of the most trusted and widely used relational database engines. The system is built around a relational database framework in which data objects may be directly accessed by users (or an application front end) through structured query language (SQL). Oracle is fully scalable relational database architecture and is often used by global enterprises, which manage and process data across wide and local area networks. The oracle database has its own network components to allow communications across networks.

**Problem Definition:** This Intranet based application saves time which earlier required a great deal of hard work and tuff maintenance for employee appraisal work manually. Each Manager can appraise and views the reports (score) of his employees. There are various parameters based on various departments which are grouped according to the departments. Employees are rated based on the factors in the group. Based on the factors scores are given and the employee is appraised.

**Functional Requirements:** - The first step of the project is to find the functional requirement of the online appraisal management system with the help of techniques like Use Cases and User Stories.

### **Use Cases:**

The first step for the functional requirement collection is the use cases. Use cases are “a description of set of sequences of actions, including variants, that a system performs that yield an observable result of value to an actor”. They are used in order to: design system from user’s perspective, communicate system behavior in user’s term and enumerate all externally visible behavior [11].

Here are the use cases for the online appraisal management system project (there are two actors for the system: a normal user and an administrator).

There are three users in this application. Those are Admin , general employee and senior employee.

### **General Employees**

The General Employees can view their own feedback and access their own controls. This module consists of login of the employee and access controls.

### **Senior Employees**

The Senior Employees will keep track and rate their subordinates on frequent intervals based on his technical performance, leadership and social behavior. The performance is measured on a scale. This module consists of Employee information (attendance, social behavior, etc) forms and Employee feedback forms from each project.

### **ADMIN**

The ADMIN will keep track of the qualities of the senior employees, register the new employee, view reports, etc. They further notice the leadership skills of the project/team leads and measure their performance on a scale.

### **Hardware requirement**

- Client side-computer with minimum 2GB RAM, intel core2duo or above, intranet connection.
- Server side- computer or laptop with minimum 8GB RAM, intel i5 or above, high speed internet connection.

### **Software Requirement**

- Client side-windows XP or above, Mozilla Firefox 2.0.0.20.
- Server side - windows 7 or above, Oracle express edition, Oracle Forms 10g or above, Mozilla Firefox 2.0.0.20 or other similar browsers.

## Database Design:

### EMPLOYEE TABLE:

E_NO	E_NAME	E_SEX	PASSWORD	E_JOB	E_SAL	E_MGR	MARKS	D_NO
1001	Arshad	1	1000	Software Engineer	20000	1006	-	10
1002	Angelina	2	1000	Tester	19000	1006	-	10
1003	Rio	1	1000	Clerk	30000	1007	-	30
1004	Tom	1	1000	Field marketing	25500	1013	-	20
1005	Prince	1	1000	Junior Scientist	29800	1008	-	40
1006	Smith	1	1000	Technical Manager	10500	-	-	10
1007	Jack	1	1000	Accounts Manager	40000	-	-	30
1008	Jones	1	1000	Research Manager	50000	-	-	40
1009	Peter	1	1000	Data Analyst	15000	1010	-	10
1010	Jennifer	2	1000	DBA	35000	1006	-	10
1011	Layka	1	1000	Tester	19000	1006	-	10
1012	Daniel	1	1000	Software Engineer	20000	1006	-	10
1013	Ronald	1	1000	Sales Manager	45000	-	-	20
1021	Ilyviy	1	1234	Tester	1234	1006	-	10
1014	sdg	1	1000	drh	1234	1006	0	10
1015	kjbj	1	oubj	Research Manager	1234	1006	-	20
1016	kh jlj	1	1000	Junior Scientist	1234	1007	-	20
1017	k kjj jlknI	1	kbokI	Tester	12344	1006	-	20
1022	khvhj	2	12343	Technical Manager	1234	-	-	10
1023	hvkH	1	vbkhb	Technical Manager	24578	-	-	10
1024	jlvkH	1	kcvjhv kh	Tester	357	1022	-	10
1025	jkgcuJgkh	1	12234	Junior Scientist	1241	1008	-	40
1018	kjbSKDVkj	1	1234	Junior Scientist	2312	1006	-	10
1019	hvjhh	1	11234	Tester	10020	1006	-	10
1020	hvjhh	1	1000	Tester	10000	1006	-	10

## DEPARTMENT:

D_NO	D_NAME	D_LOC
10	technical	Block A
20	sales	Block B
30	accounts	Block A
40	research	Block C

## FACTORS:

F_ID	F_NAME	G_ID	F_WEIGHT
1	Communication Skills	1	2
2	Confidence	1	2
3	Dedication	1	2
4	Emp-Emp Relationship	2	2
5	Initiative	2	2
6	Dependability	2	2
7	Domain Of Knowledge	3	4
8	Technical Skills	3	4
9	Quantity And Quality	3	4
10	Marketing Skills	4	4
11	Management	4	4
12	Ego-Drive	4	4
13	Time Management	5	4
14	Competency	5	4
15	Math Skills	5	4
16	Scientific Skills	6	4
17	Creativity	6	4
18	Focus	6	4

G\_DEPT:

D_NO	G_ID
10	1
10	2
10	3
20	1
20	2
20	4
30	1
30	2
30	5
40	1
40	2
40	6

TRANS:

E_NO	F_ID	SCORE	P_ID
1001	1	4	1
1001	2	3	1
1001	3	2	1
1001	4	5	1
1001	5	3	1
1001	6	5	1
1001	7	4	1
1001	8	4	1
1001	9	3	1
1003	1	5	2
1003	2	2	2
1003	3	4	2
1003	4	5	2
1003	5	4	2
1003	6	4	2
1003	10	3	2
1003	11	2	2
1003	12	3	2
1006	8	14	3
1006	7	85	4

## GRP TABLE:

G_ID	G_NAME
1	General Skills
2	Social Skills
3	Technical Skills
4	Sales Skills
5	Accounts Skills
6	Research Skills

