

```
//Server.java
```

```
/*  
 * To change this license header, choose License Headers in Project Properties.  
 * To change this template file, choose Tools | Templates  
 * and open the template in the editor.  
 */  
package a3;
```

```
import java.net.*;  
import java.io.*;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.*;  
import java.util.logging.Level;  
import java.util.logging.Logger;
```

```
public class Server  
{  
    //initialize socket and input stream  
  
    static ArrayList<String> al = new ArrayList<String>();  
        ArrayList ale = new ArrayList();  
        ArrayList prj = new ArrayList();  
    // constructor with port
```

```
    public static void main(String args[])  
    {
```

```
        Socket      socket  = null;  
        ServerSocket server = null;
```

```
        int port = 8000;  
        // starts server and waits for a connection  
        try  
        {
```

```
            server = new ServerSocket(port);  
            System.out.println(" -- Server started");
```

```
            System.out.println(" -- Waiting for a client ...");
```

```
            int i=0;  
            while(true)  
            {
```

```

i++;
socket = server.accept();
System.out.println(" Client " + i + " accepted");
MyThread newTh = new MyThread(socket,port);
}

```

```

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

```

```

public static void runServer(Socket socket,int port)
{
    try{
        DataInputStream input  = null;
        ObjectInputStream inObj = null;
        DataOutputStream out    = null;
        ObjectOutputStream outObj = null;
        DataInputStream in      = null;
        //JDBC Connection

        String path="jdbc:mysql://localhost:3306/employee";

        // takes input from the client socket
        in = new DataInputStream(
            new BufferedInputStream(socket.getInputStream()));

        inObj = new ObjectInputStream(socket.getInputStream());

        input = new DataInputStream(System.in);

        // sends output to the socket
        out  = new DataOutputStream(socket.getOutputStream());

        outObj = new ObjectOutputStream(socket.getOutputStream());


        int ch,ch2;
        Employee e;
        //Project p;
        Scanner sc = new Scanner(System.in);
        String nm,nm2;
        Combine c;
        boolean flag ;
        try{
            do{
                ch = in.read();

```

```

switch(ch)
{
    case 1 :
        flag = false;
        nm = in.readUTF();
        //login validation

        try
        {

            Connection con = DriverManager.getConnection(path,"root","root");
            Statement st = con.createStatement();
            String Query = "select * from employee.employee where name = '" + nm + "'";
            ResultSet rs = st.executeQuery(Query);
            flag = rs.next();

        }
        catch(SQLException ex)
        {
            //Failure to signUp
            System.out.println("\n\n\tSQL Exception Error !");
        }
        out.writeBoolean(flag);
        if(flag)
        {
            System.out.println(" -- Client : " + nm + " , Just Logged In");
        }

        do{
            ch2 = in.read();

            switch(ch2)
            {

                case 1 :
                    try
                    {

                        int i=0;
                        Connection con2 = DriverManager.getConnection(path,"root","root");
                        Statement st2 = con2.createStatement();
                        String Query2 = "select * from employee.employee where name = '" + nm + "'";
                        ResultSet rs2 = st2.executeQuery(Query2);
                        rs2.next();
                        Employee emp = new Employee(rs2.getString("name") , rs2.getInt("exp") , rs2.getS
tring("desig") , rs2.getString("email"),null);
                        outObj.writeObject(emp);

                    }
                    catch (Exception ex)
                    {
                        System.out.println(ex);
                    }
                }
            }
        } while(ch2 != '\n');
    }
}

