***A REPORT ON***

***Employee Management System***

Sanjivani Rural Education Society’s   
 Sanjivani College of Engineering, Kopargaon-423603  
 (An Autonomous Institute Affiliated to Savitribai Phule Pune University,Pune)   
 (NAAC ‘A’ Grade Accredited, ISO 9001:2015 Certified)   
 Department of Computer Engineering   
(NBA Accredited)



***Submitted by:-***

|  |  |
| --- | --- |
| ***Roll No*** | ***Name*** |
| 27 | Gaurav Maneshkumar Borawake |
| 59 | Rutuja Sanjay Gunjal |
| 60 | Dewanshi Lokesh Gupta |
| 62 | Adesh Arun Jadhav |

***CERTIFICATE***

This is to certify that the project report based on the **“Employee Management System** ” is successfully completed by S.Y.B.Tech computer Engineering students in the academic year 2022-23 under the respective courses.

Date:-

Place:-

Signature:-

***ACKNOWLEDGEMENT***

We would like to express our heartfelt gratitude to Dr.P.N.Kalavadekar, Dr.P.Saiprasad, Prof.V.N.Nirgude, Prof.P.N.Dhanrao under whose guidance we were able to complete the project and our honourable director Dr.A.G.Thakur who gave us this wonderful oppourtunity to do this project.

***Contents***

1.Introduction

2. Software Requirements Specification

3.Purpose of the system

4.Requirement Analysis

5. Use Case Diagrams

6.Advanced Data Structure

7.Pseudocode

9.Database Management System .

10.Operating System and Administration

11.Conclusion

***INTRODUCTION***

* An employee management system is a software, that helps your employees to give their best efforts every day to achieve the goals of your organization.
* It guides and manages employees efforts in the right direction.
* It also securely stores and manages personal and other work-related details for your employees.
* That makes it easier to store and access the data when there is a need.
* In the employee management system, you can manage admin activities in an easier and quicker way.
* Employees are an important part of your organization it is their work that ultimately contributes to the bottom line of the company.
* It is an important part of HR management.
* It also helps to employee engagement and performance management brings down costs and increases productivity.

***Software Requirements Specification***

* Every organization has different employee management needs, therefore we have designed exclusive employee management system that is adapted to the managerial requirements.
* This is designed to assist in strategic planning and will help you ensure that your organization is equipped with the right level of human resources for your future goals.
* Also, for those busy executive who are always on the go, our systems come with,remote access features, which will allow you to manage your workforce anytime, at all times.
* These systems will ultimately allow you to better manage resources.
* One of the main features in employee management system is time tracking for employees.
* Effective time tracking mechanism saves both time and money for the

organization.

***PURPOSE OF THE SYSTEM:-***

* As we know in any technical educational office,  the manager of  office used to  spare lot of  time even after the normal office hours either at home or office for preparation of daily/weekly report and other necessary record.
* Now with the help of  this system ,the manager has the information on his finger tips and can easily prepare a record based on their requirements apart  from daily/weekly report.
* Finally,We can say that this system will not only automate the process but save the valuable time of the office manager, which can be well utilized by this  institute.
* This will be an additional advantage and management of manpower based on their free time from his normal duty.

***REQUIREMENT ANALYSIS:-***

* The aim of requirement analysis is to understand the exact requirement of the customer and to document and to document them properly.
* Requirement analysis involves obtaining a clear and thorough understanding of the product to be developing with a view to remove all ambiguities and inconsistencies from the initial customer perception the problem.
* The question arising during the requirement analysis phases is: -
* What is the problem?
* Why is it important to solve the problem?
* What are the possible solutions to the problem?
* What exactly are the data inputs and data outputs by system?
* What are the likely the complex cities that might arise while solving the problem?

**During requirement analysis there exist mainly two activities.**

1.Requirement gathering

2.Analysis of gathered requirements.

1.**Requirement gathering**: -

This involves interviewing the end user and customers to collect all possible information regarding the bank.

2.**Analysis of gathered requirement**: -

The main purchase of analysis is to collect information to clearly understand the exact requirement of customer and resolve anomalies, conflicts and inconsistencies in the gathered requirement.