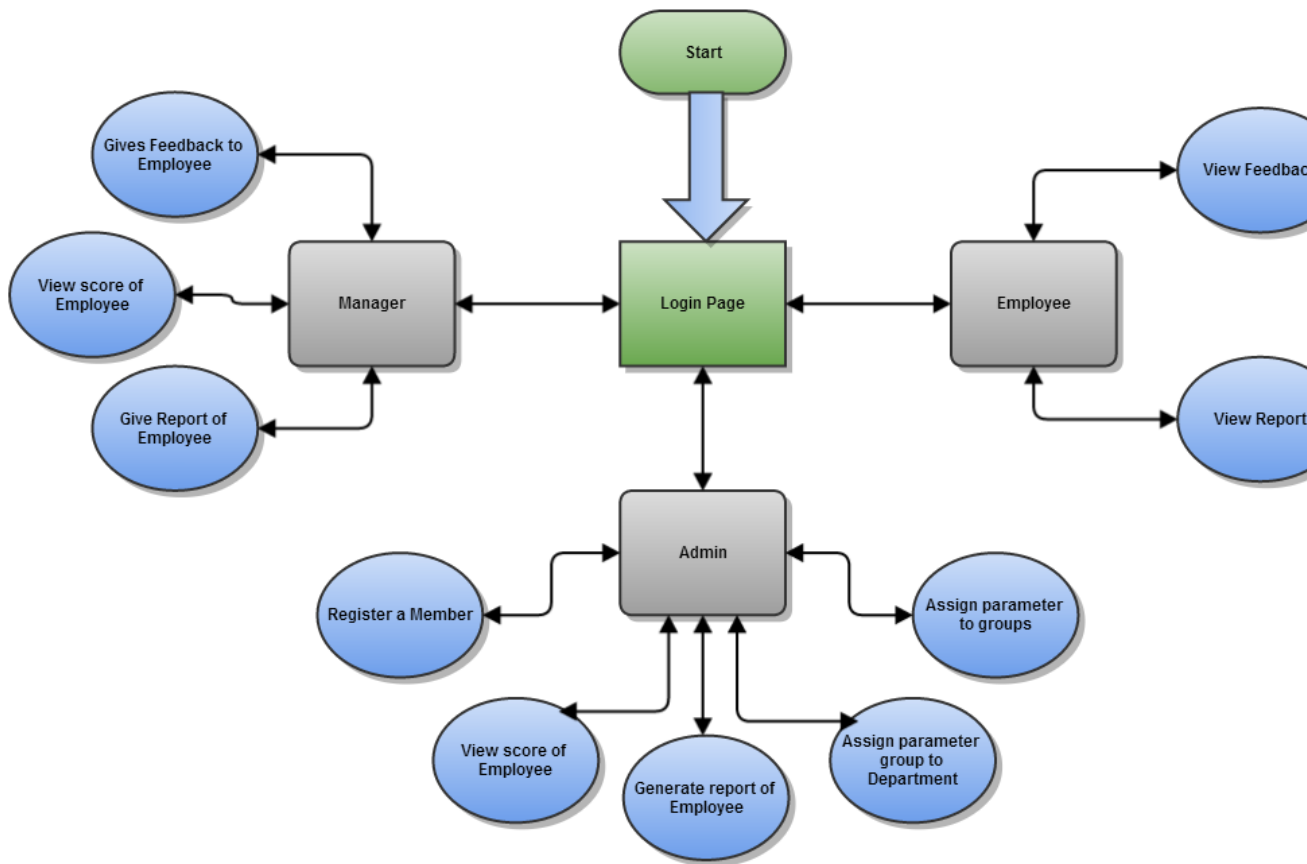
## PHASE:

P_ID	P_NAME
1	1st Quarter ,2014
2	2nd Quarter ,2014
3	3rd Quarter ,2014
4	4th Quarter ,2014

## SCORING:

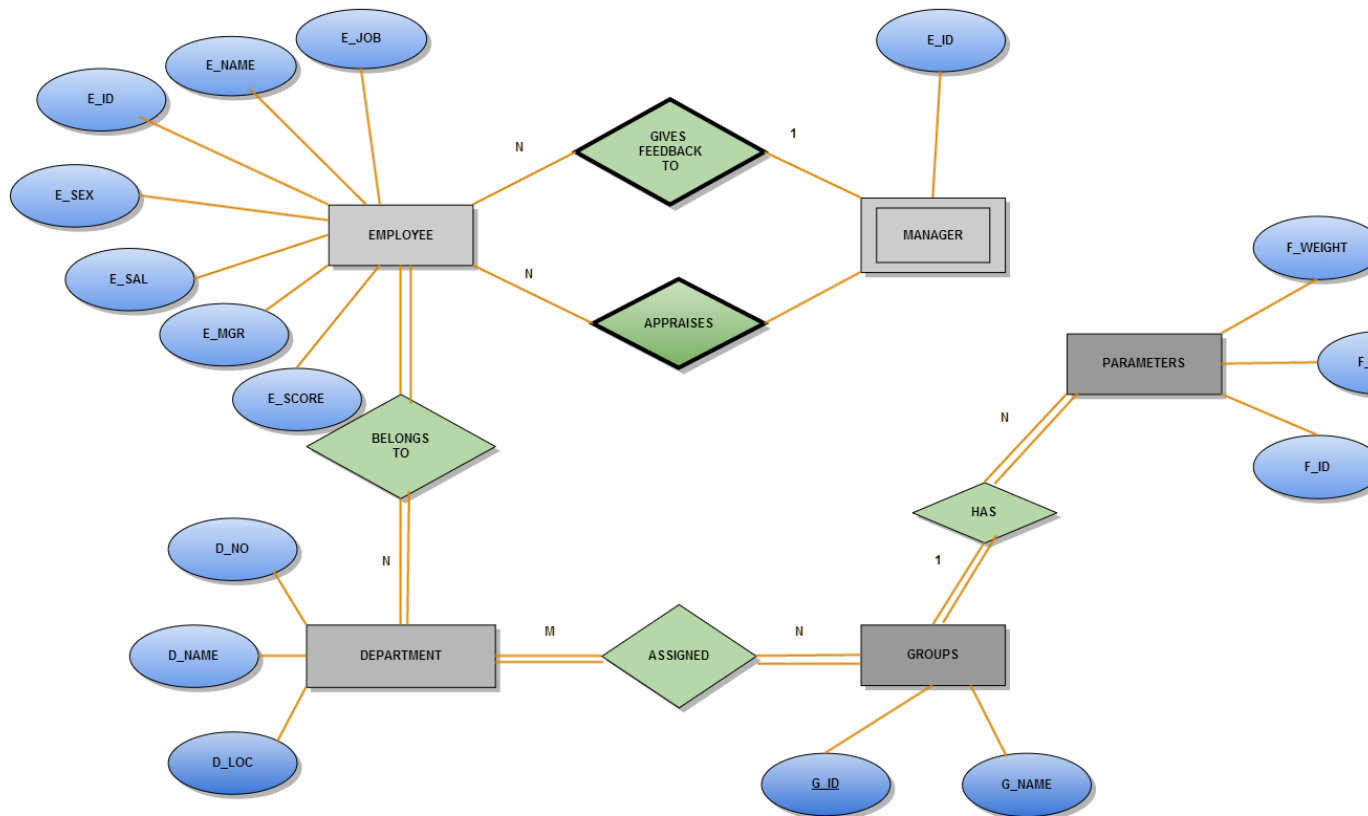
E_NO	F_ID	SCORE	F_WEIGHT	MULT
1001	1	4	2	8
1001	2	3	2	6
1001	3	2	2	4
1001	4	5	2	10
1001	5	3	2	6
1001	6	5	2	10
1001	7	4	4	16
1001	8	4	4	16
1001	9	3	4	12
1003	1	5	2	10
More than 10 rows available. Increase rows selector to view more rows.				

