

# **OBJECT ORIENTED PROGRAMMING CSL-210**

## **Project Report**

### **“EMPLOYEE COMPLAIN PORTAL”**

**Class: BS (CS)- 2A**

**Course Teacher: Miss Sameena Javed**

**Lab Teacher: Miss Salas Akbar**

**MUHAMMAD ABDULLAH KHAN (02-134221-023)**

# **Table of Contents**

01.	Introduction to system	3
02.	Database Explanation	3-4
03.	Features of the project	4
04.	Flow chart	5
05.	UML	6
06.	Screenshots of all forms	7-14
07.	Code snippets	15-56
08.	Conclusion	56

## **1. INTRODUCTION TO SYSTEM:**

A Complaint Management System is a Software Application that allows businesses to manage a more efficient way to respond to consumer queries, provide consistent service and manage complaints. We're working on a Complaint Management System that is designed for any company that would deal with complaints involving every part of the company. Implementing it in any company can be fruitful towards the development of company and can have many benefits. The main objective of the system is to identify the issues that are affecting the internal operations of the Company. It should also be able to provide the necessary help to the staff members by improving the efficiency of the organization.

### ❖ **PROPOSED SYSTEM:**

Our employee complain portal is developed to work for any Company and its employees. It provides an online way of solving the problems faced by the employees by saving time. The objective of our Complaint portal is to make complaints easier to coordinate, monitor, track and resolve, and to provide a company with an effective tool to identify and target problem areas, monitor complaints handling performance and make business improvements. By this system the employee can save his time and solve their concerns without going to an Admin Officer. They can get their problem solved by posting the problem in this proposed system.

Employee complain portal provides the following services:

- **Login form and signup form:**

Login form and signup forms for admin and users.

This login form will help the admin or user to login from their existing accounts, if any admin or user do not have an account, they can sign up via signup option and register their complaints.

Employee can register more than one complain at a time. Admin panel will manage all the complaints of employees.

## **2. DATABASE EXPLANATION:**

The class DB Connection consists of methods for connection of database, Admin login, user login, for searching, deleting, updating, fetching and adding the records in access database. The class DB connection consists of objects from our backend classes.

The database is connected to project in a way that it holds different methods of different tasks as mentioned above.

The method establishconnection () consists of complete path for access files. After giving the path of UcanAccess we have concatenated the location of our Access files.

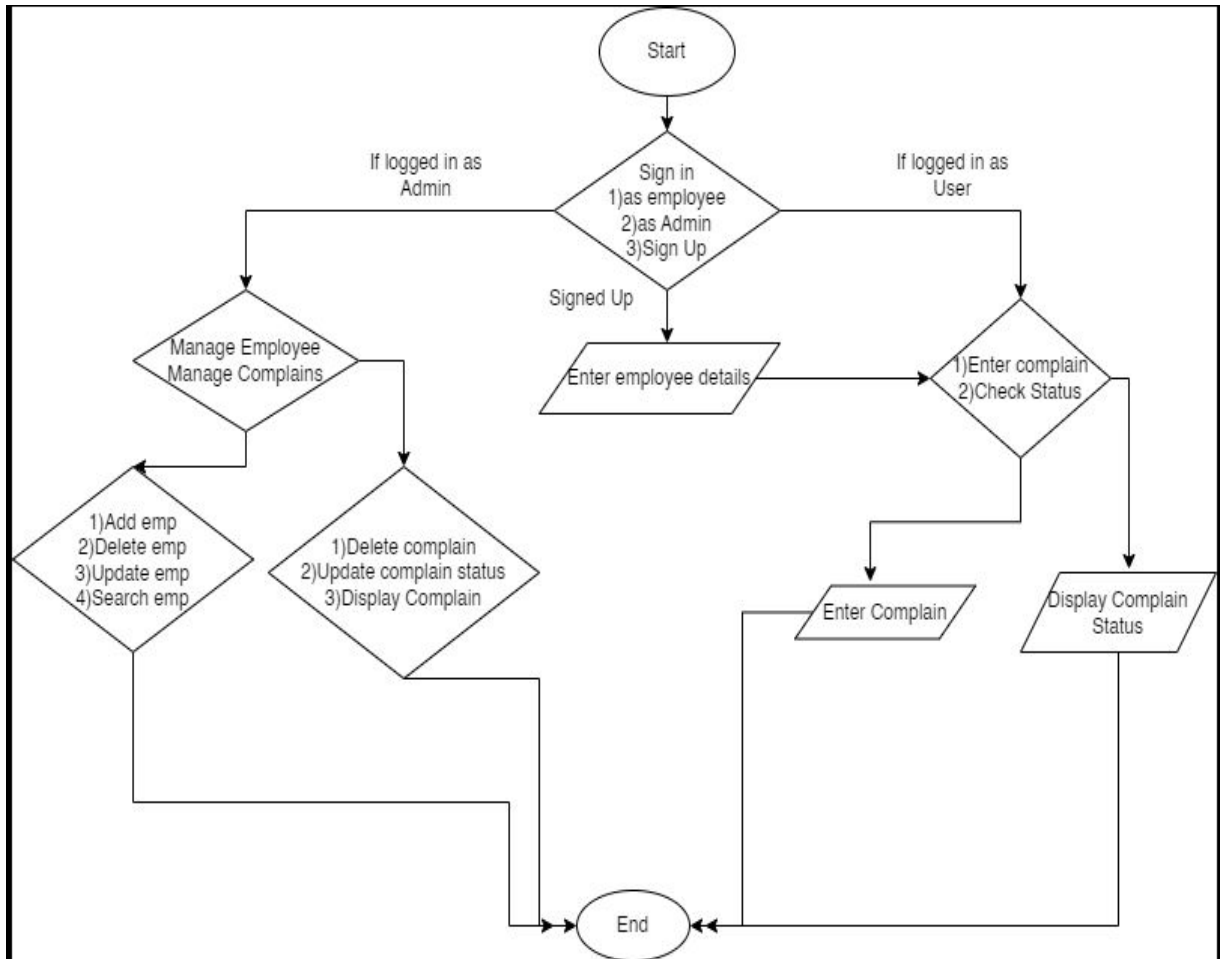
In the methods user login and admin login we pass a query to select all data from the tables in access database. The method type is Boolean which returns true if the login data matches the data in the access database and allow the following users to work on their respective pages.

The method adduser () or addEmpdet () Signs up a user and the query for inserting the data is passed to the database. The method updatepassword () passes a query for updating records like username or password using their respective ids. The method searchAdmin () searches the admin from the database and the method deleteAdmin () deletes the admin from the database table. We have made different functions in different classes and has connected those classes from database.

