

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", MACHHE, BELAGAVI – 590018



Mini Project Report on

EMPLOYEE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the V semester

Bachelor of Engineering

in

Computer Science and Engineering

of

Visvesvaraya Technological University, Belagavi

by

Ms. Tulika Paul (1CD20CS171)

Ms. Precilla Mary B (1CD20CS122)

Under the Guidance of

Ms. Bhavana P
Assistant Professor,
Dept of CSE,CITech

Ms. Priyadarshini M
Assistant Professor,
Dept of CSE,CITech



**Department of Computer Science and Engineering
CAMBRIDGE INSTITUTE OF TECHNOLOGY, BENGALURU – 560036
2022-2023**

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K.R. Puram, Bengaluru – 560036

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

Certified that **Ms. Tulika Paul** and **Ms. Precilla Mary B** bearing USN **1CD20CS171**, **1CD20CS122**, respectively are bonafide students of **Cambridge Institute of Technology**, has successfully completed Mini Project entitled “**Employee Management System**” in partial fulfillment of the requirements for V semester **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year 2022 - 2023. It is certified that all Corrections / Suggestions indicated for Internal Assessment have been incorporated in the report. The Database Project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said semester.

Internal Guides:

1) **Ms. Bhavana P**

Dept. of CSE., CiTech.

Head of the Dept.

Dr. Shashikumar D. R.

Dept. of CSE., CiTech.

2) **Ms. Priyadarshini M**

Dept. of CSE., CiTech.

Examiners:

1)

2)

ACKNOWLEDGEMENT

We are extremely thankful to **Dr. G Indumathi**, Principal, CITech., Bengaluru, for providing us the academic ambience and everlasting motivation to carry out this work and shaping our careers.

We express our sincere gratitude to **Dr. Shashikumar D. R.**, HOD, Dept. of Computer Science and Engineering, CITech., Bengaluru, for his stimulating guidance, continuous encouragement and motivation throughout the course of present work.

We also wish to extend our thanks to **Ms. Bhavana P**, Assistant Professor Dept. of Computer Science and Engineering, CITech., Bengaluru and **Ms. Priyadarshini M**, Assistant Professor, Dept. of Computer Science and Engineering, CITech., Bengaluru, for their expert guidance and constructive suggestions to improve the quality of this work.

We would also like to thank all other teaching and technical staffs of Department of Computer Science and Engineering, who have directly or indirectly helped us in the completion of this Project Work.

And lastly we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in the successful completion of this project.

Tulika Paul (1CD20CS171)

Precilla Mary B (1CD20CS122)

ABSTRACT

Introduction:

It maintains the information about the personal and official details of the employees. It is developed to override the problems prevailing in the practicing manual system. Employee Management System is a distributed application, developed to maintain the details of employees working in any organization.

Project:

The objective of this project is to provide a comprehensive approach towards the management of employee information. Provides full functional reports to the management of company. To develop a well designed database to store employee information. This project aims to simplify the task of maintaining records of the employees of Company.

Objective:

The main objective of the application is to maintain the details of employees, the working in any organization; objective of this project is to provide a comprehensive approach towards the management of employee.

CONTENTS

CHAPTERS		PAGE NO
Chapter 1	Introduction	1
Chapter 2	Requirements	3
Chapter 3	Entity Relationship Diagram	4
Chapter 4	Schema Diagram	6
Chapter 5	Implementation	8
Chapter 6	Snapshots	25
	Conclusion	29
	References	30

CHAPTER 1

INTRODUCTION

Most of the contemporary Information systems are based on the Database technology as a collection of logically related data, and DBMS as a software system allowing the users to define, create, maintain and control access to the database. The process of constructing such kind of systems is not so simple. It involves a mutual development of application program and database. The application program is actually the bridge between the users and the database, where the data is stored. Thus, the well-developed application program and database are very important for the reliability, flexibility and functionality of the system. The so defined systems differentiate to each other and their development comprises a great variety of tasks to be resolved and implemented. Information system suggests a computer technology to be used in order to provide information to users in an organization (for instance), as for the purposes of data transformation into useful information.

A particular case is the Employee Management System. This kind of systems is responsible for storing data of the staff within an organization and generating reports upon request.

1.1 Problem Definition

This Employee Management system consists of an application program, on one hand, and a database (repository of data) on the other. The program performs the basic operations upon the database as retrieving, inserting, updating and deleting data. This project is aimed to manage the information about employees in an organization.

The purpose of the project entitled as EMPLOYEE MANAGEMENT SYSTEM is to computerize the records of the employees using the Employee ID. The records of the employees are stored in Database and it can be accessed by the managers. They can generate reports about projects assigned and leave applied by the employees.

1.2 Need

We are glad to inform that this Employee Management System provides a user friendly software application for easy accessing of records of employees in an organization with proper authentication to the system. It is flexible to update, insert or delete the records of the employees. Assigning of projects, status of project progress as well. Approving the leave applied by the employees can be accessed just by one click.

