

INSURANCE POLICY MANAGEMENT SYSTEM

1. Features to Implement

1. Add Policy – Allows users to add a new insurance policy (PolicyID, HolderName, PolicyType, Premium, ExpiryDate).
 2. View Policies – Displays all insurance policies.
 3. Update Policy – Provides an interface to modify policy details.
 4. Delete Policy – Allows users to remove a policy record.
- Search Policy (Optional) – Enables searching by holder name or policy type.

2. Project Folder Structure

Folder	Contents
src/com.insurance.dao/	PolicyDAO.java, PolicyDAOImpl.java
src/com.insurance.model/	Policy.java
src/com.insurance.servlet/	AddPolicyServlet.java, ViewPolicyServlet.java, UpdatePolicyServlet.java, DeletePolicyServlet.java
src/com.insurance.util/	DBConnection.java
WebContent/view/	add_policy.jsp, view_policies.jsp, update_policy.jsp, index.jsp
WebContent/css/	styles.css

3. Database Schema

Column Name	Data Type	Constraints
policyid	INT	PRIMARY KEY, AUTO_INCREMENT
holdername	VARCHAR(100)	NOT NULL
policytype	VARCHAR(50)	NOT NULL
premium	DOUBLE	NOT NULL
expirydate	DATE	NOT NULL

CODE:

PACKAGES FILES

Insurance Policy Management System:

com.insurance.dao:

Insurance.DAO.java:

```
package com.insurance.servlet;

import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import com.insurance.dao.PolicyDAOImpl;
@WebServlet("/deletePolicy")
public class DeletePolicyServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
```

```

        new PolicyDAOImpl().deletePolicy(id);
        response.sendRedirect(request.getContextPath() + "/viewPolicy");
    }
}

```

InsuranceDAOImpl.java:

```

package com.insurance.dao;
import java.sql.*;
import java.util.*;
import com.insurance.model.Policy;
import com.insurance.util.DBConnection;
public class PolicyDAOImpl implements PolicyDAO {
    @Override
    public boolean addPolicy(Policy policy) {
        String sql = "INSERT INTO policy(holdername, policytype, premium, expirydate)
VALUES(?, ?, ?, ?)";
        try (Connection con = DBConnection.getConnection()) {
            PreparedStatement ps = con.prepareStatement(sql));
            ps.setString(1, policy.getHolderName());
            ps.setString(2, policy.getPolicyType());
            ps.setDouble(3, policy.getPremium());
            ps.setDate(4, policy.getExpiryDate());
            return ps.executeUpdate() > 0;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return false;
    }
    @Override
    public List<Policy> getAllPolicies() {
        List<Policy> list = new ArrayList<>();
        try (Connection con = DBConnection.getConnection()) {
            PreparedStatement ps = con.prepareStatement("SELECT * FROM policy");
            ResultSet rs = ps.executeQuery() {
                while (rs.next()) {
                    Policy p = new Policy();
                    p.setPolicyId(rs.getInt("policyid"));
                    p.setHolderName(rs.getString("holdername"));
                    p.setPolicyType(rs.getString("policytype"));
                    p.setPremium(rs.getDouble("premium"));
                    p.setExpiryDate(rs.getDate("expirydate"));
                    list.add(p);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        return list;
    }
    @Override
    public boolean updatePolicy(Policy policy) {
        String sql = "UPDATE policy SET holdername=?, policytype=?, premium=?, expirydate=?"

```

```

WHERE policyid=?;
try (Connection con = DBConnection.getConnection();
    PreparedStatement ps = con.prepareStatement(sql)) {
    ps.setString(1, policy.getHolderName());
    ps.setString(2, policy.getPolicyType());
    ps.setDouble(3, policy.getPremium());
    ps.setDate(4, policy.getExpiryDate());
    ps.setInt(5, policy.getPolicyId());
    return ps.executeUpdate() > 0;
} catch (Exception e) {
    e.printStackTrace();
}
return false;
}
@Override
public boolean deletePolicy(int policyId) {
    try (Connection con = DBConnection.getConnection();
        PreparedStatement ps = con.prepareStatement("DELETE FROM policy WHERE
policyid=?")) {
        ps.setInt(1, policyId);
        return ps.executeUpdate() > 0;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return false;
}
@Override
public Policy getPolicyById(int policyId) {
    Policy p = null;
    try (Connection con = DBConnection.getConnection();
        PreparedStatement ps = con.prepareStatement("SELECT * FROM policy WHERE
policyid=?")) {
        ps.setInt(1, policyId);
        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            p = new Policy();
            p.setPolicyId(rs.getInt("policyid"));
            p.setHolderName(rs.getString("holdername"));
            p.setPolicyType(rs.getString("policytype"));
            p.setPremium(rs.getDouble("premium"));
            p.setExpiryDate(rs.getDate("expirydate"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return p;
}
}