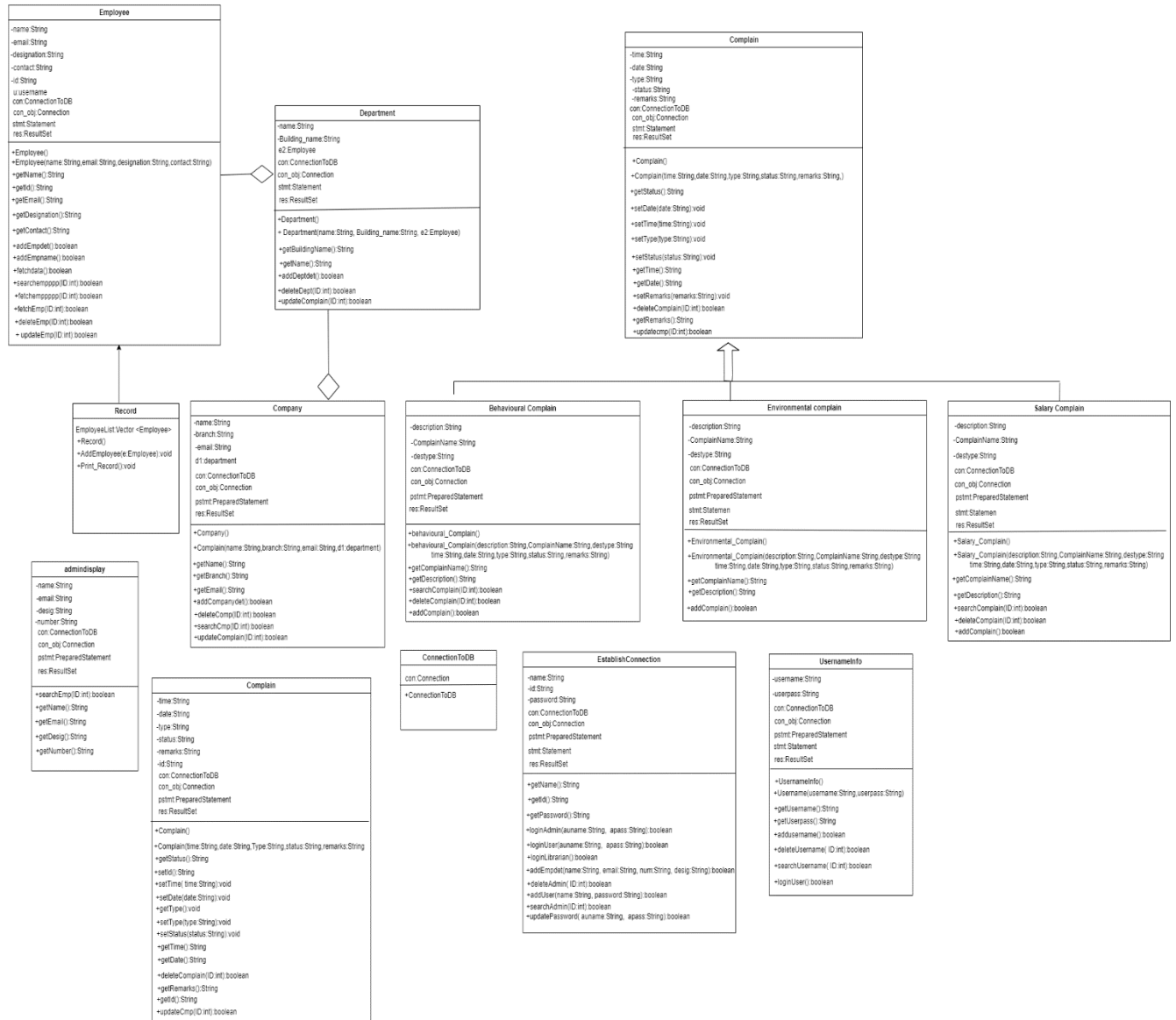
### **3. FEATURES OF PROJECT:**

- + Easy to use
- + User-friendly interface
- + Database for saving records
- + Update, delete, fetch, add and search functionalities
- + Check Status
- + Admin Login
- + User Login
- + Signup
- + Register complains

### **4. FLOWCHART:**



## 5. UML DIAGRAM:



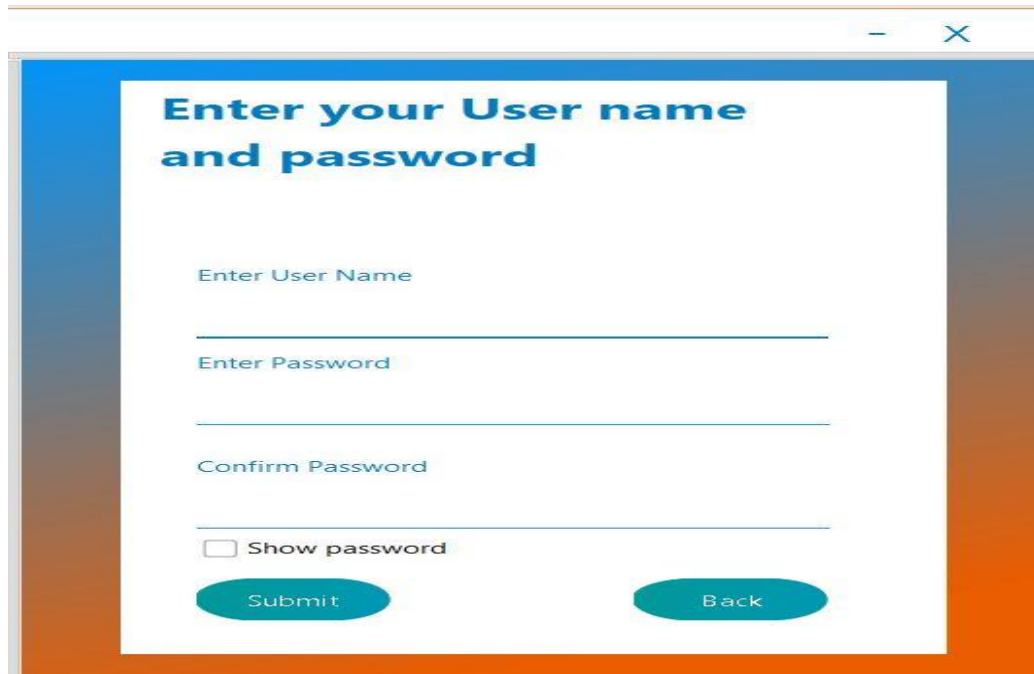
## 6. SCREENSHOTS OF ALL FORMS:

**LOGIN FORM:** Login form for users and admins.

The screenshot shows a web browser window with a blue and orange gradient background. On the left, the text "Complain Portal" is displayed in large white font, followed by "Games don't ease, safety, and comfort." in smaller white font. On the right, a white "Log in" form is overlaid. The form includes fields for "Enter Username" and "Enter Password", a dropdown menu for "Choose how you want to log in", and a checkbox for "Show password". At the bottom of the form are three teal buttons: "Login", "Sign Up", and "Quit".