CHAPTER 2

REQUIREMENTS

2.1 Software Requirement Specifications

Operating System : Windows 10

Front End: Java Swing

Back End: Oracle 11g

Documentation : Microsoft Office 2015 2.2

2.2 Hardware Requirement Specifications

Computer Processor Core i3 Processor Speed 2.3 GHz Processor Hard Disk 400 GB or more
RAM Min 2GB

CHAPTER 3

ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

Peter Chen (a.k.a. Peter Pin-Shan Chen), currently a faculty member at Carnegie-Mellon University in Pittsburgh, is credited with developing ER modeling for database design in the 1970s. While serving as an assistant professor at MIT’s Sloan School of Management, he published a seminal paper in 1976 titled “The Entity-Relationship Model: Toward a Unified View of Data.”

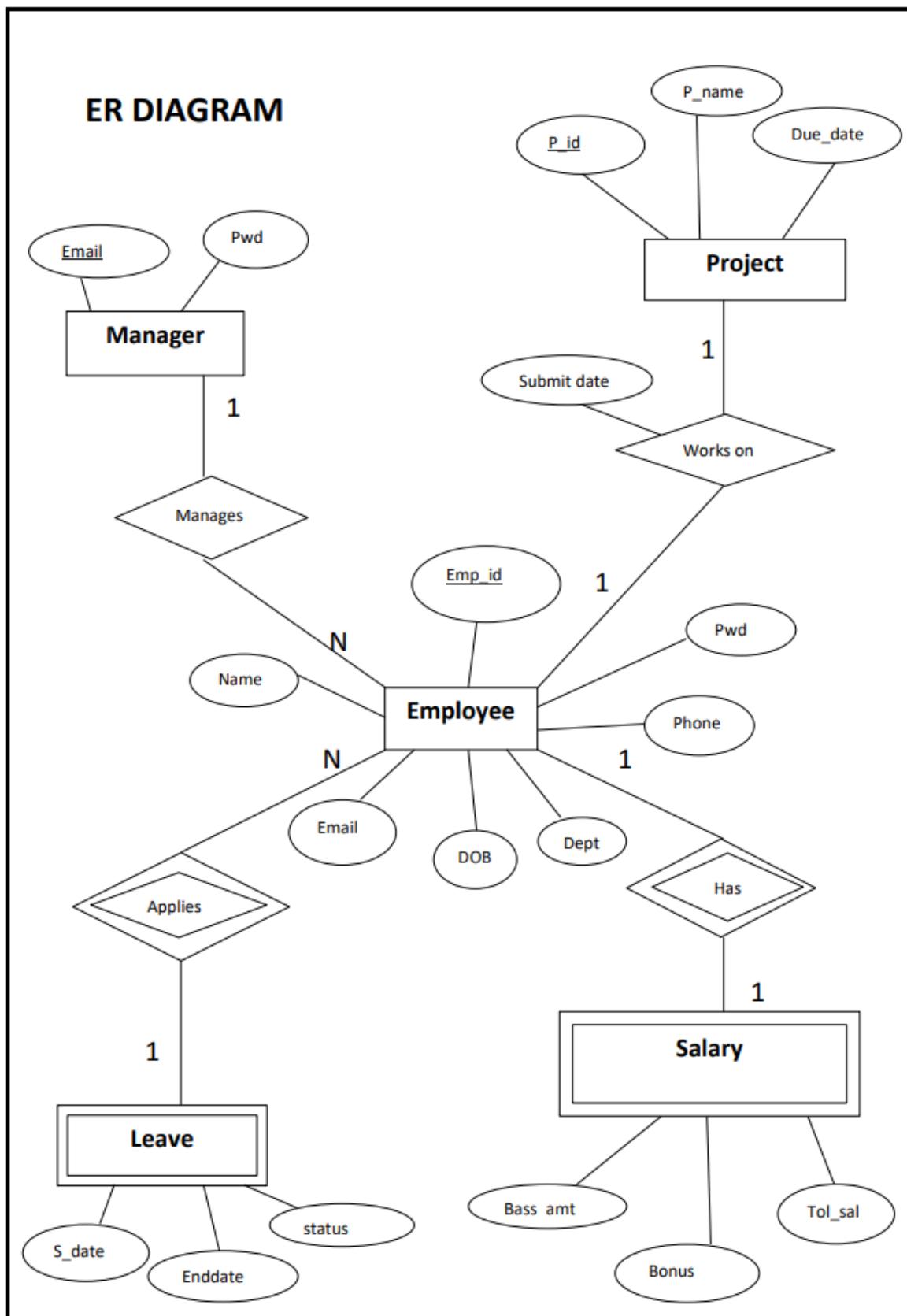
ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here’s a glossary:

Entity : A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.

Relationship : How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.

Attribute: A property or characteristic of an entity. Often shown as an oval or circle.

Cardinality : Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many.



CHAPTER 4

SCHEMA DIAGRAM

4.1 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful. It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information.

A database instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed.

