1 Click on the Edit link, Control will have to go to EditServlet and corresponding row data has to display like this

2 After changing the information, submit button. You will see that information is changed. It will go to View Servlet

3 Now, click on the delete link to delete the record. That particular row has to delete

DBCONNECTION.JAVA

```
import java.sql.Connection;
import java.sql.DriverManager;

public class DBConnection {

	private static Connection con=null;

	public DBConnection() {
		// TODO Auto-generated constructor stub
	}

	public static Connection getConnection() {

		try {
			Class.forName("com.mysql.cj.jdbc.Driver");

	con=DriverManager.getConnection("jdbc:mysql://localhost:3306/CD7","ro
ot","spectratec@22");
		}
		catch(Exception e) {
			e.printStackTrace();
		}
		return con;
	}
}
```

EDITSERVLET.JAVA

```
import jakarta.servlet.ServletException;
import jakarta.servlet.http.*;

import java.io.IOException;

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

        if (action != null && action.equals("edit")) {
            // Handle edit action
            int employeeId = Integer.parseInt(request.getParameter("id"));

            // Fetch employee data by ID from DAO
            EmployeeDao employeeDao = new EmployeeDao();
            Employee employee = employeeDao.getEmployeeById(employeeId);
```

```
            // Forward employee data to edit form
            request.setAttribute("employee", employee);

request.getRequestDispatcher("/editEmployee.jsp").forward(request,
response);
        } else if (action != null && action.equals("save")) {
            // Handle save action
            String updatedName = request.getParameter("txtName");
            String updatedPassword = request.getParameter("txtPassword");
            String updatedEmail = request.getParameter("txtEmail");
            String updatedCountry = request.getParameter("country");
            int employeeId = Integer.parseInt(request.getParameter("id"));

            // Create an updated Employee object
            Employee updatedEmployee = new Employee();
            updatedEmployee.setId(employeeId);
            updatedEmployee.setName(updatedName);
            updatedEmployee.setPassword(updatedPassword);
            updatedEmployee.setEmail(updatedEmail);
            updatedEmployee.setCountry(updatedCountry);

            // Update employee in database via DAO
            EmployeeDao employeeDao = new EmployeeDao();
            int updateCount = employeeDao.updateEmployee(updatedEmployee);

            if (updateCount > 0) {
                // Employee updated successfully, redirect to ViewServlet
                response.sendRedirect(request.getContextPath() +
"/ViewServlet");
            } else {
                // Handle update failure
                response.getWriter().println("Failed to update employee");
            }
        } else if (action != null && action.equals("delete")) {
            // Handle delete action
            int employeeId = Integer.parseInt(request.getParameter("id"));

            // Delete employee from database via DAO
            EmployeeDao employeeDao = new EmployeeDao();
            int deleteCount = employeeDao.deleteEmployee(employeeId);

            if (deleteCount > 0) {
                // Employee deleted successfully, redirect to ViewServlet
                response.sendRedirect(request.getContextPath() +
"/ViewServlet");
            } else {
                // Handle delete failure
                response.getWriter().println("Failed to delete employee");
            }
        }
    }
}
```

EMPLOYEE.JAVA

```
public class Employee {
    private int id;
    private String name;
    private String password;
    private String email;
    private String country;
```

```
    // 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 String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public String getCountry() {
        return country;
    }

    public void setCountry(String country) {
        this.country = country;
    }
}
```