**SIGN UP FORM:** sign up form for admin and user.

The screenshot shows a web browser window with a blue and orange gradient background. The "Sign Up" form is displayed in a white box. It is divided into three columns: "Employee Information", "Company Details", and "Department Details". The "Employee Information" column has fields for "Enter Full Name", "Enter Email", "Enter Phone Number", and "Designation" (with a dropdown menu). The "Company Details" column has fields for "Enter Company's Name", "Enter Company's Email", and "Choose Company's Branch" (with a dropdown menu). The "Department Details" column has fields for "Enter Department's Name" and "Enter Building Name". At the bottom right of the form are two teal buttons: "Submit" and "Back".



A screenshot of a web application window with a blue header and an orange footer. The main content area is white and contains a login form. The form has three input fields: 'Enter User Name', 'Enter Password', and 'Confirm Password'. Below the 'Enter Password' field is a checkbox labeled 'Show password'. At the bottom of the form are two buttons: 'Submit' and 'Back'.

Enter your User name and password

Enter User Name

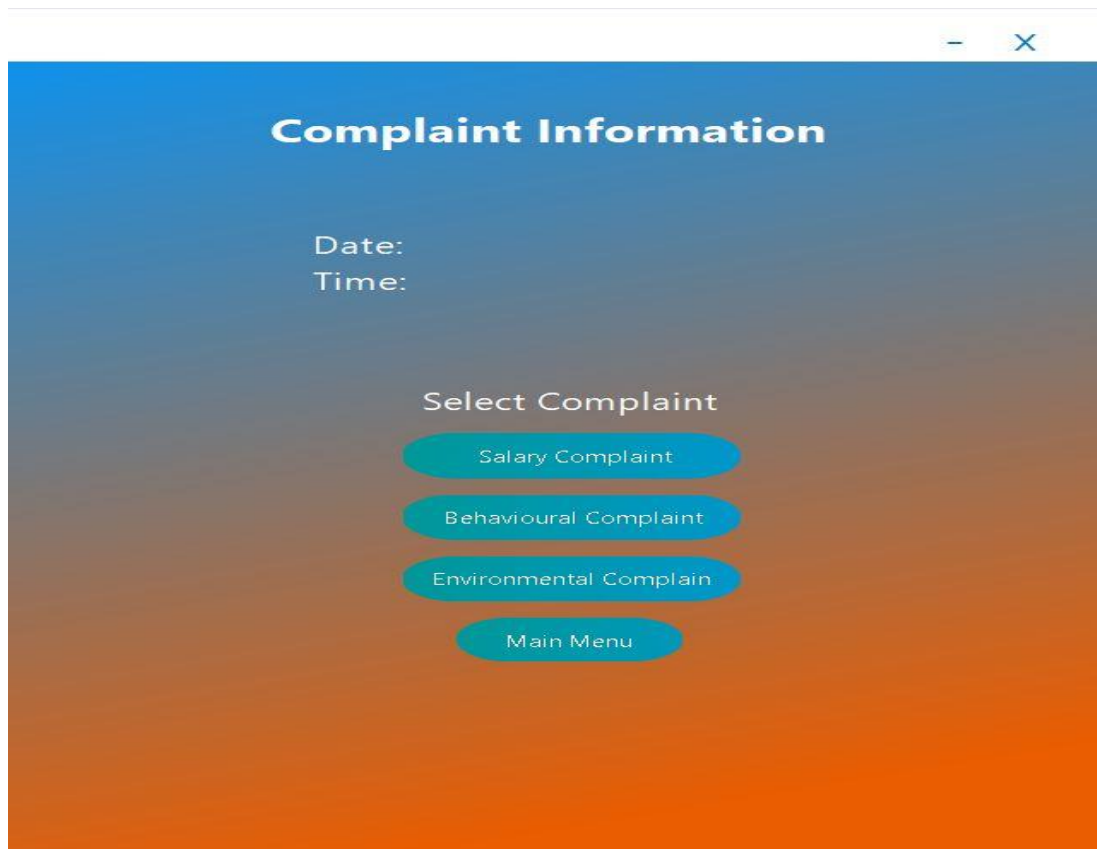
Enter Password

Confirm Password

☐ Show password

Submit Back

**Complain interface:** select the option of the type of complain you want to file.



A screenshot of a web application window with a blue header and an orange footer. The main content area is white and contains a form for entering complaint information. The form has two input fields: 'Date:' and 'Time:'. Below these fields is a section titled 'Select Complaint' with four buttons: 'Salary Complaint', 'Behavioural Complaint', 'Environmental Complain', and 'Main Menu'.

Complaint Information

Date:

Time:

Select Complaint

Salary Complaint

Behavioural Complaint

Environmental Complain

Main Menu



## SALARY COMPLAIN

Change Complain Type

Date

Time

Select your preferable complain

Select your preferable complain

Select any other complain option in the given box below if you cannot find your choice of complain

Description Type

Description

SubmitMain Menu

- X

## Behavioural Complain

Change Complain Type

Date

Time

Select your preferable complain

Select your preferable complain

Select any other complain option in the given box below if you cannot find your choice of complain

