EMPLOYEE PAYROLL MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

MANIVEER[RA2011031010127]
ADITYA [RA2011031010120]
SHRIDHAR[RA2011031010132]

Under the guidance of
Dr.C.N.S. Vinoth Kumar
(Associate Professor, Department of Networking and Communications)

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING
With specialization in <INFORMATION TECHNOLOGY>

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

APRIL 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report**"Employee Payroll Management System"**is the bonafide workof**"**MANIVEER(RA2011031010127),ADITYA(RA2011031010120),SHRIDHAR(RA2011031010132)**"** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSC303J- Database Management systems in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

SIGNATURE

Faculty name
Faculty Designation
Department name and seal

ABSTRACT

The purpose of Payroll Management System is to automate the existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Payroll Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information. that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

TABLE OF CONTENTS

Figure No.	Figure Name	Page No.

LIST OF TABLES

ABBREVIATIONS

AES	Advanced Encryption Standard
ANN	Artificial Neural Network
CSS	Cascading Style Sheet
CV	Computer Vision
DB	Data Base
DNA	Deoxyribo Neucleic Acid
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface

Payroll System Problem Statement

Our task is the creation of a new payroll system. The old system is outdated and no longer Adequately manages the payroll process and the entry of employee time card information. Therefore, manual intervention is required to process the payroll. In addition to the preexisting category of salaried employee, which the current payroll process does support, our organization now accommodates two new types of employees:
those employees working on commission, and those employees working on an hourly basis. The current payroll process doesn't support these two new employee types, which are described in detail later in this problem statement.

The system also must allow employees to query the system. Standard queries will include the Ability to view number of hours worked for a selected pay period and remaining and used vacation time.All employee information is maintained by a human resources administrator. The administrator must be able to initiate a payroll process in a fashion similar to that of the human resources representative. In addition, the administrator also will be able to add new employees, remove employees, and change personal information about employees, including an employee's category type and an employee's password. The administrator will have additional reporting capabilities beyond that of an employee, which include various administrative reports yet to be identified.

Objectives Of Payroll Management System:

The main objective of the Project on Payroll Management System is to manage the details of Payroll, Employee, Salary, Appraisals, Working Points. It manages all the information about Payroll, Payments, Working Points, Payroll. The project is totally built at administrative end and thus only the administrator is guaranteed the access. The purpose of the project is to build an application program to reduce the manual work for managing the Payroll, Employee, Payments, Salary. It tracks all the details about the Salary, Appraisals, Working Points.

Functionalist provided by Payroll Management System are as follows:

Provides the searching facilities based on various factors. Such as Payroll, Salary, Appraisals, Working Points.

Payroll Management System also manage the Payments details online for Appraisals details, Working Points details, Payroll. It tracks all the information of Employee, Payments, Appraisals etc.

Manage the information of Employee

Shows the information and description of the Payroll, Salary.
To increase efficiency of managing the Payroll, Employee .
It deals with monitoring the information and transactions of Appraisals.

Manage the information of Payroll

Editing, adding and updating of Records is improved which results in proper resource management of Payroll data.

Manage the information of Appraisals

Integration of all records of Working Points.

Summary:

1. Employee Information

Employee data is very essential in order to maintain a proper record of the employees and there personal information for various purposes like contacting them for inviting for certain summit, feedback of the company from the employee data

2. Maintaining Salary

Very important to keep this data which will help not only the managers and the HR to keep a track of the employee salaries but also help the company or its board to analyze what amount they are spending on a particular employee of a particular company

3. Work Location

It is very much important for an organization small or big to have a record of all the work locations they operate from to see how they can develop in that particular region and also increase the hiring in that region so that the organization can increase there Market Outreach that area.

4. Projects

In order to be successful company should be involved in various projects, so they also need to maintain the record of the salaries each employee is being paid for a particular type of project he/she is working on

PL/SQL features used in the project:

1. Created Explicit Cursors which shows the hourly pay of the employees associated with Accounts and Ref cursor showing the employees who are a part of a particular department.
2. Create a CDB and a PDB with users to manage the data according to the area of interest
3. Implement pre-defined exception cursor already open to demonstrate the understanding of the exceptional handling concept which shows what error will populate when we try to open a cursor which is already open
4. Also, created Relational, Inline and Materialized Views satisfying various business requirements.
5. Created Index on Account Details table.
6. Built an E-R Diagram to know how the entities are related in the payroll management system for any company.

List of Entities:

Employee

Employee table will include all the personal details of the employee and would be very much cover overall information of that particular employee .

Salary

Salary Table will cover all the current and previous salaries an employee had or currently has. This table will help a manager/ an HR to analyse which employee has been given promotion on which date or when did his salary grade changed**.**

Department

Department Table maintains the data of the all the possible departments an employee can belong to

Account-Details

Account Details Table will maintain the data regarding the accounts which the employee has connected with the company for his/her salary to be credited

Attendance

This table includes all the data of the employee's attendance which includes the number of hours an employee has worked in a week

Project

This table includes the data of all the projects a particular company is working on or the projects on which the company is going to work in the future

Education

The Education Table keeps the track of the education of the employee including his degrees achieved until now