## Application Work Flow



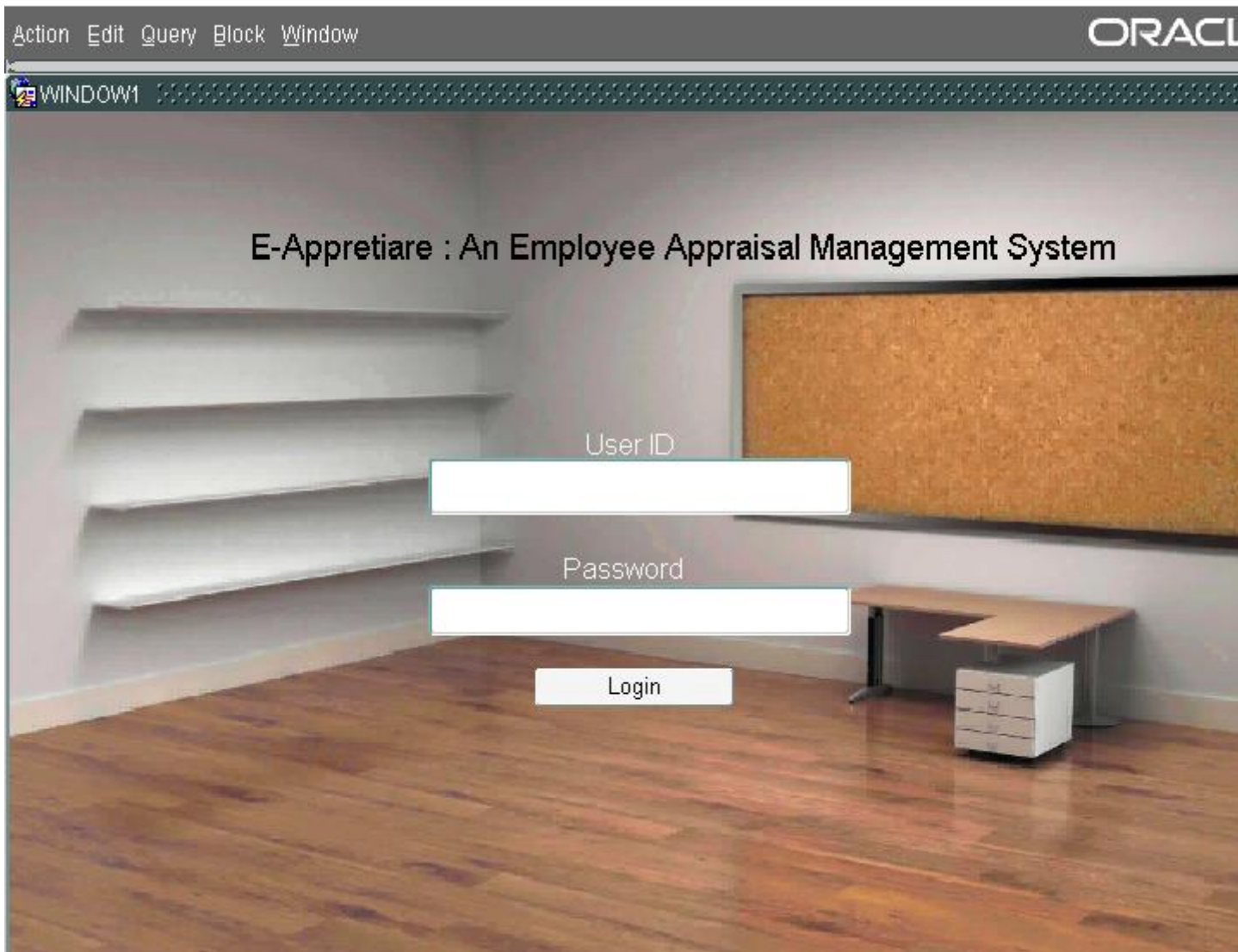


## ER DIAGRAM:



## Screenshots:

### LOGIN:



## ADMIN MASTER PAGE:

Welcome Admin

Home

Logout

Register a new Employee :