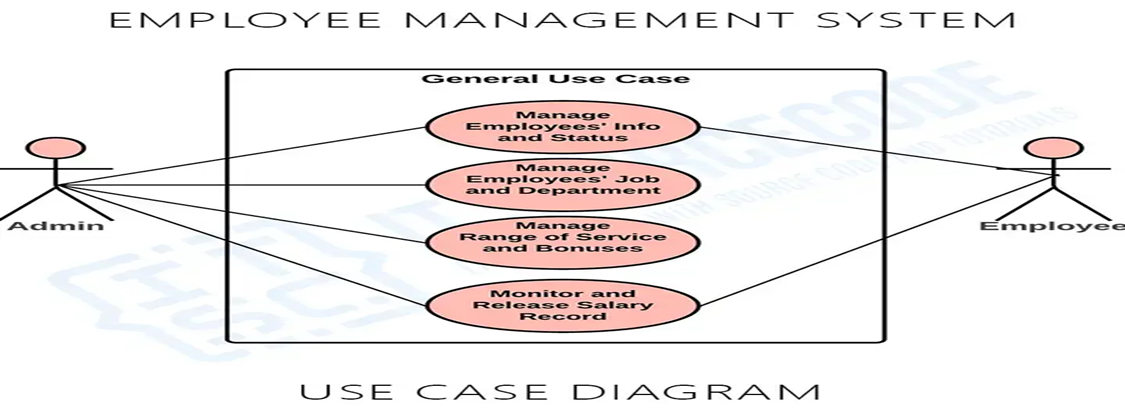
***Use Case Diagrams***

The use case diagram shows the sample behavior of the employee management system. It includes the project functions using use cases, actors, and their connections.

The diagram assists you to define and organize project needs. This also provides a clear picture of the user and system relationships. Therefore, this diagram depicts the complex functions of a system including how the user reacts to it.

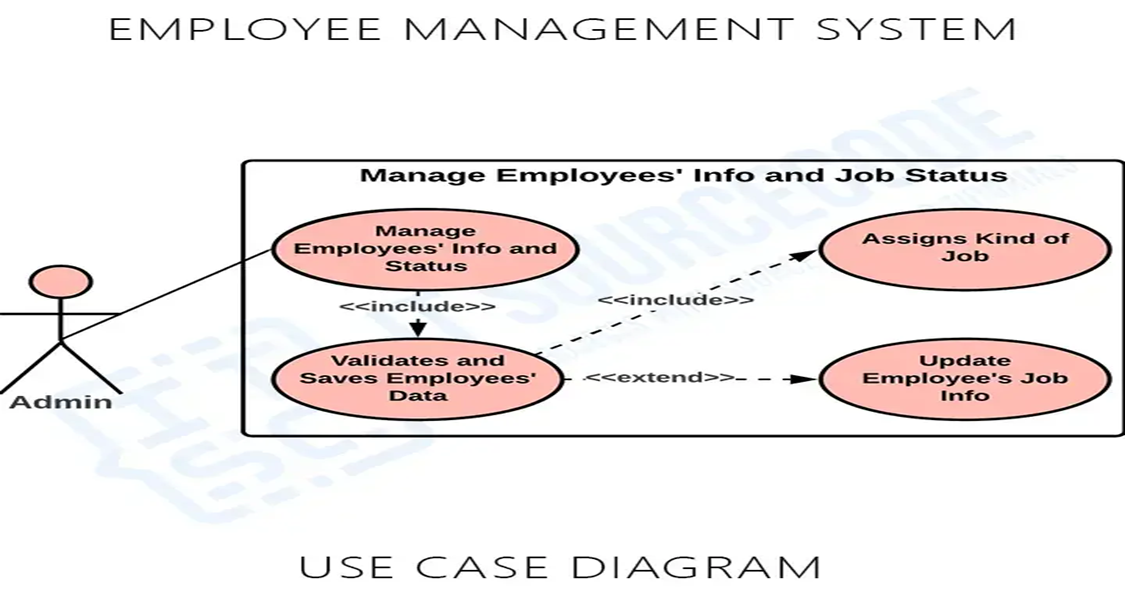
**1.General Use Case diagram:-**

This diagram shows the general processes or function that the system could do that is based on the transactions done by the admin or employer in managing the their employees.

