

Technical Design Document: Responsive Personal Finance Web App (J2EE)

Project Title: Personal Finance Web App
Technology Stack: Java, JSP, Servlets, Embedded CSS
Deployment: Apache Tomcat (Localhost)
IDE: Eclipse
Version Control: GitHub Repository: [Link](#)

1. Objective

The goal of this project is to create a responsive personal finance management application using J2EE technologies. The app allows users to track income, expenses, investments, and plan budgets, while also offering features for recurring transactions and financial summaries.

2. Core Features Implemented

Feature	Description
Record Income and Expenses	Users can input transaction type, amount, category, and tags via form.
Categories	Uses Enum (Transaction.Category) to classify financial entries.
Tags	Users can label transactions with custom tags.
Budget Planning	setBudget.jsp allows users to set monthly budgets.
Automate Monthly Transactions	Implemented with RecurringTransaction.java and corresponding servlet.
Investment Tracking	Investment.java, InvestmentServlet, and investment JSPs handle logic.
Visual Chart of Spending	Pie chart rendered using HTML5 Canvas based on category totals.
Responsive Interface	All pages styled using embedded CSS with a soft pink and white theme.

3. Project Structure

```
PersonalFinanceApp/  
|-- src/main/java/com/financeapp/  
| |-- Transaction.java  
| |-- TransactionServlet.java  
| |-- RecurringTransaction.java  
| |-- Investment.java  
| |-- BudgetManager.java  
| |-- SetBudgetServlet.java  
| |-- LoginServlet.java  
| |-- RegisterServlet.java  
|-- src/main/webapp/  
    |-- index.jsp  
    |-- login.jsp  
    |-- register.jsp  
    |-- addTransaction.jsp  
    |-- viewTransactions.jsp  
    |-- setBudget.jsp  
    |-- addRecurring.jsp  
    |-- addInvestment.jsp  
    |-- viewInvestments.jsp  
    |-- dashboard.jsp
```

4. Servlet-JSP Flow

- **Login/Register:** LoginServlet and RegisterServlet validate users and redirect to dashboard.
 - **Transaction Management:** TransactionServlet handles storing and displaying transaction data.
 - **Budget:** SetBudgetServlet manages the monthly budget data.
 - **Recurring:** AddRecurringServlet converts recurring entries into regular transactions.
 - **Investments:** InvestmentServlet manages investment records.
-

5. CSS and UI

- All pages are styled with embedded CSS.
 - Color scheme: White background with soft pink (#FAE6E7) highlights.
 - Layout is centered using div.card with responsive styling for tables and charts.
-

6. Data Management

- **No Database:** All data is held in memory using Java collections.
 - **Transaction List:** Maintained in TransactionServlet.
 - **Recurring and Investments:** Managed by corresponding servlet classes.
-

7. Limitations

- No persistent data storage; data resets on server restart.
 - No user session management or role-based access.
-

8. Future Improvements

- Add database support using MySQL and JDBC.
 - Implement user authentication with hashed passwords.
-

9. Conclusion

This web app effectively demonstrates the use of J2EE technologies for solving real-world financial management needs. With a focus on clarity, responsiveness, and core financial features, it serves as a strong portfolio project for Java developers.

Prepared by: Jasmine Fowler