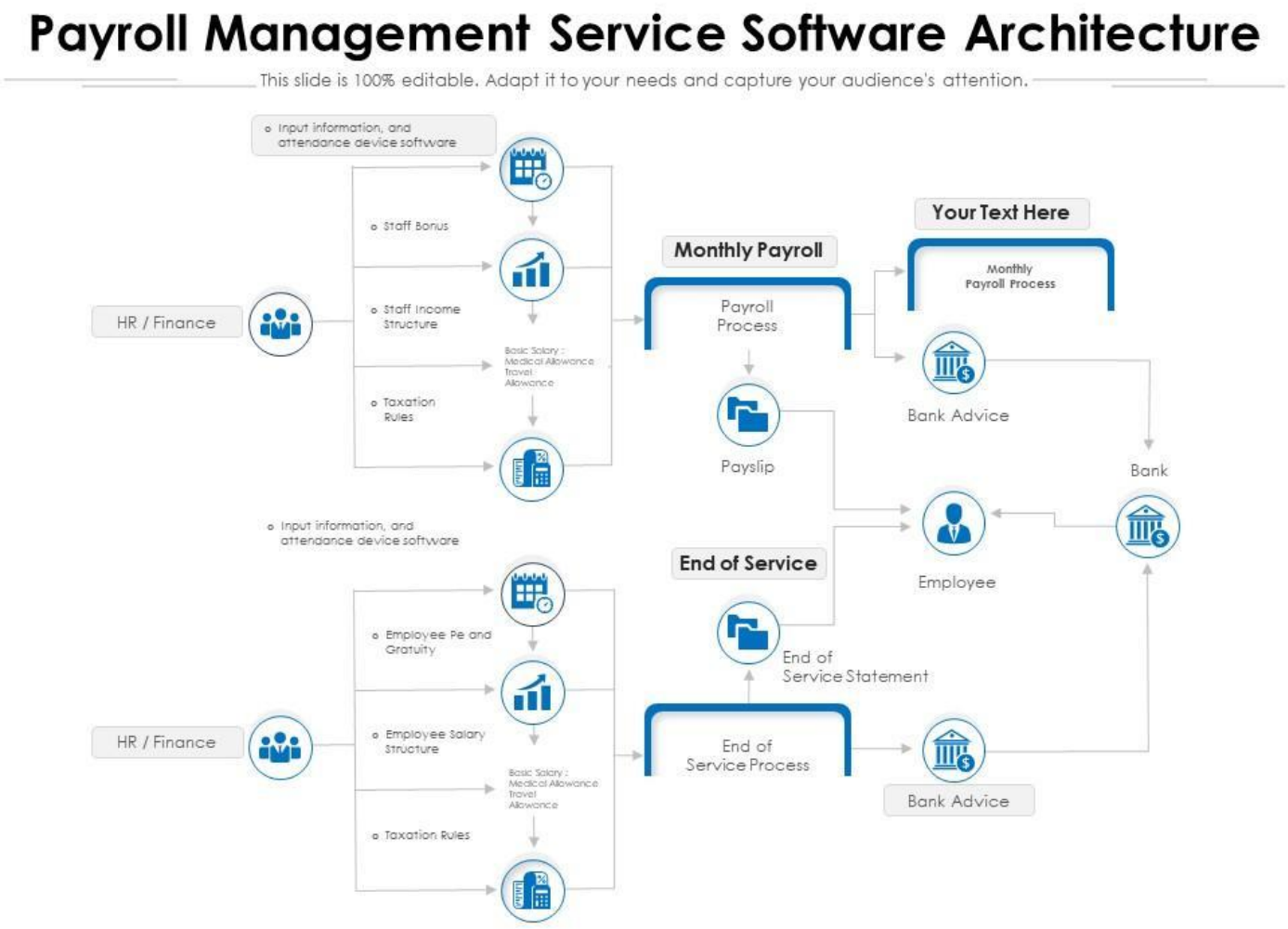
Work Location

The name of the table tells you most of the things. This table includes the location of the office, which city is it located, which state it is in and also tracks the number of employees in a particular location

Leave

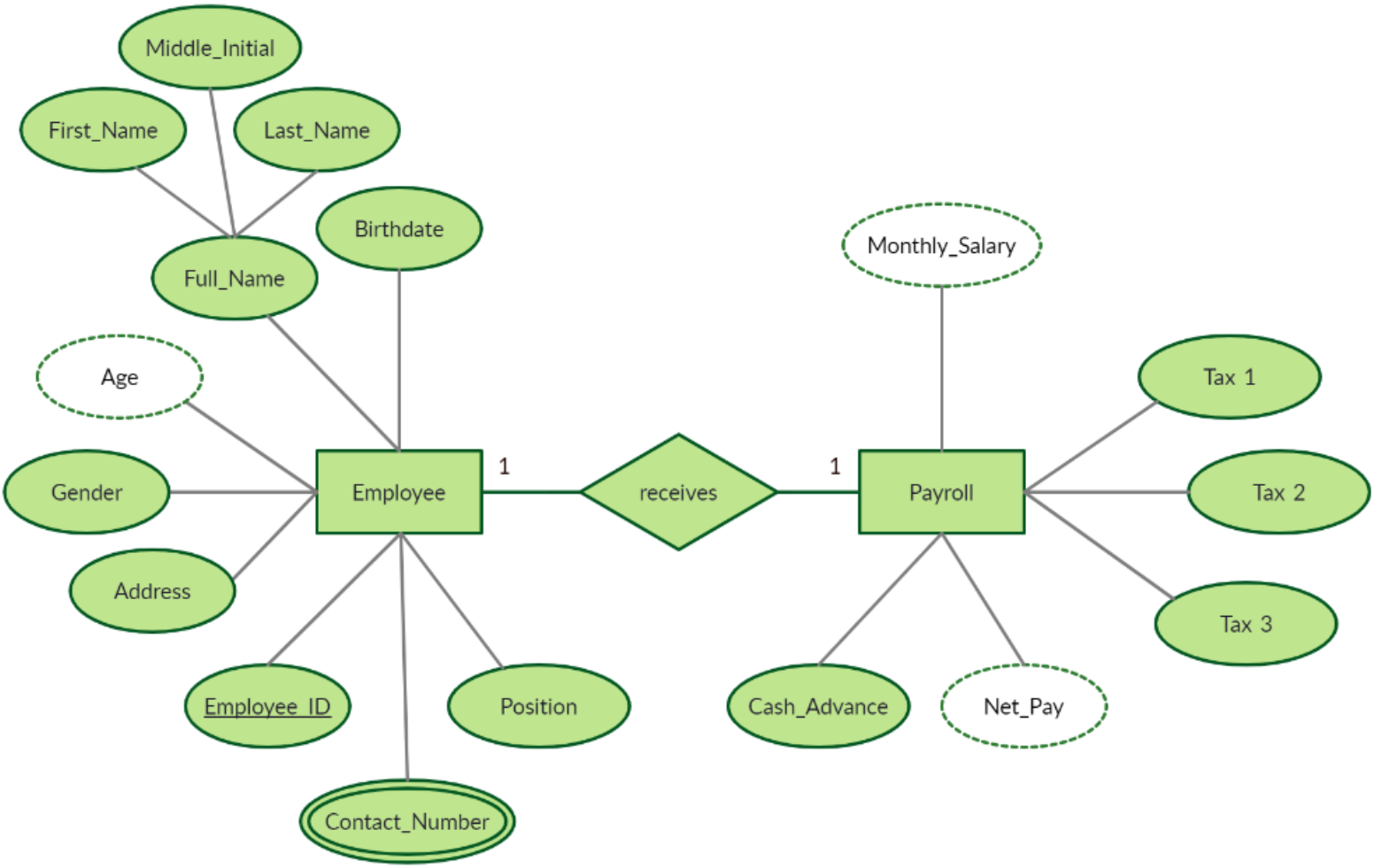
Leave table keeps the record of the number of leaves an employee takes or has taken over the course of any month or a year.

SYSTEM ARCHITECTURE :



DESIGN OF EMPLOYEE_PAYROLL_MANAGEMENT SYSTEM:

E-R Diagram:



Database used:

To pay your employees accurately and on time, you need a payroll system. A payroll database is an automated system that allows you to input employees' payroll information and compensate them accordingly. The database may be a stand-alone system that enables only payroll operations, or an integrated system that enables related business functions. Stand-Alone A stand-alone payroll database is a single payroll application that you use to perform payroll tasks. This option may come in handy if you already have HR and accounting solutions in place. An effective stand-alone database gives you a complete range of services that allows you to fully manage your payroll activities. This includes new-hire reporting, wage and deduction calculations check printing, direct deposit, wage garnishments, tax reporting and management, paycheck reconciliation, multiple company management, and electronic record-keeping.

