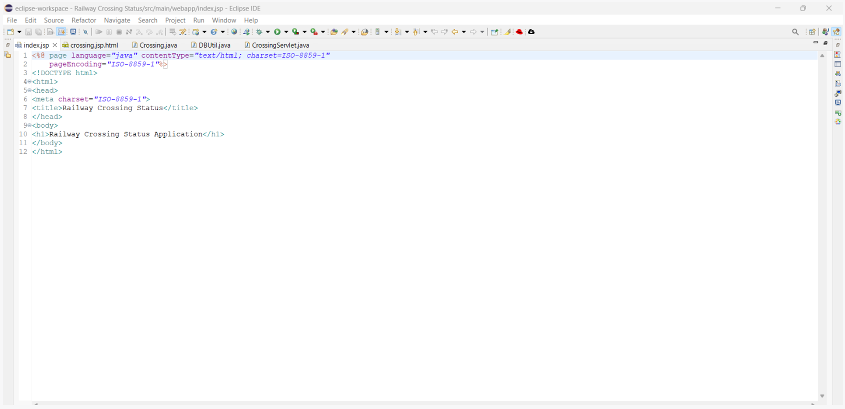
# RAILWAY CROSSING STATUS:

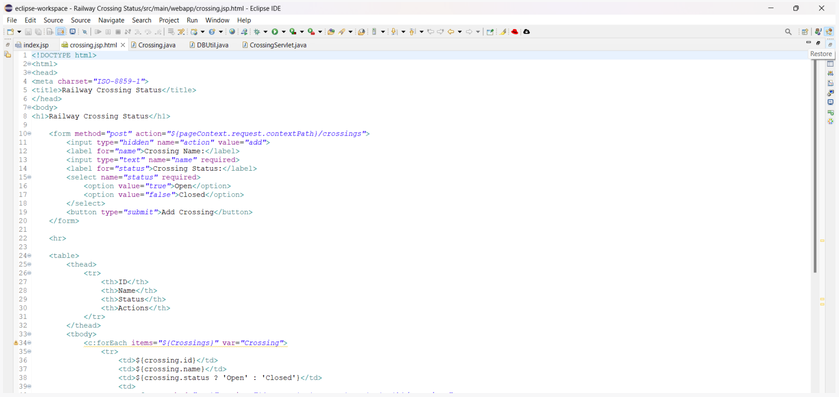
uses JSP for the frontend, Servlet for the backend, and JDBC for performing CRUD operations on a MySQL database.

`Crossing.java` (Model class):
`DBUtil.java` (Database utility class using JDBC):}
Crossing Servlet.java (Servlet for handling CRUD operations):
`crossing.jsp` (JSP view for displaying the railway crossing information):

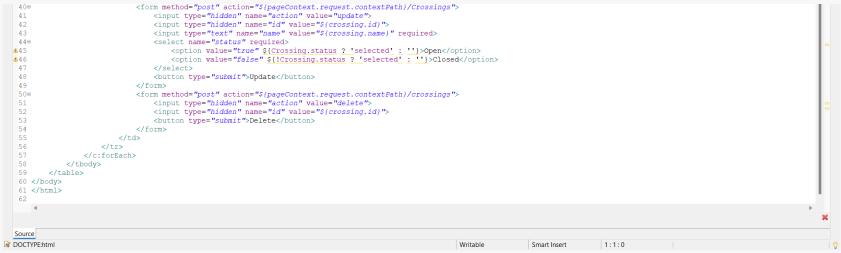
`web.xml` (Deployment descriptor for configuring the servlet


eclipse-workspace - Railway Crossing Status/src/main/webapp/index.jsp - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
index.jsp
crossing.jsp.html
Crossing.java
DBUtil.java
CrossingServlet.java