```

com.insurance.model:**Policy.java:**

```
package com.insurance.model;
import java.sql.Date;
public class Policy {
    private int policyId;
    private String holderName;
    private String policyType;
    private double premium;
    private Date expiryDate;
    public Policy() {}
    public Policy(String holderName, String policyType, double premium, Date expiryDate) {
        this.holderName = holderName;
        this.policyType = policyType;
        this.premium = premium;
        this.expiryDate = expiryDate;
    }
    public int getPolicyId() { return policyId; }
    public void setPolicyId(int policyId) { this.policyId = policyId; }
    public String getHolderName() { return holderName; }
    public void setHolderName(String holderName) { this.holderName = holderName; }
    public String getPolicyType() { return policyType; }
    public void setPolicyType(String policyType) { this.policyType = policyType; }
    public double getPremium() { return premium; }
    public void setPremium(double premium) { this.premium = premium; }
    public Date getExpiryDate() { return expiryDate; }
    public void setExpiryDate(Date expiryDate) { this.expiryDate = expiryDate; }
}
```

com.insurance.servlet:**AddPolicyServlet.java:**

```
package com.insurance.servlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.sql.Date;
import com.insurance.dao.PolicyDAOImpl;
import com.insurance.model.Policy;
@WebServlet("/addPolicy")
public class AddPolicyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest req, HttpServletResponse res) throws IOException {
        String holderName = req.getParameter("holderName");
        String policyType = req.getParameter("policyType");
        double premium = Double.parseDouble(req.getParameter("premium"));
        Date expiryDate = Date.valueOf(req.getParameter("expiryDate"));
        Policy policy = new Policy(holderName, policyType, premium, expiryDate);
        PolicyDAO dao = new PolicyDAOImpl();
```

```

        dao.addPolicy(policy);
        res.sendRedirect("viewPolicies");
    }
}

```

DeletePolicyServlet.java:

```

package com.insurance.servlet;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import com.insurance.dao.PolicyDAOImpl;
@WebServlet("/deletePolicy")
public class DeletePolicyServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int id = Integer.parseInt(request.getParameter("id"));
        new PolicyDAOImpl().deletePolicy(id);
        response.sendRedirect(request.getContextPath() + "/viewPolicy");
    }
}

```

UpdatePolicyServlet.java:

```

package com.insurance.servlet;
import com.insurance.dao.PolicyDAO;
import com.insurance.dao.PolicyDAOImpl;
import com.insurance.model.Policy;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.sql.Date;
@WebServlet("/updatePolicy")
public class UpdatePolicyServlet extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        int policyId = Integer.parseInt(request.getParameter("policyid"));
        String holderName = request.getParameter("holdername");
        String policyType = request.getParameter("policytype");
        double premium = Double.parseDouble(request.getParameter("premium"));
        Date expiryDate = Date.valueOf(request.getParameter("expirydate"));
        Policy policy = new Policy();
        policy.setPolicyId(policyId);
        policy.setHolderName(holderName);
        policy.setPolicyType(policyType);
        policy.setPremium(premium);
        policy.setExpiryDate(expiryDate);
        new PolicyDAOImpl().updatePolicy(policy);
    }
}

```

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

ViewPolicyServlet.java:

```
package com.insurance.servlet;

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

import com.insurance.dao.PolicyDAOImpl;
import com.insurance.model.Policy;

@WebServlet("/viewPolicy")
public class ViewPolicyServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        List<Policy> list = new PolicyDAOImpl().getAllPolicies();
        request.setAttribute("policies", list);
        RequestDispatcher rd = request.getRequestDispatcher("view/view_policies.jsp");
        rd.forward(request, response);
    }
}
```

com.school.util:

DBConnection.java:

```
package com.insurance.util;

import java.sql.Connection;
import java.sql.DriverManager;
public class DBConnection {

    private static final String URL = "jdbc:mysql://localhost:3306/insurance_db";
    private static final String USER = "root";
    private static final String PASSWORD = "mysql@123";
```