Integrated

An integrated database incorporates payroll with functions such as human resources and accounting. The system may integrate all three applications, or two out of the three, such as payroll and HR or payroll and accounting. For example, an integrated database with all three functions enables payroll, billing, human resources records, accounts receivable, accounts payable, general ledger, bank reconciliation, workers' compensation and risk management, benefits administration, tax filing, and unemployment reporting.

Tables

In the database are tables that allow you to input information and execute your payroll duties. Tables depend on whether the database is an integrated or stand-alone unit. Payroll tables may include deductions, leaves, pay codes, payroll calendar, job class, bank code, department, earned income credit, and salary schedule. They also provide for federal, state, local and FICA taxes. Once your payroll database is established, you can add and update employee records. Data includes each employee's name, address, birth date, Social Security number, pay rate, and withholding tax conditions, plus voluntary benefits, such as health care and retirement deduction amounts. You can also generate and print reports. For example, to create a payroll register, you would go to the report printing function and enter the pay period start and end dates. The system

would then generate the pay check data that relates to that time frame.

Implementation

A payroll software company reviews your current payroll system, and designs and engineers a database that suits your needs. If you're starting from scratch, the vendor might have integrated systems that are already designed. If you already have a system in place, the vendor can integrate what you need with your existing system. The vendor should offer a one-on-one custom experience, so the implementation is unique to your business. Regardless of whether the database is stand-alone or integrated, it should undergo testing before you use it to officially run your first payroll. This way, errors can be detected and resolved beforehand.

Access

Through the use of user identification numbers and passwords, relevant employees can access the payroll database. Access should be restricted to the appropriate personnel. For example, if your human resources and payroll departments are separate, but they share the same database, the HR department should not have access to pay check processing and the payroll department should not have access to benefits administration.

FRONT AND BACK END

List of Modules and Functionalities :

Employee Management Module:

- Employee Information Management: Store and manage employee
- information such as name, address, contact details, date of birth,and other relevant details.
- Employee Attendance Management: Track and manage employee attendance and leaves.
- Employee Performance Management: Track and manage employee performance and appraisals.

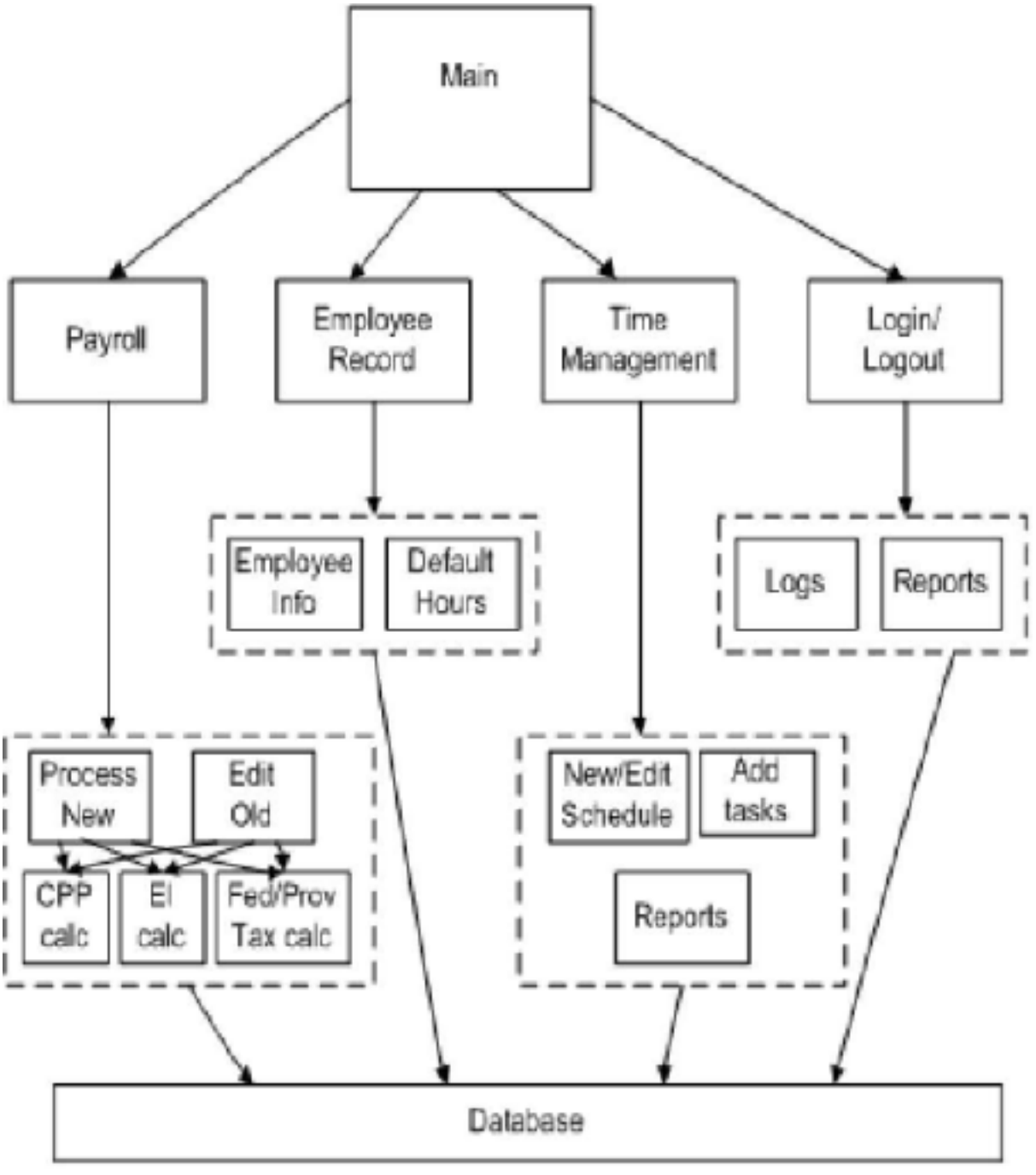
Department Management Module:

- Department Information Management: Store and manage department information such as department name, department head, and other relevant details.
- Department Budget Management: Manage departmental budgets, including expenses and revenue.

User Management Module:

- User Authentication: Authenticate users to access the system.
- User Access Management: Manage user access to different modules and Functionalities of the system.

ARCHITECTURE DIAGRAM :



The features of employee payroll management system are:

1. Login System
2. Add, Edit, Remove, View Employees record
3. Manage Department
4. Employee Personal Details
5. Employee Company Details
6. Add Payslip
7. Mange Payslip
8. PDF Format Payslip Generate
9. Settings
10. Manage Bonus and Deductions
11. List Income
12. Manage Over Time and Salary Rate

Front end (UI)design and software used
Hypertext Markup Language (HTML) is based on the
Standard Generalized Language (SGML).