Register

View Appraisal Score of Employees :

View

Add New Parameter Groups :

Submit

Assign The Parameter Groups To Department :

Submit

Generate Final Employee Report :

Submit



## ADD A NEW EMPLOYEE

Action Edit Query Block Window

ORACLE


WINDOW1

### Add a new employee

[Home](#) [Logout](#)

Employee No 1026	Sex <input checked="" type="radio"/> Male <input type="radio"/> Female	
Name Peter Jones	Enter Password ***** Password Valid	
Department Technical department	Designation Tester	
Manager Smith	E Mgr 1006	Salary 205000

Register



## VIEW APPRAISAL SCORE:

Action

Edit

Query

Block

Window

ORACLE

WINDOW1

View Employee's Score

Home

Logout

Enter Employee No

1001

Submit

Parameter ID

Score

1

4

(Rated Out of 5)

Parameter Name

F Weight

Communication Skills

2

### ADD NEW PARAMETERS GROUP:

ActionEditQueryBlockWindow

ORACLE

WINDOW1

Add a new parameter group

HomeLogout

Group Id7

Group NameIndustry Experience

Submit

### ADD PARAMETER TO DEPARTMENT:

ActionEditQueryBlockWindow

ORACLE

WINDOW1

Assign Groups To Department

HomeLogout

Select Groups

Click to see the groups

Technical Skills

Group Id

3

Select Department

Technical Departm...

Save

WINDOW1

## View Employee's Score

[Home](#)[Logout](#)

Enter Employee No

Parameter ID

Score

(Rated Out of 5)

Parameter Name

F Weight



## MANAGER:

Action Edit Query Block Window

ORACLE

WINDOW1

### Evaluate Your employee

Home Logout

E No

Select The Phase

Select The Parameters For Evaluation

Enter the score

F Id

PUSH\_BUTTON67

## FEEDBACK FOR EMPLOYEE:

## VIEW SCORE:

Action Edit Query Block Record Field Help Window

ORACLE

WINDOW1

### View Score of Your employee

[Home](#) [Logout](#)

Enter the employee no

Parameter Id	Phase	Score
<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="4"/>

Parameter Name	Weight
<input type="text" value="Communication Skills"/>	<input type="text" value="2"/>

## REPORT:

Action Edit Query Block Window

ORACLE

WINDOW1

### Generate Employee Report

[Home](#) [Logout](#)

Enter Employee No

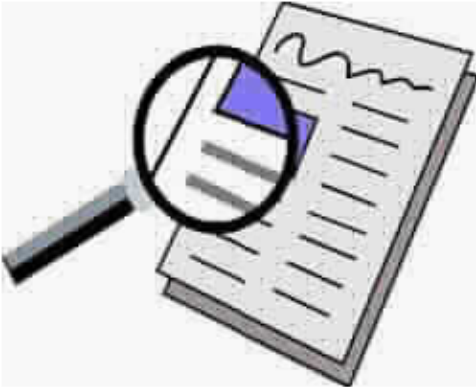
1003

Parameter Id	Score	Weight	Points
1	5	2	10
2	2	2	4
3	4	2	8
4	5	2	10
5	4	2	8

Total Points  
80


Total Credits  
24

Net Score  
66.666666666666



## EMPLOYEE:

ActionEditQueryBlockRecordFieldHelpWindow



ORACLE

Welcome Arshad

HomeLogout

My Profile

E No	E Name	D No
1001	Arshad	10
E Sal	E Job	E Mgr
20000	Software Engineer	1006

View own feedback :

Click

Generate my final report :

Click

## EMPLOYEE VIEW SCORE:

Action Edit Query Block Record Field Help Window ORACLE

View own feedback

Home Log

View Feedback

Parameter Id

Score  (Rated out of 5)

F Name

F Weight

Phase

## EMPLOYEE REPORT GENERATES:

ActionEditQueryBlockWindow

ORACLE

WINDOW1

Generate Employee Report

HomeLogout

Enter Employee No

1003

Submit

Parameter Id	Score	Weight	Points
1	5	2	10
2	2	2	4
3	4	2	8
4	5	2	10
5	4	2	8

Total Points


80

Total Credits

24

Net Score

66.66666666666666



## **Future Scope of Improvements**

- We can make the application more secure and user-friendly.
- We can add the concept of 360 degree feedback .
- Add the concept of location.
- Add more multimedia items.
- Add the concept of advanced search directly.
- Make the password encrypt in database.
- Add the concept of email notification.
- Add the concept of salary hike, promotion hike, warning, etc.
- Make the application more users friendly.

# Code:

## LOGIN

\*\*\*\*\*login form -> triggers -> keyexeqry

```
message('not allow');  
open_form('c:\final\login.fmx');
```

\*\*\*\*\*BLOCK6 -> LOGINBUTTON -> TRIGGER -> WHEN BUTTON PRESSED

```
if(:e_no=999 and :password='999') then  
    open_form('c:\final\adminmaster.fmx');
```

ELSE

DECLARE

DESIG VARCHAR2(30);

BEGIN

```
    SELECT E_NO  
    INTO :GLOBAL.USERNAME  
    FROM EMPLOYEE  
    WHERE E_NO = :E_NO  
    AND  
    PASSWORD =:PASSWORD;
```

```
    if :E_NO is null then  
        message('User Name must be entered !');  
        message('User Name must be entered !');  
        go_item('employee.e_no');  
        return;  
        RAISE Form_Trigger_Failure;  
    end if;
```

```
        if :PASSWORD is null then  
            MESSAGE('Password must be entered !');
```



```
MESSAGE('Password must be entered !');
go_item('employee.PASSWORD');
return;
end if;
```

```
SELECT E_JOB
      INTO DESIG
      FROM EMPLOYEE
      WHERE e_no = :e_no;
      IF DESIG = 'Technical Manager' or DESIG =
'Accounts Manager' or DESIG = 'Research Manager' OR DESIG = 'Sales Manager'
      THEN

open_form('C:\final\managemaster.fmx');

      ELSE

open_form('C:\final\employeeemaster.fmx');

      END IF;
```

exception

when no\_data\_found then

```
MESSAGE('Invalid User Name or Password');
MESSAGE('Invalid User Name or Password');
go_item('employee.e_no');
return;
```

end;