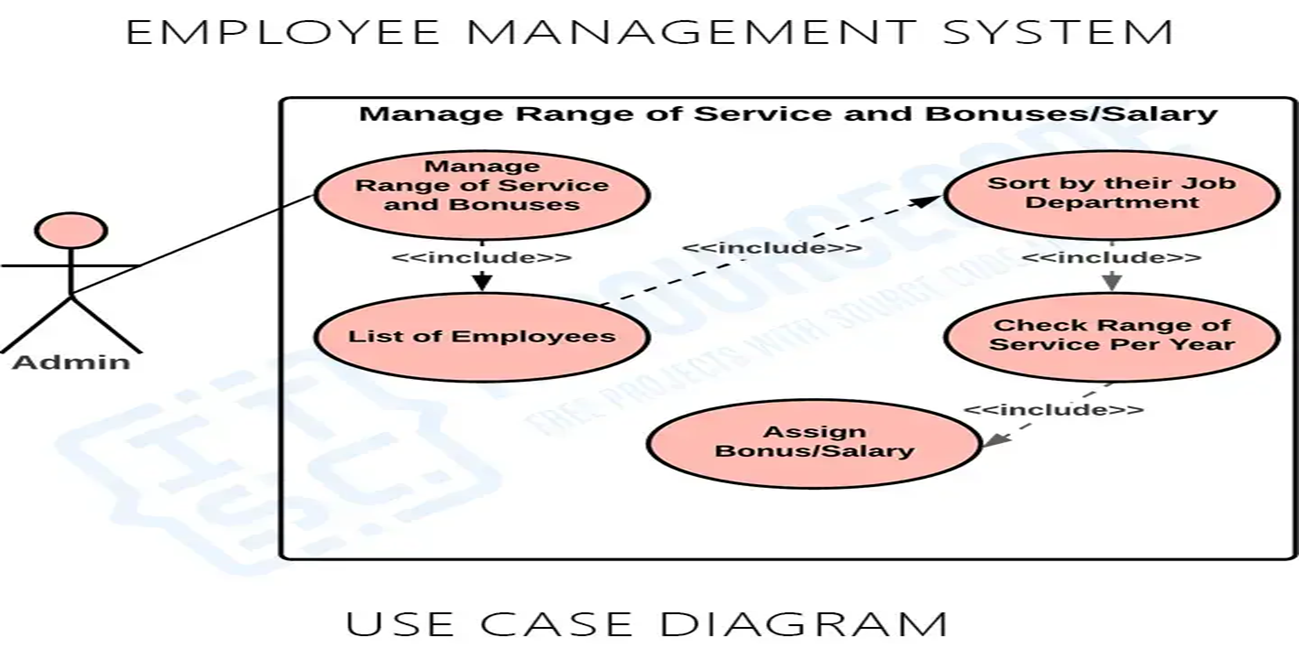
******

***2. Monitor and Manage Employees' Information and Status:-***

This is where the admin of the system could manage and monitor their employees’ information and status. In this process they were able to assign the field of work of their employee as well monitor their outputs and range of service.



**3. *Manage Employees salary Information and Status:-***



***Advanced Data Structures***

**Tree:-**It is a non linear data structure.

Trees mainly represent data containing the hierarchical relationshsip between elements.

Tree has following elements:-

**1.Node:-** Node is the information of any type.

**2.Root Node:**- The first node is called as root node.

**3.Leaf Node:-** The node which doesn’t have any other child is called the leaf node.

**Binary tree :-**

A tree in which each node has atmost two childs is known as binary tree.

**Binary tree Traversal:-**

Tree traversal means to visit each an every node exactly once.

There are three types of tree traversal:-

**1.Inorder Tree Traversal:-**

In this traversal method the left subtree is visited first then the data/information and later the right subtree is visited

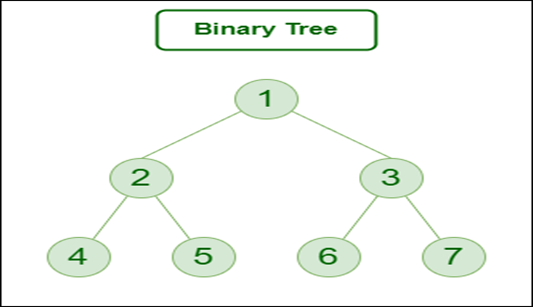
**2.Preorder Tree Traversal:-**

In this traversal method we visit the data/information first followed by left subtree and later the right subtree.

**3.Postorder Tree Traversal:-** In this traversal method we visit the left subree, then the right subtree and the data/information at last.

**Basic Operation Of Tree:**

* **Create** – Create a tree in data structure.
* **Insert** − Inserts data in a tree.
* **Search** − Searches specific data  in a tree to check it is present or not.
* **Preorder** **Traversal** – Perform Traveling a tree in a pre-order manner in data structure .
* **Inorder** **Traversal** – Perform Traveling a tree in an in-order manner.
* **Postorder** **Traversal** – Perform Traveling a tree in a post-order manner.



