Digital Assignment-2

Sub: CSE1007

SLOT: L43+L44

Faculty: Prof.Lokeshkumar R

Submitted by

Name: Rahul Kumar Sahani

Reg No: 20BDS0126

Exercise -1

Develop a java application to perform operations like insert, update, retrieve and delete in employee database using JDBC connectivity. Using Preferred DBMS (like Oracle / MySql/Others) to perform this

Steps:-

1. Create Employee Table with the following Fields
emp_id number, empname varchar2(10), email varchar2(30), city varchar2(10) salary number);
2. Connect the DB and perform the following operations
a. Insert 10 employee details
b. Use select Query to retrieve Employee with same salary
c. Delete the employee details belong to same city
d. Update operation to add the last name to the employee

Code:

```
/*
 * 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 myjdbc;


/**
 *
 * @author Asus
```

```
 */
//Kindly change the package name, class name and DBMS connnection URL as per your system
//settings before running this code.
import java.sql.*;
import java.util.Scanner;
public class MYjdbc {


   /**
    *
    * @param args
    * @throws InstantiationException
    * @throws IllegalAccessException
    * @throws ClassNotFoundException
    * @throws SQLException
    */
      public static void main(String[] args) throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException{


       try
       {    int choice=0;
          emp s = new emp();
          do
          {
              System.out.println("Select an operation \n 1- Insert \n 2-  Update Item \n 3- Delete a Record \n
4- Search for a Item \n 5- Exit");
              Scanner choicein = new Scanner(System.in);
              choice=choicein.nextInt();
              switch(choice)
              {
                  case 1:
```

```
                    s.getempDetails();
                    s.insertemp();
                    break;
                case 2:
                    s.updateempSalary();
                    break;
                case 3:
                    s.deleteempRecord();
                    break;
                case 4:
                    s.searchemp();
                    break;
                case 5:
                    break;
                default:
                    System.out.println("Select the correct choice");
              }
           }while(choice!=5);
           System.out.println("Thanks for Using our Software");
        }
        catch(Exception e)
        {
           System.out.println(e.getMessage());
        }
     }
}
class employee
{
    private String emp_id;
```

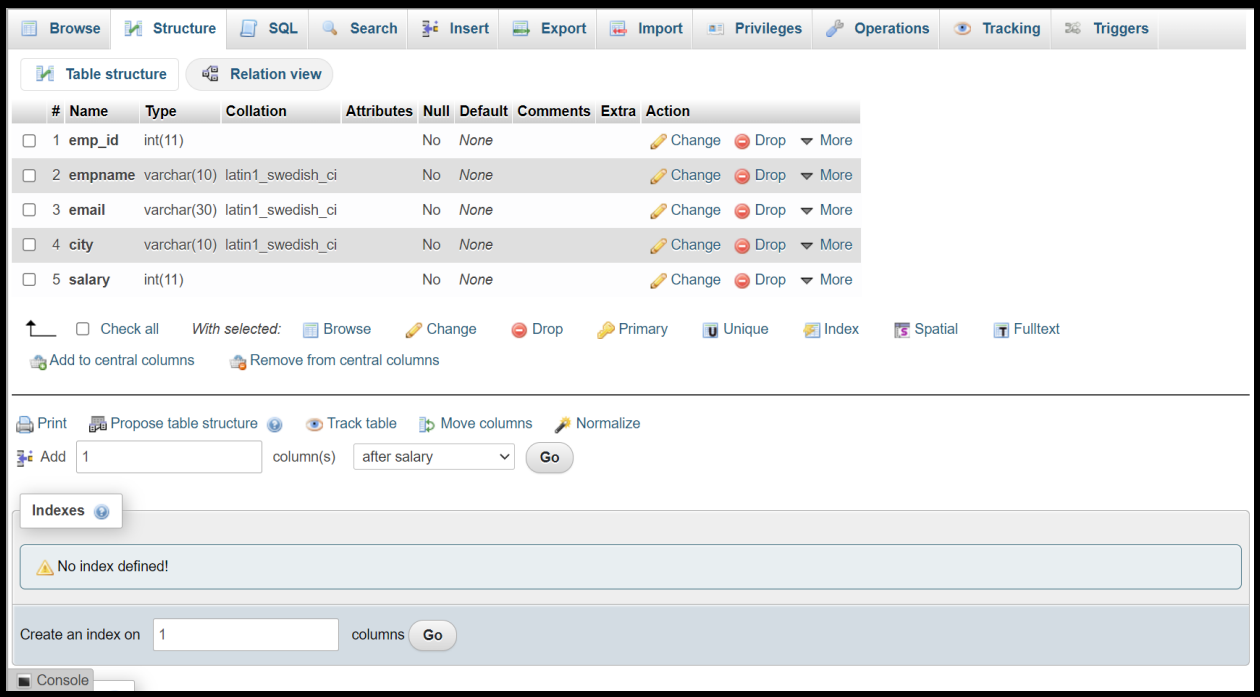
```
    private String empname ;
    private String email;
    private int city;
    private int salary;



    public void getempDetails() {
        Scanner input = new Scanner(System.in);
        System.out.println("Enter the emp_id");
        Emp_id = input.nextInt();
        System.out.println("Enter empname");
        empname = input.nextLine();
        System.out.println("Enter email");
        email = input.nextLine();
        System.out.println("Enter city");
        city = input.nextLine();
System.out.println("Enter salary");
        salary = input.nextInt();


    }


    public void insertemp() throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException {
        //here we are going to work with a database
        //we need to open a database connection
        dbmsconnection dbmsconnect = new
dbmsconnection("jdbc:mysql://localhost:3306/mypc","root","");
        Connection con = dbmsconnect.getConnection();
        String sql = "insert into emp values (?,?,?,?);";
```