END IF;

---

ADMINMASTER

ADMINMASTER -> BLOCK4 -> ITEM12 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\viewscore.fmx');

ADMINMASTER -> BLOCK4 -> ITEM14 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\addinggroup.fmx');

ADMINMASTER -> BLOCK4 -> ITEM15 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\addgrouptodept.fmx');

ADMINMASTER -> BLOCK4 -> ITEM16 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\REGISTER.fmx');

ADMINMASTER -> BLOCK4 -> ITEM18 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\LOGIN.fmx');

ADMINMASTER -> BLOCK4 -> PUSFBUTTON19 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\adminmaster.fmx');

ADMINMASTER -> BLOCK4 -> ITEM21 -> TRIGGERS -> WHEN BUTTON PRESSED  
open\_form('C:\final\reportgenerate.fmx');

---

## REGISTER

REGISTER -> BLOCK93 -> LIST98 -> TRIGGER -> WHEN LIST CHANGED  
:employee.e\_mgr := :block93.list98;

REGISTER -> EMPLOYEE -> E\_NO -> TRIGGER -> PRE-TEXT-ITEM  
select nvl(max(e\_no),0)+1 into :e\_no from employee;

REGISTER -> EMPLOYEE -> PASSWORD -> TRIGGER -> WHEN-VALIDATE-ITEM  
if length(:employee.password) < 4 then  
    select initcap('Password should be at least 4 characters') into  
:block93.display\_item96 from dual;  
    raise form\_trigger\_failure;  
  
    else  
    :block93.display\_item96 :=null;  
    select initcap('Password valid') into :block93.display\_item97 from dual;  
  
end if;

REGISTER -> EMPLOYEE -> D\_NO -> TRIGGER -> WHEN-LIST-CHANGED

```

declare
    a number;
    rg RecordGroup;
begin
    rg := Create_Group_From_Query('rg1111', 'select e_name, to_char(e_no) eno from
employee where e_mgr is null and d_no = ' || :employee.d_NO );
    a:=populate_group(rg);
    populate_list('LIST98',rg);
    delete_group(rg);

    begin
        null;
    exception
    when others then
        null;
    end;

end;

```

REGISTER -> EMPLOYEE -> REGISTER -> TRIGGER -> WHEN-BUTTON-PRESSED

```

commit_form;
clear_form;

```

REGISTER -> EMPLOYEE -> HOME -> TRIGGER -> WHEN-BUTTON-PRESSED

```

open_form('C:\final\adminmaster.fmx');

```

REGISTER -> EMPLOYEE -> LOGOUT -> TRIGGER -> WHEN-BUTTON-PRESSED

```

open_form('C:\final\LOGIN.fmx');

```

---

ADDINGGROUP

ADDINGGROUP -> BLOCK3 -> PUSHBUTTON5 -> TRIGGER -> WHEN-BUTTON-PRESSED

```

begin
    commit_form;
    --message('Group Successfully Added');
exception
    when others then

```

```
        message('ID in use already');
        message('ID in use already');
    --raise form_trigger_failure;
end;
```

```
ADDINGGROUP -> BLOCK3 -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\LOGIN.fmx');
```

```
ADDINGGROUP -> BLOCK3 -> PUSHBUTTON19 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\adminmaster.fmx');
```

---

```
ADDGROUPTODEPT
```

```
ADDGROUPTODEPT -> LISTBUTTONS -> LIST4 ->TRIGGERS -> WHEN-LIST-CHANGED
:G_DEPT.G_ID := :LISTBUTTONS.LIST4;
```

```
ADDGROUPTODEPT -> LISTBUTTONS -> PUSHBUTTON8 ->TRIGGERS -> WHEN-BUTTON-
PRESSED
```

```
declare
    a number;
    rg1 RecordGroup;
begin
    rg1 := Create_Group_From_Query('rg1111', 'select g_name , to_char(g_id) from grp');
    a:=populate_group(rg1);
```

```
begin
    null;
exception
when others then
    null;
end;
```

```
populate_list('list4',rg1);
delete_group(rg1);
```

end;

ADDGROUPTODEPT -> LISTBUTTONS -> PUSHBUTTON12 -> TRIGGERS -> WHEN-BUTTON-PRESSED

insert into G\_DEPT (D\_NO,G\_ID) values(:D\_no,:G\_id);

standard.commit;

message('Feedback Successfully Saved ');

message('Feedback Successfully Saved ');

exception

when others then

null;

message('This Group is already allocated to this department');

--message('This Group is already allocated to this department');

call\_form('C:\final\ADDGROUPTODEPT.fmx');

ADDGROUPTODEPT -> LISTBUTTONS -> ITEM18 -> TRIGGERS -> WHEN-BUTTON-PRESSED

open\_form('C:\final\LOGIN.fmx');

ADDGROUPTODEPT -> LISTBUTTONS -> ITEM18 -> TRIGGERS -> WHEN-BUTTON-PRESSED

open\_form('C:\final\adminmaster.fmx');