```

```
}
```

```
break;
```

```
case 2 :
```

```
System.out.println("-----");
```

```
String line = "";
```

```
// reads message from client until "Over" is sent
```

```
try
```

```
{
```

```
    line = "Welcome to Admin service ";
```

```
    System.out.println("Server : " + line);
```

```
    out.writeUTF(line);
```

```
}
```

```
catch(IOException i)
```

```
{
```

```
    System.out.println(i);
```

```
}
```

```
while(!line.equals("Over"))
```

```
{
```

```
    try
```

```
    {
```

```
        line = in.readUTF();
```

```
        System.out.println("Client : " + line);
```

```
        al.add(line);
```

```
        if(!line.equals("Over"))
```

```
        {
```

```
            System.out.print("Server : ");
```

```
            line = input.readLine();
```

```
            out.writeUTF(line);
```

```
        }
```

```
    }
```

```
    catch(IOException i)
```

```
    {
```

```
        System.out.println(i);
```

```
    }
```

```
}
```

```
break;
```

```
case 3 :
```

```
try
```

```
{
```

```
    Connection con = DriverManager.getConnection(path,"root","root");
```

```
    Statement st = con.createStatement();
```

```
    String Query ="SELECT * FROM employee WHERE name= " + nm +"";
```

```
    ResultSet rs2 = st.executeQuery(Query);
```

```
    if(rs2.next())
```

```
    {
```

```
        String prj_nm;
```

```

        prj_nm = rs2.getString("prj_nm");
        Query="SELECT * FROM project WHERE prj_name= '" + prj_nm + "'";
        rs2 = st.executeQuery(Query);
        if(rs2.next())
        {
            Project p = new Project(rs2.getString("prj_name") , rs2.getString("prj_detail")
, rs2.getInt("d"), rs2.getInt("m"), rs2.getInt("y"), rs2.getInt("progress") ,nm);
            outObj.writeObject(p);
        }
        else
        {
            outObj.writeObject(null);
        }
    }
    else
    {
        outObj.writeObject(null);
    }
}
catch(SQLException ex)
{
    //Failure to signUp
    System.out.println("Error at Show Project : " + "\n\n\t" + ex);
    break;
}
break;
default :
    if(ch2!= 0 )
    {
        System.out.println("Invaild Input from client");
    }
}
}while(ch2 !=0);
}
else
{
    e = null;
    outObj.writeObject(e);
    System.out.println(" -- Login failure for " + nm);
}

break;
case 2 :
    //Employee signup through dbms
    e = (Employee)inObj.readObject();
    try
    {
        Connection con = DriverManager.getConnection(path,"root","root");
        Statement st = con.createStatement();
        String Query ="INSERT INTO employee(name ,exp , desig,email,salary) VALUES ('" + e.get
Name() + "','" + e.getExp() + "','" + e.getDesig() + "','" + e.getEmail() + "','" + e.getSal() + "')";
        st.executeUpdate(Query);
    }
}

```

```

    }
    catch(SQLException ex)
    {
        //Failure to signUp
        System.out.println("SignUp Error for User : " + e.getName() + "\n\n\t" + ex);
        break;
    }

    System.out.println(" -- Client : " + e.getName() + " Signed Up Successfully ");
    System.out.println(" -- Client Details : \n " + e.giveEmp() );

    break;

case 3 :

do{
    ch2 = in.readInt();

    switch(ch2)
    {
        case 1:
            //create project
            Project prj;
            prj = (Project)inObj.readObject();
            try
            {
                Connection con = DriverManager.getConnection(path,"root","root");
                Statement st = con.createStatement();
                String Query ="INSERT INTO project(prj_name ,prj_detail , progress ,d,m,y) VALU
ES (" + prj.prjName() + " , " + prj.prjDetail() + " , " + prj.getProg() + " , " + prj.getDay() + " , " + prj.getMonth() + " , "
+ prj.getYear() + " )";

                st.executeUpdate(Query);

                Query = "UPDATE employee SET prj_nm = " + prj.prjName() + " WHERE name = "
+ prj.getEmp() + "" ;

                st.executeUpdate(Query);
            }
            catch(SQLException ex)
            {
                //Failure to Create project
                System.out.println("Error at Project Creation : " + "\n\n\t" + ex);
                break;
            }
            break;
        case 2:
            //remove emp
            String nm3;
            nm3 = in.readUTF();
            try
            {
                Connection con = DriverManager.getConnection(path,"root","root");
                Statement st = con.createStatement();
                String Query ="SELECT * FROM employee WHERE name = " + nm3 + "" ;