Here,

Node 1 is the root node

1 is the parent of 2 and 3

2 and 3 are the siblings

4, 5, 6 and 7 are the leaf nodes

1 and 2 are the ancestors of 5

***Pseudocode***

#include <iostream>

using namespace std;

struct Employee {

int id;

string name;

string position;

Employee\* left;

Employee\* right;

};

Employee\* root = NULL;

// Function to create a new employee node

Employee\* createNode(int id, string name, string position) {

Employee\* temp = new Employee;

temp->id = id;

temp->name = name;

temp->position = position;

temp->left = NULL;

temp->right = NULL;

return temp;

}

// Function to insert a new employee node

Employee\* insert(Employee\* root, int id, string name, string position)

{

if (root == NULL)

{

return createNode(id, name, position);

}

if (id < root->id)

{

root->left = insert(root->left, id, name, position);

}

else if (id > root->id)

{

root->right = insert(root->right, id, name, position);

}

return root;

}

// Function to search for an employee

Employee\* search(Employee\* root, int id)

{

if (root == NULL || root->id == id)

{

return root;

}

if (root->id < id)

{

return search(root->right, id);

}

return search(root->left, id);

}

**// Function to find the minimum value in a binary search tree**

Employee\* minValueNode(Employee\* node)

{

Employee\* current = node;

while (current && current->left != NULL)

{

current = current->left;

}

return current;

}

// Function to delete an employee node

Employee\* deleteNode(Employee\* root, int id) {

if (root == NULL) {

return root;

}

if (id < root->id) {

root->left = deleteNode(root->left, id);

}

else if (id > root->id) {

root->right = deleteNode(root->right, id);

}

else {

if (root->left == NULL) {

Employee\* temp = root->right;

delete root;

return temp;

}

else if (root->right == NULL) {

Employee\* temp = root->left;

delete root;

return temp;

}

Employee\* temp = minValueNode(root->right);

root->id = temp->id;

root->name = temp->name;

root->position = temp->position;

root->right = deleteNode(root->right, temp->id);

}

return root;

}

// Function to display the employee tree in order

void display(Employee\* root) {

if (root != NULL) {

display(root->left);

cout << "ID: " << root->id << ", Name: " << root->name << ", Position: " << root->position << endl;

display(root->right);

}

}

int main() {

int choice, id;

string name, position;

Employee\* temp;

do{

cout << "--------------------------" << endl;

cout << "Employee Management System" << endl;

cout << "--------------------------" << endl;

cout << "1. Add an Employee" << endl;

cout << "2. Search for an Employee" << endl;

cout << "3. Delete an Employee" << endl;

cout << "4. Display all Employees" << endl;

cout << "5. Exit" << endl;

cout << "Enter your choice: ";

cin >> choice;

switch (choice) {

case 1:

cout << "Enter Employee ID: ";

cin >> id;

cout << "Enter Employee Name: ";

cin >> name;

cout << "Enter Employee Position: ";

cin >> position;

root = insert(root, id, name, position);

break;

case 2:

cout << "Enter Employee ID to Search: ";

cin >> id;

temp = search(root, id);

if (temp != NULL)

{

cout << "ID: " << temp->id << ", Name: " << temp->name << ", Position: " << temp->position << endl;

}

else

{

cout << "Employee not found." << endl;

}

break;

case 3:

cout << "Enter Employee ID to Delete: ";

cin >> id;

root = deleteNode(root, id);

break;

case 4:

cout << "Employee List" << endl;

cout << "-------------" << endl;

