Software Requirement Specification (SRS)

1. Project Title

Employee Performance Management System (EPMS)

2. Objective

The goal is to develop a console-based Java application using Object-Oriented Programming concepts such as Encapsulation, Inheritance, Polymorphism, and Abstraction to manage employee performance. No database or CRUD operations should be used.

3. Functional Requirements

- Create an abstract class 'Employee' with private fields: id, name, designation, baseSalary.
- Create subclasses: Manager, Developer, Intern which inherit from Employee.
- Use Polymorphism to override the method calculateBonus() in each subclass.
- Implement an interface 'PerformanceReviewable' with a method reviewPerformance().
- Each subclass should implement the reviewPerformance() method differently.
- Use getters and setters to encapsulate fields in the Employee class.
- Optional: Add a PerformanceCalculator using an interface to apply strategy-based scoring.
- Optional: Add a ReportGenerator to generate employee reports.

4. Non-Functional Requirements

- The system must be written in pure Java.
- The program should be run on the command line (console).
- There should be no use of CRUD operations or external file/database access.
- The structure should follow clean object-oriented design.

5. Class Breakdown

- Abstract Class: Employee
  - Fields: id, name, designation, baseSalary
  - Methods: calculateBonus() [abstract], displayDetails()

- Subclasses: Manager, Developer, Intern
  - Each subclass overrides calculateBonus()
  - Each implements reviewPerformance()

- Interface: PerformanceReviewable
  - Method: reviewPerformance()

- Optional Class: PerformanceCalculator with ScoreStrategy interface

- Optional Class: EmployeeReport for displaying consolidated report

6. Sample Output

ID: 101

Name: Alice

Designation: Developer

Base Salary: 50000

Bonus: 7500

Performance Score: 4

Remarks: Good performer

7. Constraints

- The program should avoid any CRUD or persistence logic.
- Only in-memory class objects should be used.
- Console-based dummy data is acceptable.

8. Deliverables

- Java source code with clear class separation.
- Console-based output showing performance and bonuses.
- Use of all major OOP concepts must be evident.