Description Type

Description

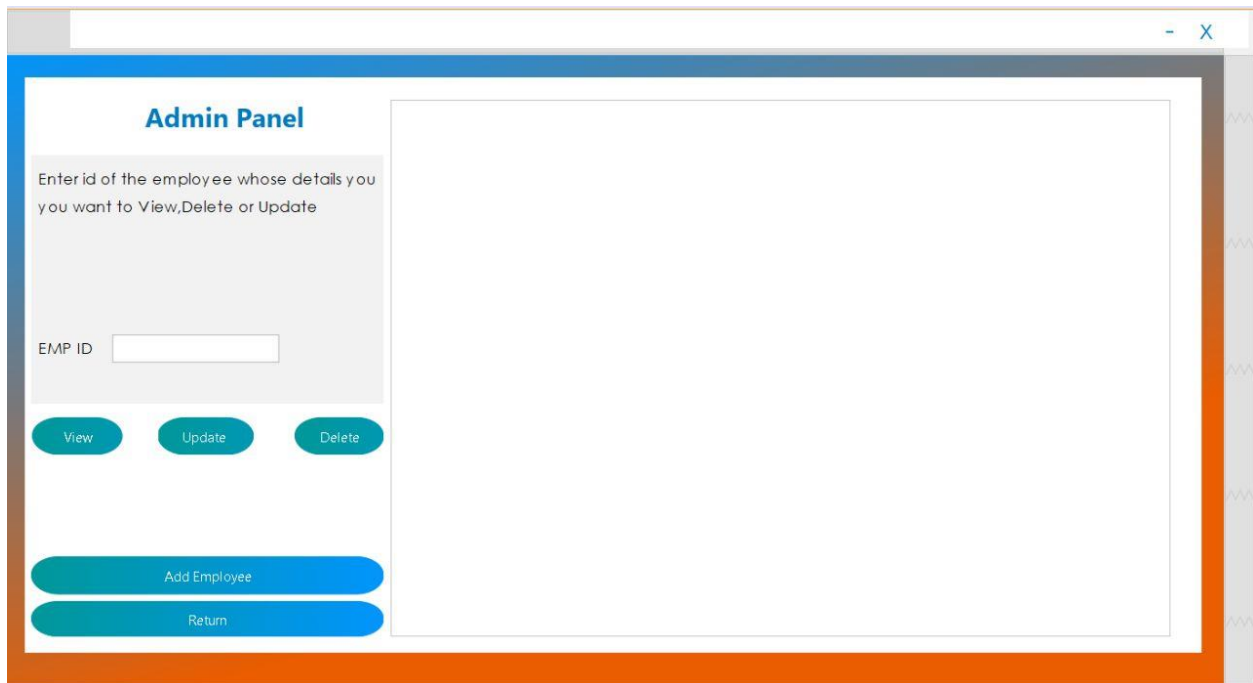
SubmitMain Menu

The screenshot shows a web form titled "Environmental Complain" within a window. The form has a blue header bar with the title and a "Change Complain Type" button. Below the header, there are input fields for "Date" and "Time". A section titled "Select your preferable complain" contains a dropdown menu labeled "Select your complain". Below this, a text prompt says "Select any other complain option in the given box below if you cannot find your choice of complain", followed by another dropdown menu labeled "Description Type". A large text area labeled "Description" is provided for further input. At the bottom right of the form, there are two buttons: "Submit" and "Main Menu".

**ADMIN PANEL INTERFACE:** Admin will select whether he has to manage employees or complains.

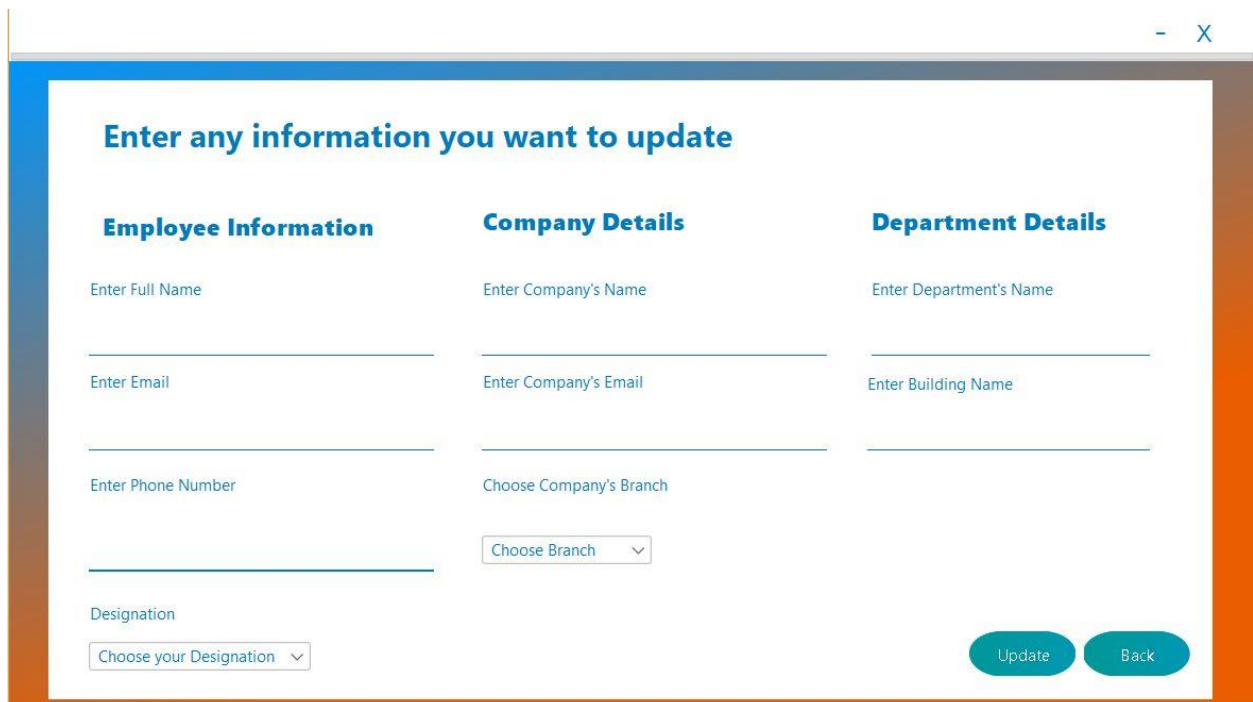
The screenshot shows the "Admin Panel" interface. On the left, a large blue and orange gradient box contains the text "Admin Panel" in large white font, followed by "Here an admin can manage employees, and complains." in smaller white font. On the right, a white panel titled "Admin Panel" in blue font contains three stacked teal buttons: "Manage employees", "Manage Complains", and "Main Menu". The entire interface is displayed within a window with standard OS controls.

This is an admin panel which will delete, view and update complains.



The screenshot shows a web browser window with a title bar containing a minus sign and an 'X' icon. The page has a blue header with the text "Admin Panel". Below the header, there is a light gray box containing the text "Enter id of the employee whose details you want to View,Delete or Update". Below this box is a text input field labeled "EMP ID". Underneath the input field are three teal buttons: "View", "Update", and "Delete". At the bottom of the left sidebar, there are two more teal buttons: "Add Employee" and "Return". The main content area on the right is a large, empty white rectangle.

You can update the information of employee by entering all the information.



The screenshot shows a web browser window with a title bar containing a minus sign and an 'X' icon. The page has a blue header with the text "Enter any information you want to update". Below the header, the form is divided into three columns: "Employee Information", "Company Details", and "Department Details".  
Under "Employee Information", there are three text input fields labeled "Enter Full Name", "Enter Email", and "Enter Phone Number", followed by a "Designation" dropdown menu with the text "Choose your Designation".  
Under "Company Details", there are two text input fields labeled "Enter Company's Name" and "Enter Company's Email", followed by a "Choose Company's Branch" dropdown menu with the text "Choose Branch".  
Under "Department Details", there are two text input fields labeled "Enter Department's Name" and "Enter Building Name".  
At the bottom right of the form, there are two teal buttons: "Update" and "Back".

A screenshot of a web application window titled "Update username and password of the employee". The window has a blue header bar and a white body. The title is in bold blue text. Below the title, there are three input fields: "Enter User Name", "Enter Password", and "Confirm Password". Each field has a blue underline. Below the "Enter Password" field, there is a checkbox labeled "Show password". At the bottom, there are two teal buttons: "Submit" and "Back".

You can add any employee by jotting down all the information of the desired employee.

A screenshot of a web application window titled "Add Employee". The window has a blue header bar and a white body. The title is in bold blue text. Below the title, there are three columns of input fields: "Employee Information", "Company Details", and "Department Details". Each column has three input fields. The "Employee Information" column has "Enter Full Name", "Enter Email", and "Enter Phone Number". The "Company Details" column has "Enter Company's Name", "Enter Company's Email", and "Choose Company's Branch" (with a dropdown menu labeled "Choose Branch"). The "Department Details" column has "Enter Department's Name", "Enter Building Name", and an empty input field. At the bottom right, there are two teal buttons: "Add" and "Back".