```

```

ResultSet rs2 = st.executeQuery(Query);
if(rs2.next())
{
    String prj_nm = rs2.getString("prj_nm");
    if(prj_nm != null)
    {
        Query="DELETE FROM employee WHERE name = '" + nm3 + "'";
        st.executeUpdate(Query);
        Query="DELETE FROM project WHERE prj_name = '" + prj_nm + "'";
        st.executeUpdate(Query);
        out.writeUTF("\n\n\tEmployee\t:\t"+nm3+" Removed!");
    }
    else
    {
        Query="DELETE FROM employee WHERE name = '" + nm3 + "'";
        st.executeUpdate(Query);
        out.writeUTF("\n\n\tEmployee\t:\t"+nm3+" Removed!");
    }
}
else
{
    out.writeUTF("\n\n\tEmployee\t:\t"+nm3+" Doesn't Exist !");
}
}
catch(SQLException ex)
{
    //Failure to Create project
    System.out.println("Error at Employee Removal : " + "\n\n\t" + ex);
    break;
}
break;
case 3:
    //project removed

nm3 = in.readUTF();
try
{
    Connection con = DriverManager.getConnection(path,"root","root");
    Statement st = con.createStatement();
    String Query ="SELECT * FROM employee WHERE name = '" + nm3 + "'";
    ResultSet rs2 = st.executeQuery(Query);
    if(rs2.next())
    {
        String prj_nm = rs2.getString("prj_nm");
        if(prj_nm != null)
        {
            Query="UPDATE employee SET prj_nm = " + null + " WHERE name = '" + nm
3 + "'";

            st.executeUpdate(Query);
            Query="DELETE FROM project WHERE prj_name = '" + prj_nm + "'";
            st.executeUpdate(Query);
            out.writeUTF("\n\n\tProject\t:\t"+prj_nm+" Removed!");
        }
    }
}

```

```

        else
        {
            out.writeUTF("\n\n\tAlready No Projects for \t :\t"+nm3+" !!");
        }
    }
    else
    {
        out.writeUTF("\n\n\tEmployee\t :\t"+nm3+" Doesn't Exist !");
    }

}
catch(SQLException ex)
{
    //Failure to Create project
    System.out.println("Error at Employee Removal :  " + "\n\n\t" + ex);
    break;
}
break;

case 5 :
    ch2=0;
    break;
}

}while(ch2!=0);

break;

default:
    if(ch != 0)
    {
        System.out.println(" -- Invalid Input from Client !");
    }
}

}while(ch != 0);
}
catch(IOException i)
{
    System.out.println(i);
}
catch(ClassNotFoundException cnf)
{
    System.out.println(cnf);
}

System.out.println("Closing connection");

// close connection
socket.close();
in.close();
input.close();

```



```

        out.close();
    }
    catch(IOException ex)
    {
        Logger.getLogger(Server.class.getName()).log(Level.SEVERE, null,ex);
    }
}
}

```

// Mythread.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```
package a3;
```

```
import java.net.Socket;
```

```

/**
 *
 * @author hp
 */

```

```
public class MyThread extends Thread
```

```

{
    Thread t;
    Socket socket;
    int port;
    MyThread(Socket soc , int p)
    {
        socket = soc;
        port =p;
        t= new Thread(this);
        t.start();
    }

```

