**Personal Finance Management System:**

**Objective**

To develop a Personal Finance Management System that allows users to track their earnings and expenses, calculate their savings, and gain a clear insight into their financial behaviour. The system empowers users to manage finances effectively through simple analysis and reporting.

**Introduction**

Managing personal finances is an essential part of modern life. People often struggle with tracking their income and expenses and making data-driven decisions. A Personal Finance Management System automates and organizes financial activities, helping users become more financially disciplined.

This system allows users to:

* Record their income and expenses.
* View summarized reports of earnings, expenses, and savings.
* Get a dashboard view of their financial health.

| **Layer** | **Technology Used** |
| --- | --- |
| Frontend | JSP, HTML, CSS |
| Backend | Spring Boot with Servlet-style coding |
| Build Tool | Maven |
| Database | MySQL |
| Server | Apache Tomcat |
| Language | Java |

**Methodology / Implementation Summary**

1. **User Authentication**
   * Users log in via the login.jsp page.
   * Login servlet validates credentials and stores userId in session.
2. **Dashboard**
   * After login, users are redirected to dashboard.jsp.
   * Backend fetches total income and expense from the database.
   * Savings are calculated and displayed.
3. **Add Income/Expense**
   * JSP pages allow users to add income or expenses.
   * Data is submitted via forms and inserted into the database via servlets.

Short Theory of Methods and Technologies Used

* **Spring Boot:** A widely-used Java framework that simplifies the creation of stand-alone, production-ready Spring applications by providing auto-configuration and starter dependencies.
* **Maven:** A project management and comprehension tool used to manage dependencies, build, and document the project.
* **Servlets:** Java programs running on a web server that handle HTTP requests and generate dynamic responses.
* **JSP (JavaServer Pages):** A technology that helps to create dynamic web content with embedded Java code.
* **JDBC (Java Database Connectivity):** An API to connect and execute SQL queries on databases like MySQL.
* **MySQL:** An open-source relational database system for storing user data.
* **HTML & CSS:** Markup and styling languages used for creating the user interface.

**Implementation Summary:**

The Personal Finance Management System follows the Model-View-Controller (MVC) architecture:

* **Model:** Represents database tables such as users, incomes, and expenses.
* **View:** JSP pages provide the UI with forms to enter data and dashboards to view summaries.
* **Controller:** Servlets handle business logic, process requests, and interact with the database using JDBC.

The system workflow includes:

* User authentication and session management.
* Adding income and expense records with validation.
* Displaying summarized financial data in the dashboard.
* Redirecting unauthenticated users to login.

Spring Boot manages dependencies and runs the embedded server for quick startup and deployment.

**Project Structure:**

**PersonalFinanceSystem/**

**│**

**|── src/main/java/com/example/personalfinance/**

**│ |── controller/**

**│ │ |── LoginServlet.java**

**│ │ |── AddIncomeServlet.java**

**│ │ |── AddExpenseServlet.java**

**│ │ |── DashboardServlet.java**

**│ |── model/**

**│ │ |─ User.java**

**│ │ |── Income.java**

**│ │ |── Expense.java**

**│ |── repository/**

**│ │ |── DBUtil.java**

**│ |── PersonalFinanceApplication.java**

**│**

**|── src/main/resources/**

**│ |── application.properties**

**│**

**|── src/main/webapp/**

**│ |── WEB-INF/**

**│ │ |── web.xml**

**│ │ |── views/**

**│ │ |── login.jsp**

**│ │ |── dashboard.jsp**

**│ │ |── addIncome.jsp**

**│ │ |── addExpense.jsp**

**│ |── css/**

**│ |── styles.css**

**│**

**|── pom.xml**

**|── README.md**

**Database Schema:**

**CREATE DATABASE finance\_db;**

**USE finance\_db;**

**CREATE TABLE users (**

**id INT PRIMARY KEY AUTO\_INCREMENT,**

**username VARCHAR(50) UNIQUE NOT NULL,**

**password VARCHAR(100) NOT NULL**

**);**

**CREATE TABLE incomes (**

**id INT PRIMARY KEY AUTO\_INCREMENT,**

**user\_id INT,**

**amount DECIMAL(10,2),**

**source VARCHAR(100),**

**date DATE,**

**FOREIGN KEY (user\_id) REFERENCES users(id)**

**);**

**CREATE TABLE expenses (**

**id INT PRIMARY KEY AUTO\_INCREMENT,**

**user\_id INT,**

**amount DECIMAL(10,2),**

**category VARCHAR(100),**

**date DATE,**

**FOREIGN KEY (user\_id) REFERENCES users(id)**

**);**

**Code Files:**

LoginServlet.java:

package com.example.personalfinance.controller;

import jakarta.servlet.\*;

import jakarta.servlet.http.\*;

import jakarta.servlet.annotation.\*;

import java.io.IOException;

import java.sql.\*;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username = request.getParameter("username");