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

JSP Files:

index.jsp:

```
<html>  
<head>  
    <title>Insurance Policy Management</title>  
</head>  
<body>  
    <h2>Insurance Policy Management</h2>  
    <a href="add_policy.jsp">Add Policy</a><br><br>  
    <a href="../viewPolicy">View Policies</a>  
</body>  
</html>
```

add_policy.jsp:

```
<html>  
<head>  
    <title>Add Policy</title>  
</head>  
<body>
```

```
<h2>Add Policy</h2>

<form action="<%request.getContextPath()%>/addPolicy" method="post">

    Holder Name: <input type="text" name="holdername" required><br><br>

    Policy Type: <input type="text" name="policytype" required><br><br>

    Premium: <input type="number" name="premium" required><br><br>

    Expiry Date: <input type="date" name="expirydate" required><br><br>

    <input type="submit" value="Add Policy">

</form>
```

```
</body>
```

```
</html>
```

update_student.jsp:

```
<%@ page import="com.insurance.model.Policy" %>

<%
    Policy p = (Policy) request.getAttribute("policy");
    if (p == null) {
        out.println("<h3>Policy not found!</h3>");
        return;
    }
%>

<html>
    <head>
        <title>Update Policy</title>
    </head>
    <body>
        <h2>Update Insurance Policy</h2>
        <form action="${pageContext.request.contextPath}/updatePolicy" method="post">
            <input type="hidden" name="policyid" value="<%p.getId()%>">
```

Holder Name:

```
<input type="text" name="holdername" value="<% p.getHolderName() %>" required>
```

```
<br><br>
```

Policy Type:

```
<input type="text" name="policytype" value="<% p.getPolicyType() %>" required>
```

```
<br><br>
```

Premium:

```
<input type="number" name="premium" value="<% p.getPremium() %>" required>
```

```
<br><br>
```

Expiry Date:

```
<input type="date" name="expirydate" value="<% p.getExpiryDate() %>" required>
```

```
<br><br>
```

```
<input type="submit" value="Update Policy">
```

```
</form>
```

```
<br>
```

```
<a href="#"><% pageContext.request.contextPath %>/viewPolicies">
```

Back to Policy List

```
</a>
```

```
</body>
```

```
</html>
```

view_student.jsp:

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
```

```
pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Insert title here</title>
```

```

</head>

<body>

<%@ page import="java.util.*,com.insurance.model.Policy" %>

<h2>Policy List</h2>

<a href="<%="request.getContextPath()%>/add_policy.jsp">Add Policy</a><br><br>

<table border="1">

<tr>

    <th>ID</th>

    <th>Holder Name</th>

    <th>Policy Type</th>

    <th>Premium</th>

    <th>Expiry Date</th>

    <th>Action</th>

</tr>

<%

List<Policy> list = (List<Policy>) request.getAttribute("policies");

for (Policy p : list) {

%>

<tr>

    <td><%= p.getPolicyId() %></td>

    <td><%= p.getHolderName() %></td>

    <td><%= p.getPolicyType() %></td>

    <td><%= p.getPremium() %></td>

    <td><%= p.getExpiryDate() %></td>

    <td>

        <a href="$ {pageContext.request.contextPath}/editPolicy?id=$ {p.id}">

            Edit /<br>

        </a>

    </td>


```

```
<a href="<%="request.getContextPath()%>/deletePolicy?id=<%= p.getPolicyId()%>">Delete</a>

</td>
</tr>
<% } %>
</table>
</body>
</html>
```

delete_student.jsp:

```
<%
String id = request.getParameter("id");
%>
<html>
<head>
<title>Delete Policy</title>
</head>
<body>
<h2>Delete Policy Confirmation</h2>
<p>Are you sure you want to delete Policy ID: <b><%= id %></b>?</p>
<a href="../deletePolicy?id=<%= id %>">
    <button>Yes, Delete</button>
</a>
<a href="../viewPolicies">
    <button>No, Cancel</button>
</a>
</body>
</html>
```

HOME PAGE:

Insurance Policy Management

[Add Policy](#) | [View Policies](#)

ADD POLICY PAGE:

Add Policy

Holder Name:

Policy Type:

Premium:

Expiry Date:

UPDATE POLICY PAGE:

Update Policy

Holder Name: Alice Smith

Policy Type: Vehicle

Premium: 980

Expiry Date: 2027-12-31

Update Policy**POLICY PAGE AFTER UPDATE:**

Policy List

ID	Name	Type	Premium	Expiry	Action
1	John Doe	Health	1200	2024-12-31	Delete
2	Alice Smith	Vehicle	980	2025-06-30	Delete

[Add New](#)**POLICY PAGE AFTER DELETE:**

Policy List

ID	Name	Type	Premium	Expiry	Action

[Add New](#)