HTML is a language for describing the structure of a document, not its presentation. HTML defines a set of common styles for web pages: headings, paragraphs, lists and tables.

HTML provides a means by which a documents main content can be annotated with various kinds of meta-data and rendering hints. Adobe Dream weaver is the leading software tools for editing HTML.

Content and presentation can be combined using server-side scripting languages like PHP and ASP.Net to make the final HTML Chosen Web Application Technologies.

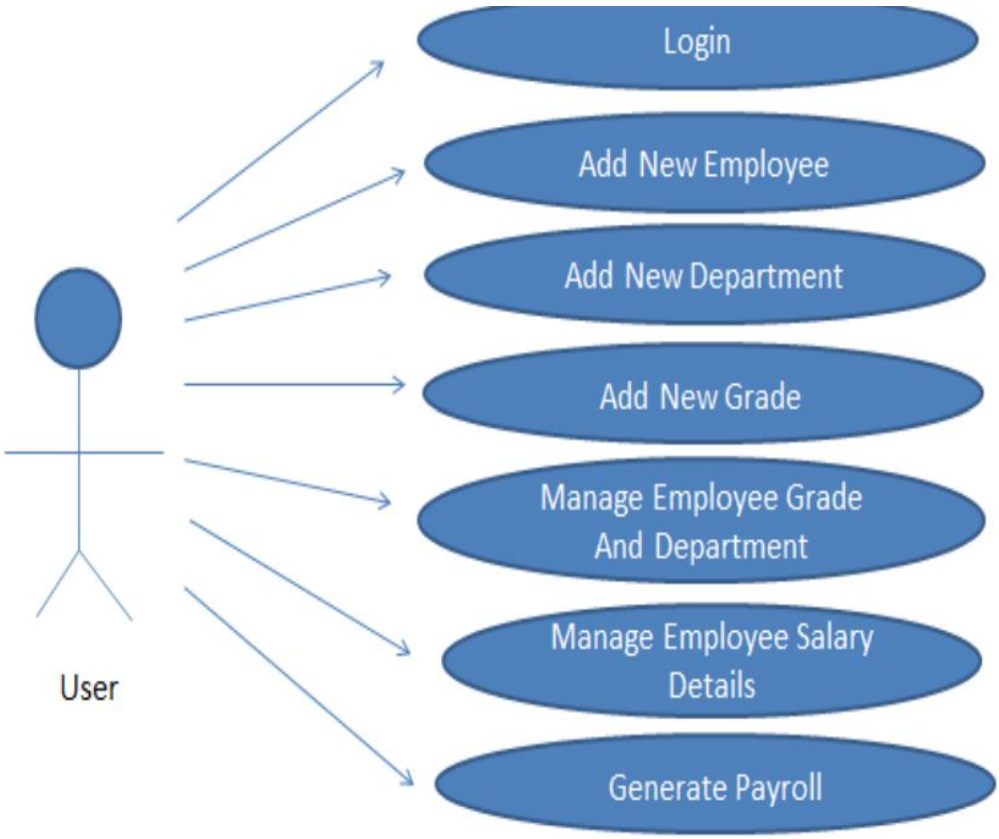
The combination of PHP and MySQL was chosen for this research project because of the following reasons:

1. They are open source which implies that they are cheap to get since one just need to download them from the net.

2. PHP is a rapid application development environment and is known for its ease of use. It enables most developers get involved with dynamic web applications without having to learn entirely new set of functions because it is very similar to structured programming languages.

3. MySQL has very fast database management system and is also easier to use than many other database system**s.**

USE CASE DIAGRAM :



CODE FOR EMPLOYEE_PAYROLL_MANAGEMENT SYSTEM:

Create Table Statements
Employee

```
CREATE TABLE Employee(
Employee_Id NUMBER(6),
 First_Name VARCHAR2(25),
 Last_Name VARCHAR2(25),
Hire_Date DATE,
City VARCHAR2(25),
State VARCHAR2(25),
CONSTRAINT EMPLOYEE_PK PRIMARY KEY (Employee_Id));
```

Department

```
CREATE TABLE Department(
Department_Id NUMBER,
Department_Name VARCHAR2(30),
CONSTRAINT DEPARTMENT_PK PRIMARY KEY (Department_Id) );
```


Salary

```
CREATE TABLE Salary(
Salary_Id NUMBER,
 Gross_Salary NUMBER,
Hourly_Pay NUMBER,
 State_Tax NUMBER,
Federal_Tax NUMBER,
Account_Id NUMBER,
CONSTRAINT SALARY_PK PRIMARY KEY (Salary_Id),
FOREIGN KEY (Account_Id)
 REFERENCES ACCOUNTDETAILS(Account_Id)
 );
```

Department Project Bridge

```
CREATE TABLE DepartmentProject(
Department_Id NUMBER,
```

```
Project_Id NUMBER,
CONSTRAINT DEPTPROJECT_PK PRIMARY KEY (Department_Id,Project_Id),
 FOREIGN KEY (Department_Id)
    REFERENCES Department(Department_Id),
FOREIGN KEY (Project_Id)
 REFERENCES Project(Project_Id)
 );
```

Project

```
CREATE TABLE Project(
Project_Id NUMBER,
Project_Name VARCHAR2(50),
Project_Description VARCHAR2(50),
CONSTRAINT Project_PK PRIMARY KEY (Project_Id)
);
```

AccountDetails

```
CREATE TABLE AccountDetails(
Account_Id NUMBER,
Bank_Name VARCHAR2(50),
Account_Number VARCHAR2(50),
Employee_Id NUMBER,
CONSTRAINT Account_PK PRIMARY KEY (Account_Id),
FOREIGN KEY (Employee_Id)
      REFERENCES Employee(Employee_Id)
 );
```

Education

```
CREATE TABLE Education(
Education_Id NUMBER,
Employee_Id NUMBER,
Degree VARCHAR(30),
Graduation_Year NUMBER(4),
CONSTRAINT Location_PK PRIMARY KEY (Education_Id),
FOREIGN KEY (Employee_Id)
REFERENCES Employee(Employee_Id)
 );
```

Leave

```
CREATE TABLE Leave(
 Leave_Id NUMBER,
Employee_Id NUMBER,
 Leave_date DATE,
 CONSTRAINT Leave_PK PRIMARY KEY (Leave_Id),
FOREIGN KEY (Employee_Id)
  REFERENCES Employee(Employee_Id)
 );
```

Employee Attendance Bridge

```
 CREATE TABLE Employee_Attendance(
 Employee_Id NUMBER,
Attendance_Id NUMBER,
CONSTRAINT DEPARTMENTPROJECT_PK PRIMARY KEY (Employee_Id,Attendance_Id),
 FOREIGN KEY (Employee_Id)
 REFERENCES Employee(Employee_Id),
FOREIGN KEY (Attendance_Id)
REFERENCES Attendance(Attendance_Id)
);
```

