

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>Murach's Java Servlets and JSP</title>
6     <link rel="stylesheet" href="css/main.css" type
    ="text/css"/>
7 </head>
8 <body>
9     <h1>Join our email list</h1>
10
11     <p>To join our email list, enter your name and
12         email address below.</p>
13
14     <form action="emailList" method="post">
15         <input type="hidden" name="action" value="
    add">
16         <label class="pad_top">Email:</label>
17         <input type="email" name="email" required><
    br>
18         <label class="pad_top">First Name:</label>
19         <input type="text" name="firstName"
    required><br>
20         <label class="pad_top">Last Name:</label>
21         <input type="text" name="lastName" required
    ><br>
22         <label>&nbsp;</label>
23         <input type="submit" value="Join Now" class
    ="margin_left">
24     </form>
25
26 </body>
27 </html>
```

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <title>Murach's Java Servlets and JSP</title>
6     <link rel="stylesheet" href="css/main.css" type
    ="text/css"/>
7 </head>
8
9 <body>
10     <h1>Thanks for joining our email list</h1>
11
12     <p>Here is the information that you entered:</p>
13
14     <label>Email:</label>
15     <span>${user.email}</span><br>
16     <label>First Name:</label>
17     <span>${user.firstName}</span><br>
18     <label>Last Name:</label>
19     <span>${user.lastName}</span><br>
20
21     <p>To enter another email address, click on the
    Back
22     button in your browser or the Return button
    shown
23     below.</p>
24
25     <form action="emailList" method="post">
26         <input type="hidden" name="action" value="
    returnOrManage">
27         <input type="submit" name="return" value="
    Return">
```

```
28         <input type="submit" name="manage" value="
    Manage joining list">
29     </form>
30
31 </body>
32 </html>
```

```
1 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/
  jstl/core" %>
2 <%--
3   Created by IntelliJ IDEA.
4   User: mhm85
5   Date: 11/4/2019
6   Time: 2:36 PM
7   To change this template use File | Settings |
  File Templates.
8 --%>
9 <%@ page contentType="text/html; charset=UTF-8"
  language="java" %>
10 <html>
11 <head>
12     <title>Manage joining list</title>
13     <link rel="stylesheet" href="css/management.css
  ">
14 </head>
15 <body>
16 <div class="modify-div">
17     <table border="1">
18         <tr>
19             <th>Userid</th>
20             <th>Firstname</th>
21             <th>Lastname</th>
22             <th>Email</th>
23         </tr>
24         <c:forEach var="user" items="${users}">
25             <tr>
26                 <td>${user.id}</td>
27                 <td>${user.firstName}</td>
28                 <td>${user.lastName}</td>
29                 <td>${user.email}</td>
```

```
30         </tr>
31     </c:forEach>
32 </table>
33 <p>Modify data</p>
34 <form action="emailList" method="post">
35     <input type="hidden" name="action" value="
modify">
36     <table border="1">
37         <tr>
38             <th>Firstname</th>
39             <th>Lastname</th>
40             <th>Email</th>
41         </tr>
42         <tr>
43             <td>
44                 <input type="text" name="
firstName">
45             </td>
46             <td>
47                 <input type="text" name="
lastName">
48             </td>
49             <td>
50                 <input type="text" name="email"
>
51             </td>
52         </tr>
53         <tr>
54             <td colspan="3" align="center">
55                 <input type="submit" name="
update" value="Update">
56                 <input type="submit" name="
delete" value="Delete">
```

```
57                 </td>
58             </tr>
59         </table>
60     </form>
61 </div>
62 </body>
63 </html>
64
```

```
1 body {
2     font-family: Arial, Helvetica, sans-serif;
3     font-size: 85%;
4     margin-left: 2em;
5     margin-right: 2em;
6     width: 400px;
7 }
8
9 h1 {
10     font-size: 140%;
11     color: teal;
12     margin-bottom: .5em;
13 }
14
15 h2 {
16     font-size: 120%;
17     color: teal;
18     margin-bottom: .5em;
19 }
20
21 label {
22     float: left;
23     width: 7em;
24     margin-bottom: 0.5em;
25     font-weight: bold;
26 }
27
28 input[type="text"], input[type="email"] { /* An
    attribute selector */
29     width: 15em;
30     margin-left: 0.5em;