A database schema can be divided broadly into two categories –

- Physical Database Schema – This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.
- Logical Database Schema – This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

Manager

<u>Email</u>	pwd
--------------	-----

Project

<u>Pid</u>	P_name	Due_date	Status	Emp_id
------------	--------	----------	--------	--------

Employee

<u>Emp_id</u>	name	email	DOB	gender	Phone_no	pwd
---------------	------	-------	-----	--------	----------	-----

Leave

S_date	End_date	status	<u>Emp_id</u>
--------	----------	--------	---------------

Salary

sal	<u>Emp_id</u>
-----	---------------

CHAPTER 5

IMPLEMENTATION

5.1 Backend Implementation (database)

```
--create table admin(email varchar2(20) primary key, password varchar2(20));  
--insert into admin values('admin@123','1234');  
  
--create table employee(emp_id number primary key, name varchar2(20),email varchar2(20),  
password varchar2(20),gender varchar2(10),dob date,address varchar2(20),contact  
number);  
  
--insert into employee(1, 'john', 'john@123','5678','male',  
'12-dec-97','bangalore',987654321);  
  
--create table salary(emp_id1 referenece employee(emp_id) on delete cascade,  
Sal number,bonus number,primary key(emp_id1));  
  
--insert into salary(1,30000,12);  
  
--create table project(pr_id number, pname varchar2(20), due date, status varchar2(20),  
emp_id2 referenece employee(emp_id) on delete cascade,primary key(pr_id,emp_id2));  
  
--insert into project(101,'bigdata','12-jan-23','active',1);  
  
--create table leave(emp_id3 referenece employee(emp_id) on delete cascade, reason  
varchar2(20), status varchar2(20), sdate date,edate date, primary key(emp_id3));  
  
--insert into leave values(1,'vacation','in process','12-feb-23','01-mar-23');
```

5.2 Frontend Implementation

Welcome.java

```
public class welcompage extends javax.swing.JFrame {  
  
    public welcompage() {  
  
        initComponents();  
  
    }  
  
    private void initComponents() {  
  
        jLabel1 = new javax.swing.JLabel();  
  
        jLabel3 = new javax.swing.JLabel();  
  
        jPanel1 = new javax.swing.JPanel();  
  
        jLabel4 = new javax.swing.JLabel();  
  
        jButton2 = new javax.swing.JButton();  
  
        jLabel5 = new javax.swing.JLabel();  
  
        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
  
            adminpage ad=new adminpage();  
  
            ad.setVisible(true);  
  
            dispose();  
  
        }  
  
        public static void main(String args[]) {  
  
            for (javax.swing.UIManager.LookAndFeelInfo info :  
                javax.swing.UIManager.getInstalledLookAndFeels()) {  
  
        }
```

```
if ("Nimbus".equals(info.getName())) {  
  
    javax.swing.UIManager.setLookAndFeel(info.getClassName());  
  
    break;  
  
}  
  
java.awt.EventQueue.invokeLater(new Runnable() {  
  
    public void run() {  
  
        new welcompage().setVisible(true);  
  
    }  
  
})
```

Login.java

```
import java.sql.*;  
  
import javax.swing.JOptionPane;  
  
public class adminpage extends javax.swing.JFrame {  
  
    public adminpage() {  
  
        initComponents();  
  
        Connect();  
  
    }  
  
    Connection con;  
  
    PreparedStatement pst;  
  
    ResultSet rs;  
  
    public void Connect()  
  
    {  
  
        try {  
  
            Class.forName("oracle.jdbc.OracleDriver");  
  
        }
```

```
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","tulika","tulika")
;} catch (SQLException ex) {
    Logger.getLogger(adminpage.class.getName()).log(Level.SEVERE, null, ex); }

}

private void initComponents() {
    jPanel2 = new javax.swing.JPanel();
    jButton2 = new javax.swing.JButton();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    txtmail = new javax.swing.JTextField();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    jLabel5 = new javax.swing.JLabel();
    jButton1 = new javax.swing.JButton();
    txtpass = new javax.swing.JPasswordField();
    jLabel1 = new javax.swing.JLabel();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    getContentPane().setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    welcompage w=new welcompage();
    w.setVisible(true);
    dispose();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    try {

```

```
String e=txtemail.getText();
String p=txtpass.getText();

pst = con.prepareStatement("select * from admin where email='"+e+"' and
password='"+p+"');

rs=pst.executeQuery();

if(rs.next())
{
    adminarea a=new adminarea();
    a.setVisible(true);
    dispose();
}

else
{
    JOptionPane.showMessageDialog(this,"invalid user or password");
}

} catch (SQLException ex) {
    Logger.getLogger(adminpage.class.getName()).log(Level.SEVERE, null, ex);
}

}

public static void main(String args[]) {
    for (javax.swing.UIManager.LookAndFeelInfo info :
        javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }

    java.awt.EventQueue.invokeLater(new Runnable() {
```

```
public void run() {  
    new adminpage().setVisible(true);  
}  
}
```

Adminarea.java

```
public class adminarea extends javax.swing.JFrame {  
  
    public adminarea() {  
        initComponents();  
    }  
  
    private void initComponents() {  
  
        jPanel1 = new javax.swing.JPanel();  
        jLabel2 = new javax.swing.JLabel();  
        jLabel3 = new javax.swing.JLabel();  
        jLabel4 = new javax.swing.JLabel();  
        jLabel5 = new javax.swing.JLabel();  
        jLabel6 = new javax.swing.JLabel();  
        jButton1 = new javax.swing.JButton();  
        jButton2 = new javax.swing.JButton();  
        jButton3 = new javax.swing.JButton();  
        jButton4 = new javax.swing.JButton();  
  
        jButton3ActionPerformed(evt) {  
            leave_page lp=new leave_page();  
            lp.setVisible(true);  
            dispose();  
        }  
  
        jButton1ActionPerformed(evt)  
        admin_details ad=new admin_details();  
    }  
}
```

```
ad.setVisible(true);

dispose();

}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

welcompage w=new welcompage();

w.setVisible(true);

dispose();

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

salary_page sp=new salary_page();

sp.setVisible(true);

dispose();

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

project_page pp=new project_page();

pp.setVisible(true);

dispose();

}

java.awt.EventQueue.invokeLater(new Runnable() {

    public void run() {

        new adminarea().setVisible(true);

    }
})
```