Attendance

```
CREATE TABLE Attendance(
Attendance_Id NUMBER,
 Hours_Worked NUMBER,
CONSTRAINT Attendance_PK PRIMARY KEY (Attendance_Id)
 );
```

WorkLocation

```
CREATE TABLE Work_Location(
Location_Id NUMBER,
Location VARCHAR2(25),
Number_Of_Employees NUMBER,
City VARCHAR2(25),
State VARCHAR2(25),
```

CONSTRAINT Loc_PK PRIMARY KEY (Location_Id)
);

Insert Statements

INSERT INTO Employee VALUES (101,'Ojas','Phansekar',to_date('14-APR-16', 'dd-MON-yyyy'),'New York City','New York',1);

INSERT INTO Employee VALUES (102,'Vrushali','Patil',to_date('21-JUN-18', 'ddMONyyyy'),'Boston','Massachusetts',2);

INSERT INTO Employee VALUES (103,'Pratik','Parija',to_date('13-SEP-19', 'ddMONyyyy'),'Chicago','Illinois',3);

INSERT INTO Employee VALUES (104,'Chetan','Mistry',to_date('12-APR-11', 'ddMONyyyy'),'Miami','Florida',4);

INSERT INTO Employee VALUES (105,'Anugraha','Varkey',to_date('16-AUG-17', 'ddMONyyyy'),'Atlanta','Georgia',5);

INSERT INTO Employee VALUES (106,'Rasagnya','Reddy',to_date('25-JUL-18', 'dd-MON-yyyy'),'San Mateo','California',6);

INSERT INTO EmployeVALUES(107,'Aishwarya','Boralkar',to_date('18-DEC-10', 'dd-MON-yyyy'),'San Francisco','California',7);

INSERT INTO Employee VALUES (108,'Shantanu','Savant',to_date('27-NOV-15', 'ddMONyyyy'),'Seattle','Washington',8);

INSERT INTO Employee VALUES(109,'Kalpita','Malvankar',to_date('24-APR-16', 'ddMONyyyy'),'Boston','Massachusetts',8);

INSERT INTO Employee VALUES (110,'Saylee','Bhagat',to_date('21-MAY-14', 'dd-MON-yyyy'),'San Francisco','California',7);

INSERT INTO Department VALUES (1,'Human Resources');

INSERT INTO Department VALUES (2,'Software Development'); INSERT INTO Department VALUES (3,'Data Analysis');

INSERT INTO Department VALUES (4,'Data Science');

INSERT INTO Department VALUES (5,'Business Intelligence');

INSERT INTO Department VALUES (6,'Data Engineering');

INSERT INTO Department VALUES (7,'Manufacturing');

INSERT INTO Department VALUES (8,'Quality Control');

INSERT INTO Project VALUES (21,'Dev','Whatever');

INSERT INTO Project VALUES (22,'Prod','do something');

INSERT INTO Project VALUES (23,'Test','focus');

INSERT INTO Project VALUES (24,'Nothing','do nothing');

INSERT INTO Project VALUES (25,'Research','focus on everything'); INSERT INTO Project VALUES (26,'Next Steps','find some way out');

INSERT INTO AccountDetails VALUES (40,'Santander','S12344',101); INSERT INTO AccountDetails VALUES (41,'Santander','S12345',102); INSERT INTO AccountDetails VALUES (42,'Santander','S12346',103); INSERT INTO AccountDetails VALUES (43,'Santander','S12347',104); INSERT INTO AccountDetails VALUES (44,'Chase','C12344',105); INSERT INTO AccountDetails VALUES (45,'Chase','C12345',106); INSERT INTO AccountDetails VALUES (46,'Chase','C12347',107); INSERT INTO AccountDetails VALUES (47,'Chase','C12334',108); INSERT INTO AccountDetails VALUES (48,'BOFA','C12378',109); INSERT INTO AccountDetails VALUES (49,'BOFA','C12390',110);

```
INSERT INTO Education VALUES (10,101,'MS',2017);
INSERT INTO Education VALUES (11,102,'MS',2019);
INSERT INTO Education VALUES (12,104,'MS',2011);
INSERT INTO Education VALUES (13,108,'MS',2015);
INSERT INTO Education VALUES (14,109,'Bachelor',2013);
INSERT INTO Education VALUES (15,107,'Bachelor',2008);
INSERT INTO Education VALUES (16,106,'Bachelor',2007);



INSERT INTO Leave VALUES (51,104,to_date('1-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (52,108,to_date('2-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (53,109,to_date('3-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (54,107,to_date('4-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (55,106,to_date('5-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (56,104,to_date('6-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (57,108,to_date('7-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (58,109,to_date('7-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (59,107,to_date('8-DEC-19', 'dd-MON-yyyy'));
INSERT INTO Leave VALUES (60,106,to_date('9-DEC-19', 'dd-MON-yyyy'));

INSERT INTO Attendance VALUES (90,10);
INSERT INTO Attendance VALUES (91,20);
INSERT INTO Attendance VALUES (92,30);
INSERT INTO Attendance VALUES (93,40);
INSERT INTO Attendance VALUES (94,45);
INSERT INTO Attendance VALUES (95,56);
INSERT INTO Attendance VALUES (96,58);



INSERT INTO Work_Location VALUES (71,'North',4,'New York City','New York',101);
INSERT INTO Work_LocationVALUES (72,'North',4,'Boston','Massachusetts',102);
INSERT INTO Work_Location VALUES (73,'North',4,'Chicago','Illinois',103);
INSERT INTO Work_Location VALUES (74,'North',89,'Miami','Florida',104);
INSERT INTO Work_Location VALUES (75,'South',90,'Atlanta','Georgia',105);
INSERT INTO Work_Location VALUES (76,'South',100,'San Mateo','California',106);
INSERT INTO Work_Location VALUES (77,'South',4,'San Francisco','California',107);
INSERT INTO Work_Location VALUES (78,'South',2,'Seattle','Washington',108);
INSERT INTO Work_Location VALUES (79,'South',25,'Alpharetta','Georgia',109);
INSERT INTO Work_Location VALUES (80,'South',20,'Keene','New Hampshire',110);
INSERT INTO Work_Location VALUES (81,'South',22,'Hampton','New Hampshire',109);
```

```
INSERT INTO Employee_Attendance VALUES (101,90);
INSERT INTO Employee_Attendance VALUES (102,91);
INSERT INTO Employee_Attendance VALUES (103,92);
INSERT INTO Employee_Attendance VALUES (104,93);
INSERT INTO Employee_Attendance VALUES (105,94);
INSERT INTO Employee_Attendance VALUES (106,95);
INSERT INTO Employee_Attendance VALUES (107,96);
INSERT INTO Employee_Attendance VALUES (108,91);
INSERT INTO Employee_Attendance VALUES (109,92);
INSERT INTO Employee_Attendance VALUES (110,93);
```

```
INSERT INTO DepartmentProject VALUES (1,21);
INSERT INTO DepartmentProject VALUES (2,22);
INSERT INTO DepartmentProject VALUES (3,23);
INSERT INTO DepartmentProject VALUES (4,24);
INSERT INTO DepartmentProject VALUES (5,25);
INSERT INTO DepartmentProject VALUES (6,26);
INSERT INTO DepartmentProject VALUES (7,21);
INSERT INTO DepartmentProject VALUES (8,24);
```

```
INSERT INTO Salary VALUES (1,57600,30,200,1000,40);
INSERT INTO Salary VALUES (2,76800,40,300,1300,41);
INSERT INTO Salary VALUES (3,96000,50,400,1500,42);
INSERT INTO Salary VALUES (4,115200,60,500,1700,43);
INSERT INTO Salary VALUES (5,57600,30,200,1000,44);
INSERT INTO Salary VALUES (6,76800,40,300,1300,45);
INSERT INTO Salary VALUES (7,96000,50,400,1500,46);
INSERT INTO Salary VALUES (8,115200,60,500,1700,47);
INSERT INTO Salary VALUES (9,57600,30,200,1000,48);
INSERT INTO Salary VALUES (10,76800,40,300,1300,49);
```

```
Inline View select Department_name count(*)

to_char((count(*)/No_of_Employees.cnt)*100, '90.99') Percentages from
 Department,Employee, ( select count(*) cnt from Employee ) No_of_Employees
where Department.Department_Id = Employee.Department_Id group by Department_Name,
No_of_Employees.cnt
```