String password = request.getParameter("password");

try {

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/finance\_db", "root", "your\_password");

PreparedStatement ps = conn.prepareStatement("SELECT id FROM users WHERE username=? AND password=?");

ps.setString(1, username);

ps.setString(2, password);

ResultSet rs = ps.executeQuery();

if(rs.next()) {

int userId = rs.getInt("id");

HttpSession session = request.getSession();

session.setAttribute("userId", userId);

response.sendRedirect("dashboard.jsp");

} else {

request.setAttribute("error", "Invalid username or password");

request.getRequestDispatcher("login.jsp").forward(request, response);

}

conn.close();

} catch(Exception e) {

e.printStackTrace();

}

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

response.sendRedirect("login.jsp");

}

}

Dashboard.jsp:

<%@ page import="java.sql.\*, javax.servlet.http.\*" %>

<%@ page session="true" %>

<%

HttpSession session = request.getSession(false);

if(session == null || session.getAttribute("userId") == null) {

response.sendRedirect("login.jsp");

return;

}

int userId = (int) session.getAttribute("userId");

double totalIncome = 0, totalExpense = 0;

try {

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/finance\_db", "root", "your\_password");

PreparedStatement psIncome = conn.prepareStatement("SELECT SUM(amount) FROM incomes WHERE user\_id=?");

psIncome.setInt(1, userId);

ResultSet rsIncome = psIncome.executeQuery();

if(rsIncome.next()) totalIncome = rsIncome.getDouble(1);

PreparedStatement psExpense = conn.prepareStatement("SELECT SUM(amount) FROM expenses WHERE user\_id=?");

psExpense.setInt(1, userId);

ResultSet rsExpense = psExpense.executeQuery();

if(rsExpense.next()) totalExpense = rsExpense.getDouble(1);

conn.close();

} catch(Exception e) {

e.printStackTrace();

}

double savings = totalIncome - totalExpense;

%>

<!DOCTYPE html>

<html>

<head>

<title>Dashboard</title>

<link rel="stylesheet" href="css/styles.css" />

</head>

<body>

<h2>Welcome to Your Personal Finance Dashboard</h2>

<p>Total Income: ₹<%= totalIncome %></p>

<p>Total Expense: ₹<%= totalExpense %></p>

<p><strong>Savings: ₹<%= savings %></strong></p>

<a href="addIncome.jsp">Add Income</a> |

<a href="addExpense.jsp">Add Expense</a> |

<a href="logout.jsp">Logout</a>

</body>

</html>

addIncome.jsp:

<%@ page session="true" %>

<%

if(session == null || session.getAttribute("userId") == null) {

response.sendRedirect("login.jsp");

return;

}

%>

<!DOCTYPE html>

<html>

<head>

<title>Add Income</title>

<link rel="stylesheet" href="css/styles.css" />

</head>

<body>

<h2>Add Income</h2>

<form action="addIncome" method="post">

<label>Amount: </label>

<input type="number" name="amount" step="0.01" required/><br/><br/>

<label>Source: </label>

<input type="text" name="source" required/><br/><br/>

<label>Date: </label>

<input type="date" name="date" required/><br/><br/>

<button type="submit">Add</button>

</form>

<br/>

<a href="dashboard.jsp">Back to Dashboard</a>

</body>

</html>

addIncome Servlet:

@WebServlet("/addIncome")

public class AddIncomeServlet extends HttpServlet {

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

HttpSession session = request.getSession(false);

if(session == null || session.getAttribute("userId") == null) {

response.sendRedirect("login.jsp");

return;

}

int userId = (int) session.getAttribute("userId");

double amount = Double.parseDouble(request.getParameter("amount"));

String source = request.getParameter("source");

String date = request.getParameter("date");

try {

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

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/finance\_db", "root", "your\_password");

PreparedStatement ps = conn.prepareStatement("INSERT INTO incomes (user\_id, amount, source, date) VALUES (?, ?, ?, ?)");

ps.setInt(1, userId);

ps.setDouble(2, amount);

ps.setString(3, source);

ps.setDate(4, Date.valueOf(date));

ps.executeUpdate();

conn.close();

response.sendRedirect("dashboard.jsp");