31     margin-bottom: 0.5em;
32 }
```

```
33
34 span {
35     margin-left: 0.5em;
36     margin-bottom: 0.5em;
37 }
38
39 br {
40     clear: both;
41 }
42
43 .pad_top {
44     padding-top: 0.25em;
45 }
46
47 .margin_left {
48     margin-left: 0.5em;
49 }
```



```
1 .modify-div {
2     width: 75%;
3     margin: auto;
4 }
5
6 table {
7     width: 100%;
8 }
9
10 p {
11     margin: 0;
12 }
13
14 input[type="text"] {
15     width: 100%;
16 }
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
3           xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
4           xsi:schemaLocation="http://xmlns.jcp.org/
xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/
web-app_4_0.xsd"
5           version="4.0">
6
7     <servlet>
8       <servlet-name>MyServlet</servlet-name>
9       <servlet-class>MyServlet</servlet-class>
10    </servlet>
11
12    <servlet-mapping>
13      <servlet-name>MyServlet</servlet-name>
14      <url-pattern>/emailList</url-pattern>
15    </servlet-mapping>
16
17 </web-app>
```

```
1 import java.io.Serializable;
2
3 // An example of JavaBean that implements an
  interface.
4 public class User implements Serializable {
5
6     private int id;
7     private String firstName;
8     private String lastName;
9     private String email;
10
11     public User(int id, String firstName, String
  lastName, String email) {
12         this.id = id;
13         this.firstName = firstName;
14         this.lastName = lastName;
15         this.email = email;
16     }
17
18     public User() {
19         id = 0;
20         firstName = "";
21         lastName = "";
22         email = "";
23     }
24
25     public int getId() {
26         return id;
27     }
28
29     public void setId(int id) {
30         this.id = id;
31     }
```

```
32
33     public String getFirstName() {
34         return firstName;
35     }
36
37     public void setFirstName(String firstName) {
38         this.firstName = firstName;
39     }
40
41     public String getLastName() {
42         return lastName;
43     }
44
45     public void setLastName(String lastName) {
46         this.lastName = lastName;
47     }
48
49     public String getEmail() {
50         return email;
51     }
52
53     public void setEmail(String email) {
54         this.email = email;
55     }
56 }
57
```

```
1 import java.io.IOException;
2 import java.io.InputStream;
3 import java.sql.*;
4 import java.util.ArrayList;
5 import java.util.Properties;
6
7 public class DatabaseTools {
8     public static Connection getConnected() {
9         Connection conn = null;
10
11         try {
12             //Load MySQL database driver
13             Class.forName("com.mysql.jdbc.Driver");
14
15             // Connection string
16             String dbURL = getConnectionString();
17
18             // Credentials
19             String username = "root";
20             String password = "mysql";
21
22             // Set connection to database to
23             Connection object
24             conn = DriverManager.getConnection(
25             dbURL, username, password);
26         } catch (ClassNotFoundException e) {
27             e.printStackTrace();
28         } catch (SQLException e) {
29             e.printStackTrace();
30         }
31
32         return conn;
33     }
34 }
```

```
32
33     public static void insert(User user) {
34         // Connection to database;
35         Connection myConn = DatabaseTools.
getConnection();
36
37         PreparedStatement ps = null;
38
39         // Insert query
40         String queryInsert = "INSERT INTO userdata
(firstname, lastname, email) VALUES (?, ?, ?)";
41
42         try {
43             // Set PreparedStatement object to
instance with query
44             ps = myConn.prepareStatement(
queryInsert);
45
46             // Set values for parameters
47             ps.setString(1, user.getFirstName());
48             ps.setString(2, user.getLastName());
49             ps.setString(3, user.getEmail());
50
51             // Execute insert on database
52             ps.execute();
53         } catch (SQLException e) {
54             e.printStackTrace();
55         } finally {
56             DatabaseTools.closePreparedStatement(ps
);
57             DatabaseTools.closeConnection(myConn);
58         }
59     }
```

```
60
61     public static ArrayList<User> selectAllUsers()
62     {
63         // Connection to database
64         Connection conn = DatabaseTools.
65         getConnected();
66
67         PreparedStatement ps = null;
68         ResultSet rs = null;
69         ArrayList<User> myUserList = new ArrayList<
70         >();
71
72         // Select statement
73         String selectQuery = "SELECT userid,
74         firstname, lastname, email FROM userdata;";
75
76         try {
77             // Create prepared statement object
78             ps = conn.prepareStatement(selectQuery)
79             ;
80
81             // Execute query and return result set
82             rs = ps.executeQuery();
83
84             // Loop through all rows in result set
85             while (rs.next()) {
86                 // Create user object to add to
87                 list.
88
89                 User user = new User();
90
91                 // Add values for row of result set
92                 to properties of User object
93                 user.setId(rs.getInt("userid"));
```

```
86         user.setFirstName(rs.getString("
    firstname"));
87         user.setLastName(rs.getString("
    lastname"));
88         user.setEmail(rs.getString("email"
    ));
89
90         // Add user to list
91         myUserList.add(user);
92     }
93     } catch (SQLException e) {
94         e.printStackTrace();
95     } finally {
96         DatabaseTools.closePreparedStatement(
    ps);
97         DatabaseTools.closeResultSet(rs);
98         DatabaseTools.closeConnection(conn);
99     }
100
101     return myUserList;
102 }
103
104 public static void update(User user) {
105     // Connection to database;
106     Connection myConn = DatabaseTools.
    getConnected();
107
108     PreparedStatement ps = null;
109
110     // Update query
111     String queryUpdate = "UPDATE userdata SET
    firstname = ?, lastname = ?, email = ? WHERE
    firstname = ? AND lastname = ?;";
```



```
112
113         try {
114             // Set PreparedStatement object to
instance with query
115             ps = myConn.prepareStatement(
queryUpdate);
116
117             // Set values for parameters
118             ps.setString(1, user.getFirstName());
119             ps.setString(2, user.getLastName());
120             ps.setString(3, user.getEmail());
121             ps.setString(4, user.getFirstName());
122             ps.setString(5, user.getLastName());
123
124             // Execute update on database
125             ps.executeUpdate();
126         } catch (SQLException e) {
127             e.printStackTrace();
128         } finally {
129             DatabaseTools.closePreparedStatement(
ps);
130             DatabaseTools.closeConnection(myConn);
131         }
132     }
133
134     public static void delete(String firstName,
String lastName) {
135         // Connection to database;
136         Connection myConn = DatabaseTools.
getConnection();
137
138         PreparedStatement ps = null;
139
```

```
140         // Update query
141         String queryUpdate = "DELETE FROM userdata
    WHERE firstname = ? AND lastname = ?;";
142
143         try {
144             // Set PreparedStatement object to
    instance with query
145             ps = myConn.prepareStatement(
    queryUpdate);
146
147             // Set values for parameters
148             ps.setString(1, firstName);
149             ps.setString(2, lastName);
150
151             // Execute update on database
152             ps.executeUpdate();
153         } catch (SQLException e) {
154             e.printStackTrace();
155         } finally {
156             DatabaseTools.closePreparedStatement(
    ps);
157             DatabaseTools.closeConnection(myConn);
158         }
159     }
160
161     public static void closePreparedStatement(
    Statement ps) {
162         try {
163             if (ps != null) {
164                 ps.close();
165             }
166         } catch (SQLException e) {
167             e.printStackTrace();
```

```
168         }
169     }
170
171     public static void closeConnection(Connection
conn) {
172         try {
173             if (conn != null) {
174                 conn.close();
175             }
176         } catch (SQLException e) {
177             e.printStackTrace();
178         }
179     }
180
181     public static void closeResultSet(ResultSet rs
) {
182         try {
183             if (rs != null) {
184                 rs.close();
185             }
186         } catch (SQLException e) {
187             e.printStackTrace();