/

Materialized View
Number of Employees with different degrees

```
create materialized view Education_View
 build immediate refresh on
commit
 as select Degree, count(Degree) from
Education group by Degree;
```

Procedure

Locations with less number of employees

```
CREATE OR REPLACE PROCEDURE Unimportant_Locations(l_NOFEmployees IN Number)
 IS
 l_wl NUMBER;
 l_emp NUMBER;

 BEGIN
 SELECT COUNT(*) INTO l_wl
 FROM Work_Location
 WHERE Number_Of_Employees LIKE l_NOFEmployees;

select count(*)
into l_emp from
Employee e inner
 Join
 Work_Location w
 on e.Employee_Id = w.Employee_Id
 where
 w.Number_Of_E mployees LIKE l_NOFEmployees;

 IF l_wl < 5 THEN
 DELETE FROM Work_Location
 WHERE Number_Of_Employees = l_NOFEmployees;
 END IF;
```

```
EXCEPTION WHEN no_data_found THEN
DBMS_OUTPUT.PUT_LINE('No Such Data Available');
END;
```

Explicit Cursor

```
declare cursor salaries(p_hourly in number)
is select * from Salary where
 Hourly_Pay=p_hourly;

            l_sal Salary%rowtype;
            begin

      dbms_output.put_line(' Extracting hourly pay'); open
salaries(30);
           loop
             fetch salaries into l_sal;
                  exit when salaries%notfound;
                  dbms_output.put('For Account ' || l_sal.Account_Id || ' Hourly Pay is ');
dbms_output.put_line(l_sal.hourly_pay);
               end loop;
               close salaries;
      end;
/
```

Pre-Defined Exception

```
declare
l_attendance Attendance%rowtype; New_Exception
exception;
begin l_attendance.Attendance_Id := 90; l_attendance.Hours_Worked :=
 'AS'; insert into Attendance (Attendance_Id,Hours_Worked) values
( l_attendance.Attendance_Id, l_attendance.Hours_Worked ); exception
when VALUE_ERROR then dbms_output.put_line('We encountered the VALUE_ERROR
exception');
end;
/
```

Explicit Cursor and Pre-Defined Cursor Together

```
declare cursor salaries(p_hourly in number) is
select * from Salary where Hourly_Pay=p_hourly;
l_sal Salary%rowtype;
 begin
 dbms_output.put_line('Gett ing hourly pay');
open salaries(30);
 loop fetch salaries into l_sal;
 exit when salaries%notfound;
dbms_output.put('For Account ' || l_sal.Account_Id || ' Hourly Pay is ');
 dbms_output.put_line(l_sal.hourly_pay);
end loop;
open salaries(30);
exception when CURSOR_ALREADY_OPEN then
dbms_output.put_line('No Need to open cursor again'); close salaries;
end;
/
```

External table

```
create table Salary_External
( Salary_Id NUMBER,
Gross_Salary NUMBER,
 Hourly_Pay NUMBER,
 State_Tax NUMBER,
 Federal_Tax NUMBER,
 Account_Id NUMBER
 )


organization external ( type
        oracle_loader default
        directory ext_Salaries access
        parameters (
```

```
                fields terminated by ',' )
        location ('Salary.csv')
   ) reject limit
unlimited
/
```

Ref cursor

```
declare type emp_dept_rec is
record(
        Employee_Id number,
        First_Name varchar2(66),
        Department_Name varchar2(37)
         );


          type emp_dept_refcur_type is ref cursor
                  return emp_dept_rec;

    employee_refcur emp_dept_refcur_type;
```

```
    emp_dept emp_dept_rec;

begin

open employee_refcur for
 select e.Employee_Id,

                                e.First_Name || ' ' || e.Last_Name "Employee Name",
                                d.Department_Name
                        from Employee e, Department d
                        where e.Department_Id =
              d.Department_Id and rownum < 5
                    order by e.Employee_Id;



                fetch employee_refcur into emp_dept; while employee_refcur%FOUND
loop            dbms_output.put(emp_dept.First_Name || "'s department is ');
dbms_output.put_line(emp_dept.Department_Name); fetch employee_refcur
into emp_dept;
end loop;
end;
/
```

Transaction

```
INSERT INTO Employee VALUES (111,'Priyanka','Jonas',to_date('14-NOV-16', 'dd-MON-yyyy'),'New York
City','New York',1);



commit;



INSERT INTO Employee VALUES (112,'John','Vincent',to_date('21-JUN-18',
'ddMONyyyy'),'Boston','Massachusetts',2);


SAVEPOINT A1;
INSERT INTO Employee VALUES (113,'Pratik','Panhale',to_date('13-SEP-19',
'ddMONyyyy'),'Chicago','Illinois',3);



SAVEPOINT A2;
```

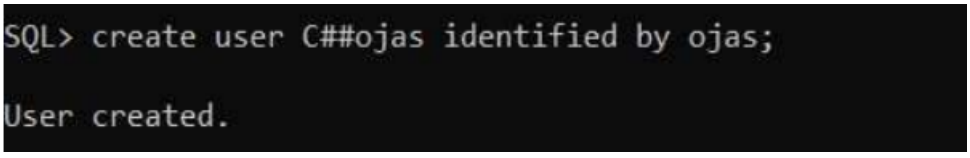
ROLLBACK A1;