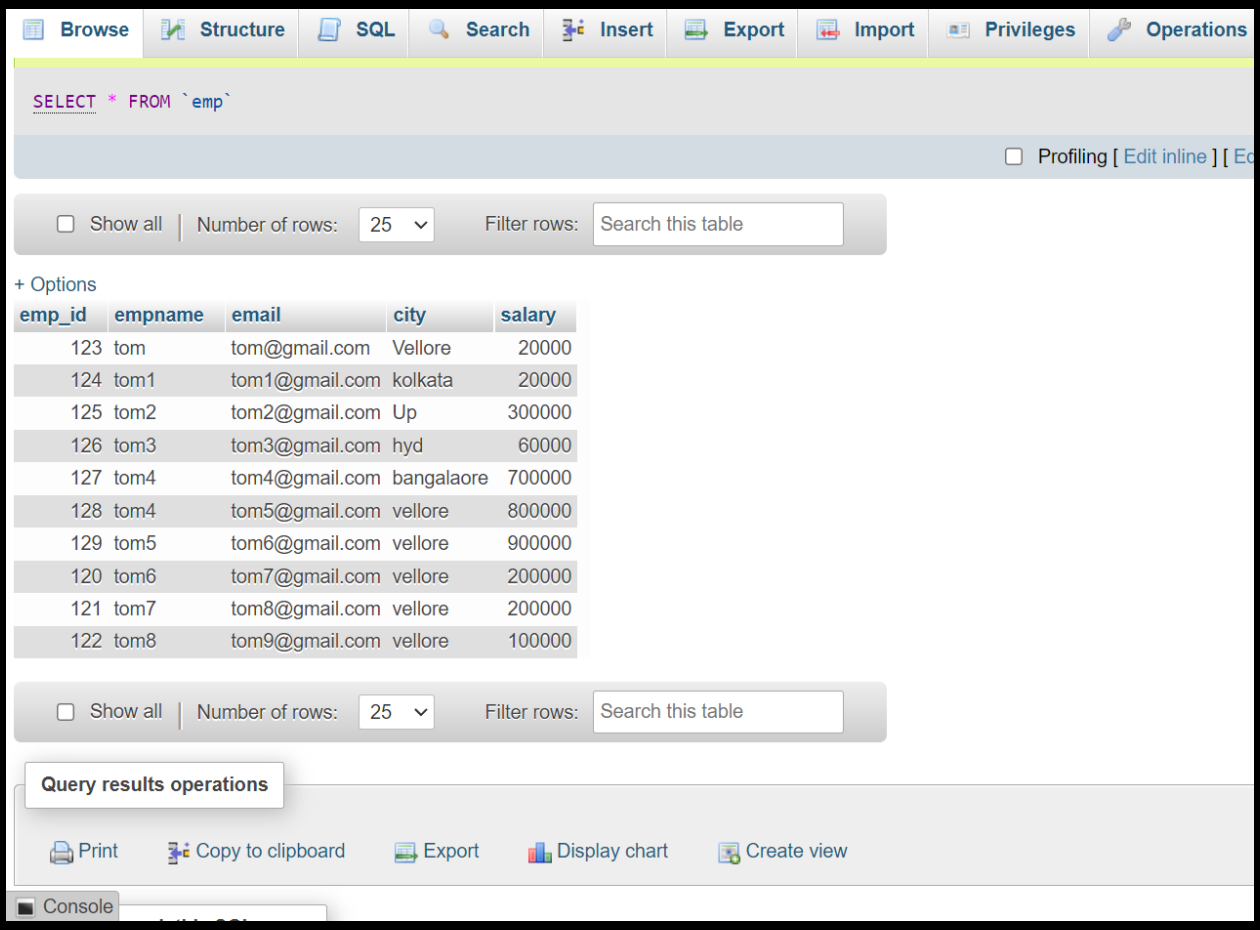
```
        PreparedStatement stmt = con.prepareStatement(sql);
        stmt.setInt(1, emp_id);
        stmt.setString(2, empname);
        stmt.setString(3, email);
        stmt.setString(4, city);
              stmt.setInt(5, salary);

        int i = stmt.executeUpdate();
        System.out.println("Record  inserted successfully");
        dbmsconnect.closeConnection(con, stmt);
    }

    public void updateStudentPassword() throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException {
        dbmsconnection dbmsconnect = new
dbmsconnection("jdbc:mysql://localhost:3306/mypc","root","");
        Connection con = dbmsconnect.getConnection();
        System.out.println("Enter emp_id");
        Scanner input = new Scanner(System.in);
        String inputitem=input.nextLine();
        System.out.println("Enter the new salary");
        String inputcost=input.nextLine();
        String sql = "update emp set salary= ? where item = ?;";
        PreparedStatement stmt = con.prepareStatement(sql);
        stmt.setString(1, inputemp_id);
        stmt.setString(2, inputsalary);
        int i = stmt.executeUpdate();
        if(i>0)
        {
```