EMPLOYEEDAO.JAVA

```
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class EmployeeDao {
    private Connection con;
    private PreparedStatement ps;

    public EmployeeDao() {
        con = DBConnection.getConnection();
    }

    public List<Employee> getAllEmployees() {
        List<Employee> allEmployees = new ArrayList<>();
```

```
        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE");
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                Employee employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));

                allEmployees.add(employee);
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return allEmployees;
    }

    public Employee getEmployeeById(int id) {
        Employee employee = null;

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE WHERE ID =
?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return employee;
    }

    public int updateEmployee(Employee employee) {
        int count = 0;

        try {
            ps = con.prepareStatement("UPDATE EMPLOYEE SET NAME=?,
PASSWORD=?, EMAIL=?, COUNTRY=? WHERE ID=?");
            ps.setString(1, employee.getName());
            ps.setString(2, employee.getPassword());
```

```
            ps.setString(3, employee.getEmail());
            ps.setString(4, employee.getCountry());
            ps.setInt(5, employee.getId());

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }

    public int deleteEmployee(int id) {
        int count = 0;

        try {
            ps = con.prepareStatement("DELETE FROM EMPLOYEE WHERE ID = ?");
            ps.setInt(1, id);

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }
}
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class EmployeeDao {
    private Connection con;
    private PreparedStatement ps;

    public EmployeeDao() {
        con = DBConnection.getConnection();
    }

    public List<Employee> getAllEmployees() {
        List<Employee> allEmployees = new ArrayList<>();

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE");
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                Employee employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));

                allEmployees.add(employee);
            }
```

```
            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return allEmployees;
    }

    public Employee getEmployeeById(int id) {
        Employee employee = null;

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE WHERE ID =
?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return employee;
    }

    public int updateEmployee(Employee employee) {
        int count = 0;

        try {
            ps = con.prepareStatement("UPDATE EMPLOYEE SET NAME=?,
PASSWORD=?, EMAIL=?, COUNTRY=? WHERE ID=?");
            ps.setString(1, employee.getName());
            ps.setString(2, employee.getPassword());
            ps.setString(3, employee.getEmail());
            ps.setString(4, employee.getCountry());
            ps.setInt(5, employee.getId());

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }

    public int deleteEmployee(int id) {
```

```
        int count = 0;

        try {
            ps = con.prepareStatement("DELETE FROM EMPLOYEE WHERE ID = ?");
            ps.setInt(1, id);

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }
}
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class EmployeeDao {
    private Connection con;
    private PreparedStatement ps;

    public EmployeeDao() {
        con = DBConnection.getConnection();
    }

    public List<Employee> getAllEmployees() {
        List<Employee> allEmployees = new ArrayList<>();

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE");
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                Employee employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));

                allEmployees.add(employee);
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return allEmployees;
    }

    public Employee getEmployeeById(int id) {
        Employee employee = null;

        try {
```

```
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE WHERE ID =
?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return employee;
    }

    public int updateEmployee(Employee employee) {
        int count = 0;

        try {
            ps = con.prepareStatement("UPDATE EMPLOYEE SET NAME=?,
PASSWORD=?, EMAIL=?, COUNTRY=? WHERE ID=?");
            ps.setString(1, employee.getName());
            ps.setString(2, employee.getPassword());
            ps.setString(3, employee.getEmail());
            ps.setString(4, employee.getCountry());
            ps.setInt(5, employee.getId());

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }

    public int deleteEmployee(int id) {
        int count = 0;

        try {
            ps = con.prepareStatement("DELETE FROM EMPLOYEE WHERE ID = ?");
            ps.setInt(1, id);

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
```

```
    }
}
import java.sql.*;
import java.util.ArrayList;
import java.util.List;

public class EmployeeDao {
    private Connection con;
    private PreparedStatement ps;

    public EmployeeDao() {
        con = DBConnection.getConnection();
    }

    public List<Employee> getAllEmployees() {
        List<Employee> allEmployees = new ArrayList<>();

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE");
            ResultSet rs = ps.executeQuery();

            while (rs.next()) {
                Employee employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));

                allEmployees.add(employee);
            }

            rs.close();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return allEmployees;
    }

    public Employee getEmployeeById(int id) {
        Employee employee = null;

        try {
            ps = con.prepareStatement("SELECT * FROM EMPLOYEE WHERE ID =
?");
            ps.setInt(1, id);
            ResultSet rs = ps.executeQuery();

            if (rs.next()) {
                employee = new Employee();
                employee.setId(rs.getInt("ID"));
                employee.setName(rs.getString("NAME"));
                employee.setPassword(rs.getString("PASSWORD"));
                employee.setEmail(rs.getString("EMAIL"));
                employee.setCountry(rs.getString("COUNTRY"));
            }

            rs.close();
```

```
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return employee;
    }

    public int updateEmployee(Employee employee) {
        int count = 0;

        try {
            ps = con.prepareStatement("UPDATE EMPLOYEE SET NAME=?,
PASSWORD=?, EMAIL=?, COUNTRY=? WHERE ID=?");
            ps.setString(1, employee.getName());
            ps.setString(2, employee.getPassword());
            ps.setString(3, employee.getEmail());
            ps.setString(4, employee.getCountry());
            ps.setInt(5, employee.getId());

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }

    public int deleteEmployee(int id) {
        int count = 0;

        try {
            ps = con.prepareStatement("DELETE FROM EMPLOYEE WHERE ID = ?");
            ps.setInt(1, id);

            count = ps.executeUpdate();
        } catch (SQLException ex) {
            ex.printStackTrace(); // Handle or log the exception properly
        } finally {
            // Close PreparedStatement here if needed
        }

        return count;
    }
}
```