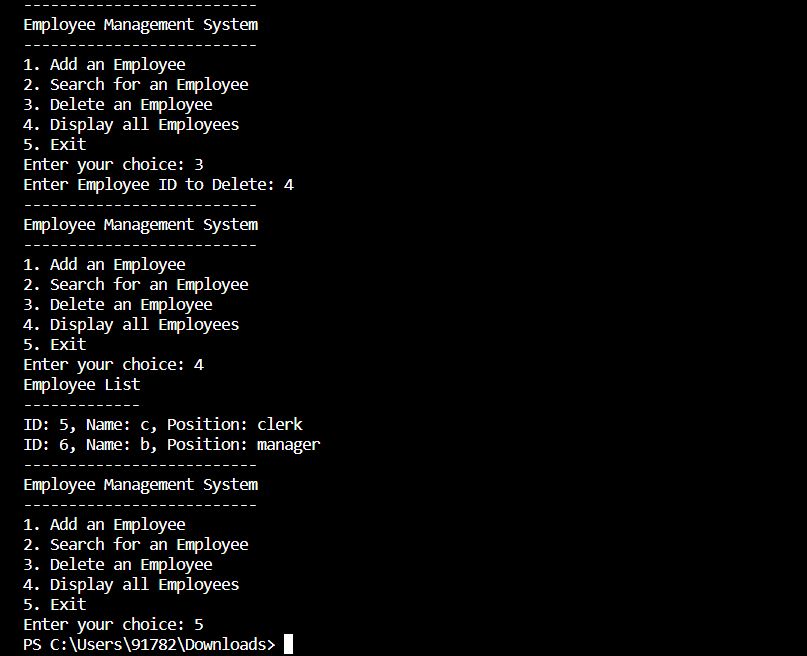
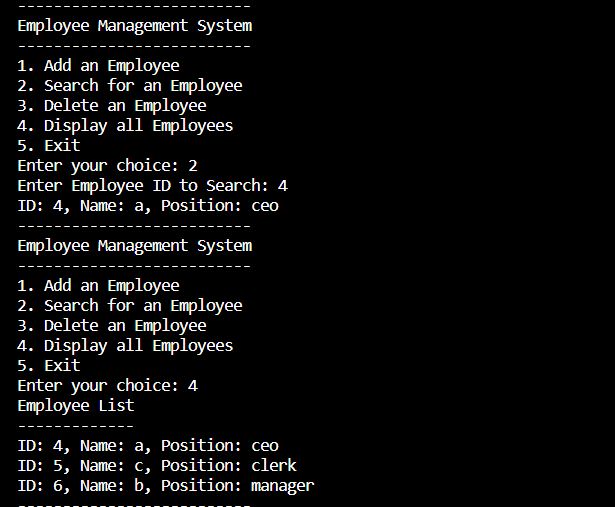
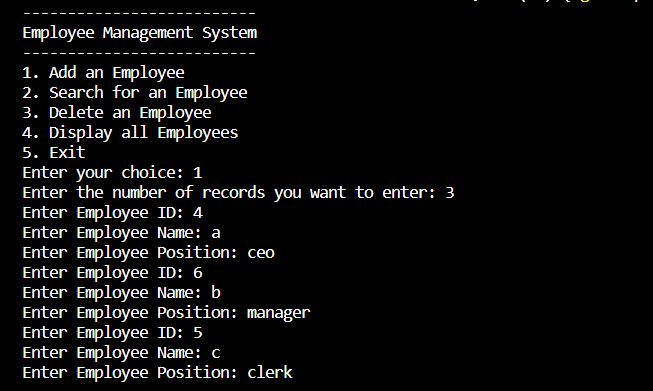
display(root);

break;

}

}while(choice!=5);

return 0;}



***Database Management System***

**Basic Commands:**

* SELECT - extracts data from a database
* UPDATE - updates data in a database
* DELETE - deletes data from a database
* INSERT INTO - inserts new data into a database
* CREATE DATABASE - creates a new database
* ALTER DATABASE - modifies a database
* CREATE TABLE - creates a new table
* ALTER TABLE - modifies a table
* DROP TABLE - deletes a table
* CREATE INDEX - creates an index (search key)
* DROP INDEX - deletes an index

**Procedure:-**

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again.

So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

**Stored Procedure Syntax**

CREATE PROCEDURE procedure\_name  
AS  
sql\_statement  
GO;

Execute a Stored Procedure

EXEC procedure\_name;

**Triggers in MYSQL:**

A trigger is a set of actions that are run automatically when a specified change operation (SQL INSERT, UPDATE, or DELETE statement) is performed on a specified table.

Triggers are useful for tasks such as enforcing business rules, validating input data, and keeping an audit trail.

**How to create triggers in MySQL?**

We can use the CREATE TRIGGER statement for creating a new trigger in MySQL. Below is the syntax of creating a trigger in MySQL:

CREATE TRIGGER trigger\_name

    (AFTER | BEFORE) (INSERT | UPDATE | DELETE)