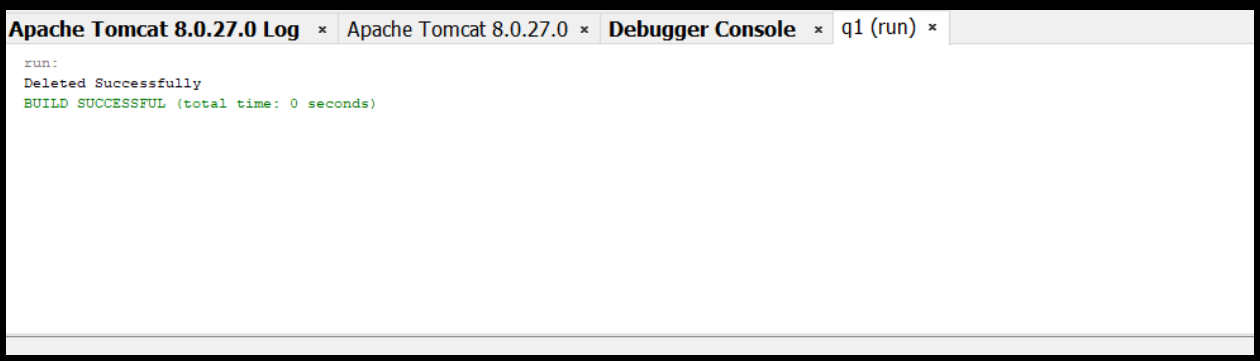
```
            System.out.println("updated sucessfully");
        }else
        {
            System.out.println("No Such record in the Database");
        }
        dbmsconnect.closeConnection(con, stmt);
    }



public void deleteempRecord() throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException {
    dbmsconnection dbmsconnect = new
dbmsconnection("jdbc:mysql://localhost:3306/mypc","root","");
    Connection con = dbmsconnect.getConnection();
    System.out.println("Enter the Name ");
    Scanner input = new Scanner(System.in);
    String inputemp_id=input.nextLine();
    String sql = "delete from Item where item = ?;";
    PreparedStatement stmt = con.prepareStatement(sql);
    stmt.setString(1, inputitem);
    int i = stmt.executeUpdate();
    if(i>0)
    {
        System.out.println("Record Deleted Successfully");
    }
    else
    {
        System.out.println("No Such Record in the Database");
    }
```

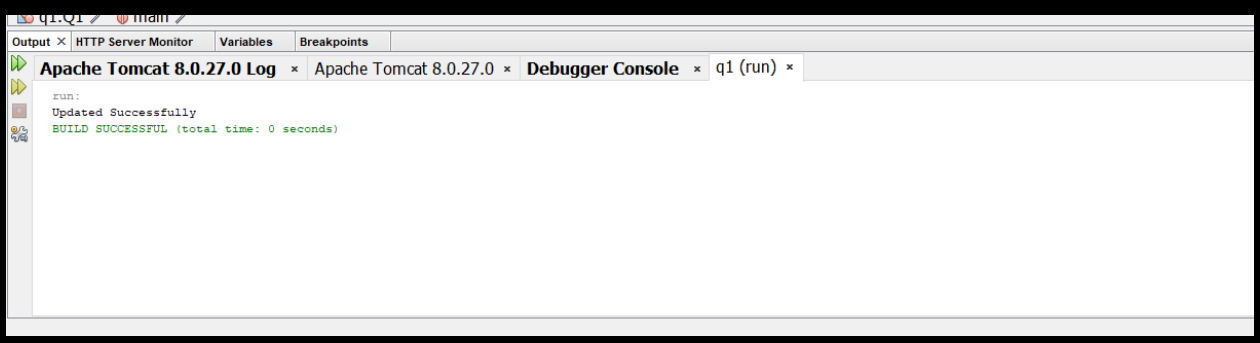
```
    dbmsconnect.closeConnection(con, stmt);
}

public void searchStudent() throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException {
    dbmsconnection dbmsconnect = new
dbmsconnection("jdbc:mysql://localhost:3306/mypc","root","");
    Connection con = dbmsconnect.getConnection();
    System.out.println("Enter emp_id");
    Scanner input = new Scanner(System.in);
    String inputname=input.nextLine();
    String sql = "select * from emp where empid=?";
    PreparedStatement stmt = con.prepareStatement(sql);
    stmt.setString(1, inputname);
    ResultSet rs = stmt.executeQuery();
    if(rs.next()==false)
    {
       System.out.println("No such record found in the database");
    }
    else
    {
       System.out.println(rs.getString(1)+rs.getString(2)+rs.getString(3)+rs.getInt(4));

    }
    dbmsconnect.closeConnection(con, stmt);
}
}

class dbmsconnection
```