```

    public void run()
    {
        Server.runServer(socket, port);
    }

```

```
}
```

// Client.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

```

```
package a3;
```

```
// A Java program for a Client
```

```
import java.net.*;
```

```
import java.io.*;
```

```
import java.util.*;
```

```
class Client
```

```
{
```

```
    // initialize socket and input output streams
```

```
    private Socket socket      = null;
```

```
    private BufferedReader input = null;
```

```
    private DataOutputStream out  = null;
```

```
    private DataInputStream in    = null;
```

```
    private ObjectOutputStream outObj = null;
```

```
    private ObjectInputStream inObj = null;
```

```
    // constructor to put ip address and port
```

```
    public Client(String address, int port)
```

```
    {
```

```
        // establish a connection
```

```
        try
```

```
        {
```

```
            socket = new Socket(address, port);
```

```
            System.out.println("Connected");
```

```
            // takes input from terminal
```

```
            input = new BufferedReader(new InputStreamReader(System.in));
```

```
            // sends output to the socket
```

```
            out = new DataOutputStream(socket.getOutputStream());
```

```
            outObj = new ObjectOutputStream(socket.getOutputStream());
```

```
            in = new DataInputStream(
```

```
                new BufferedInputStream(socket.getInputStream());
```

```
            inObj = new ObjectInputStream(socket.getInputStream());
```

```
        }
```

```
        catch(UnknownHostException u)
```

```
        {
```

```
            System.out.println(u);
```

```
        }
```

```
        catch(IOException i)
```

```
        {
```

```
            System.out.println(i);
```

```
        }
```

```
    int ch,ch2;
```

```
    Employee e;
```

```
    boolean f;
```

```
    Combine c;
```

```
String nm,msg;
Employee response;
Scanner sc = new Scanner(System.in);
try{
    do{
        System.out.println("\n\n\t1.Login \n\t2.SignUp \n\t3.Employee Manager \n\t0.Exit");
        System.out.print("\n\tEnter Your Choice  : ");
        ch = sc.nextInt();
        out.write(ch);
        switch(ch)
        {
            case 1 :
                //Login
                System.out.print("\n\n\tEnter Name  : ");
                nm = sc.nextLine();
                nm = sc.nextLine();
                out.writeUTF(nm);
                f = in.readBoolean();

                if(f)
                {
                    System.out.println("Login Successful !");

                    do{
                        System.out.println("\n\n\t1.Profile \n\t2.Help \n\t3.Projects \n\t0.LogOut");
                        System.out.print("Enter Your Choice  : ");
                        ch2 = sc.nextInt();
                        out.write(ch2);
                        switch(ch2)
                        {
                            case 1 :
                                try{
                                    response = (Employee)inObj.readObject();
                                    System.out.println(response.giveEmp());
                                }
                                catch(Exception ex)
                                {
                                    System.out.println(ex);
                                }
                                break;
                            case 2 :
                                String line = "";
                                System.out.println("\n\t-----");
                                while(!line.equals("Over"))
                                {
                                    try
                                    {
                                        line = in.readUTF();
                                        System.out.println("Server  : " + line);
                                        if(!line.equals("Over"))
                                        {
                                            System.out.print("Client  : ");
                                            line = input.readLine();
                                            out.writeUTF(line);
                                        }
                                    }
                                }
                            case 3 :
                                System.out.print("\n\tEnter Project Name : ");
                                nm = sc.nextLine();
                                nm = sc.nextLine();
                                out.writeUTF(nm);
                                f = in.readBoolean();
                                if(f)
                                {
                                    System.out.println("Project Added Successfully");
                                }
                                else
                                {
                                    System.out.println("Project Name already exists");
                                }
                                break;
                            case 0 :
                                System.out.println("Exit");
                                break;
                        }
                    }
                }
            case 2 :
                System.out.print("\n\tEnter Email : ");
                nm = sc.nextLine();
                nm = sc.nextLine();
                out.writeUTF(nm);
                f = in.readBoolean();
                if(f)
                {
                    System.out.println("SignUp Successful");
                }
                else
                {
                    System.out.println("Email already exists");
                }
                break;
            case 3 :
                System.out.print("\n\tEnter Employee ID : ");
                nm = sc.nextLine();
                nm = sc.nextLine();
                out.writeUTF(nm);
                f = in.readBoolean();
                if(f)
                {
                    System.out.println("Employee Manager");
                }
                else
                {
                    System.out.println("Employee ID does not exist");
                }
                break;
            case 0 :
                System.out.println("Exit");
                break;
        }
    }
}
catch(Exception ex)
{
    System.out.println(ex);
}
}
```

```

        }
    }
    catch(IOException i)
    {
        System.out.println(i);
    }
}
break;
case 3 :
try
{
    Project p = (Project) inObj.readObject();
    if(p!=null)
        p.showPrj();
    else
        System.out.println("No Projects Available !");
}
catch(Exception exp)
{
    System.out.println(exp);
}

break;
default :
    if(ch2 != 0)
    {
        System.out.println("Invalid Choice !");
    }

}

}while(ch2!=0);
}
else
{
    System.out.println("Login Failed !");
}
break;
case 2 :
    //SignUp
    e = new Employee();
    e.getEmp();
    outObj.writeObject(e);
    System.out.println("SignUp Successful !");
    break;

case 3 :
    System.out.println("\n\t----- Employee Manager Section -----");
    System.out.print("\n\tEnter UserName : ");
    nm = sc.nextLine();
    nm = sc.nextLine();
    if(nm.equals("admin") || nm.equals("Admin") || nm.equals("ADMIN"))
    {