---

VIEWSCORE

VIEWSCORE -> EMPLOYEE -> TRIGGER -> ON-POPULATE-DETAILS

--

-- Begin default relation declare section

--

DECLARE

recstat VARCHAR2(20) := :System.record\_status;

startitm VARCHAR2(61) := :System.cursor\_item;

```

    rel_id    Relation;
--
-- End default relation declare section
--
--
-- Begin default relation program section
--
BEGIN
    IF ( recstat = 'NEW' or recstat = 'INSERT' ) THEN
        RETURN;
    END IF;
--
-- Begin TRANS detail program section
--
    IF ( (:EMPLOYEE.E_NO is not null) ) THEN
        rel_id := Find_Relation('EMPLOYEE.EMPLOYEE_TRANS');
        Query_Master_Details(rel_id, 'TRANS');
    END IF;
--
-- End TRANS detail program section
--

    IF ( :System.cursor_item <> startitm ) THEN
        Go_Item(startitm);
        Check_Package_Failure;
    END IF;
END;
--
-- End default relation program section
--

VIEWSCORE -> EMPLOYEE -> TRIGGER -> ON-CHECK-DELETE-MASTER
--
-- Begin default relation declare section
--
DECLARE
    Dummy_Define CHAR(1);
--
-- Begin TRANS detail declare section
--
CURSOR TRANS_cur IS

```

```

SELECT 1 FROM TRANS T
WHERE T.E_NO = :EMPLOYEE.E_NO;
--
-- End TRANS detail declare section
--
--
-- End default relation declare section
--
--
-- Begin default relation program section
--
BEGIN
--
-- Begin TRANS detail program section
--
OPEN TRANS_cur;
FETCH TRANS_cur INTO Dummy_Define;
IF ( TRANS_cur%found ) THEN
    Message('Cannot delete master record when matching detail records exist. ');
    CLOSE TRANS_cur;
    RAISE Form_Trigger_Failure;
END IF;
CLOSE TRANS_cur;
--
-- End TRANS detail program section
--
END;
--
-- End default relation program section
--

```

```

VIEWSCORE -> EMPLOYEE -> B1 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\LOGIN.fmx');

```

```

VIEWSCORE -> EMPLOYEE -> BIN_OPEN_FORM -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\adminmaster.fmx');

```

```

VIEWSCORE -> EMPLOYEE -> PUSH_BUTTON40 -> TRIGGER -> WHEN-BUTTON-PRESSED
go_block('trans');
set_block_property('trans',onetime_where, 'e_no = ' || :employee.e_no);

```

```
execute_query(no_validate);
```

```
VIEWSCORE -> TRANS -> TRIGGER -> ON-POPULATE-DETAILS
```

```
--  
-- Begin default relation declare section  
--  
DECLARE  
    recstat   VARCHAR2(20) := :System.record_status;  
    startitm   VARCHAR2(61) := :System.cursor_item;  
    rel_id     Relation;  
--  
-- End default relation declare section  
--  
--  
-- Begin default relation program section  
--  
BEGIN  
    IF ( recstat = 'NEW' or recstat = 'INSERT' ) THEN  
        RETURN;  
    END IF;  
--  
-- Begin FACTORS detail program section  
--  
    IF ( (:TRANS.F_ID is not null) ) THEN  
        rel_id := Find_Relation('TRANS.TRANS_FACTORS');  
        Query_Master_Details(rel_id, 'FACTORS');  
    END IF;  
--  
-- End FACTORS detail program section  
--  
  
    IF ( :System.cursor_item <> startitm ) THEN  
        Go_Item(startitm);  
        Check_Package_Failure;  
    END IF;  
END;  
--  
-- End default relation program section  
--
```



VIEWSCORE -> TRANS -> TRIGGER -> ON-CHECK-DELETE-MASTER

```
--  
-- Begin default relation declare section  
--  
DECLARE  
    Dummy_Define CHAR(1);  
--  
-- Begin FACTORS detail declare section  
--  
CURSOR FACTORS_cur IS  
    SELECT 1 FROM FACTORS F  
    WHERE F.F_ID = :TRANS.F_ID;  
--  
-- End FACTORS detail declare section  
--  
--  
-- End default relation declare section  
--  
--  
-- Begin default relation program section  
--  
BEGIN  
--  
-- Begin FACTORS detail program section  
--  
OPEN FACTORS_cur;  
FETCH FACTORS_cur INTO Dummy_Define;  
IF ( FACTORS_cur%found ) THEN  
    Message('Cannot delete master record when matching detail records exist.');    CLOSE FACTORS_cur;  
    RAISE Form_Trigger_Failure;  
END IF;  
CLOSE FACTORS_cur;  
--  
-- End FACTORS detail program section  
--  
END;  
--  
-- End default relation program section  
--
```

---

## REPORTGENERATE

REPORTGENERATE -> SCOREPERCENT -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED

```
open_form('C:\final\LOGIN.fmx');
```

REPORTGENERATE -> SCOREPERCENT -> PUSH\_BUTTON24-> TRIGGER -> WHEN-BUTTON-PRESSED

```
go_block('scoring');
```

```
set_block_property('scoring',onetime_where, 'e_no='||:employee.e_no);
```

```
execute_query(no_validate);
```

REPORTGENERATE -> EMPLOYEE -> TRIGGER -> ON-POPULATE-DETAILS

```
--
```

```
-- Begin default relation declare section
```

```
--
```

```
DECLARE
```

```
  recstat  VARCHAR2(20) := :System.record_status;
```

```
  startitm VARCHAR2(61) := :System.cursor_item;
```

```
  rel_id   Relation;
```

```
--
```

```
-- End default relation declare section
```

```
--
```

```
--
```

```
-- Begin default relation program section
```

```
--
```

```
BEGIN
```

```
  IF ( recstat = 'NEW' or recstat = 'INSERT' ) THEN
```

```
    RETURN;
```

```
  END IF;
```

```
--
```

```
-- Begin SCORING detail program section
```

```
--
```

```
  IF ( (:EMPLOYEE.E_NO is not null) ) THEN
```

```
    rel_id := Find_Relation('EMPLOYEE.EMPLOYEE_SCORING');
```

```
    Query_Master_Details(rel_id, 'SCORING');
```

```
  END IF;
```

```
--
```

```
-- End SCORING detail program section
```

```

--

IF ( :System.cursor_item <> startitm ) THEN
    Go_Item(startitm);
    Check_Package_Failure;
END IF;
END;
--
-- End default relation program section
--

REPORTGENERATE -> EMPLOYEE -> TRIGGER -> ON-CHECK-DELETE-MASTER
--
-- Begin default relation declare section
--
DECLARE
    Dummy_Define CHAR(1);
--
-- Begin SCORING detail declare section
--
CURSOR SCORING_cur IS
    SELECT 1 FROM SCORING S
    WHERE S.E_NO = :EMPLOYEE.E_NO;
--
-- End SCORING detail declare section
--
--
-- End default relation declare section
--
--
-- Begin default relation program section
--
BEGIN
--
-- Begin SCORING detail program section
--
OPEN SCORING_cur;
FETCH SCORING_cur INTO Dummy_Define;
IF ( SCORING_cur%found ) THEN
    Message('Cannot delete master record when matching detail records exist. ');
CLOSE SCORING_cur;