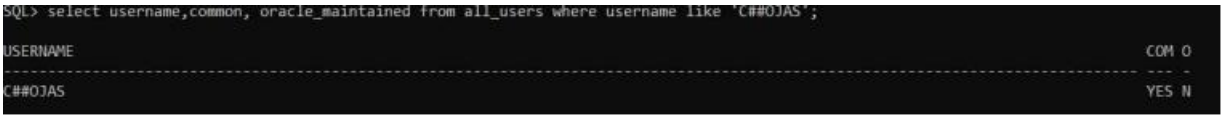
Relational View

```
create or replace view
salary_range_calculator as select
 e.First_Name, s.Hourly_Pay
from Employee e
 inner join AccountDetails a on
 e.Employee_Id = a.Employee_Id
inner join Salary s on a.Account_Id = s.Account_Id
where s.Hourly_Pay = 30;
```

SNAPSHOTS:

1.Created Command User on sysdba





```
SQL> connect sys@orcl as sysdba
Enter password:
Connected.
SQL> grant all privileges to C##OJAS;

Grant succeeded.

SQL> disconnect
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
SQL> connect C##OJAS
Enter password:
Connected.
```

2.Create Pluggable Database

```
SQL> create pluggable database payroll_management_system
  2  ADMIN USER HR_ADMIN identified by hr;
ADMIN USER HR_ADMIN identified by hr
                   *
ERROR at line 2:
ORA-65016: FILE_NAME_CONVERT must be specified


SQL> alter system set pdb_file_name_convert = 'C:\Users\phansekar.o\Oracle\oradata\ORCL\pdbseed\','C:\Users\phansekar.o\Oracle\oradata\ORCL\payroll_management_system\' scope=both;

System altered.

SQL> create pluggable database payroll_management_system
  2  ADMIN USER HRADMIN identified by hradmin;

Pluggable database created.
```

```
SQL> show pdbs;

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         5 PAYROLL_MANAGEMENT_SYSTEM      READ WRITE NO
```

```
SQL> connect sys@orcl as sysdba
Enter password:
Connected.
SQL> show pdbs;

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB                        MOUNTED
         4 OJPDB                          READ WRITE NO
         5 PAYROLL_MANAGEMENT_SYSTEM      MOUNTED
SQL> alter pluggable database payroll_management_system open read write;

Pluggable database altered.

SQL> select status from v$instance;

STATUS
------------
OPEN
```

```
SQL> show pdbs;

    CON_ID CON_NAME                       OPEN MODE  RESTRICTED
---------- ------------------------------ ---------- ----------
         2 PDB$SEED                       READ ONLY  NO
         3 ORCLPDB                        MOUNTED
         4 OJPDB                          READ WRITE NO
         5 PAYROLL_MANAGEMENT_SYSTEM      READ WRITE NO
SQL> disconnect
Disconnected from Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
Version 19.3.0.0.0
SQL> connect C##OJAS
Enter password:
```

```
SQL> connect sys@payroll_management_system as sysdba
Enter password:
Connected.
```

3.Inline View

```
SQL> select Department_Name, count(*),
  2  to_char((count(*)/No_of_Employees.cnt)*100, '90.99') Percentages
  3  from Department,Employee, ( select count(*) cnt from Employee ) No_of_Employees
  4  where Department.Department_Id = Employee.Department_Id
  5  group by Department_Name, No_of_Employees.cnt
  6  /

DEPARTMENT_NAME                 COUNT(*) PERCEN
------------------------------ ---------- ------
Data Analysis                           1  10.00
Data Science                            1  10.00
Data Engineering                        1  10.00
Human Resources                         1  10.00
Software Development                    1  10.00
Business Intelligence                   1  10.00
Manufacturing                           2  20.00
Quality Control                         2  20.00

8 rows selected.
```

4.Materialized Views

```
SQL> select * from Education_View;

DEGREE                          COUNT(DEGREE)
------------------------------ -------------
Bachelor                                   3
MS                                         4
```

5.Explicit Cursor

```
SQL> declare
  2    cursor salaries(p_hourly in number)
  3    is select *
  4    from Salary
  5    where Hourly_Pay=p_hourly;
  6
  7    l_sal Salary%rowtype;
  8    begin
  9     dbms_output.put_line(' Extracting hourly pay');
 10     open salaries(30);
 11     loop
 12      fetch salaries into l_sal;
 13  exit when salaries%notfound;
 14  dbms_output.put('For Account ' || l_sal.Account_Id || ' Hourly Pay is ');
 15          dbms_output.put_line(l_sal.hourly_pay);
 16  end loop;
 17  close salaries;
 18         end;
 19        /
Extracting hourly pay
For Account 40 Hourly Pay is 30
For Account 44 Hourly Pay is 30
For Account 48 Hourly Pay is 30

PL/SQL procedure successfully completed.
```

6.Index

```
SQL> create index account_ix
  2  on AccountDetails(Bank_Name);

Index created.
```

7.Relational Views

```
SQL> create or replace view salary_range_calculator
  2  as
  3  select e.First_Name, s.Hourly_Pay
  4  from Employee e
  5  inner join AccountDetails a
  6  on e.Employee_Id = a.Employee_Id
  7  inner join Salary s
  8  on a.Account_Id = s.Account_Id
  9  where s.Hourly_Pay = 30;

View created.

SQL> select * from salary_range_calculator;

FIRST_NAME                HOURLY_PAY
------------------------- ----------
Ojas                              30
Anugraha                          30
Kalpita                           30
```

8.Transaction

```
SQL> INSERT INTO Employee VALUES (111,'Priyanka','Jonas',to_date('14-NOV-16', 'dd-MON-yyyy'),'New York City','New York',1);

1 row created.

SQL>
SQL> commit;

Commit complete.

SQL>
SQL> INSERT INTO Employee VALUES (112,'John','Vincent',to_date('21-JUN-18', 'dd-MON-yyyy'),'Boston','Massachusetts',2);

1 row created.

SQL>
SQL> SAVEPOINT A1;

Savepoint created.

SQL>
SQL> INSERT INTO Employee VALUES (113,'Pratik','Panhale',to_date('13-SEP-19', 'dd-MON-yyyy'),'Chicago','Illinois',3);

1 row created.

SQL>
SQL> SAVEPOINT A2;

Savepoint created.

SQL>
SQL> ROLLBACK A1;
ROLLBACK A1
         *
ERROR at line 1:
ORA-02181: invalid option to ROLLBACK WORK


SQL> ROLLBACK TO A1;

Rollback complete.
```