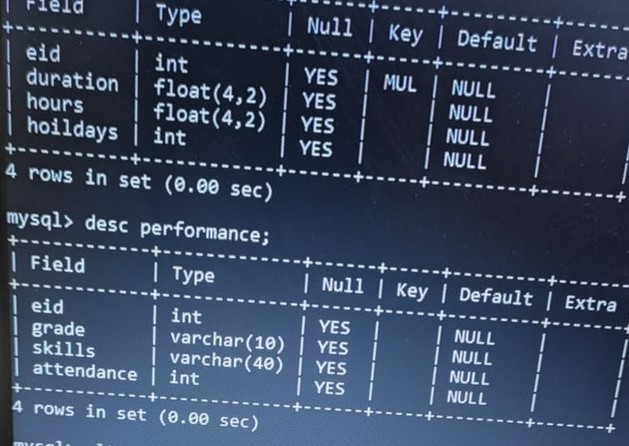
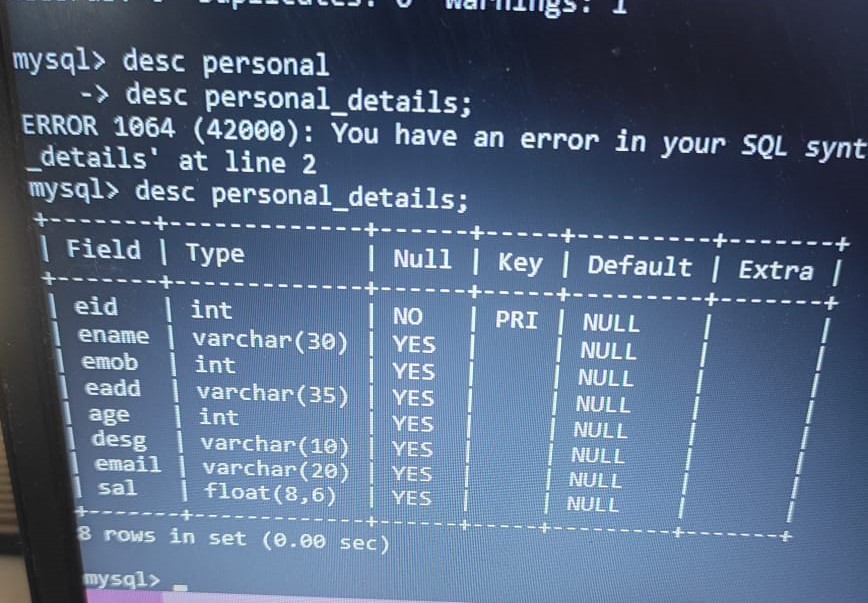
         ON table\_name FOR EACH ROW

         BEGIN

        --variable declarations

        --trigger code

        END;

***Operating System and Administration***

**Concepts Used:**

Ubuntu OS installed and used throughout the project

Shell script done using OSA and ubantu shell commands, taught in the practical sessions

**Shell Script:-**

insert\_personal()

{

echo "Enter the personal details of the employees (ID,Name,Mobile,Address,Age and Salary)"

read i;

read n;

read m;

read add;

read age;

read sal;

echo "use emps" > insert.sql

echo "insert into personal\_details values($i,'$n',$m,'$add',$age,$sal);">> insert.sql

echo "Record stored"

echo "Select \* from personal\_details;" >> insert.sql

sudo mysql -u root < insert.sql

}

insert\_schedule()

{

echo "Enter the work schedule of the employees(Duration and no. of holidays)"

read d;

read h;

echo "use emps" > insert1.sql

echo "insert into work\_schedule values($d,$sh);">> insert1.sql

echo "Record stored"

echo "Select \* from work\_details;" >> insert1.sql

sudo mysql -u root < insert1.sql

}

insert\_performance()

{

echo "Enter the skills and attendance"

read s;

read a;

echo "use emps" > insert3.sql

echo "insert into performance values($s,$a);">> insert3.sql

echo "Record stored"

echo "Select \* from performance;" >> insert3.sql

sudo mysql -u root < insert3.sql

}

while true

do

echo "1.Enter the personal details"

echo "2.Enter the work schedule of the employee"

echo "3.Enter the performance of the employee"

echo "4.Exit"

echo "Enter your choice"

read ch

case $ch in

1)insert\_personal

;;

2)insert\_schedule

;;

3)insert\_performance

;;

4) echo "Program Terminated"

exit

;;

esac

done

***Conclusion***

* From the above project we have gained a through knowledge about every course.
* In software Engineering we learnt about the Software requirement specifications,various use case diagrams and test cases.
* In Operating System and Administration we learnt about writing shell scripts and its connecctivity with the tables in mysql.
* In Database management System we learnt about the create the tables and writing procedures and triggers.
* In Advanced Data structures we leant about the concept of tree and basic operations performed on trees.

***References used :-***

1.https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/

2. https://www.cyberciti.biz/faq/using-mysql-in-shell-scripts/

***Thank You!***