

## **Project Title:** Builder Portfolio Management System

### **Problem Statement:**

The Builders' Association aims to transition from manual, spreadsheet-based project record management to a robust, real-time, console-based Java application. The current manual process leads to data inconsistency, loss, errors, and inefficient communication with clients and stakeholders. To address these issues, you are tasked with designing and developing a **Builder Portfolio Management System** that supports real-time data handling, multiple user roles, and reliable project management functionalities.

---

### **Development Guidelines:**

#### **1. System Overview:**

- Develop a Java console application that functions as a real-time management system for construction projects.
- Ensure the system supports concurrent operations for multiple users, reflecting real-time updates.

#### **2. Core Features:**

- **User Authentication:**
  - Implement secure registration and login for different user roles.
  - Support role-based access control (e.g., Builders, Project Managers).
- **Project Management:**
  - Add new projects with details such as project name, description, start date, end date, assigned client, and project manager.
  - Update existing project details.
  - Delete projects when necessary.
- **Project Status Management:**
  - Assign and update project statuses: **Upcoming, In Progress, Completed.**
- **Associations:**
  - Link projects with clients and project managers.
  - Maintain client details and associate them with projects.
- **User Roles & Permissions:**
  - Restrict certain operations based on user roles.
  - e.g., Only project managers can update project statuses or delete projects; builders can view projects assigned to them.

### **3. Design Considerations:**

- Develop a **modular architecture** using design patterns such as **MVC** (Model-View-Controller) or **Service DAO** pattern.
- Maintain separation of concerns:
  - **Model Layer:** Data classes for Users, Projects, Clients.
  - **DAO Layer:** Data access objects to handle data storage and retrieval.
  - **Service Layer:** Business logic for operations.
  - **View Layer:** Console menus and user interactions.

### **4. Data Storage:**

- For simplicity, use in-memory data structures (like Lists, Maps).
- Data should persist during the application's runtime.
- Optionally, demonstrate saving/loading from files for persistence across sessions.

### **5. Real-Time Aspects:**

- The system should reflect updates immediately upon user actions.
- Support multiple sequential user sessions within the application runtime.

### **6. Development Tips:**

- Use clear, reusable classes and methods.
- Implement input validation and exception handling to ensure robustness.
- Provide a user-friendly console interface with menus and prompts.
- Document your code with comments for clarity.

### **7. Optional Enhancements (if time permits):**

- Implement simple search and filtering options.
- Add feature to generate basic reports (e.g., project status summary).
- Implement login session management.

---

### **Summary:**

Build a **console-based Java application** that acts as a **real-time project portfolio management system** for the Builders' Association. Focus on modular design, role-based access, data integrity, and user-friendly interaction to facilitate efficient construction project tracking and management.