VIEWSERVLET.JAVA

```
import jakarta.servlet.*;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.util.List;

public class ViewServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private EmployeeDao employeeDao;
```

```
    public void init() throws ServletException {
        super.init();
        employeeDao = new EmployeeDao(); // Initialize EmployeeDao on
servlet initialization
    }

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

        if (action != null && action.equals("delete")) {
            // Handle deletion action
            int employeeId = Integer.parseInt(request.getParameter("id"));
            int deleteCount = employeeDao.deleteEmployee(employeeId);

            if (deleteCount > 0) {
                // Employee deleted successfully, redirect to refresh
employee list
                response.sendRedirect(request.getContextPath() +
"/ViewServlet");
            } else {
                // Handle delete failure
                response.getWriter().println("Failed to delete employee");
            }
        } else {
            // Fetch all employees and forward to viewEmployees.jsp
            List<Employee> allEmployees = employeeDao.getAllEmployees();
            request.setAttribute("allEmployees", allEmployees);
            RequestDispatcher dispatcher =
request.getRequestDispatcher("/viewEmployees.jsp");
            dispatcher.forward(request, response);
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
        // Redirect POST requests to doGet method
        doGet(request, response);
    }
}
```

EDITEMPLOYEE.JSP

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Edit Employee</title>
</head>
<body>
    <h1>Edit Employee</h1>
    <form action="EditServlet" method="post">
        <input type="hidden" name="action" value="save">
        <input type="hidden" name="id" value="${employee.id}">
```

EMPLOYEEVIEW,JSP

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>View All Employees</title>
</head>
<body>
    <h1>View All Employees</h1>

    <table border="2" width="100%">
        <tr style='background-color:#EF6079FF; color:white;'>
            <th>ID</th>
            <th>Name</th>
            <th>Password</th>
            <th>Email</th>
            <th>Country</th>
            <th colspan="2">Action</th>
        </tr>

        <c:forEach items="${allEmployees}" var="employee">
            <tr style='color:black;'>
                <td>${employee.id}</td>
                <td>${employee.name}</td>
                <td>${employee.password}</td>
                <td>${employee.email}</td>
                <td>${employee.country}</td>
                <td><a
href="EditServlet?action=edit&id=${employee.id}">Edit</a></td>
                <td><a
href="ViewServlet?action=delete&id=${employee.id}">Delete</a></td>
            </tr>
        </c:forEach>
    </table>
</body>
</html>
```

UPDATEFORM.JSP

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Update Employee</title>
</head>
<body>
    <h1>Update Employee</h1>
    <form action="EditServlet" method="post">
        <table>
            <tr>
                <td>Name:</td>
                <td><input type="text" name="txtName"></td>
            </tr>
            <tr>
                <td>Password:</td>
                <td><input type="password" name="txtPassword"></td>
            </tr>
```

```
            <tr>
                <td>Email:</td>
                <td><input type="email" name="txtEmail"></td>
            </tr>
            <tr>
                <td>Country:</td>
                <td>
                    <select name="country">
                        <option value="India">India</option>
                        <option value="Russia">Russia</option>
                        <option value="China">China</option>
                        <option value="UK">UK</option>
                        <option value="USA">USA</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td colspan="2">
                    <input type="submit" value="EDit&save">
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

OUTPUT: