**📁 Project Directory Overview**

**Personal\_Finance\_Management-Java-Swing/**

**│**

**├── database\_setup/ # SQL scripts and setup instructions for MySQL**

**├── images/ # UI icons and background images**

**├── lib/ # External libraries like MySQL connector and JCalendar**

**├── nbproject/ # NetBeans project configuration files**

**├── build/classes/ # Core application source code**

**│ ├── Chart/**

**│ │ └── IncomeExpenseChart.java**

**│ ├── Database/**

**│ │ ├── DatabaseManager.java**

**│ │ └── UserSession.java**

**│ ├── Home/**

**│ │ ├── HomePage.java**

**│ │ └── HomePage.form**

**│ ├── Icon/**

**│ │ └── icons.png (and other icon images)**

**│ ├── Login/**

**│ │ ├── Login.java**

**│ │ ├── Login.form**

**│ │ ├── SignUp.java**

**│ │ └── SignUp.form**

**│ └── personalfinancemanagement/**

**│ └── PersonalFinanceManagement.java**

**├── .gitignore**

**└── README.md**

**🔄 Flow Diagram**

**[PersonalFinanceManagement.java] --> Launches the app**

**|**

**[Login.java / SignUp.java] --(uses)--> [DatabaseManager.java]**

**|**

**[HomePage.java] <---------> [UserSession.java]**

**|**

**+---> Tabs: Income, Expense, Budget, History**

**|**

**+---> [IncomeExpenseChart.java] (renders monthly charts)**

**Logic & Frontend Code Components**

* **Login.java / SignUp.java**:
  + Built with Java Swing forms (\*.form files).
  + Takes user input and validates via DatabaseManager.java.
  + After successful login → sets UserSession → opens HomePage.
* **HomePage.java**:
  + The dashboard with tabs for:
    - Account Summary
    - Income Entry
    - Expense Entry
    - Savings Target
    - History of Transactions
  + Uses **AbsoluteLayout** for pixel-precise UI component placement.
* **IncomeExpenseChart.java**:
  + Uses **XChart** to generate monthly bar charts.
  + Connected to backend via DatabaseManager to pull income/expense data.
* **DatabaseManager.java**:
  + Handles all JDBC operations:
    - Connecting to MySQL, then

public static ResultSet executeQuery(String query) {

ResultSet resultSet = null;

try {

Statement statement = connection.createStatement();

resultSet = statement.executeQuery(query);

} catch (SQLException e) {

e.printStackTrace();

}

return resultSet;

}

* + - This function is used for performing CRUD operations for users, transactions, accounts and Used across Login, Signup, and HomePage by calling DatabaseManager.prepareStatement()
* **UserSession.java**:
  + Stores the active user ID and session info.
  + Shared globally across components to fetch or store user-specific data.
* **Icons & Images**:
  + Enhance the Swing GUI with icons, backgrounds, and feedback images.