- X

### Complete Employee Information

Full Name	Company's Name
Email	Company's Email
Phone Number	Company's Branch Name
Designation	Department's Name
User Name	Building Name
Password	

Show

Back

This is an admin panel which will delete, view and update complains.

### Admin Panel

Enter id of the complain whose details you want to View,Delete or Update status

COMPLAIN ID

View

Update Status

Delete

Return

It will update complain status.

The screenshot shows a web form titled "Update Complain Status". It features a table with the following fields: "Complain ID", "Date", "Time", "Complain Name", and "Complain Type". Below the table is a large text area labeled "Description". Underneath the description is a "Complain Status" dropdown menu, which currently shows "Not Reviewed Yet" with a downward arrow. Below the status dropdown is a "Remarks" text area. At the bottom right of the form, there are two teal buttons: "Update" and "Back".

This will show complete complain information.

The screenshot shows a web form titled "Complete Complain Information". It features a table with the following fields: "Complain ID", "Date", "Time", "Complain Status", "Complain Name", and "Complain Type". Below the table is a large text area labeled "Description". Underneath the description is a "Remarks" text area. At the bottom right of the form, there is a single teal button labeled "Back".

## **7. CODE SNIPPETS:**

### **CLASS EMPLOYEE:**

```
package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Employee {

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj = con.EstablishConnection();
    PreparedStatement pstmt = null;
    Statement stmt = null;
    ResultSet res = null;
    private String name, email, designation, contact, id;

    public Employee() {
    }

    public Employee(String name, String email, String designation, String contact) {

        this.name = name;
        this.email = email;
        this.designation = designation;
        this.contact = contact;
    }

    public String getName() {
```

```

        return name;
    }

    public String getId() {
        return id;
    }

    public String getEmail() {
        return email;
    }

    public String getDesignation() {
        return designation;
    }

    public String getContact() {
        return contact;
    }

    public boolean addEmpdet() {
        boolean b = false;

        String sql = "insert into emp_details(Fullname,email,number,Designation)values('" + this.name + "','" + this.email + "','" +
this.contact + "','" + this.designation + "')";

        try {

            stmt = con_obj.createStatement();
            int res = stmt.executeUpdate(sql);

            if (res > 0) {

                b = true;
            }
        }
    }

```



```

    } else {

        JOptionPane.showMessageDialog(null, "Error");

        b = false;

    }

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, ex);

}

return b;

}

```

```

public boolean fetchdata() {

    String str = "SELECT * FROM emp_details ";

    boolean b = false;

    try {

        pstmt = con_obj.prepareStatement(str);

        res = pstmt.executeQuery();

        while (res.next()) {

            name = res.getString("Fullname");

            email = res.getString("email");

            contact = res.getString("number");

            designation = res.getString("Designation");

            id = res.getString("ID");

            b = true;

            return b;

        }

    }

```

```

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
        b = false;
    }
    return b;
}

public boolean searchEmp(int ID) {
    String str = "select * from emp_details where ID = '" + ID + "'";
    boolean b = false;

    try {
        pstmt = con_obj.prepareStatement(str);
        res = pstmt.executeQuery();

        while (res.next()) {
            name = res.getString("Fullname");
            email = res.getString("email");
            contact = res.getString("number");
            designation = res.getString("Designation");
            id = res.getString("ID");

            b = true;
            return b;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
        b = false;
    }
    return b;
}

public boolean deleteEmp(int ID) {

```

```

String str = "delete from emp_details where ID = '" + ID + "'";

boolean b = false;

try {

    stmt = con_obj.createStatement();

    int res = stmt.executeUpdate(str);

    if (res > 0) {

        b = true;

    } else {

        b = false;

    }

} catch (Exception e) {

    JOptionPane.showMessageDialog(null, e);

}

return b;

}

public boolean updateEmp(int ID) {

    boolean b = false;

    String sql = "UPDATE emp_details SET Fullname = '" + this.name + "',email = '" + this.email + "',number = '" + this.contact +
    "',Designation = '" + this.designation + "' WHERE ID = '" + ID + "'";

    try {

        stmt = con_obj.createStatement();

        int res = stmt.executeUpdate(sql);

        if (res > 0) {

            b = true;

        } else {

    
```

```

        JOptionPane.showMessageDialog(null, "Error");

        b = false;
    }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }

    return b;

}
}

```

### **CLASS BEHAVIOURAL COMPLAIN:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Behavioural_Complain extends Complain {

    String description,ComplainName,des_type;

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj = con.EstablishConnection();
    PreparedStatement pstmt = null;
    Statement stmt = null;
    ResultSet res = null;

```

```

public Behavioural_Complain() {
}

public Behavioural_Complain(String description, String ComplainName, String des_type, String time, String date, String type,
String status, String remarks) {
    super(time, date, type, status, remarks);
    this.description = description;
    this.ComplainName = ComplainName;
    this.des_type = des_type;
}

public String getComplainName() {
    return ComplainName;
}

public String getDescription() {
    return description;
}

public boolean searchComplain(int ID) {
    String str = "select * from Complains where ID = '" + ID + "'";
    boolean b = false;

    try {
        pstmt = con_obj.prepareStatement(str);
        res = pstmt.executeQuery();

        while (res.next()) {
            super.setType(res.getString("ComplainType"));
            ComplainName = res.getString("ComplainName");
            des_type=res.getString("DescriptionType");

```

```

        description = res.getString("Description");
        super.setDate(res.getString("Date"));
        super.setTime(res.getString("Time"));
        super.setStatus(res.getString("Status"));
        super.setRemarks(res.getString("Remarks"));

        b = true;
        return b;
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);

    b = false;
}
return b;
}

```

```

public boolean deleteComplain(int ID) {
    String str = "delete from Complains where ID = " + ID + "";
    boolean b = false;

    try {
        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(str);

        if (res > 0) {

            b = true;
        } else {

            b = false;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }

    return b;
}

```

```

    }

    public boolean addComplain() {

        boolean b = false;

        String sql = "insert into
Complains(ComplainType,ComplainName,DescriptionType,Description,Date,Time,Status,Remarks)values('" + super.getType() +
";'" + this.ComplainName + ";'" + this.des_type + ";'" + this.description + ";'" + super.getDate() + ";'" + super.getTime() + ";'" +
super.getStatus() + ";'" + super.getRemarks() + "');";

        try {

            stmt = con_obj.createStatement();

            int res = stmt.executeUpdate(sql);

            if (res > 0) {

                b = true;

            } else {

                JOptionPane.showMessageDialog(null, "Error");

                b = false;

            }

        } catch (Exception ex) {

            JOptionPane.showMessageDialog(null, ex);

        }

        return b;