9.External Table

```
SQL> create directory ext_Salaries
  2  as 'C:\Users\phansekar.o\Desktop\Salary.csv'
  3  /

Directory created.

SQL> grant all on directory ext_Salaries to HRADMIN
  2  /

Grant succeeded.

SQL> create table Salary_External (
  2     Salary_Id NUMBER,
  3     Gross_Salary NUMBER,
  4     Hourly_Pay NUMBER,
  5     State_Tax NUMBER,
  6     Federal_Tax NUMBER,
  7     Account_Id NUMBER
  8  )
  9  organization external (
 10  type oracle_loader
 11  default directory ext_Salaries
 12  access parameters (
 13  fields terminated by ',' )
 14  location ('Salary.csv')
 15  )
 16  reject limit unlimited
 17  /

Table created.
```

```
SQL> desc Salary_External;
Name                                      Null?    Type
----------------------------------------- -------- ----------------------------
SALARY_ID                                          NUMBER
GROSS_SALARY                                       NUMBER
HOURLY_PAY                                         NUMBER
STATE_TAX                                          NUMBER
FEDERAL_TAX                                        NUMBER
ACCOUNT_ID                                         NUMBER
```

10.Ref cursor

```
SQL> declare
  2  type emp_dept_rec is record(
  3  Employee_Id number,
  4  First_Name varchar2(66),
  5  Department_Name varchar2(37)
  6  );
  7
  8  type emp_dept_refcur_type is ref cursor
  9  return emp_dept_rec;
 10
 11  employee_refcur emp_dept_refcur_type;
 12
 13  emp_dept emp_dept_rec;
 14  begin
 15  open employee_refcur for
 16  select e.Employee_Id,
 17     e.First_Name || ' ' || e.Last_Name "Employee Name",
 18     d.Department_Name
 19  from Employee e, Department d
 20  where e.Department_Id = d.Department_Id
 21  and rownum < 5
 22  order by e.Employee_Id;
 23
 24  fetch employee_refcur into emp_dept;
 25  while employee_refcur%FOUND loop
 26  dbms_output.put(emp_dept.First_Name || '''s department is ');
 27  dbms_output.put_line(emp_dept.Department_Name);
 28  fetch employee_refcur into emp_dept;
 29  end loop;
 30  end;
 31  /
Ojas Phansekar's department is Human Resources
Vrushali Patil's department is Software Development
Pratik Parija's department is Data Analysis
Chetan Mistry's department is Data Science

PL/SQL procedure successfully completed.
```

11.Pre-defined Exception

```
SQL> declare
  2  l_attendance Attendance%rowtype;
  3  begin
  4  l_attendance.Attendance_Id := 90;
  5  l_attendance.Hours_Worked := 'AS';
  6  insert into Attendance (Attendance_Id,Hours_Worked)
  7  values ( l_attendance.Attendance_Id, l_attendance.Hours_Worked );
  8  exception
  9  when VALUE_ERROR then
 10  dbms_output.put_line('We encountered the VALUE_ERROR exception');
 11  end;
 12  /
We encountered the VALUE_ERROR exception

PL/SQL procedure successfully completed.
```

12.Procedure

```
SQL> CREATE OR REPLACE PROCEDURE Unimportant_Locations(l_NOFEmployees IN Number)
  2  IS
  3    l_wl NUMBER;
  4    l_emp NUMBER;
  5
  6  BEGIN
  7    SELECT COUNT(*) INTO l_wl
  8    FROM Work_Location
  9    WHERE Number_Of_Employees LIKE l_NOFEmployees;
 10
 11
 12    select count(*)
 13    into l_emp
 14    from Employee e
 15    inner join Work_Location w
 16    on e.Employee_Id = w.Employee_Id
 17    where w.Number_Of_Employees LIKE l_NOFEmployees;
 18
 19    IF l_wl < 5 THEN
 20      DELETE FROM Work_Location
 21      WHERE Number_Of_Employees = l_NOFEmployees;
 22  END IF;
 23
 24    EXCEPTION WHEN no_data_found THEN
 25    DBMS_OUTPUT.PUT_LINE('No Such Data Available');
 26  END;
 27  /

Procedure created.

SQL> execute Unimportant_Locations(5);

PL/SQL procedure successfully completed.

SQL> select * from Work_Location;

LOCATION_ID LOCATION                 NUMBER_OF_EMPLOYEES CITY                 STATE                EMPLOYEE_ID
----------- ------------------------ ------------------- -------------------- -------------------- -----------
         71 North                                      4 New York City        New York                     101
         72 North                                      4 Boston               Massachusetts                102
         73 North                                      4 Chicago              Illinois                     103
         74 North                                     89 Miami                Florida                      104
         75 South                                     90 Atlanta              Georgia                      105
         76 South                                    100 San Mateo            California                   106
         77 South                                      4 San Francisco        California                   107
```

13.Predefined Exception and Explicit Cursor

```
SQL> declare
  2    cursor salaries(p_hourly in number)
  3    is select *
  4    from Salary
  5    where Hourly_Pay=p_hourly;
  6
  7    l_sal Salary%rowtype;
  8    begin
  9     dbms_output.put_line('Getting hourly pay');
 10     open salaries(30);
 11     loop
 12      fetch salaries into l_sal;
 13  exit when salaries%notfound;
 14  dbms_output.put('For Account ' || l_sal.Account_Id || ' Hourly Pay is ');
 15          dbms_output.put_line(l_sal.hourly_pay);
 16  end loop;
 17  open salaries(30);
 18  exception
 19  when CURSOR_ALREADY_OPEN then
 20  dbms_output.put_line('No Need to open cursor again');
 21  close salaries;
 22         end;
 23        /
Getting hourly pay
For Account 40 Hourly Pay is 30
For Account 44 Hourly Pay is 30
For Account 48 Hourly Pay is 30
No Need to open cursor again

PL/SQL procedure successfully completed.
```

Conclusion of the Project Employee Payroll Management System:

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding has also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the school. The objective of software planning is to provide a frame work that enables the manger to make

reasonable estimates made within a limited time frame at the beginning of the software

project and should be updated regularly as the project progress. At the end it is concluded that we have made effort on following points...

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project
- The description of Purpose. Scope, and applicability. We define the problem on which we are working in the project
- We describe the requirement Specifications of the system and the actions that can be done on these things.
- We understand the problem domain and produce a model of the system, which describes operations that can be performed on the system. We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system.
- Finally, the system is implemented and tested according to test cases.

Future Scope of the Project:

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can add printer in future.
- We can give more advance software for Payroll Management System including more facilities. We will host the platform on online servers to make it accessible worldwide
- Integrate multiple load balancers to distribute the loads of the system Create the master and slave database structure to reduce the overload of the • database queries
- Implement the backup mechanism for taking backup of codebase and database • on regular basis on different servers
- The above-mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of Payroll and Employee. Also, as it can be seen that now-a-days the players are versatile. i.e. so there is a scope for introducing a method to maintain the Payroll Management System Enhancements can be done to maintain all the Payroll, Employee, Salary, Appraisals, Working Points.
- We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them. In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process

References

Websites:

• www.w3schools.com

• www.tutorialspoint.com

• www.youtube.com