

Assignment No : 10

#BankBean.java

package com.bank;

```
public class BankBean implements BankInterface {  
    private double balance;
```

```
    public BankBean(double initialBalance) {  
        this.balance = initialBalance;  
    }
```

```
    public double getBalance() {  
        return balance;  
    }
```

```
    @Override
```

```
    public String deposit(double amount) {  
        if (amount <= 0) {  
            return "⚠️ Deposit amount must be greater than ₹0.";  
        }  
        balance += amount;  
        return "Deposited ₹" + String.format("%.2f", amount) + ". New Balance: ₹" +  
String.format("%.2f", balance);  
    }
```

```
    @Override
```

```
    public String withdraw(double amount) {  
        if (amount <= 0) {  
            return "⚠️ Withdrawal amount must be greater than ₹0.";  
        }  
        if (amount > balance) {  
            return "❌ Withdrawal Failed. Insufficient Balance: ₹" + String.format("%.2f", balance);  
        }  
        balance -= amount;  
        return "Withdrawn ₹" + String.format("%.2f", amount) + ". Remaining Balance: ₹" +  
String.format("%.2f", balance);  
    }  
}
```

#BankInterface.java

package com.bank;

```
public interface BankInterface {  
    String deposit(double amount);
```

```
String withdraw(double amount);  
}
```

#index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  
<html>  
<head>  
  <meta charset="UTF-8">  
  <title>Bank App</title>  
  <style>  
    body {  
      font-family: Arial;  
      background: #f0f2f5;  
      text-align: center;  
      padding-top: 60px;  
    }  
    .box {  
      background: white;  
      display: inline-block;  
      padding: 30px;  
      border-radius: 8px;  
      box-shadow: 0 0 10px #ccc;  
    }  
    input, select {  
      padding: 8px;  
      margin: 10px;  
      width: 200px;  
    }  
    .btn {  
      background: #28a745;  
      color: white;  
      border: none;  
      cursor: pointer;  
    }  
    .btn:hover {  
      background: #218838;  
    }  
  </style>  
</head>  
<body>  
  <div class="box">  
    <h2>&img alt="Bank icon" data-bbox="178 801 205 821"/> Welcome to Simple Bank</h2>  
    <form action="result.jsp" method="post">  
      <input type="text" name="accNo" placeholder="Account No" required /><br/>  
      <input type="number" step="0.01" name="amount" placeholder="Amount" required /><br/>  
>  
      <select name="action" required>  
        <option value="deposit">Deposit</option>
```

```

        <option value="withdraw">Withdraw</option>
    </select><br/>
    <input type="submit" value="Submit" class="btn" />
</form>
</div>
</body>
</html>

```

#result.jsp

```

<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ page import="com.bank.BankBean" %>
<%@ page import="java.util.*, java.text.*" %>

```

```

<%

```

```

    DecimalFormat df = new DecimalFormat("0.00");
    SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");

```

```

    HttpSession sessionObj = request.getSession();
    String result = "";
    int accNo = 0;
    double amount = 0.0;
    String action = request.getParameter("action");

```

```

    try {
        accNo = Integer.parseInt(request.getParameter("accNo"));
        amount = Double.parseDouble(request.getParameter("amount"));

```

```

        Map<Integer, BankBean> accountMap = (Map<Integer, BankBean>)
sessionObj.getAttribute("accountMap");
        if (accountMap == null) accountMap = new HashMap<>();

```

```

        BankBean bank = accountMap.get(accNo);
        if (bank == null) {
            bank = new BankBean(1000); // Default initial balance
            accountMap.put(accNo, bank);
        }

```

```

        // Transaction logic with validation
        if (amount <= 0) {
            result = "❌ Amount must be greater than ₹0.";
        } else if ("deposit".equalsIgnoreCase(action)) {
            result = bank.deposit(amount);
        } else if ("withdraw".equalsIgnoreCase(action)) {
            result = bank.withdraw(amount);
        } else {
            result = "❌ Invalid transaction type.";
        }


```