    }
}

```

### **CLASS COMPANY:**

```
package com.mycompany.logininterface;
```

```
import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;
```

```
public class Company {

    private String name,branch,email;

    Department d1;
```

```
    public Company() {

    }

}
```

```
    public Company(String name, String branch, String email, Department d1) {

        this.name = name;

        this.branch = branch;

        this.email = email;

        this.d1 = d1;

    }

}
```

```
    public String getName() {

        return name;

    }

}
```

```
    public String getBranch() {

        return branch;

    }

}
```

```
    public String getEmail() {

        return email;

    }

}
```



```

    }

    ConnectionToDB con = new ConnectionToDB();

    Connection con_obj=con.EstablishConnection();

    PreparedStatement pstmt=null;

    Statement stmt=null;

    ResultSet res=null;

    public boolean addCompanydet() {

        boolean b = false;

        String sql = "insert into company_details(cmpName,cmpBranch,cmpEmail)values('" + this.name + "','" + this.branch + "','" +
this.email + "')";

        try {

            stmt = con_obj.createStatement();

            int res = stmt.executeUpdate(sql);

            if (res > 0) {

                b = true;

            } else {

                JOptionPane.showMessageDialog(null, "Error");

                b = false;

            }

        } catch (Exception ex) {

            JOptionPane.showMessageDialog(null, ex);

        }

        return b;
    }

```

```

    }

    public boolean deleteComp(int ID) {

        String str = "delete from company_details where ID = '" + ID + "'";

        boolean b = false;

        try {

            stmt = con_obj.createStatement();

            int res = stmt.executeUpdate(str);

            if (res > 0) {

                b = true;

            } else {

                b = false;

            }

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }

        return b;

    }

    public boolean searchCmp(int ID) {

        String str = "select * from company_details where ID = '" + ID + "'";

        boolean b = false;

        try {

            pstmt = con_obj.prepareStatement(str);

            res = pstmt.executeQuery();

            while (res.next()) {

                this.name = res.getString("cmpName");

                this.email = res.getString("cmpEmail");

                this.branch = res.getString("cmpBranch");

            }

        } catch (Exception e) {

            JOptionPane.showMessageDialog(null, e);

        }

        return b;

    }

```

```

        b = true;

        return b;

    }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

        b = false;

    }

    return b;

}

public boolean updateComplain(int ID) {

    boolean b = false;

    String sql = "UPDATE company_details SET cmpName = '" + this.name + "',cmpEmail = '" + this.email + "',cmpBranch = '" +
this.branch + "' WHERE ID = '" + ID + "'";

    try {

        stmt = con_obj.createStatement();

        int res = stmt.executeUpdate(sql);

        if (res > 0) {

            b = true;

        } else {

            JOptionPane.showMessageDialog(null, "Error");

            b = false;

        }

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, ex);

    }

    return b;
}

```

```
}  
}
```

### **CLASS COMPLAIN:**

```
package com.mycompany.logininterface;
```

```
import java.sql.Connection;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.Statement;
```

```
import javax.swing.JOptionPane;
```

```
public class Complain {
```

```
    private String time;
```

```
    private String date;
```

```
    private String type;
```

```
    private String status;
```

```
    private String remarks;
```

```
    ConnectionToDB con = new ConnectionToDB();
```

```
    Connection con_obj = con.EstablishConnection();
```

```
    PreparedStatement pstmt = null;
```

```
    Statement stmt = null;
```

```
    ResultSet res = null;
```

```
    public Complain() {
```

```
    }
```

```
public Complain(String time, String date, String type, String status, String remarks) {  
    this.time = time;  
    this.date = date;  
    this.type = type;  
    this.status = status;  
    this.remarks = remarks;  
}
```

```
public String getStatus() {  
    return status;  
}
```

```
public void setTime(String time) {  
    this.time = time;  
}
```

```
public void setDate(String date) {  
    this.date = date;  
}
```

```
public String getType() {  
    return type;  
}
```

```
public void setType(String type) {  
    this.type = type;  
}
```

```
public void setStatus(String status) {  
    this.status = status;  
}
```

```
public String getTime() {  
    return time;  
}
```

```
}
```

```
public String getDate() {
```

```
    return date;
```

```
}
```

```
public void setRemarks(String remarks) {
```

```
    this.remarks = remarks;
```

```
}
```

```
public boolean deleteComplain(int ID) {
```

```
    String str = "delete from Complains where ID = " + ID + "";
```

```
    boolean b = false;
```

```
    try {
```

```
        stmt = con_obj.createStatement();
```

```
        int res = stmt.executeUpdate(str);
```

```
        if (res > 0) {
```

```
            b = true;
```

```
        } else {
```

```
            b = false;
```

```
        }
```

```
    } catch (Exception e) {
```

```
        JOptionPane.showMessageDialog(null, e);
```

```
    }
```

```
    return b;
```

```
}
```

```
public String getRemarks() {
```

```

        return remarks;
    }

    public boolean updateCmp(int ID) {
        boolean b = false;

        String sql = "UPDATE Complains SET Status = '" + this.status + "',Remarks = '" + this.remarks + "' WHERE ID = '" + ID + "'";

        try {

            stmt = con_obj.createStatement();
            int res = stmt.executeUpdate(sql);

            if (res > 0) {
                b = true;
            } else {

                JOptionPane.showMessageDialog(null, "Error");
                b = false;
            }

        } catch (Exception ex) {
            JOptionPane.showMessageDialog(null, ex);
        }

        return b;
    }
}

CLASS DEPARTMENT:
package com.mycompany.logininterface;

```

```

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

import javax.swing.JOptionPane;


public class Department {

    private String name,Bulding_name;

    Employee e2;


    public Department() {

    }


    public Department(String name, String Building_name, Employee e2) {

        this.name = name;

        this.e2 = e2;

        this.Bulding_name = Building_name;

    }

    public String getBuildingName() {

        return this.Bulding_name;

    }

    public String getName() {

        return name;

    }

    ConnectionToDB con = new ConnectionToDB();

    Connection con_obj=con.EstablishConnection();

    PreparedStatement pstmt=null;

    Statement stmt=null;

    ResultSet res=null;

    public boolean addDeptdet() {

        boolean b = false;

```



```

String sql = "insert into dept_details(deptName,buildingName)values('" + this.name + "','" + this.Bulding_name + "')";

try {

    stmt = con_obj.createStatement();
    int res = stmt.executeUpdate(sql);

    if (res > 0) {

        b = true;

    } else {

        JOptionPane.showMessageDialog(null, "Error");
        b = false;
    }

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, ex);
}

return b;

}

public boolean deleteDept(int ID) {
    String str = "delete from dept_details where ID = '" + ID + "'";
    boolean b = false;
    try {
        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(str);
        if (res > 0) {

```

```

        b = true;
    } else {

        b = false;
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);

