Solid_Act36

✅ Team Task: Job Application Platform (SOLID Principles)

Level: Intermediate – Team project
Goal: Apply all 5 SOLID principles to design and code a mini job application system.

🧠 Scenario:

Your team works for a recruitment platform. You're tasked with creating the logic (backend) that handles:

- Applicants
- Job posts
- Applications
- Recruiters
- Notifications

🛠️ Instructions Per Principle

---

1. SRP – Single Responsibility Principle

👉 Start by creating separate classes that each do one specific job:

Class	Responsibility
Applicant	Holds applicant info
JobPost	Stores job details
Application	Links applicant to a job
ReportGenerator	Generates summary reports

➡ **Your Task:** Make sure each class does **only one thing**.
Don't mix logic (e.g., don’ t calculate stats inside the `Applicant` class).

---

2. OCP – Open/Closed Principle

👉 Make your filtering system **extendable** without changing existing logic.

Example Filters
Filter by job title
Filter by location
Filter by skills/salary (bonus)

➡ **Your Task:**

- Use inheritance or strategy pattern.
- Don’ t use a long if-else block.
- Add at least **two filters** without touching your filter logic later.

---

3. LSP – Liskov Substitution Principle

👉 Treat all job types as `JobPost` safely.

Job Types
FullTimeJob
PartTimeJob
FreelanceJob

➡ **Your Task:**

- Create a base class `JobPost`
- Let each job type inherit from it
- Test that `JobPost` methods work the same on any type

---

4. ISP – Interface Segregation Principle

👉 Only give users the actions they need.

Interface	Who uses it
CanPostJob	Recruiter
CanApplyJob	Applicant

➡ **Your Task:**

- Create small interfaces for specific roles
- Don’ t force `Applicant` to implement job-posting behavior
- Add at least 1 more role if you can (e.g., `Admin` or `HR`)

---

5. DIP – Dependency Inversion Principle

👉 The alert system should work with **any type of notifier**.

Notifiers
EmailNotifier
SMSNotifier
InAppNotifier

➡ **Your Task:**

- Create a `NotificationService` interface
- The system should only depend on that interface, not the notifier type
- Add support for a new notifier without editing the alert class

---

🧩 Bonus Challenge (Optional):

- Add a `MockDatabase` and `RealDatabase` that both follow a common interface
- Support an `Admin` role that can view all reports