```

```

        RAISE Form_Trigger_Failure;
    END IF;
    CLOSE SCORING_cur;
    --
    -- End SCORING detail program section
    --
END;
--
-- End default relation program section
--

```

---

## MANAGERMASTER

```

MANAGERMASTER -> BLOCK3 -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\LOGIN.fmx');

```

```

MANAGERMASTER -> BLOCK3 -> PUSHBUTTON19 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\managermaster.fmx');

```

```

MANAGERMASTER -> BLOCK3 -> ITEM16 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\givefeedbackforemp.fmx');

```

```

MANAGERMASTER -> BLOCK3 -> ITEM7 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\managerviewscore.fmx');

```

```

MANAGERMASTER -> BLOCK3 -> ITEM8 -> TRIGGER -> WHEN-BUTTON-PRESSED
open_form('C:\final\reportgenerate.fmx');

```

---

## GIVEFEEDBACKFOREMP

```

GIVEFEEDBACKFOREMP -> LISTBUTTONS -> PUSH_BUTTON12 -> TRIGGER -> WHEN-BUTTON-
PRESSED
declare
    a number;
    desig varchar2(30);
    rg RecordGroup;

```

```

        abc number(4);

begin
    rg := Create_Group_From_Query('rg1111', 'select f_name, to_char(f_id) fid from
factors where g_id in(select g_id from g_dept where d_no=(select d_no from employee where
e_no= ' || :trans.e_no || '))' );
    a:=populate_group(rg);

--select e_name into :text_item62 from employee where e_no=:trans1.e_no;

    SELECT E_JOB
    INTO DESIG
    FROM EMPLOYEE
    WHERE e_no = :e_no;

    select e_mgr into abc from employee where e_no = :trans.e_no;

IF DESIG = 'Technical Manager' or DESIG = 'Accounts Manager' or DESIG = 'Research Manager'
OR DESIG = 'Sales Manager'
    THEN
        message('You Cannot Rate a Manager');
        --message('You Cannot Rate a Manager');
        call_form('C:\final\GIVEFEEDBACKFOREMP.fmx');
        else
            if abc = :global.username    then
                populate_list('F_LIST',rg);
                delete_group(rg);
            else
                message('You Can give feedback to your own
subordinates only');
                call_form('C:\final\GIVEFEEDBACKFOREMP.fmx');
            end if;
        END IF;

        begin
            null;
        exception
        when others then
            null;
        end;

```

end;

GIVEFEEDBACKFOREMP -> LISTBUTTONS -> F\_LIST -> TRIGGER -> WHEN-LIST-CHANGED  
:TRANS.F\_ID := :LISTBUTTONS.F\_LIST;

GIVEFEEDBACKFOREMP -> LISTBUTTONS -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED  
open\_form('C:\final\LOGIN.fmx');

GIVEFEEDBACKFOREMP -> LISTBUTTONS -> PUSHBUTTON67 -> TRIGGER -> WHEN-BUTTON-PRESSED

declare

score1 number(4);

begin

select score

into score1

from trans

where

e\_no = :e\_no and f\_id =:f\_id and p\_id = :p\_id;

update trans set score=:score where e\_no=:e\_no and f\_id=:f\_id and p\_id=:p\_id;

exception

when no\_data\_found then

insert into trans (e\_no,f\_id,score,p\_id) values(:e\_no,:f\_id,:score,:p\_id);

standard.commit;

message(sqlerrm);

message(sqlerrm);

end;

standard.commit;

message('Feedback Successfully Saved ');

message('Feedback Successfully Saved ');

GIVEFEEDBACKFOREMP -> LISTBUTTONS -> ITEM73 -> TRIGGER -> WHEN-BUTTON-PRESSED  
open\_form('C:\final\managermaster.fmx');

MANAGERVIEWSCORE -> BLOCK11 -> ITEM18 -> TRIGGERS -> WHEN-BUTTON-PRESSED  
open\_form('C:\final\LOGIN.fmx');

MANAGERVIEWSCORE -> BLOCK11 -> ITEM73 -> TRIGGERS -> WHEN-BUTTON-PRESSED  
open\_form('C:\final\managermaster.fmx');