```



sessionObj.setAttribute("accountMap", accountMap);

// Transaction history tracking
Map<Integer, List<String>> historyMap = (Map<Integer, List<String>>)
sessionObj.getAttribute("historyMap");
if (historyMap == null) historyMap = new HashMap<>();

List<String> history = historyMap.getDefault(accNo, new ArrayList<>());

//  Add entry ONLY if successful
if (result.startsWith("Deposited") || result.startsWith("Withdrawn")) {
    history.add(sdf.format(new Date()) + " | " + action.toUpperCase() + " ₹" +
df.format(amount));
}

historyMap.put(accNo, history);
sessionObj.setAttribute("historyMap", historyMap);

} catch (NumberFormatException e) {
    result = " Invalid input format. Please enter numeric values for Account Number and
Amount.";
} catch (Exception e) {
    result = " Error processing transaction: " + e.getMessage();
}
}
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Transaction Result</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background: #f5f5f5;
        text-align: center;
        padding-top: 50px;
    }
    .card {
        background: #fff;
        display: inline-block;
        padding: 30px;
        border-radius: 10px;
        box-shadow: 0 0 12px rgba(0,0,0,0.1);
    }
    .btn {
        margin-top: 20px;

```

```

        display: inline-block;
        padding: 10px 20px;
        background: #007bff;
        color: white;
        text-decoration: none;
        border-radius: 5px;
    }
    ul {
        text-align: left;
        margin-top: 20px;
        padding: 0;
        list-style: none;
    }
    li {
        margin-bottom: 5px;
    }
</style>
</head>
<body>
    <div class="card">
        <h2>✅ Transaction Status</h2>
        <p><strong><%= result %></strong></p>

        <% if (result.startsWith("Deposited") || result.startsWith("Withdrawn")) { %>
            <p>Account No: <%= accNo %></p>
            <p>Current Balance: ₹<%= df.format(((Map<Integer, BankBean>)
sessionObj.getAttribute("accountMap")).get(accNo).getBalance()) %></p>

            <h4>📄 Transaction History</h4>
            <ul>
                <% for (String entry : ((Map<Integer, List<String>>)
sessionObj.getAttribute("historyMap")).get(accNo)) { %>
                    <li><%= entry %></li>
                <% } %>
            </ul>
            <% } %>
        </div>
</body>
</html>

```

#web.xml

```

<web-app>
    <display-name>Simple Bank App</display-name>
</web-app>

```

Output :-

The screenshot shows the WildFly console interface in a web browser. The address bar displays `localhost:9990/console/index.html#deployments;path=deployment-dply-simplebankappwar`. The console has tabs for Homepage, Deployments, Configuration, Runtime, and Access Control. The Deployments tab is active, showing a list of deployments on the left with 'SimpleBankApp...' selected. The main area displays the details for 'SimpleBankApp.war'.

SimpleBankApp.war

✓ The deployment **SimpleBankApp.war** is enabled and active. [Disable](#)

Main Attributes

Name:	SimpleBankApp.war
Runtime Name:	SimpleBankApp.war
Context Root:	/SimpleBankApp
Hash:	4968d562b38058db3defa5db47cab507409a8874
Enabled, Managed, Exploded:	✓ ✓ ✗
Status:	OK
Last enabled at:	4/12/25, 2:16 AM
Last disabled at:	n/a

At the bottom right, there are links for '3.7.10.Final', 'Tools', and 'Settings'.

The screenshot shows the Simple Bank application running in a web browser. The address bar displays `127.0.0.1:8080/SimpleBankApp/index.jsp`. The application has a light purple background. In the center, there is a white card titled 'Welcome to Simple Bank' with a bank icon. The card contains a form with three input fields: a text field with the value '2', a dropdown menu with the value '10', and a dropdown menu with the value 'Deposit'. Below these fields is a green 'Submit' button.

Welcome to Simple Bank

2

10

Deposit

Submit