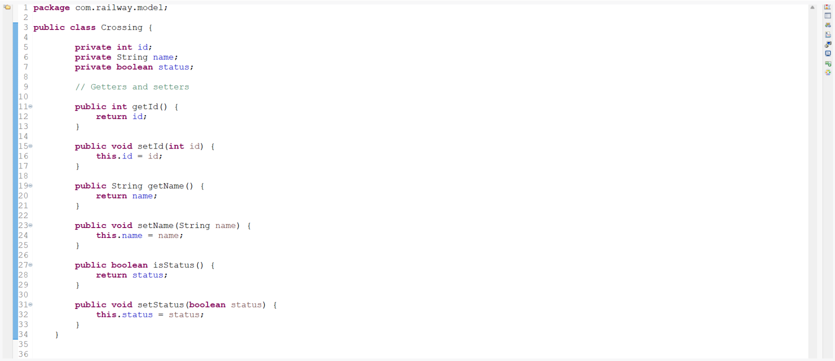
```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Railway Crossing Status</title>
</head>
<body>
<h1>Railway Crossing Status Application</h1>
</body>
</html>
```


eclipse-workspace - Railway Crossing Status/src/main/webapp/crossing.jsp.html - Eclipse IDE
File Edit Source Source Navigate Search Project Run Window Help
index.jsp
crossing.jsp.html
Crossing.java
DBUtil.java
CrossingServlet.java
Restore

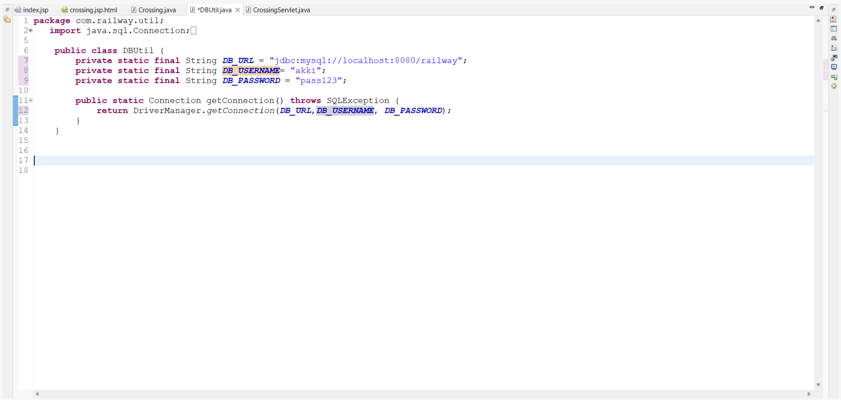
```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Railway Crossing Status</title>
</head>
<body>
<h1>Railway Crossing Status</h1>

    <form method="post" action="${pageContext.request.contextPath}/crossings">
        <input type="hidden" name="action" value="add">
        <label for="name">Crossing Name:</label>
        <input type="text" name="name" required>
        <label for="status">Crossing Status:</label>
        <select name="status" required>
            <option value="true">Open</option>
            <option value="false">Closed</option>
        </select>
        <button type="submit">Add Crossing</button>
    </form>

    <hr>

    <table>
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Status</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${Crossings}" var="Crossing">
                <tr>
                    <td>${crossing.id}</td>
                    <td>${crossing.name}</td>
                    <td>${crossing.status ? 'Open' : 'Closed'}</td>
                    <td>
```



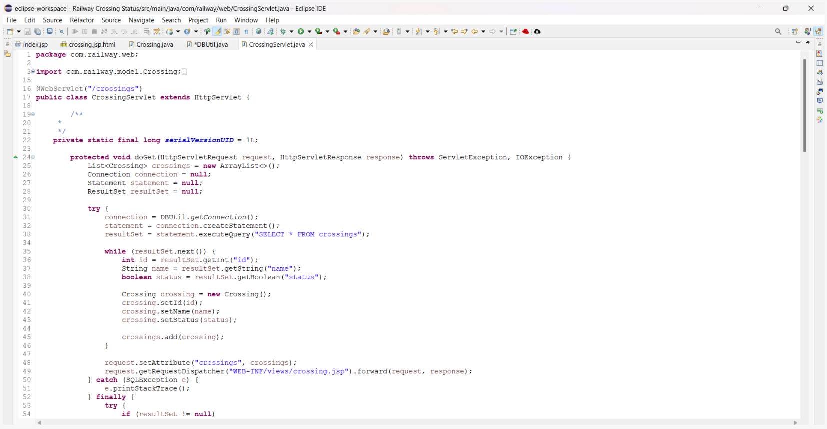
```
                        <form method="post" action="${pageContext.request.contextPath}/Crossings">
                            <input type="hidden" name="action" value="update">
                            <input type="hidden" name="id" value="${crossing.id}">
                            <input type="text" name="name" value="${crossing.name}" required>
                            <select name="status" required>
                                <option value="true" ${Crossing.status ? 'selected' : ''}>Open</option>
                                <option value="false" ${!Crossing.status ? 'selected' : ''}>Closed</option>
                            </select>
                            <button type="submit">Update</button>
                        </form>
                        <form method="post" action="${pageContext.request.contextPath}/crossings">
                            <input type="hidden" name="action" value="delete">
                            <input type="hidden" name="id" value="${crossing.id}">
                            <button type="submit">Delete</button>
                        </form>
                    </td>
                </tr>
            </c:forEach>
        </tbody>
    </table>
</body>
</html>
```



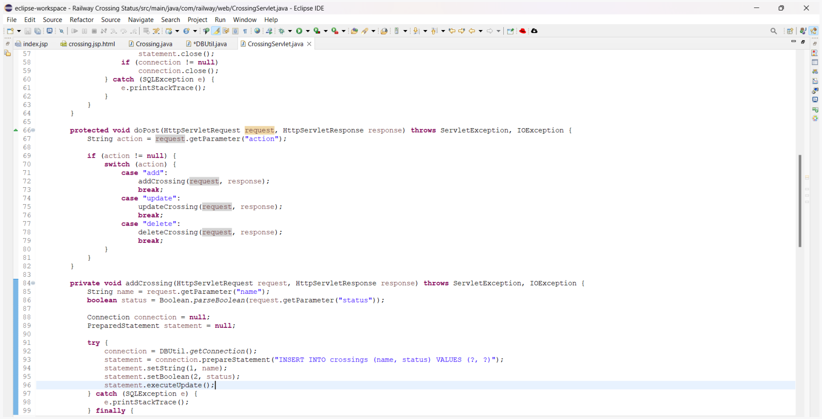
```
package com.railway.model;

public class Crossing {

        private int id;
        private String name;
        private boolean status;

        // 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 boolean isStatus() {
            return status;
        }

        public void setStatus(boolean status) {
            this.status = status;
        }
    }


```



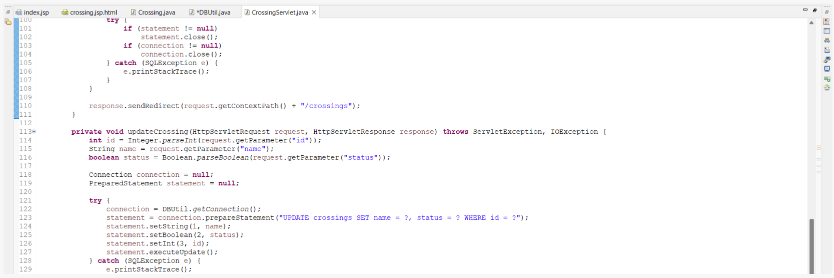
```
package com.railway.util;
   import java.sql.Connection;

    public class DBUtil {
        private static final String DB_URL = "jdbc:mysql://localhost:8080/railway";
        private static final String DB_USERNAME= "akki";
        private static final String DB_PASSWORD = "pass123";

        public static Connection getConnection() throws SQLException {
            return DriverManager.getConnection(DB_URL,DB_USERNAME, DB_PASSWORD);
        }
    }




```



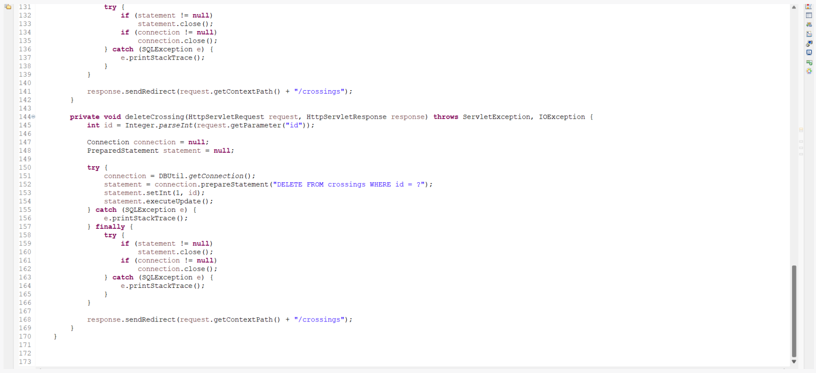
```
package com.railway.web;

import com.railway.model.Crossing;

@WebServlet("/crossings")
public class CrossingServlet extends HttpServlet {

        /**
     *
     */
    private static final long serialVersionUID = 1L;

        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            List<Crossing> crossings = new ArrayList<>();
            Connection connection = null;
            Statement statement = null;
            ResultSet resultSet = null;

            try {
                connection = DBUtil.getConnection();
                statement = connection.createStatement();
                resultSet = statement.executeQuery("SELECT * FROM crossings");

                while (resultSet.next()) {
                    int id = resultSet.getInt("id");
                    String name = resultSet.getString("name");
                    boolean status = resultSet.getBoolean("status");

                    Crossing crossing = new Crossing();
                    crossing.setId(id);
                    crossing.setName(name);
                    crossing.setStatus(status);

                    crossings.add(crossing);
                }

                request.setAttribute("crossings", crossings);
                request.getRequestDispatcher("WEB-INF/views/crossing.jsp").forward(request, response);
            } catch (SQLException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (resultSet != null)
```



```
                        statement.close();
                    if (connection != null)
                        connection.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }
        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            String action = request.getParameter("action");

            if (action != null) {
                switch (action) {
                    case "add":
                        addCrossing(request, response);
                        break;
                    case "update":
                        updateCrossing(request, response);
                        break;
                    case "delete":
                        deleteCrossing(request, response);
                        break;
                }
            }
        }

        private void addCrossing(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            String name = request.getParameter("name");
            boolean status = Boolean.parseBoolean(request.getParameter("status"));

            Connection connection = null;
            PreparedStatement statement = null;

            try {
                connection = DBUtil.getConnection();
                statement = connection.prepareStatement("INSERT INTO crossings (name, status) VALUES (?, ?)");
                statement.setString(1, name);
                statement.setBoolean(2, status);
                statement.executeUpdate();
            } catch (SQLException e) {
                e.printStackTrace();
            } finally {
```

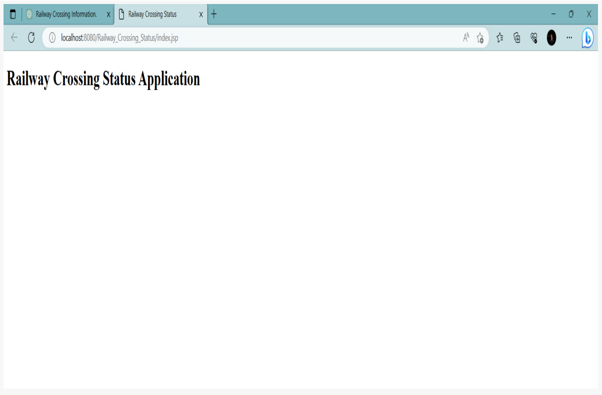


```
                try {
                    if (statement != null)
                        statement.close();
                    if (connection != null)
                        connection.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }

            response.sendRedirect(request.getContextPath() + "/crossings");
        }

        private void updateCrossing(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            int id = Integer.parseInt(request.getParameter("id"));
            String name = request.getParameter("name");
            boolean status = Boolean.parseBoolean(request.getParameter("status"));

            Connection connection = null;
            PreparedStatement statement = null;

            try {
                connection = DBUtil.getConnection();
                statement = connection.prepareStatement("UPDATE crossings SET name = ?, status = ? WHERE id = ?");
                statement.setString(1, name);
                statement.setBoolean(2, status);
                statement.setInt(3, id);
                statement.executeUpdate();
            } catch (SQLException e) {
                e.printStackTrace();
```

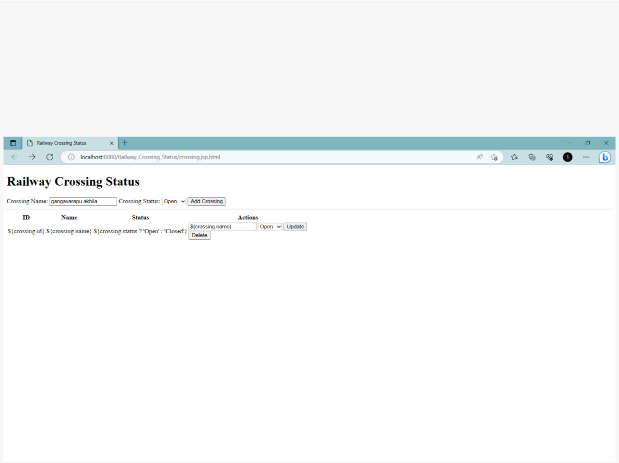


```
                try {
                    if (statement != null)
                        statement.close();
                    if (connection != null)
                        connection.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }

            response.sendRedirect(request.getContextPath() + "/crossings");
        }

        private void deleteCrossing(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            int id = Integer.parseInt(request.getParameter("id"));

            Connection connection = null;
            PreparedStatement statement = null;

            try {
                connection = DBUtil.getConnection();
                statement = connection.prepareStatement("DELETE FROM crossings WHERE id = ?");
                statement.setInt(1, id);
                statement.executeUpdate();
            } catch (SQLException e) {
                e.printStackTrace();
            } finally {
                try {
                    if (statement != null)
                        statement.close();
                    if (connection != null)
                        connection.close();
                } catch (SQLException e) {
                    e.printStackTrace();
                }
            }

            response.sendRedirect(request.getContextPath() + "/crossings");
        }
    }
```

# OUTPUT:


Railway Crossing Information.
Railway Crossing Status
localhost:8080/Railway_Crossing_Status/index.jsp

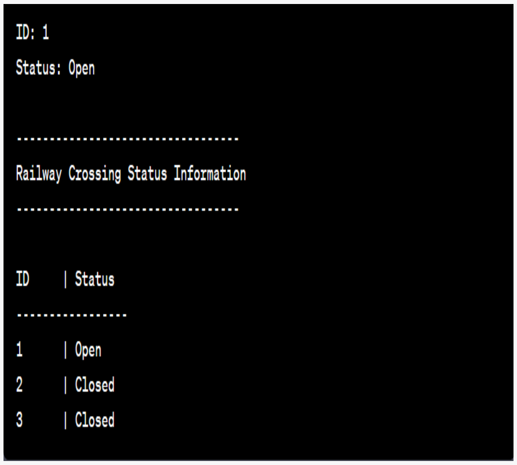
Railway Crossing Status Application



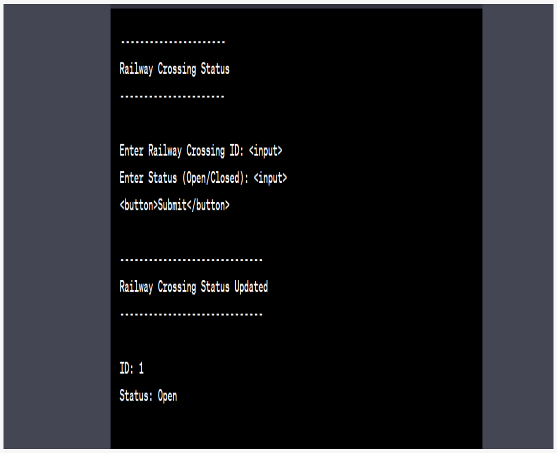
Railway Crossing Status

Crossing Name: gangavarapu akhila Crossing Status: Open Add Crossing

| ID | Name | Status | Actions |
|---|---|---|---|
| ${crossing.id} | ${crossing.name} | ${crossing.status ? 'Open' : 'Closed'} | ${crossing.name} Open Update<br>Delete |



```
ID: 1
Status: Open

-----------------------------------
Railway Crossing Status Information
-----------------------------------

ID    | Status
-----------------
1     | Open
2     | Closed
3     | Closed
```



```
----------------------
Railway Crossing Status
----------------------

Enter Railway Crossing ID: <input>
Enter Status (Open/Closed): <input>
<button>Submit</button>

------------------------------
Railway Crossing Status Updated
------------------------------

ID: 1
Status: Open
```