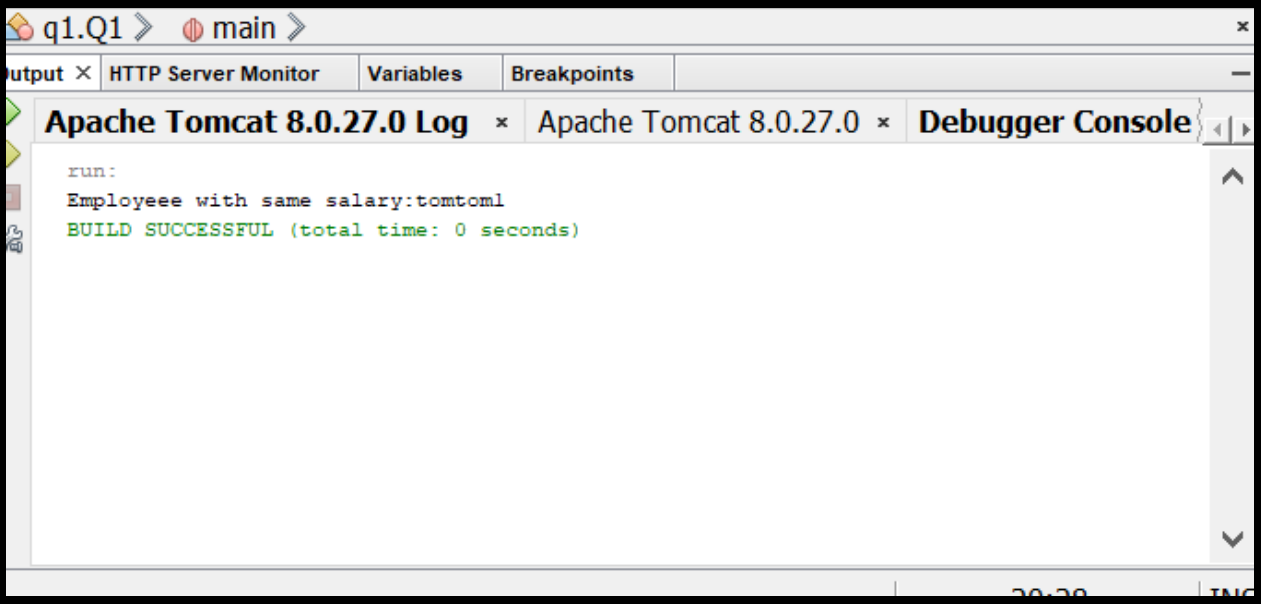
```
{
    String url;
    String username;
    String password;

    public dbmsconnection(String url, String username, String password) {
        this.url = url;
        this.username = username;
        this.password = password;
    }

    public Connection getConnection() throws InstantiationException, IllegalAccessException,
ClassNotFoundException, SQLException {
        Connection con=null;
        Class.forName("com.mysql.rh.jdbc.Driver").newInstance();
        con = DriverManager.getConnection(url,username,password);
        System.out.println("Connection Established Successfully");
        return con;
    }

    public void closeConnection(Connection con,Statement stmt) throws SQLException
    {
        stmt.close();
        con.close();
        System.out.println("The connection is closed");
    }
}
```


Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Table structure
Relation view
Name Type Collation Attributes Null Default Comments Extra Action
1 emp_id int(11) No None Change Drop More
2 empname varchar(10) latin1_swedish_ci No None Change Drop More
3 email varchar(30) latin1_swedish_ci No None Change Drop More
4 city varchar(10) latin1_swedish_ci No None Change Drop More
5 salary int(11) No None Change Drop More
Check all
With selected:
Browse
Change
Drop
Primary
Unique
Index
Spatial
Fulltext
Add to central columns
Remove from central columns
Print
Propose table structure
Track table
Move columns
Normalize
Add
1
column(s)
after salary
Go
Indexes
No index defined!
Create an index on
1
columns
Go
Console