}

return b;
}

public boolean searchDep(int ID) {
    String str = "select * from dept_details where ID = '" + ID + "'";
    boolean b = false;

    try {
        pstmt = con_obj.prepareStatement(str);
        res = pstmt.executeQuery();

        while (res.next()) {
            this.name = res.getString("deptName");
            this.Bulding_name = res.getString("buildingName");

            b = true;
            return b;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
        b = false;
    }
    return b;
}

```

```

public boolean updateComplain(int ID) {

    boolean b = false;

    String sql = "UPDATE dept_details SET deptName = '" + this.name + "',buildingName = '" + this.Bulding_name + "' WHERE ID
= '" + ID + "'";

    try {

        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(sql);

        if (res > 0) {
            b = true;
        } else {

            JOptionPane.showMessageDialog(null, "Error");
            b = false;
        }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }

    return b;
}

```

### **CLASS ENVIRONMENTAL COMPLAIN:**

```

package com.mycompany.logininterface;

```

```

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;

```

```

import javax.swing.JOptionPane;

public class Environmental_Complain extends Complain {

    String description,ComplainName,des_type;

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj = con.EstablishConnection();
    PreparedStatement pstmt = null;
    Statement stmt = null;
    ResultSet res = null;

    public Environmental_Complain() {
    }

    public Environmental_Complain(String description, String ComplainName, String des_type, String time, String date, String
type, String status, String remarks) {
        super(time, date, type, status, remarks);
        this.description = description;
        this.ComplainName = ComplainName;
        this.des_type = des_type;
    }

    public String getComplainName() {
        return ComplainName;
    }

    public String getDescription() {
        return description;
    }

```

```
}
```

```
public boolean addComplain() {
```

```
    boolean b = false;
```

```
    String sql = "insert into  
Complains(ComplainType,ComplainName,DescriptionType,Description,Date,Time,Status,Remarks)values('" + super.getType() +  
"', '" + this.ComplainName + "', '" + this.des_type + "', '" + this.description + "', '" + super.getDate() + "', '" + super.getTime() + "', '" +  
super.getStatus() + "', '" + super.getRemarks() + "')";
```

```
    try {
```

```
        stmt = con_obj.createStatement();
```

```
        int res = stmt.executeUpdate(sql);
```

```
        if (res > 0) {
```

```
            b = true;
```

```
        } else {
```

```
            JOptionPane.showMessageDialog(null, "Error");
```

```
            b = false;
```

```
        }
```

```
    } catch (Exception ex) {
```

```
        JOptionPane.showMessageDialog(null, ex);
```

```
    }
```

```
    return b;
```

```
}
```

```
}
```

### **CLASS RECORD:**

```
package com.mycompany.logininterface;
```

```
import java.util.Vector;
```

```
public class Record {
```

```
    Vector <Employee> EmployeeList = new Vector<>();
```

```
    public Record() {
```

```
        EmployeeList = new Vector<>();
```

```
    }
```

```
    public void AddEmployee(Employee e){
```

```
        EmployeeList.add(e);
```

```
    }
```

```
    public void Print_Record(){
```

```
        for(int i=0;i<EmployeeList.size();i++){
```

```
            Employee employeeList= EmployeeList.elementAt(i);
```

```
            System.out.println("EMPLOYEE NAME:" + employeeList.getName());
```

```
            System.out.println("EMPLOYEE CONTACT:" + employeeList.getContact());
```

```
            System.out.println("EMPLOYEE EMAIL: " + employeeList.getEmail());
```

```
            System.out.println("EMPLOYEE DESIGNATION: " + employeeList.getDesignation());
```

```
        }
```

```
    }
```

```
}
```

### **CLASS SALARY COMPLAIN:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class Salary_Complain extends Complain {

    String description,ComplainName,des_type;

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj = con.EstablishConnection();
    PreparedStatement pstmt = null;
    Statement stmt = null;
    ResultSet res = null;

    public Salary_Complain() {
    }

    public Salary_Complain(String description, String ComplainName, String des_type, String time, String date, String type, String
status, String remarks) {
        super(time, date, type, status, remarks);
        this.description = description;
        this.ComplainName = ComplainName;
        this.des_type = des_type;
    }

    public String getComplainName() {
        return ComplainName;
    }

```

```
}
```

```
public String getDescription() {  
    return description;  
}
```

```
public boolean searchComplain(int ID) {  
    String str = "select * from Complains where ID = '" + ID + "'";  
    boolean b = false;  
  
    try {  
        pstmt = con_obj.prepareStatement(str);  
        res = pstmt.executeQuery();  
  
        while (res.next()) {  
            super.setType(res.getString("ComplainType"));  
            ComplainName = res.getString("ComplainName");  
            des_type=res.getString("DescriptionType");  
            description = res.getString("Description");  
            super.setDate(res.getString("Date"));  
            super.setTime(res.getString("Time"));  
            super.setStatus(res.getString("Status"));  
            b = true;  
            return b;  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null, e);  
        b = false;  
    }  
    return b;  
}
```



```

public boolean deleteComplain(int ID) {

    String str = "delete from Complains where ID = '" + ID + "'";

    boolean b = false;

    try {

        stmt = con_obj.createStatement();

        int res = stmt.executeUpdate(str);

        if (res > 0) {

            b = true;

        } else {

            b = false;

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

    return b;

}

```

```

public boolean addComplain() {

    boolean b = false;

    String sql = "insert into
Complains(ComplainType,ComplainName,DescriptionType,Description,Date,Time,Status,Remarks)values('" + super.getType() +
"','" + this.ComplainName + "','" + this.des_type + "','" + this.description + "','" + super.getDate() + "','" + super.getTime() + "','" +
super.getStatus() + "','" + super.getRemarks() + "')";

    try {

        stmt = con_obj.createStatement();

        int res = stmt.executeUpdate(sql);

        if (res > 0) {

```

```

        b = true;

    } else {

        JOptionPane.showMessageDialog(null, "Error");

        b = false;

    }

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, ex);

    }

    return b;

}

}

```

### **CLASS USERNAME INFO:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

public class UsernameInfo {

    private String username,userpass;

    ConnectionToDB con = new ConnectionToDB();

    Connection con_obj=con.EstablishConnection();

    PreparedStatement pstmt=null;

    Statement stmt=null;

```

```
ResultSet res=null;
```

```
public UsernameInfo() {  
}
```

```
public UsernameInfo(String username, String userpass) {  
    this.username = username;  
    this.userpass = userpass;  
}
```

```
public String getUsername() {  
    return username;  
}
```

```
public String getUserpass() {  
    return userpass;  
}
```

```
public boolean addusername() {  
    boolean b = false;
```

```
    String sql = "insert into user_login(username,password)values('" + this.username + "','" + this.userpass + "')";
```

```
    try {
```

```
        stmt = con_obj.createStatement();
```

```
        int res = stmt.executeUpdate(sql);
```

```
        if (res > 0) {
```

```
            b = true;
```

```

        } else {

            JOptionPane.showMessageDialog(null, "Error");

            b = false;