188         }
189     }
190
191     public static String getConnectionString() {
192         Properties properties = new Properties();
193         InputStream input = Thread.currentThread()
.getContextClassLoader().getResourceAsStream("
settings.properties");
194         try {
195             properties.load(input);
196         } catch (IOException e) {
```

```
197             e.printStackTrace();
198         }
199
200         return properties.getProperty("dbURL").
    toString();
201     }
202 }
203
```

```
1  import javax.servlet.ServletException;
2  import javax.servlet.annotation.WebServlet;
3  import javax.servlet.http.HttpServlet;
4  import javax.servlet.http.HttpServletRequest;
5  import javax.servlet.http.HttpServletResponse;
6  import java.io.DataInput;
7  import java.io.IOException;
8  import java.util.ArrayList;
9
10 @WebServlet(name = "MyServlet")
11 public class MyServlet extends HttpServlet {
12     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
13     ServletException, IOException {
14         String action = request.getParameter("
15         action");
16         String url = "/index.jsp";
17
18         if (action == null) {
19             action = "home";
20         }
21
22         if (action.equals("home")) {
23             url = "/index.jsp";
24         }
25         else if (action.equals("add")) {
26             String firstName = request.getParameter
27             ("firstName");
28             String lastName = request.getParameter(
29             "lastName");
30             String email = request.getParameter("
31             email");
32         }
33     }
34 }
```

```
28         User user = new User(0, firstName,
    lastName, email);
29
30         DatabaseTools.insert(user);
31
32         request.setAttribute("user", user);
33
34         url = "/thanks.jsp";
35     }
36     else if (action.equals("returnOrManage")) {
37         if (request.getParameter("return") !=
    null) {
38             url = "/index.jsp";
39         }
40         else if (request.getParameter("manage")
    != null) {
41             ArrayList<User> users =
    DatabaseTools.selectAllUsers();
42             request.setAttribute("users", users
    );
43             url = "/modify.jsp";
44         }
45     }
46     else if (action.equals("modify")) {
47         if (request.getParameter("update") !=
    null) {
48             String firstName = request.
    getParameter("firstName");
49             String lastName = request.
    getParameter("lastName");
50             String email = request.getParameter
    ("email");
51
```

```
52         User user = new User(0, firstName,
    lastName, email);
53
54         DatabaseTools.update(user);
55
56         ArrayList<User> users =
    DatabaseTools.selectAllUsers();
57
58         request.setAttribute("users", users
    );
59
60         url = "/modify.jsp";
61     }
62     else if (request.getParameter("delete")
    != null) {
63         String firstName = request.
    getParameter("firstName");
64         String lastName = request.
    getParameter("lastName");
65
66         DatabaseTools.delete(firstName,
    lastName);
67
68         ArrayList<User> users =
    DatabaseTools.selectAllUsers();
69
70         request.setAttribute("users", users
    );
71
72         url = "/modify.jsp";
73     }
74 }
75
```

```
76         getServletContext().getRequestDispatcher(  
            url).forward(request, response);  
77     }  
78  
79     protected void doGet(HttpServletRequest request  
        request, HttpServletResponse response) throws  
        ServletException, IOException {  
80         doPost(request, response);  
81     }  
82 }  
83
```



```
1 dbURL=jdbc:mysql://localhost:3306/registered_users
```

test2-maxwellmartin [C:\Users\maxma\Desktop\college-pcfd\fall-2019\CIS

File Edit View Navigate Code Analyze Refactor Build Run Tools

test2-maxwellmartin > src > settings.properties

Project

- test2-maxwellmartin C:\Users\maxma\Desktop\college-pcfd\fa 1
 - .idea
 - out
 - print-out
 - src
 - DatabaseTools
 - MyServlet
 - settings.properties
 - User
 - web
 - css
 - main.css
 - management.css
 - WEB-INF
 - web.xml
 - index.jsp
 - modify.jsp
 - thanks.jsp
 - test2-maxwellmartin.iml
- External Libraries
 - < 12 > C:\Program Files\Java\jdk-12.0.2
 - jstl:jstl:1.2
 - mysql:mysql-connector-java:5.1.48
 - Tomcat 9.0.24
- Scratches and Consoles

Learn

1: Project

2: Structure

3: Favorites

Web