Details.java

```
public class admin_details extends javax.swing.JFrame {  
    public admin_details() {  
        initComponents();  
        connect();  
        tableload();  
    }  
  
    Connection con,con1;  
  
    PreparedStatement pst,pst2,insert,up,de;  
  
    ResultSet rs;  
  
    Statement stmt;  
  
    private void tableload()  
    {  
        int cc;  
        try {  
            Class.forName("oracle.jdbc.OracleDriver");  
            con1=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","tulika","tulika");  
            insert=con.prepareStatement("select  
            emp_id,name,sex,dob,email,password,contact,address from employee");  
            ResultSet Rs=insert.executeQuery();  
            ResultSetMetaData RSMD=Rs.getMetaData();  
            cc=RSMD.getColumnCount();  
            DefaultTableModel DFT=(DefaultTableModel) jTable1.getModel();  
            DFT.setRowCount(0);  
            while(Rs.next()){  
                Vector v2=new Vector();  
                for (int i=1;i<=cc;i++){  
                    v2.add(Rs.getString(i));  
                }  
                DFT.addRow(v2);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
v2.add(Rs.getString("emp_id"));

v2.add(Rs.getString("name"));

v2.add(Rs.getString("sex"));

v2.add(Rs.getString("dob"));

v2.add(Rs.getString("email"));

v2.add(Rs.getString("password"));

v2.add(Rs.getString("contact"));

v2.add(Rs.getString("address"));

}

DFT.addRow(v2);

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

String emp_id=txtemp_id.getText();

String name=txtname.getText();

String gender=txtgender.getText();

String dob=txtdob.getText();

String email=txtemail.getText();

String pass=txtpass.getText();

String conc=txtcon.getText();

String addr=txtaddr.getText();

try {

pst=con.prepareStatement("insert into
employee(emp_id,name,email,password,dob,sex,address,contact) values(?,?,?,?,?,?)");

pst.setString(1, emp_id);

pst.setString(2, name);

pst.setString(3, email);

pst.setString(4, pass);

pst.setString(5, dob);
```

```
        pst.setString(6, gender);

        pst.setString(7, addr);

        pst.setString(8, conc);

        pst2=con.prepareStatement("insert into salary(emp_id2)values(?");

        pst2.setString(1, emp_id);

        int k=pst.executeUpdate();

        int k1=pst2.executeUpdate();

        if(k==1){

            JOptionPane.showMessageDialog(this,"record added");

            txtemp_id.setText("");

            txtname.setText("");

            txtgender.setText("");

            txtdob.setText("");

            txtemail.setText("");

            txtpass.setText("");

            txtcon.setText("");

            txtaddr.setText("");

            tableload();

            txtemp_id.requestFocus();

        }

        else{

            JOptionPane.showMessageDialog(this,"failed!!!");

        }

    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {

        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

        int selectedIndex=jTable1.getSelectedRow();

        txtemp_id.setText(model.getValueAt(selectedIndex,0).toString());

        txtname.setText(model.getValueAt(selectedIndex,1).toString());
```

```
txtgender.setText(model.getValueAt(selectedIndex,2).toString());  
  
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
    int selectedIndex=jTable1.getSelectedRow();  
    int emp_id=Integer.parseInt(model.getValueAt(selectedIndex,0).toString());  
    int dialogResult=JOptionPane.showConfirmDialog(null,"Do you want to  
delete","Warning",JOptionPane.YES_NO_OPTION);  
    if(dialogResult==JOptionPane.YES_OPTION){  
        try {  
            de=con.prepareStatement("delete from employee where emp_id=?");  
            de.setInt(1,emp_id);  
            de.executeUpdate();  
            JOptionPane.showMessageDialog(this,"record deleted");  
            txtemp_id.setText("");  
            txtname.setText("");  
            txtgender.setText("");  
            txtdob.setText("");  
            txtemail.setText("");  
            txtpass.setText("");  
            txtcon.setText("");  
            txtaddr.setText("");  
            tableload();  
        }  
    }  
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
        int selectedIndex=jTable1.getSelectedRow();  
        int emp_id=Integer.parseInt(model.getValueAt(selectedIndex,0).toString());
```

```
String name=txtname.getText();
String gender=txtgender.getText();
String dob=txtdob.getText();
String email=txtemail.getText();
String pass=txtpass.getText();
String connn=txtcon.getText();
String addr=txtaddr.getText();

    up=con.prepareStatement("update employee set
name=?,email=?,password=?,sex=?,address=?,dob=?,contact=? where emp_id=?");

    up.setString(1, name);
    up.setString(2, email);
    up.setString(3, pass);
    up.setString(4, gender);
    up.setString(5, addr);
    up.setString(6, dob);
    up.setString(7,connn);
    up.setInt(8, emp_id);
    up.executeUpdate();
    JOptionPane.showMessageDialog(this, "record updated");
    txtemp_id.setText("");
    txtname.setText("");
    txtgender.setText("");
    txtdob.setText("");
    txtemail.setText("");
    txtpass.setText("");
    txtcon.setText("");
    txtaddr.setText("");
    tableload();
```

```
public static void main(String args[]) {  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new admin_details().setVisible(true);  
        }  
    });  
}
```

Salary.java

```
public class salary_page extends javax.swing.JFrame {  
  
    public salary_page() {  
        initComponents();  
        connect();  
        tableload();  
    }  
  
    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
        int selectedIndex=jTable1.getSelectedRow();  
        txtemp.setText(model.getValueAt(selectedIndex,0).toString());  
        txtsal.setText(model.getValueAt(selectedIndex,1).toString());  
        txtbb.setText(model.getValueAt(selectedIndex,2).toString());  
    }  
  
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
        int selectedIndex=jTable1.getSelectedRow();  
        int emp_id=Integer.parseInt(model.getValueAt(selectedIndex,0).toString());  
        String sal=txtsal.getText();  
        String bonus=txtbb.getText();  
        up=con.prepareStatement("update salary set sal=?,bonus=? where emp_id=?");  
    }  
}
```

```
        up.setString(1, sal);
        up.setString(2, bonus);
        up.setInt(3, emp_id);
        up.executeUpdate();
        JOptionPane.showMessageDialog(this, "record updated");
        txtemp.setText("");
        txtsal.setText("");
        txtbb.setText("");
        tableload();
    }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new salary_page().setVisible(true);
        }
    });
}
```

Project.java

```
public class leave_page extends javax.swing.JFrame {
    public leave_page() {
        initComponents();
        connect();
        tableload();
    }
    Connection con,con1;
    PreparedStatement pst,up,de;
    ResultSet rs;
    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();
        int selectedIndex=jTable1.getSelectedRow();
```

```

txtemp.setText(model.getValueAt(selectedIndex,0).toString());

txts.setText(model.getValueAt(selectedIndex,1).toString());

txte.setText(model.getValueAt(selectedIndex,2).toString());

txtre.setText(model.getValueAt(selectedIndex,3).toString());

txtst.setText(model.getValueAt(selectedIndex,4).toString());

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

DefaultTableModel model=(DefaultTableModel) jTable1.getModel();

int selectedIndex=jTable1.getSelectedRow();

int emp=Integer.parseInt(model.getValueAt(selectedIndex,0).toString());

String sdate=txts.getText();

String edate=txte.getText();

String status=txtst.getText();

String rea=txtre.getText();

up=con.prepareStatement("update leave set status=?,sdate=?,edate=?,reason=? where
emp_id4=?");

up.setString(1, status);

up.setString(2, sdate);

up.setString(3, edate);

up.setString(4, rea);

up.setInt(5, emp);

up.executeUpdate();

JOptionPane.showMessageDialog(this, "record updated");

txtemp.setText("");

txts.setText("");

txte.setText("");

txtre.setText("");

txtst.setText("");

tableload();

```

```
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new leave_page().setVisible(true);  
    }  
});
```

leave.java

```
public class leave_page extends javax.swing.JFrame {  
  
    public leave_page() {  
        initComponents();  
        connect();  
        tableload();  
    }  
  
    Connection con,con1;  
    PreparedStatement pst,up,de;  
    ResultSet rs;  
  
    private void jTable1MouseClicked(java.awt.event.MouseEvent evt) {  
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
        int selectedIndex=jTable1.getSelectedRow();  
        txttemp.setText(model.getValueAt(selectedIndex,0).toString());  
        txts.setText(model.getValueAt(selectedIndex,1).toString());  
        txtc.setText(model.getValueAt(selectedIndex,2).toString());  
        txtre.setText(model.getValueAt(selectedIndex,3).toString());  
        txtst.setText(model.getValueAt(selectedIndex,4).toString());  
    }  
  
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
        DefaultTableModel model=(DefaultTableModel) jTable1.getModel();  
        int selectedIndex=jTable1.getSelectedRow();  
        int emp=Integer.parseInt(model.getValueAt(selectedIndex,0).toString());  
        String sdate=txts.getText();  
        String edate=txtc.getText();  
        String redate=txtre.getText();  
        String stdate=txtst.getText();  
        String update="update employee set sdate=?,edate=?,redate=?,stdate=? where id=?";  
        try{  
            pst=conn.prepareStatement(update);  
            pst.setString(1,sdate);  
            pst.setString(2,edate);  
            pst.setString(3,redate);  
            pst.setString(4,stdate);  
            pst.setInt(5,emp);  
            pst.executeUpdate();  
            JOptionPane.showMessageDialog(null,"Record Updated");  
        }catch(Exception e){  
            JOptionPane.showMessageDialog(null,e);  
        }  
    }  
}
```

```
String status=txtst.getText();
String rea=txtre.getText();
up=con.prepareStatement("update leave set status=?,sdate=?,edate=?,reason=? where
emp_id4=?");
up.setString(1, status);
up.setString(2, sdate);
up.setString(3, edate);
up.setString(4, rea);
up.setInt(5, emp);
up.executeUpdate();
JOptionPane.showMessageDialog(this, "record updated");
txtemp.setText("");
txts.setText("");
txte.setText("");
txtre.setText("");
txtst.setText("");
tableload();
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new leave_page().setVisible(true);
    }
});
```