        }

    } catch (Exception ex) {

        JOptionPane.showMessageDialog(null, ex);

    }

    return b;

}

public boolean deleteUserName(int ID) {

    String str = "delete from user_login where ID = '" + ID + "'";

    boolean b = false;

    try {

        stmt = con_obj.createStatement();

        int res = stmt.executeUpdate(str);

        if (res > 0) {

            b = true;

        } else {

            b = false;

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

    }

    return b;

}

public boolean searchUsername(int ID) {

```

```

String str = "select * from user_login where ID = '" + ID + "'";
boolean b = false;

try {
    pstmt = con_obj.prepareStatement(str);
    res = pstmt.executeQuery();

    while (res.next()) {
        this.username = res.getString("username");
        this.userpass = res.getString("password");

        b = true;
        return b;
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, e);
    b = false;
}
return b;
}

public boolean loginUser() {

    String loginString = "Select * from user_login where username='" + this.username + "' and password='" + this.userpass + "'";

    boolean b;

    try {

        pstmt = con_obj.prepareStatement(loginString);
        res = pstmt.executeQuery();

        if (res.next()) {

```

```

        b = true;
    } else {
        b = false;
    }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
        b = false;
    }

    return b;

}
}

```

### **CLASS ADMIN DISPLAY:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

/**
 *
 * @author hp
 */
public class adminispaly {

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj=con.EstablishConnection();
    PreparedStatement pstmt=null;

```

```

Statement stmt=null;

ResultSet res=null;

private String name,email,desig,number;

public boolean searchEmp(int ID) {

    String str = "select * from tbl_login where ID = '" + ID + "'";

    boolean b = false;

    try {

        pstmt = con_obj.prepareStatement(str);

        res = pstmt.executeQuery();

        while (res.next()) {

            name = res.getString("Fullname");

            email = res.getString("email");

            number = res.getString("number");

            desig = res.getString("designation");

            b = true;

            return b;

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, e);

        b = false;

    }

    return b;

}

public String getName() {

    return name;

}

public String getEmail() {

```

```

        return email;
    }

    public String getDesig() {
        return desig;
    }

    public String getNumber() {
        return number;
    }

}

```

## **DATABASE CONNECTION CLASSES:**

### **CONNECTION TO DB CLASS:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.DriverManager;
import javax.swing.JOptionPane;

public class ConnectionToDB {
    Connection con = null;

    public Connection EstablishConnection() {
        try {
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");
            con= DriverManager.getConnection("jdbc:ucanaccess://E:\\test.accdb");
            // JOptionPane.showMessageDialog(null, "connected");
        }
        catch(Exception ex)
        {

```



```

        JOptionPane.showMessageDialog(null, ex);
    }
    return con;
}
}

```

### **CLASS ESTABLISH CONNECTION:**

```

package com.mycompany.logininterface;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;

/**
 *
 * @author hp
 */
public class EstablishConnection {

    ConnectionToDB con = new ConnectionToDB();
    Connection con_obj=con.EstablishConnection();
    PreparedStatement pstmt=null;
    Statement stmt=null;
    ResultSet res=null;
    private String name, id, password;

    public String getName() {
        return name;
    }
}

```

```

public String getId() {
    return id;
}

public String getPassword() {
    return password;
}

public boolean loginAdmin(String auname, String apass) {

    String loginString = "Select * from admin_login where username='" +auname+ "' and password='" +apass+ "'";

    boolean b;
    try {

        pstmt = con_obj.prepareStatement(loginString);
        res = pstmt.executeQuery();

        if (res.next()) {
            b = true;
        } else {
            b = false;
        }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
        b = false;
    }

    return b;
}

public boolean loginUser(String auname, String apass) {

```

```

String loginString = "Select * from user_login where username='" +auname+ "' and password='" +apass+ "'";

boolean b;

try {

    pstmt = con_obj.prepareStatement(loginString);
    res = pstmt.executeQuery();

    if (res.next()) {
        b = true;
    } else {
        b = false;
    }

} catch (Exception ex) {
    JOptionPane.showMessageDialog(null, ex);
    b = false;
}

return b;

}

public boolean loginLibrarian() {

    String loginString = "Select * from Librarian where Librarian_Name='" +this.name+ "' and Librarian_Password='"
+this.password+ "'";

    boolean b;

    try {

        pstmt = con_obj.prepareStatement(loginString);
        res = pstmt.executeQuery();

```

```

        if (res.next()) {
            b = true;
        } else {
            b = false;
        }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
        b = false;
    }

    return b;

}

public boolean addEmpdet(String name, String email, String num, String desig) {
    boolean b = false;

    String sql = "insert into Table1(Fullname,email,number,Designation)values('" + name + "','" + email + "','" + num + "','" + desig
+ "')";

    try {

        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(sql);

        if (res > 0) {

            b = true;

        } else {

            JOptionPane.showMessageDialog(null, "Error");

```

```

        b = false;
    }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }

    return b;

}

public boolean deleteAdmin(int ID) {
    String str = "delete from tbl_login where ID = '" + ID + "'";
    boolean b = false;
    try {
        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(str);
        if (res > 0) {

            b = true;
        } else {

            b = false;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
    }

    return b;
}

```

```

public boolean addUser(String name, String password) {

```

```

boolean b = false;

String sql = "insert into user_login(username,password)values('" + name + "','" + password + "')";

try {

    stmt = con_obj.createStatement();
    int res = stmt.executeUpdate(sql);

    if (res > 0) {

        b = true;

    } else {

        JOptionPane.showMessageDialog(null, "Error");
        b = false;
    }

} catch (Exception ex) {

    JOptionPane.showMessageDialog(null, ex);
}

return b;

}

public boolean searchAdmin(int ID) {

    String str = "select * from tbl_login where ID = '" + ID + "'";

    boolean b = false;

    try {

```

```

        pstmt = con_obj.prepareStatement(str);
        res = pstmt.executeQuery();

        while (res.next()) {
            name = res.getString("Name");
            password = res.getString("Password");
            id = res.getString("UserName");

            b = true;
            return b;
        }
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e);
        b = false;
    }
    return b;
}

public boolean updatePassword(String auname, String apass) {
    boolean b = false;

    String sql = "UPDATE tbl_login set Password = '" + apass + "' WHERE UserName = '" + auname + "' ";

    try {

        stmt = con_obj.createStatement();
        int res = stmt.executeUpdate(sql);

        if (res > 0) {
            b = true;
        } else {

            JOptionPane.showMessageDialog(null, "Error");
            b = false;
        }
    }
}

```

```
    }

    } catch (Exception ex) {
        JOptionPane.showMessageDialog(null, ex);
    }

    return b;

}

}
```

## **8. CONCLUSION:**

Our Complaint Management System's easy access and flexibility allow the admin to maintain a constant contact with the employees even if they are not in the office. It also saves their time and money by allowing them to communicate with their employees through an online platform.

We learned a lot while coding this project, hope you like it.