Q7) Write a java Program to create a servlet to read information from client Registration page

Java Code:

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class DataSubmission extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
        // Get form data
        String name = req.getParameter("name");
        String email = req.getParameter("email");
        String phonenumber = req.getParameter("phonenumber");
        String lastname = req.getParameter("last");
        String gender = req.getParameter("gender");
        String pwd = req.getParameter("password");

        // Set response content type
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        System.out.println("Received Data:");
        System.out.println("Name: " + name);
        System.out.println("Gender: " + gender);
        System.out.println("Mail: " + email);
        System.out.println("Phone Number: " + phonenumber);
        System.out.println("Password:"+pwd);
        System.out.println("LastName:"+lastname);
        // Database
        String url = "jdbc:mysql://127.0.0.1:3306/servletdatabase";
        String user = "root";
        String password = "kcm@123";

        try {
            // Load JDBC driver
            Class.forName("com.mysql.cj.jdbc.Driver");

            // Connect to the database
            Connection con = DriverManager.getConnection(url, user, password);

            // SQL query
            String query = "INSERT INTO table1 (user, gender, mail,
phonenumber,password,lastname) VALUES (?, ?, ?, ?,?,?)";
```

```
            PreparedStatement pstmt = con.prepareStatement(query);
            pstmt.setString(1, name);
            pstmt.setString(2, gender);
            pstmt.setString(3, email);
            pstmt.setString(4, phonenumber);
            pstmt.setString(5, pwd);
            pstmt.setString(6,lastname);

            // Execute query
            int rowsInserted = pstmt.executeUpdate();

            // Close resources
            pstmt.close();
            con.close();

            // Response to the user
            if (rowsInserted > 0) {
                out.println("<h3>Data inserted successfully!</h3>");
                out.println("<p>Name: " + name + "</p>");
                out.println("<p>Gender: " + gender + "</p>");
                out.println("<p>Email: " + email + "</p>");
                out.println("<p>Phone Number: " + phonenumber + "</p>");
                out.println("<p>Password:"+pwd+"</p>");
                out.println("<p>LastName:"+lastname+"</p>");
            } else {
                out.println("<h3>Data insertion failed!</h3>");
            }
        } catch (Exception e) {
            out.println("<h3>Error:</h3> " + e.getMessage());
        } finally {
            out.close();
        }
    }
}
```

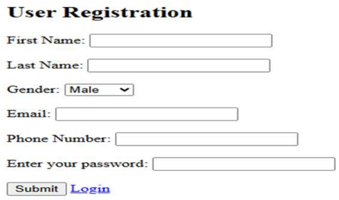
Index.hml

```
<!DOCTYPE html>
<html>
<head>
    <title>Register User</title>
</head>
<body>
    <h2>User Registration</h2>
    <form action="add" method="post">
        <label>First Name:</label>
        <input type="text" name="name" required><br><br>

        <label>Last Name:</label>
        <input type="text" name="last" required><br><br>
        <label>Gender:</label>
        <select name="gender" required>
          <option value="Male">Male</option>
          <option value="Female">Female</option>
        </select><br><br>

        <label>Email:</label>
        <input type="email" name="email" required><br><br>

        <label>Phone Number:</label>
        <input type="text" name="phonenumber" required><br><br>

        <label>Enter your password:</label>
        <input type="text" name="password" required><br><br>

        <input type="submit" value="Submit">
        <a href="login.html">Login</a>
    </form>
</body>
</html>
```

web.xml

```
<servlet>
            <servlet-name>ApplicationForm</servlet-name>
            <servlet-class>com.servletparas.DataSubmission</servlet-class>
      </servlet>
      <servlet-mapping>
            <servlet-name>ApplicationForm</servlet-name>
            <url-pattern>/add</url-pattern>
      </servlet-mapping>
<servlet>
```

Output:



User Registration

First Name:

Last Name:

Gender: Male

Email:

Phone Number:

Enter your password:

Submit Login

Q8) Write a program in hibernate with CFG and HBM file to perform a CRUD operation.

Java Code:

```
import org.hibernate.Session;
import org.hibernate.Transaction;

public class Main {
    public static void main(String[] args) {
        // Create an Employee
        Employee employee = new Employee("John Doe", 50000);
        saveEmployee(employee);

        // Read Employee
        Employee emp = getEmployee(employee.getId());
        System.out.println("Retrieved: " + emp);

        // Update Employee
        emp.setSalary(60000);
        updateEmployee(emp);
        System.out.println("Updated: " + getEmployee(emp.getId()));