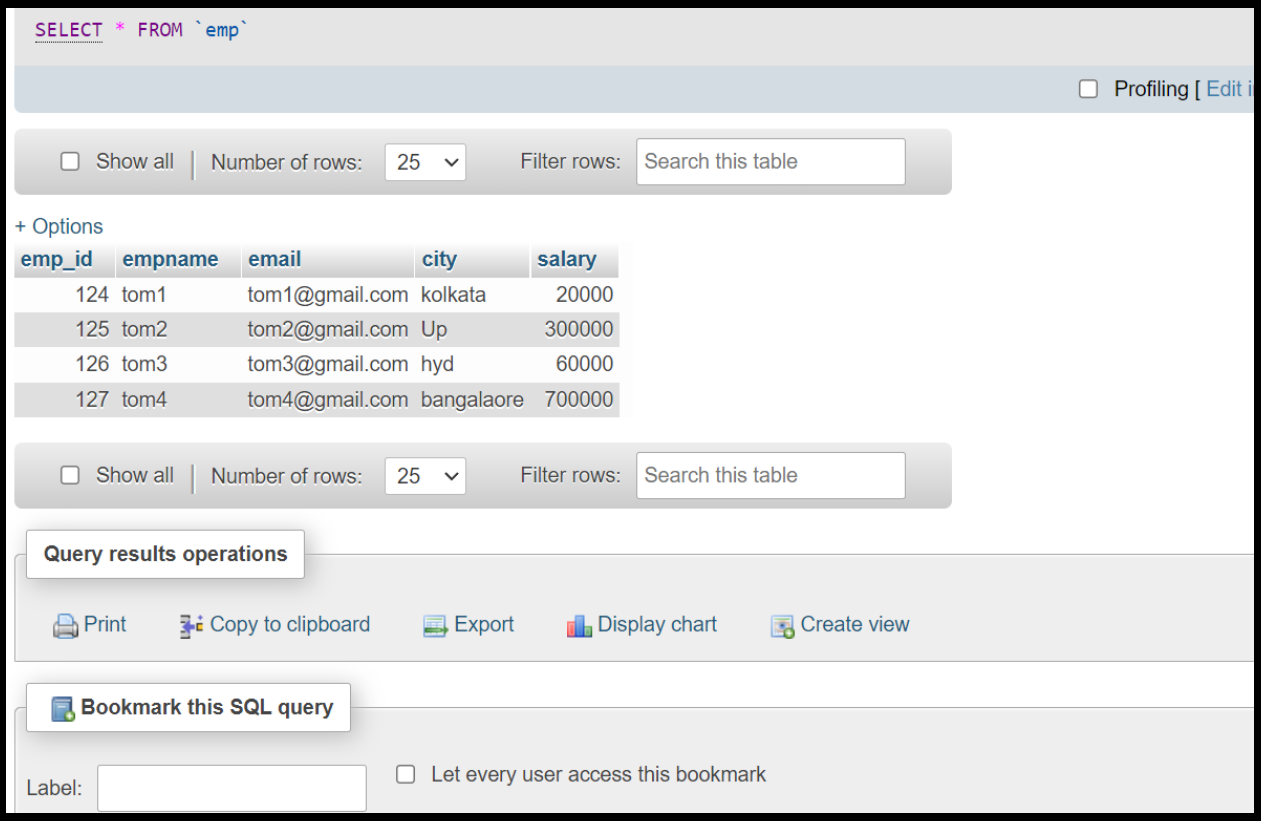

Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
SELECT * FROM `emp`
Profiling [Edit inline] [Ed
Show all | Number of rows: 25 Filter rows: Search this table
+ Options
emp_id empname email city salary
123 tom tom@gmail.com Vellore 20000
124 tom1 tom1@gmail.com kolkata 20000
125 tom2 tom2@gmail.com Up 300000
126 tom3 tom3@gmail.com hyd 60000
127 tom4 tom4@gmail.com bangalaore 700000
128 tom4 tom5@gmail.com vellore 800000
129 tom5 tom6@gmail.com vellore 900000
120 tom6 tom7@gmail.com vellore 200000
121 tom7 tom8@gmail.com vellore 200000
122 tom8 tom9@gmail.com vellore 100000
Show all | Number of rows: 25 Filter rows: Search this table
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Console


Apache Tomcat 8.0.27.0 Log
Apache Tomcat 8.0.27.0
Debugger Console
q1 (run)
run:
Deleted Successfully
BUILD SUCCESSFUL (total time: 0 seconds)


Output
HTTP Server Monitor
Variables
Breakpoints
Apache Tomcat 8.0.27.0 Log
Apache Tomcat 8.0.27.0
Debugger Console
q1 (run)
run:
Updated Successfully
BUILD SUCCESSFUL (total time: 0 seconds)



Deleted where city is vellore



Add last name.

empname
tom1 lastn
tom1 lastn
tom1 lastn
tom1 lastn

Exercise -2

Get the following details from the user in a HTML form
Name :
Mobile :
E-Mail :
Model : (Add 2 more fields of your choice for Bike description)
Loan Amount Required:
Duration in Months :
ROI : (get from the user) (Users choice)

Write a servlet program to calculate the Two wheeler EMI and display the User Details along with the EMI per month, Total Amount paid with Interest after the Tenure / Duration in the output screen.