CHAPTER 6

SNAPSHOTS

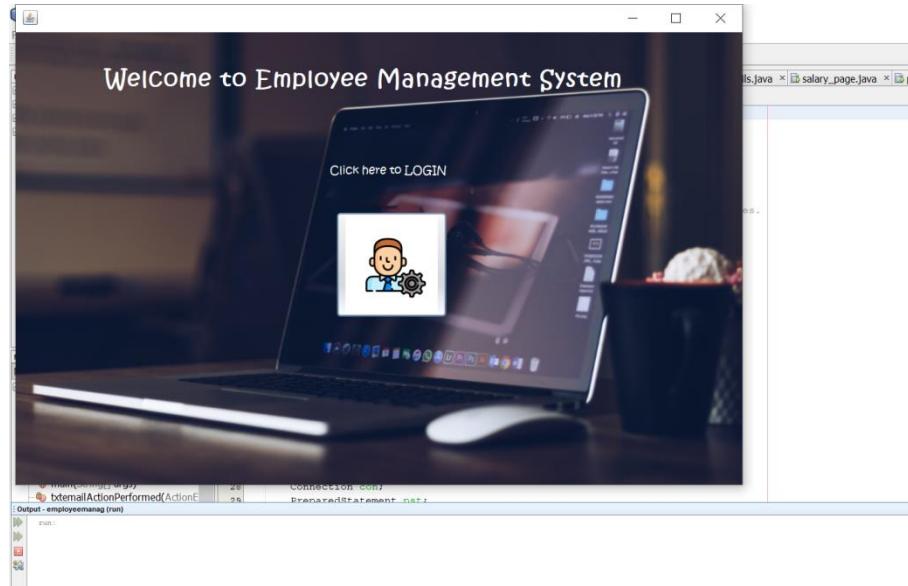


Fig 6.1 welcome page

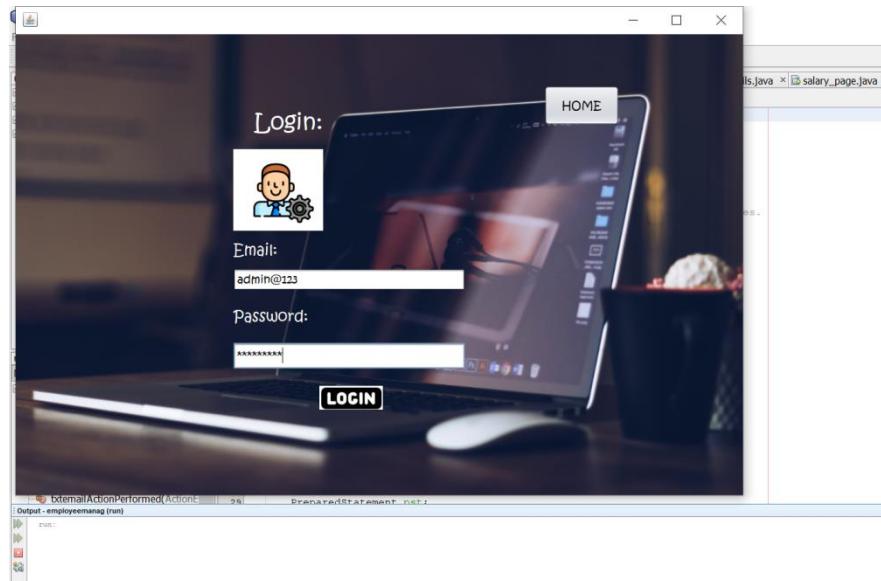


Fig 6.2 login page

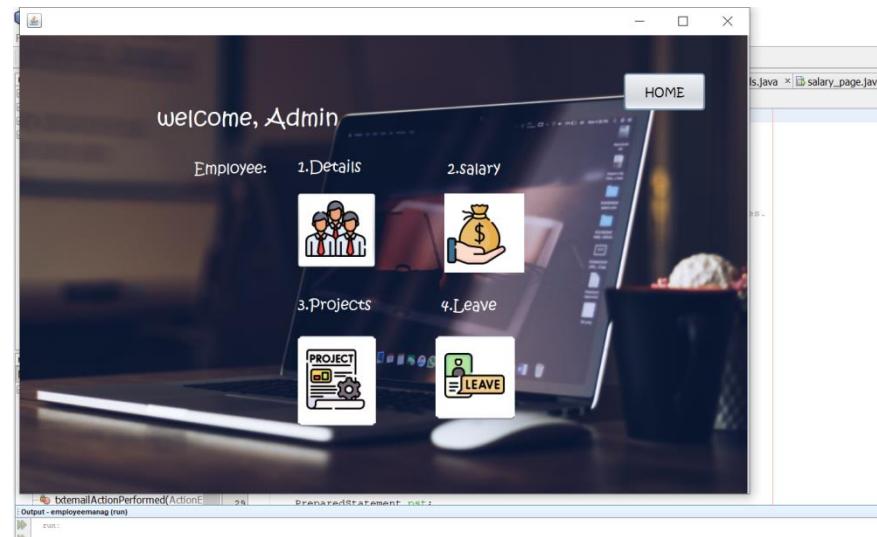


Fig 6.3 Menu page

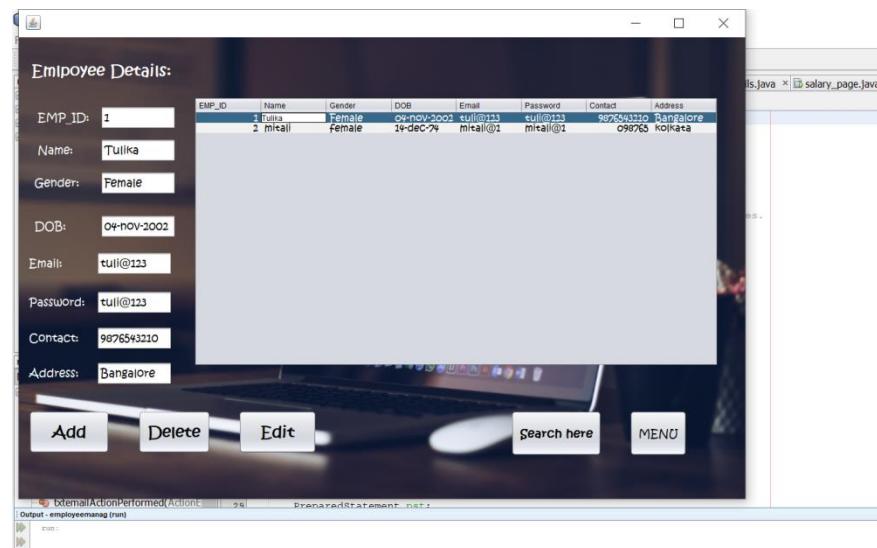


Fig 6.4 employee detail page

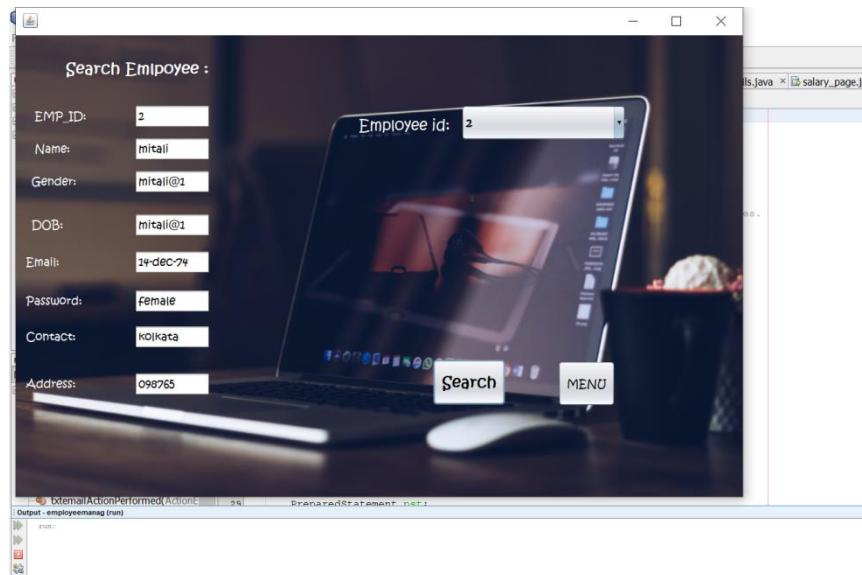


Fig 6.5 employee search page

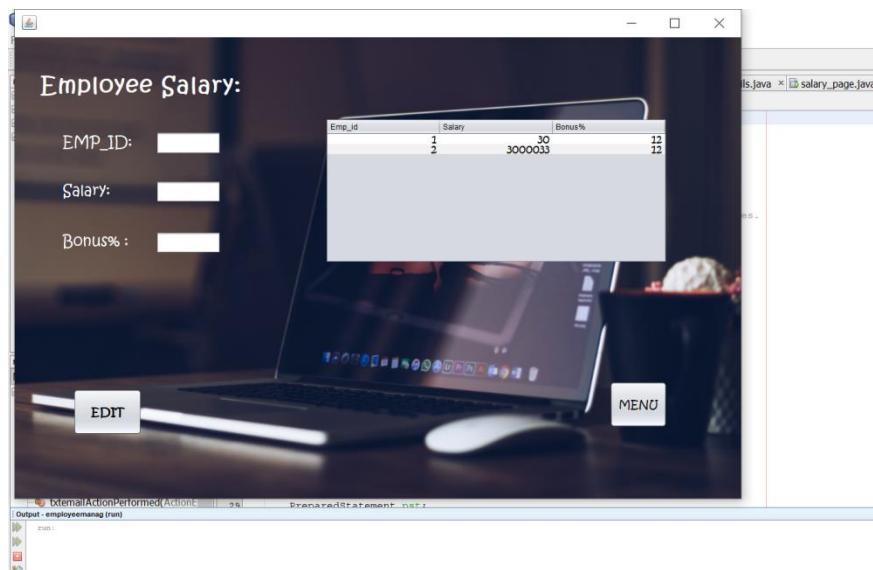


Fig 6.4 employee salary page

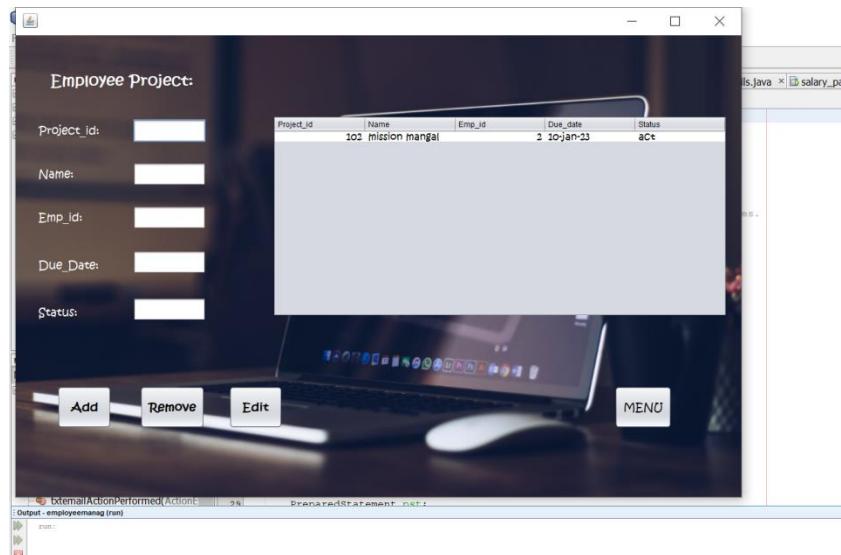


Fig 6.3 employee project page

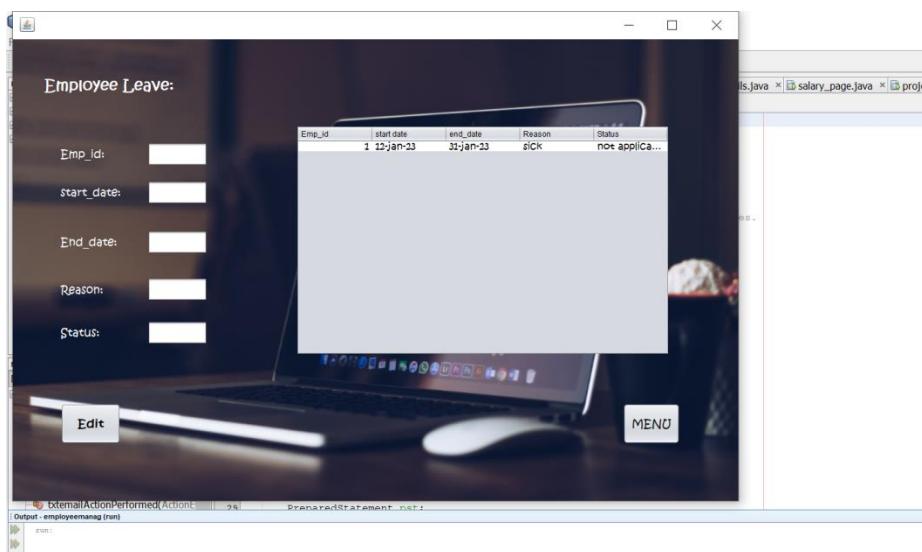


Fig 6.4 employee leave page

CONCLUSION

This project enables the managers of the organization to maintain a proper record of the employees. This project gives a user based application interface which is easy to use and easy to access data from the database without any prior knowledge