13. Develop a JAX-RS service that provides CRUD database operation interface for a phonebook. Call appropriate service from servlet/jsp pages to perform data oriented operations.

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Phonebook Management</title>
//Don’t write style tag in your journal this is for design only
    <style>
        body {
            font-family: Arial, sans-serif;
            text-align: center;
            margin: 50px;
        }
        h1 {
            color: #333;
        }
        .menu {
            margin-top: 20px;
        }
        .menu a {
            display: block;
            margin: 10px;
            padding: 10px;
            text-decoration: none;
            color: white;
            background-color: #007BFF;
            border-radius: 5px;
            width: 200px;
            margin-left: auto;
            margin-right: auto;
        }
        .menu a:hover {
            background-color: #0056b3;
        }
    </style>
</head>
<body>
    <h1>Phonebook Management System</h1>
    <div class="menu">
        <a href="addContact.jsp">Add Contact</a>
        <a href="updateContact.jsp">Update Contact</a>
        <a href="deleteContact.jsp">Delete Contact</a>
        <a href="http://localhost:8080/JerseyPhonebookCrud/ws/phonebook/list">View Contacts</a>
    </div>
</body>
</html>
```

addContact.jsp

```
<form action="/JerseyPhonebookCrud/ws/phonebook/add" method="post">
    Name: <input type="text" name="name" required>
    Phone: <input type="text" name="phone" required>
    <button type="submit">Add Contact</button>
</form>
```

updateContact.jsp

```
<form action="/JerseyPhonebookCrud/ws/phonebook/update" method="post">
   ID: <input type="number" name="id" required>
   Name: <input type="text" name="name" required>
   Phone: <input type="text" name="phone" required>
   <button type="submit">Update Contact</button>
</form>
```

deleteContact.jsp

```
<form action="/JerseyPhonebookCrud/ws/phonebook/delete" method="post">
   ID: <input type="number" name="id" required>
   <button type="submit">Delete Contact</button>
</form>
```

DatabaseConnection.java

```
package in.ga.config;


import java.sql.Connection;
import java.sql.DriverManager;
public class DatabaseConnection {
   private static final String URL = "jdbc:mysql://localhost:3306/phonebookdb";
   private static final String USER = "root";
   private static final String PASSWORD = "Gaurav@123";


   public static Connection getConnection(){
      try{
         Class.forName("com.mysql.cj.jdbc.Driver");
         Connection con = DriverManager.getConnection(URL,USER,PASSWORD);
         return con;
      }catch(Exception e){
            e.printStackTrace();
        }
        return null;
    }
}
```

PhonebookService.java

```
package in.ga.service;


import in.ga.config.DatabaseConnection;
import jakarta.ws.rs.*;
import jakarta.ws.rs.core.MediaType;
import java.sql.*;


@Path("/phonebook")
public class PhonebookService {


    @POST
    @Path("/add")
    @Produces(MediaType.TEXT_PLAIN)
    public String addContact(@FormParam("name") String name, @FormParam("phone") String phone) {
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "INSERT INTO phonebook (name, phone) VALUES (?, ?)";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, name);
            stmt.setString(2, phone);
            int rows = stmt.executeUpdate();
            return rows > 0 ? "Contact Added Successfully" : "Failed to Add Contact";
        } catch (SQLException e) {
            return "Error: " + e.getMessage();
        }
    }

    @POST
    @Path("/update")
    @Produces(MediaType.TEXT_PLAIN)
    public String updateContact(@FormParam("id") int id, @FormParam("name") String name, @FormParam("phone") String phone) {
        try (Connection conn = DatabaseConnection.getConnection()) {
            String sql = "UPDATE phonebook SET name=?, phone=? WHERE id=?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, name);
            stmt.setString(2, phone);
            stmt.setInt(3, id);
            int rows = stmt.executeUpdate();
            return rows > 0 ? "Contact Updated Successfully" : "Failed to Update Contact";
        } catch (SQLException e) {
            return "Error: " + e.getMessage();
        }
    }

    @POST
    @Path("/delete")
    @Produces(MediaType.TEXT_PLAIN)
    public String deleteContact(@FormParam("id") int id) {
        try (Connection conn = DatabaseConnection.getConnection()) {
```

```
        String sql = "DELETE FROM phonebook WHERE id=?";
        PreparedStatement stmt = conn.prepareStatement(sql);
        stmt.setInt(1, id);
        int rows = stmt.executeUpdate();
        return rows > 0 ? "Contact Deleted Successfully" : "Failed to Delete Contact";
    } catch (SQLException e) {
        return "Error: " + e.getMessage();
    }
}

@GET
@Path("/list")
@Produces(MediaType.TEXT_HTML)
public String listContacts() {
    StringBuilder html = new StringBuilder("<h2>Phonebook Contacts</h2><ul>");
    try (Connection conn = DatabaseConnection.getConnection();
         Statement stmt = conn.createStatement();
         ResultSet rs = stmt.executeQuery("SELECT * FROM phonebook")) {

        while (rs.next()) {
            html.append("<li>").append(rs.getInt("id"))
                .append(": ").append(rs.getString("name"))
                .append(" - ").append(rs.getString("phone"))
                .append("</li>");
        }
    } catch (SQLException e) {
        return "Error: " + e.getMessage();
    }
```

```
        html.append("</ul>");

        return html.toString();

    }

}
```