        // Delete Employee
        deleteEmployee(emp.getId());
        System.out.println("Employee deleted.");
    }

    public static void saveEmployee(Employee employee) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            transaction = session.beginTransaction();
            session.save(employee);
            transaction.commit();
        }
    }

    public static Employee getEmployee(int id) {
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            return session.get(Employee.class, id);
        }
    }

    public static void updateEmployee(Employee employee) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            transaction = session.beginTransaction();
            session.update(employee);
            transaction.commit();
        }
    }

    public static void deleteEmployee(int id) {
        Transaction transaction = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            transaction = session.beginTransaction();
            Employee employee = session.get(Employee.class, id);
            if (employee != null) {
                session.delete(employee);
            }
            transaction.commit();
        }
    }
}
```

Getter and Setter for Employees:

```
public class Employee {
    private int id;
    private String name;
    private double salary;

    public Employee() {}

    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    // Getters and Setters
    public int getId() { return id; }
    public void setId(int id) { this.id = id; }

    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    public double getSalary() { return salary; }
    public void setSalary(double salary) { this.salary = salary; }

    @Override
    public String toString() {
        return "Employee [id=" + id + ", name=" + name + ", salary=" + salary + "]";
    }
}
```

Hibernate session factory:

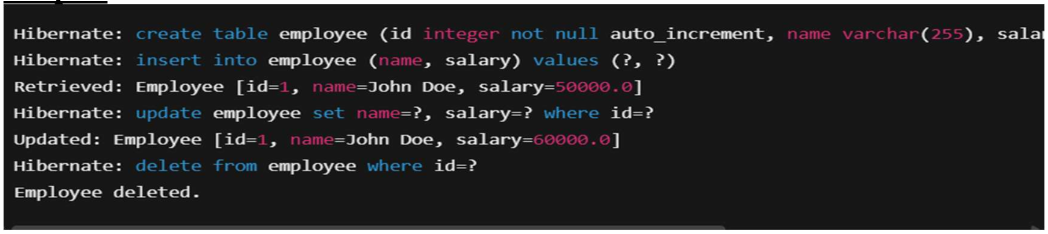
```
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
```

```
public class HibernateUtil {
    private static final SessionFactory sessionFactory;

    static {
        try {
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}
```

hibernate.cfg.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernatedb</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password">kcm@123</property>

        <property
name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="hibernate.hbm2ddl.auto">update</property>
        <property name="hibernate.show_sql">true</property>

        <mapping resource="Employee.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

Output:



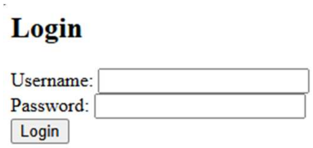
```
Hibernate: create table employee (id integer not null auto_increment, name varchar(255), sala
Hibernate: insert into employee (name, salary) values (?, ?)
Retrieved: Employee [id=1, name=John Doe, salary=50000.0]
Hibernate: update employee set name=?, salary=? where id=?
Updated: Employee [id=1, name=John Doe, salary=60000.0]
Hibernate: delete from employee where id=?
Employee deleted.
```

Q9) Develop a simple JSP program for user login form and authenticate user from data-base entries

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8" %>
<%@ page import="java.sql.*" %>
<%@ page import="javax.servlet.http.*, javax.servlet.*" %>

<%
    String msg = "";
    if (request.getMethod().equalsIgnoreCase("POST")) {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // Database connection
        Connection con = null;
        PreparedStatement pst = null;
        ResultSet rs = null;

        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/userdb", "root",
"password");

            pst = con.prepareStatement("SELECT * FROM users WHERE username=?
AND password=?");
            pst.setString(1, username);
            pst.setString(2, password);
            rs = pst.executeQuery();

            if (rs.next()) {
                session.setAttribute("username", username);
                response.sendRedirect("welcome.jsp"); // Redirect to welcome page on
success
            } else {
                msg = "Invalid Username or Password!";
            }
        } catch (Exception e) {
            e.printStackTrace();
            msg = "Database Connection Error!";
        } finally {
            try { if (rs != null) rs.close(); } catch (Exception e) {}
            try { if (pst != null) pst.close(); } catch (Exception e) {}
            try { if (con != null) con.close(); } catch (Exception e) {}
        }
    }
%>

<!DOCTYPE html>
```

```
<html>
<head>
  <title>Login Page</title>
</head>
<body>
  <h2>Login</h2>
  <form method="post">
    <label>Username:</label>
    <input type="text" name="username" required><br>
    <label>Password:</label>
    <input type="password" name="password" required><br>
    <input type="submit" value="Login">
  </form>
  <p style="color:red;"><%= msg %></p>
</body>
</html>
```

<u>Output:</u>


Login
Username:
Password:
Login