of Sql queries. This project has guided our path through various aspects of computer science where developing application plays a major role. Cloud Based Medical Consultation System is free to prescribe, generate and use the functioning and maintaining the quality of services.

REFERENCES

- [1] Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, 7th edition, 2017, Perason.
- [2] www.w3school.com
- [3] www.tutussfunny.com/javajdbc-crud/
- [4] www.myproject.com
- [5] www.javapoint.com
- [6] www.geeksforgeeks.org

Vision

To become a premier institute transforming our students to be global professionals.

Mission

M1: Develop competent Human Resources, and create state-of-the-art infrastructure to impart quality education and to support research.

M2: Adopt tertiary approach in teaching – learning pedagogy that transforms students to become professionally competent technocrats and entrepreneurs.

M3: Nurture and train students to develop the qualities of global professionals.

Department of Computer Science and Engineering

Vision

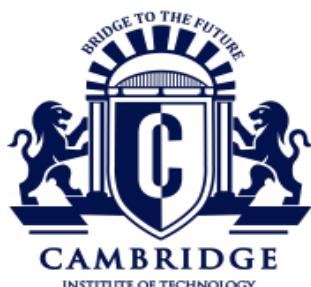
To impart quality education in the field of Computer Science and Engineering with emphasis on innovative thinking, communication and leadership skills to meet the global challenges in IT paradigm.

Mission

M1: Focus on student centric approach through experiential learning and necessary infrastructure.

M2: Develop innovative thinking, communication and leadership skills by creating conducive environment and relevant training.

M3: Enrich students by developing the traits of global professionals.



CAMBRIDGE INSTITUTE OF TECHNOLOGY
K. R. PURAM, BENGALURU - 560036