MANAGERVIEWSCORE -> TRANS-> TRIGGERS -> ON-POPULATE-DETAILS

```
--  
-- Begin default relation declare section  
--  
DECLARE  
    recstat   VARCHAR2(20) := :System.record_status;  
    startitm   VARCHAR2(61) := :System.cursor_item;  
    rel_id     Relation;  
--  
-- End default relation declare section  
--  
--  
-- Begin default relation program section  
--  
BEGIN  
    IF ( recstat = 'NEW' or recstat = 'INSERT' ) THEN  
        RETURN;  
    END IF;  
--  
-- Begin FACTORS detail program section  
--  
    IF ( (:TRANS.F_ID is not null) ) THEN  
        rel_id := Find_Relation('TRANS.TRANS_FACTORS');  
        Query_Master_Details(rel_id, 'FACTORS');  
    END IF;  
--  
-- End FACTORS detail program section  
--  
  
    IF ( :System.cursor_item <> startitm ) THEN  
        Go_Item(startitm);  
        Check_Package_Failure;  
    END IF;
```

```

END;
--
-- End default relation program section
--

MANAGERVIEWSCORE -> TRANS-> TRIGGERS -> ON-CHECK-DELETE-MASTER
--
-- Begin default relation declare section
--
DECLARE
    Dummy_Define CHAR(1);
--
-- Begin FACTORS detail declare section
--
CURSOR FACTORS_cur IS
    SELECT 1 FROM FACTORS F
    WHERE F.F_ID = :TRANS.F_ID;
--
-- End FACTORS detail declare section
--
-- End default relation declare section
--
-- Begin default relation program section
--
BEGIN
--
-- Begin FACTORS detail program section
--
OPEN FACTORS_cur;
FETCH FACTORS_cur INTO Dummy_Define;
IF ( FACTORS_cur%found ) THEN
    Message('Cannot delete master record when matching detail records exist. ');
    CLOSE FACTORS_cur;
    RAISE Form_Trigger_Failure;
END IF;
CLOSE FACTORS_cur;
--
-- End FACTORS detail program section
--

```



END;

--

-- End default relation program section

--

---

## EMPLOYEEMASTER

EMPLOYEEMASTER -> BLOCK3 -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED

open\_form('C:\final\LOGIN.fmx');

EMPLOYEEMASTER -> BLOCK3 -> PUSFBUTTON19 -> TRIGGER -> WHEN-BUTTON-PRESSED

open\_form('C:\final\employeeemaster.fmx');

EMPLOYEEMASTER -> BLOCK3 -> ITEM16 -> TRIGGER -> WHEN-BUTTON-PRESSED

open\_form('C:\final\employeescore.fmx');

EMPLOYEEMASTER -> BLOCK3 -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED

open\_form('C:\final\reportgenerate.fmx');

---

## EMPLOYEESCORE

EMPLOYEESCORE -> BLOCK17 -> PUSHBUTTON18 -> TRIGGER -> WHEN-BUTTON-PRESSED

go\_block('trans');

set\_block\_property('trans',onetime\_where,'e\_no = '|| :global.username);

execute\_query(no\_validate);

EMPLOYEESCORE -> TRANS -> TRIGGER -> ON-POPULATE-DETAILS

--

-- Begin default relation declare section

--

DECLARE

recstat VARCHAR2(20) := :System.record\_status;

startitm VARCHAR2(61) := :System.cursor\_item;

rel\_id Relation;

--

-- End default relation declare section

--

```

--
-- Begin default relation program section
--
BEGIN
  IF ( recstat = 'NEW' or recstat = 'INSERT' ) THEN
    RETURN;
  END IF;
  --
  -- Begin FACTORS detail program section
  --
  IF ( (:TRANS.F_ID is not null) ) THEN
    rel_id := Find_Relation('TRANS.TRANS_FACTORS');
    Query_Master_Details(rel_id, 'FACTORS');
  END IF;
  --
  -- End FACTORS detail program section
  --

  IF ( :System.cursor_item <> startitm ) THEN
    Go_Item(startitm);
    Check_Package_Failure;
  END IF;
END;
--
-- End default relation program section
--

EMPLOYEESCORE -> TRANS -> TRIGGER -> ON-CHECK-DELETE-MASTER
--
-- Begin default relation declare section
--
DECLARE
  Dummy_Define CHAR(1);
  --
  -- Begin FACTORS detail declare section
  --
  CURSOR FACTORS_cur IS
    SELECT 1 FROM FACTORS F
    WHERE F.F_ID = :TRANS.F_ID;
  --
  -- End FACTORS detail declare section

```

```

--
--
-- End default relation declare section
--
--
-- Begin default relation program section
--
BEGIN
--
-- Begin FACTORS detail program section
--
OPEN FACTORS_cur;
FETCH FACTORS_cur INTO Dummy_Define;
IF ( FACTORS_cur%found ) THEN
    Message('Cannot delete master record when matching detail records exist.');
```

CLOSE FACTORS\_cur;
 RAISE Form\_Trigger\_Failure;
END IF;
CLOSE FACTORS\_cur;
--
-- End FACTORS detail program section
--
END;
--
-- End default relation program section
--

EMPLOYEESCORE -> TRANS -> ITEM18 -> TRIGGER -> WHEN-BUTTON-PRESSED  
 open\_form('C:\final\LOGIN.fmx');

---

## Certificate

This is to certify that Mr. *Prince Kumar Singh* of *JIS college of Engineering*, registration number: *111230110182*, has successfully completed a project on *Intranet based appraisal management system* using *oracle 10g*.  
under the guidance of Mr. *Suman Aich*.

-----  
[*Suman Aich*]

**Globsyn Finishing School**  
(*a division of Globsyn Skills*)



## Certificate

This is to certify that Mr. Md. Arshad Feeroz of *Guru Nanak Institute Of Technology*,  
registration number: 111430110053 , has successfully completed a project on *Intranet based  
Appraisal Management System* using *Oracle 10g* under the guidance of Mr. *Suman Aich*.

-----  
[*Suman Aich*]

**Globsyn Finishing School**  
(*a division of Globsyn Skills*)

## Certificate

This is to certify that Mr. *Surajdev Kumar Pandey* of *Guru Nanak Institute of Technology*, registration number: 111430110118 , has successfully completed a project on *Intranet based Appraisal Management System* using *Oracle* under the guidance of Mr. *Suman Aich*.

-----  
[*Suman Aich*]

**Globsyn Finishing School**  
(*a division of Globsyn Skills*)





## Certificate

This is to certify that Ms. Suman Kumar Maity of *Guru Nanak Institute of Technology*, registration number: 111430110114, has successfully completed a project on *Intranet based Appraisal Management System* using Oracle 10g under the guidance of Mr. *Suman Aich*.

[*Suman Aich*]

**Globsyn Finishing School**  
(a division of *Globsyn Skills*)



[www.globs`a](http://www.globs`a)

[ynskills.com](http://ynskills.com)