Code:

Index.html

20BDS0126_Rahul kumar Sahani

Name:

Mobile :

Email

Loan Amount Required:

Duration in months:

ROI:

EMI

Submit

```
<!DOCTYPE html>
<!--
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.
-->
<html>
  <head>
    <title>EMI</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body style="background-color:#000000; color: white" >
    <fieldset>
```

```
<form action="./serv" method="get">
   <div style="text-align:center; color: #993300">20BDS0126_Rahul kumar Sahani</div>
   <table>
      <tr><td><div><label>Name:</label>
            <input type ="text" name="Name"></div></td></tr>
<tr><td><div><label>Mobile :</label>
<input type ="text" name="Mobile"></div>
<tr><td><div><label>Email</label>
<input type ="email" name="email"></div></tr></td>
<tr><td><div><label>Loan Amount Required: </label>
<input type ="text" name="Loan Amount Required"></div></tr></td>
<tr><td><div><label>Duration in months:</label>
<input type ="text" name="Duration"></div></tr></td>
<tr><td><div><label>ROI:</label>
   <input type ="text" name="ROI" placeholder="(get from the user) (User
choice)"></div></tr></td>
   <tr><td><select name="op">

      <option value ="EMI"> EMI</option></tr></td>
   </select>
   </table>


<tr><td><div><input type ="submit" value="Submit"></div></tr></td>
</fieldset>
      </form>

<style>
   form{
```

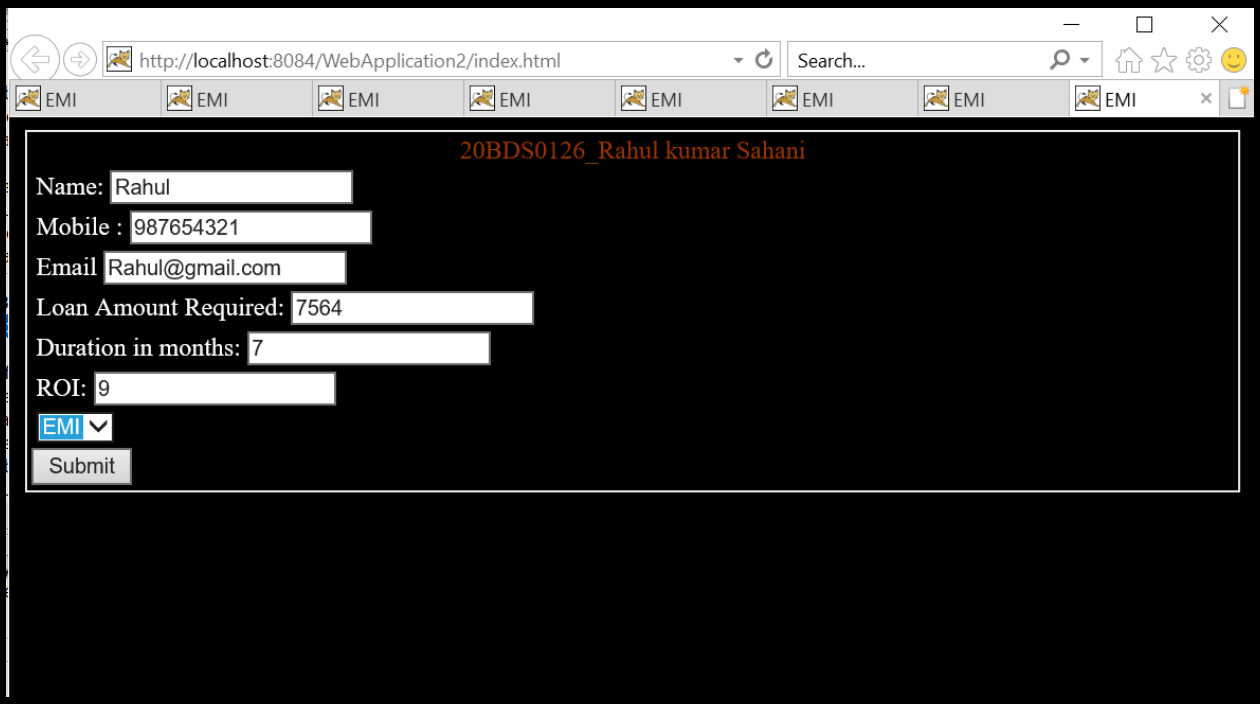
```
            text-align: inherit;
        }
    </style>
    <div></div>

</body>
</html>
```

