1. Write a Servlet application for fetching the entire data from the database and showing it as tablein response webpage.Use the following query in MySQL for creating a table which contains employee details.create table employee(empid varchar(10), empname varchar(20), age integer, salary integer);

•Connection Interface

Package com.jdbc.demo.connection;

Connection interface public interface dBDetails {

String CONSTR = "jdbc:mysql://localhost:3306/cdac_tvm?useSSL=false";

String DBDDRIVER = "com.mysql.cj.jdbc.Driver"; String USERNAME = "root";

String PASSWORD = "radhesham";

}

•Connection class

package com.jdbc.demo.connection;

import java.sql.Connection;

```
import java.sql.DriverManager;
import java.sql.SQLException;

public class DbConnection {
        public static Connection getDbConnection()
        {
                try {
                Class.forName(dBDetails.DBDDRIVER);
                Connection con=
DriverManager.getConnection(dBDetails.CONSTR,dBDetails.US

        ERNAME,dBDetails.PASSWORD);
                return con;
                  }
                catch(ClassNotFoundException |SQLException exc) {
                  exc.printStackTrace(); return null;
                  }
        }
}
```

- **EMPLOYEE POJO CLASS**

```
package com.jdbc.demo.pojo;

Employee class public class Employee {
	private int id;
	private String ename;
	private int age;
	private int salary;

	public int getId() { return id;
	}
	public void setId(int id) {
	this.id = id;
	}
	public String getEname() {
	return ename;
	}
	public void setEname(String ename) {
	this.ename = ename;
	}
	public int getAge() {
	return age;
```

```
	}
	public void setAge(int age) {
	this.age = age;
	}
	public int getSalary() {
	return salary;
	}
	public void setSalary(int salary) {
	this.salary = salary;
	}
	@Override
	public String toString() { return "Employee [id=" + id + ",
ename=" + ename + ",
	age=" + age + ", salary=" + salary + "]";
	}
}
```

•Employee DAO CLASS

```
package com.jdbc.demo.dao;
import java.util.List;
```

```
import com.jdbc.demo.pojo.Employee;

public interface Employeedao {

        List<Employee> getAllEmployee();
        Employee searchEmployee(int EmpId);

        boolean addNewEmployee(Employee Empmloyee);
        boolean updateEmployee(Employee Employee);
        boolean deleteEmployee(Employee EmpId);
}
```

- **IMPLEMENTATION OF EMPLOYEE DAO CLASSS**

```
package com.jdbc.demo.empImp;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet; import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
```

```
import com.jdbc.demo.connection.DbConnection;
  import com.jdbc.demo.dao.Employeedao;
import com.jdbc.demo.pojo.Employee;


public class EmployeeDaoImp implements Employeedao{
	@Override
	public List<Employee> getAllEmployee() { List<Employee>
lst=new ArrayList<>();
	try(Connection con=DbConnection.getDbConnection())
	{
	PreparedStatement pst=con.prepareStatement("SELECT * FROM
Employee");


	ResultSet rs=pst.executeQuery();
	while(rs.next()) {
		Employee emp=new Employee();
		emp.setId(rs.getInt("eid"));
		emp.setEname(rs.getString("ename"));
		emp.setAge(rs.getInt("age"));
		emp.setSalary(rs.getInt("salary"));
		lst.add(emp);
		}
		return lst;
```

```
        }
        catch(NullPointerException |SQLException exc)
        {
                exc.printStackTrace(); return null;
        }
        }

@Override
public Employee searchEmployee(int EmpId) {
        Employee emp=null; try(Connection
con=DbConnection.getDbConnection())
        {
                PreparedStatement pst=con.prepareStatement("SELECT
*FROM Employee WHERE eid=?");

        ResultSet rs=pst.executeQuery();
        if(rs.isBeforeFirst()) {
                rs.next();
                emp=new Employee();
                emp.setId(rs.getInt("eid"));
                emp.setEname(rs.getString("ename"));
                emp.setAge(rs.getInt("age"));
                emp.setSalary(rs.getInt("salary"));
```

```
            return emp;
                                }
        return emp;
        }
        catch(SQLException|NullPointerException exc)
        {
            exc.printStackTrace(); return null;
                    }
        }
@Override
public boolean addNewEmployee(Employee Employee)
{
        try(Connection con=DbConnection.getDbConnection())
        {
            PreparedStatement pst=con.prepareStatement("INSERT
            INTO Employee(ename,age,salary)VALUES (?,?,?)",
            Statement.RETURN_GENERATED_KEYS);
            pst.setString(1,Employee.getEname());
            pst.setInt(2,Employee.getAge());
            pst.setInt(3, Employee.getSalary());
            int count=pst.executeUpdate();
            ResultSet rs=pst.getGeneratedKeys();
```

```
            rs.next();
            System.out.println("generated id is"+rs.getInt(1));
            if(count>0) {
                return true;
                }
            else {
                return false;
            }
        catch(SQLException | NullPointerException exc)
        {
            exc.printStackTrace(); return false;
        }
}

@Override
public boolean updateEmployee(Employee Employee) {
        try(Connection con=DbConnection.getDbConnection())
        {
        PreparedStatement pst=con.prepareStatement("UPDATE
Employee SET ename=?,age=?,salary=?"
                                                    + " WHERE eid=?");
        pst.setString(1,Employee.getEname());
```

```
        pst.setInt(2, Employee.getAge());
        pst.setInt(3, Employee.getSalary());
        pst.setInt(4, Employee.getId());
        int count =pst.executeUpdate();
                if(count>0) {
                        return true;
                }
                else {
                        return false;
                }
        }
        catch(SQLException | NullPointerException exc)
        {
                exc.printStackTrace();
                return false;
        }
}
@Override
public boolean deleteEmployee(Employee EmpId)
{
                // TODO Auto-generated method stub return false;
}
```