```

");

```
do{
    System.out.println("\n\n\t1.Create Project \n\t2.Remove Employee \n\t3.Delete Project\n\t0.Exit
```

```
System.out.print("\n\tEnter your choice  :  ");
```

```
ch2 = sc.nextInt();
```

```
out.writeInt(ch2);
```

```
switch(ch2)
```

```
{
```

```
    case 1 :
```

```
        //create project
```

```
        Project prj=new Project();
```

```
        prj.getProject();
```

```
        outObj.writeObject(prj);
```

```
        System.out.println("\n\n\tProject Created Successfully!!");
```

```
        break;
```

```
    case 2 :
```

```
        //remove employee
```

```
        System.out.print("\n\n\tEnter Employee Name  :  ");
```

```
        nm = sc.nextLine();
```

```
        nm = sc.nextLine();
```

```
        out.writeUTF(nm);
```

```
        System.out.println(in.readUTF());
```

```
        break;
```

```
    case 3 :
```

```
        //remove project
```

```
        System.out.print("\n\n\tEnter Employee Name  :  ");
```

```
        nm = sc.nextLine();
```

```
        nm = sc.nextLine();
```

```
        out.writeUTF(nm);
```

```
        System.out.println(in.readUTF());
```

```
        break;
```

```
    default :
```

```
        if(ch2 !=0)
```

```
        {
```

```
            System.out.println("\n\tInvalid Choice");
```

```
        }
```

```
    }
```

```
    }while(ch2!=0);
```

```
}
```

```
else{
```

```
    out.writeInt(5);
```

```
}
```

```
break;
```

```
default:
```

```
if(ch!=0)
```

```
{
```

```
    System.out.println("Invalid Input");
```

```
}
```

```
break;
```

```
}
```

```
}while(ch != 0);
```

```
}
```

```

        catch(IOException i)
        {
            System.out.println(i);
        }

        // close the connection
        try
        {
            input.close();
            out.close();
            socket.close();
            in.close();
        }
        catch(IOException i)
        {
            System.out.println(i);
        }
    }

    public static void main(String args[])
    {
        System.out.println("Client");
        Client c = new Client("localhost" , 8000);

    }
}

```

//emmployee.java

```

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package a3;
import java.io.Serializable;
import java.util.Scanner;

public class Employee implements Serializable {
    String name;
    int exp;
    int id;
    String desig;
    String email;
    float sal;
}