} catch(Exception e) {

e.printStackTrace();

}

}

}

CSS (styles.css):

body {

font-family: Arial, sans-serif;

margin: 40px;

background-color: #f4f6f7;

color: #333;

}

h2 {

color: #2c3e50;

}

a {

text-decoration: none;

color: #2980b9;

margin-right: 15px;

}

a:hover {

text-decoration: underline;

}

form label {

display: inline-block;

width: 100px;

font-weight: bold;

}

input, button {

padding: 8px;

margin: 5px 0;

}

button {

background-color: #2980b9;

color: white;

border: none;

cursor: pointer;

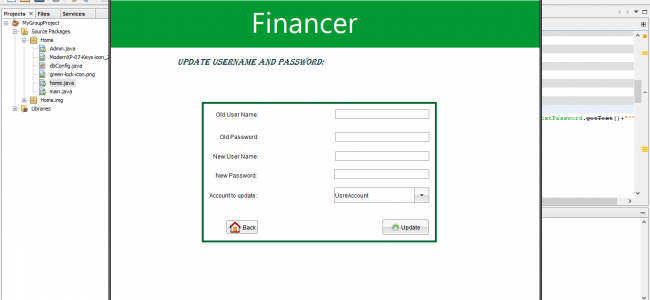
}

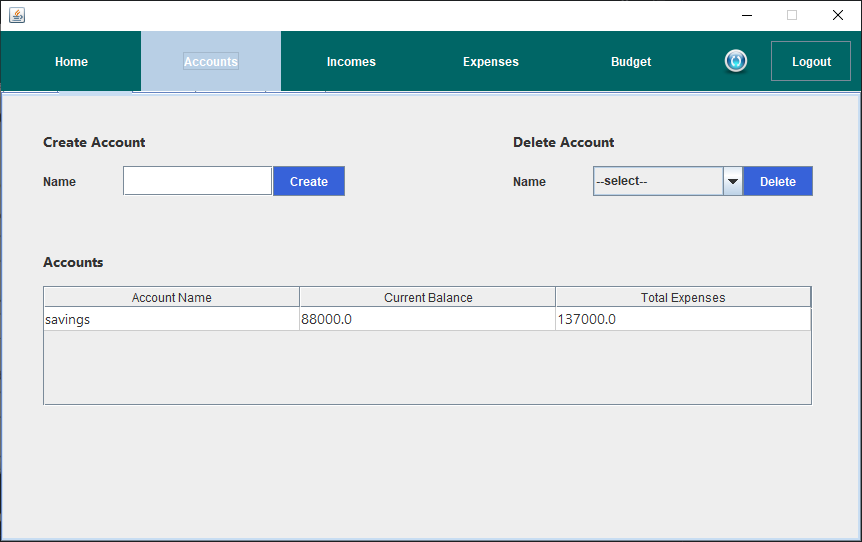
button:hover {

background-color: #3498db;

}

**Home Page Screenshots:-**





**Result**

The Personal Finance Management System offers users an easy-to-use interface for tracking income and expenses. Users can securely log in, add their financial transactions, and view summaries showing total income, expenses, and savings. This helps users analyze their spending habits and manage their finances more effectively.

**Conclusion**

The project successfully demonstrates the implementation of a personal finance tracker using Java web technologies with Spring Boot, Servlets, JSP, and MySQL. The MVC architecture ensures clear separation of concerns, making the system maintainable and scalable.

Further enhancements could include:

* Visualizing data with charts and graphs.
* Adding robust user authentication with encrypted passwords.
* Enabling expense categorization with filters.
* Implementing mobile-friendly responsive design.
* Adding notifications for budgeting or upcoming payments.

Overall, this system provides a foundational platform for personal financial management.

**References:-**

**Java (Desktop – Swing/JavaFX)**

* [**Personal Finance Manager (Java Swing)**](https://github.com/litzvi/personal-finance-manager)  
  Features: Budget tracking, transaction management, income/expense logs.  
  Technologies: Java Swing, MySQL.
* [**Finance Tracker JavaFX**](https://github.com/zeeshanejaz/Finance-Tracker)  
  Features: Monthly expense tracking, charts.

**For Java Desktop (Swing/JavaFX)**

* **"Build a Personal Budget Application in JavaFX" – CodeJava.net**  
  Walks you through building a finance app with JavaFX.
* **"Java Personal Finance Tracker"** on GeeksforGeeks  
  Basic version using Java Swing with full code samples.