}

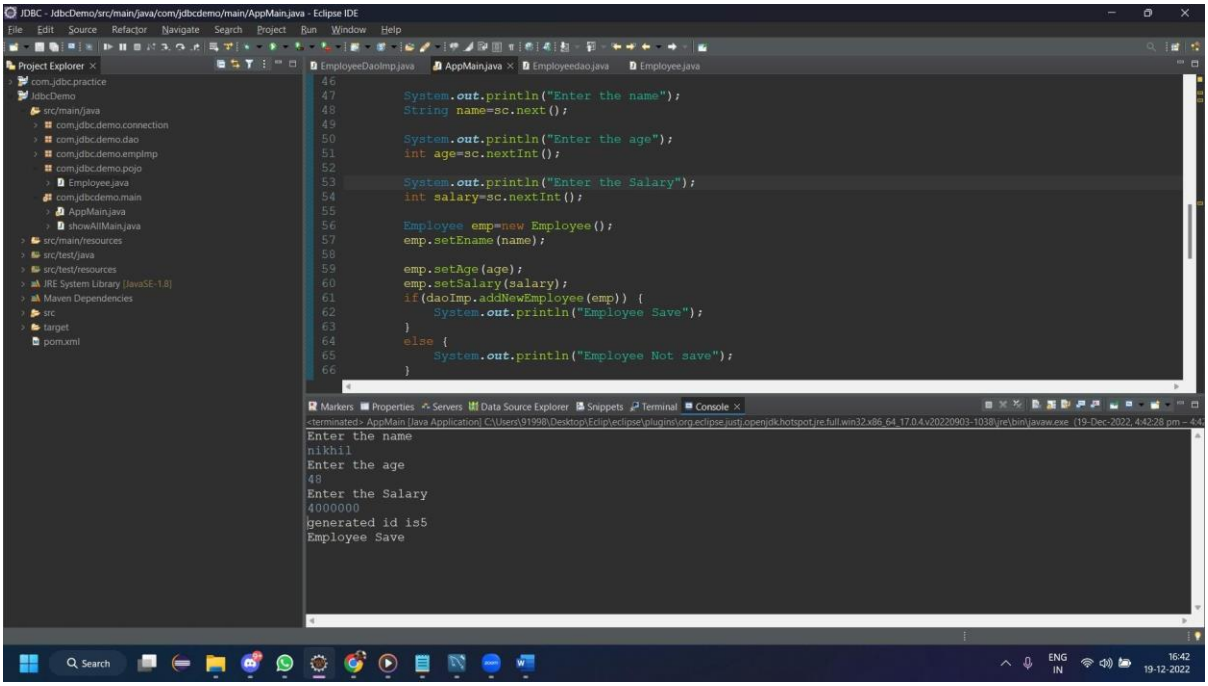
• Main class

```
package com.jdbcdemo.main;
import java.util.List;
import java.util.Scanner;
import com.jdbc.demo.dao.Employeedao;
import com.jdbc.demo.empImp.EmployeeDaoImp;
import com.jdbc.demo.pojo.Employee;

public class AppMain {
	public static void main(String[] args)
	{

		EmployeeDaoImp daoImp=new EmployeeDaoImp();
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the name");
		String name=sc.next();
```

```
System.out.println("Enter the age");
  int age=sc.nextInt();
System.out.println("Enter the Salary");
int salary=sc.nextInt();
Employee emp=new Employee();
emp.setEname(name);
emp.setAge(age);
emp.setSalary(salary);
if(daoImp.addNewEmployee(emp))
  {
        System.out.println("Employee Save");
}
else
{
        System.out.println("Employee Not save");
}
  }
}
```



	eid	ename	age	salary
▶	1	atharva	23	1000
	2	Kshitij	21	500000
	3	pranit	24	2000
	4	RAHUL	33	5000
	5	nikhil	48	4000000
*	NULL	NULL	NULL	NULL