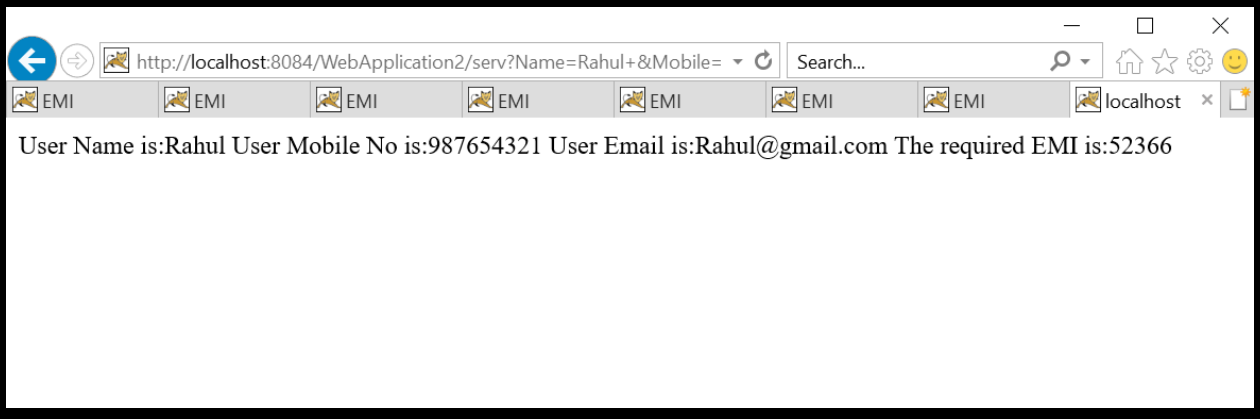
Serv.java

```
import java.io.IOException;
import javax.servlet.RequestDispatcher;
import java.io.PrintWriter;
import java.util.Enumeration;
import static java.lang.System.out;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.annotation.WebServlet;

@WebServlet(urlPatterns = {"/index"})
public class serv extends HttpServlet {
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    }
```

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    PrintWriter out = response.getWriter();
    String n1= request.getParameter("Loan Amount Required");
    String n2= request.getParameter("Duration");
    String n3= request.getParameter("ROI");
    String op  = request.getParameter("op");
    String n5= request.getParameter("Name");
    String n6 = request.getParameter("Mobile");
    String n7 =request.getParameter("email");
    int a = Integer.parseInt(n1);
    int b= Integer.parseInt(n2);
    int c = Integer.parseInt(n3);
    int d = 0;
    //emi = la*rate*(1+rate)months/((1+rate)months-1))
    switch(op){
       case "EMI":

          out.println((" \n User Name is:")+n5+'\n');
          out.println((" \n User Mobile No is:")+n6+'\n');
          out.println((" \n User Email is:")+n7+ '\n');

          out.println("The required EMI is:" + ((a*c*(1+c)^b)/((1+c)^b)));



    }
```

```
}


}
```


http://localhost:8084/WebApplication2/index.html
Search...
EMI
20BDS0126_Rahul kumar Sahani
Name: Rahul
Mobile : 987654321
Email Rahul@gmail.com
Loan Amount Required: 7564
Duration in months: 7
ROI: 9
EMI
Submit


http://localhost:8084/WebApplication2/serv?Name=Rahul+&Mobile=
Search...
EMI
localhost
User Name is:Rahul User Mobile No is:987654321 User Email is:Rahul@gmail.com The required EMI is:52366