```

```
static int st_id = 0;
String prj;
```

```
public Employee(){
    name="";
    exp=0;
    desig="";
    email="";
    sal = exp<=5 ? 30000 : 50000;
    id=st_id;
    st_id++;
    prj = "";
}
```

```
public Employee(String nm , int ex , String des ,String mail, String p){
    name = nm;
    exp=ex;
    desig = des;
    email = mail;
    sal = exp<=5 ? 30000 : 50000;
    id = st_id;
    st_id++;
    prj = p;
}
```

```
public Employee(Employee e)
{
    name = e.name;
    exp = e.exp;
    desig = e.desig;
    email = e.email;
    sal = exp<=5 ? 30000 : 50000;
    prj = e.prj;
    id = st_id;
    st_id++;
}
```

```
public String givePrj()
{
    return prj;
}
```

```
public void setPrj(String p)
{
    prj = p;
}
```

```
public String getName()
{
    return name;
}
```

```
public String getDesig()
{
    return desig;
}
```

```
public String getEmail()
{
    return email;
}
```

```
public int getExp()
{
    return exp;
}
```

```
public float getSal()
{
    return sal;
}
```

```
public String giveEmp()
{
    String msg;
    msg = "-----" +
        "\n\tName      : " + name +
        "\n\tExpierence  : " + exp +
        "\n\tDesignation : " + desig +
        "\n\tContact    : " + email +
        "\n\tSalary     : Rs. " + sal +
        "\n\n-----";
    return msg;
}
```

```
public void getEmp()
{
    Scanner sc = new Scanner(System.in);
    System.out.print("\n\n\tEnter Name      : ");
    name = sc.nextLine();
    System.out.print("\n\tEnter Experience  : ");
    exp = sc.nextInt();
    System.out.print("\n\tEnter Designation : ");
    desig = sc.nextLine();
    desig = sc.nextLine();
    System.out.print("\n\tEnter Email      : ");
    email = sc.nextLine();
    sal = exp<=5 ? 30000 : 50000;

}
```

```
public void showEmp()
{
    System.out.println("-----");
}
```



```

        System.out.println("\tName      : " + name);
        System.out.println("\tExpierence   : " + exp + " Years");
        System.out.println("\tDesignation : " + desig);
        System.out.println("\tContact    : " + email);
    }

    public static void main(String[] args) {

    }
}

//Project.java

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package a3;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.Scanner;

public class Project implements Serializable{
    String prjTaskName;
    String emp;
    String prjTaskDetail;
    int d,m,y;
    int progress;

    public Project()
    {
        prjTaskName = "";
        prjTaskDetail = "";
        d=1;m=1,y=2020;
        progress = 0;
        emp="";
    }

    public Project(String prj_name ,String prj_detail,int dd,int mm,int yy,int prg ,String emp_nm)
    {
        prjTaskName = prj_name;
        prjTaskDetail = prj_detail;
        d=dd;m=mm;y=yy;
        progress = prg;
        emp = emp_nm;
    }

    public Project(Project pr)
    {
        prjTaskName = pr.prjTaskName;
        prjTaskDetail = pr.prjTaskDetail;
    }
}

```

```
    d = pr.d;
    m = pr.m;
    y = pr.y;
    progress = pr.progress;
    emp = pr.emp;
}
```

```
public Project(Project pr , int p)
{
    prjTaskName = pr.prjTaskName;
    prjTaskDetail = pr.prjTaskDetail;
    d = pr.d;
    m = pr.m;
    y = pr.y;
    progress = p;
    emp = pr.emp;
}
```

```
public int getProg()
{
    return progress;
}
```

```
public void setProg(int p)
{
    progress = p;
}
```

```
public String prjName()
{
    return prjTaskName;
}
```

```
public String prjDetail()
{
    return prjTaskDetail;
}
```

```
public int getDay()
{
    return d;
}
```

```
public int getMonth()
{
    return m;
}
```

```
public int getYear()
{
    return y;
}
```

```
public String getEmp()
{
    return emp;
}
```

```
public void showPrj()
```

```

{
    System.out.println("\n\tProject Task Name   : " + prjTaskName);
    System.out.println("\n\tProject Task Detail : " + prjTaskDetail);
    System.out.println("\n\tProject Deadline   : " + d + "/" + m + "/" + y);
}

public void getProject()
{
    Scanner sc = new Scanner(System.in);

    System.out.print("\n\n\tEnter Employee Name   : ");
    emp = sc.nextLine();

    System.out.print("\n\tEnter Task Name   : ");
    prjTaskName = sc.nextLine();
    System.out.print("\n\tEnter Project Task Details : ");
    prjTaskDetail = sc.nextLine();

    System.out.println("\n\n\tEnter Deadline   : ");
    System.out.print("\n\t\tDay : ");
    d = sc.nextInt();
    System.out.print("\n\t\tMonth : ");
    m = sc.nextInt();
    System.out.print("\n\t\tYear : ");
    y = sc.nextInt();

}

}

```