Assessment Case Study for Advance Java

Module Overview:

- Module 15: Java - Web Technologies in Java (Servlets, JSP, RESTful APIs, etc.)
- Module 16: Java - Software Design Patterns and Project (MVC, Singleton, Factory, etc.)

Case Overview

Scenario:

You are a Java Developer at **TechSolutions** Ltd. working on a system named **TaskMaster** for your client Global HR Services. The application aims to automate key HR functions such as employee management, task tracking, attendance, and payroll via a web-based portal.

Core Functionality Requirements

1. Employee & Task Management

CRUD operations for employee profiles (Add, Update, Delete, View)
Assign tasks to employees and mark task status
Maintain lists of ongoing and completed tasks

2. Attendance Tracking

Admin can mark daily attendance (Present/Absent)
View attendance summary per employee
Calculate leaves and working hours

3. Payroll Management

Monthly payroll calculated as: Basic Salary + Task Completion Bonus – Leave Deductions = Net Salary
Admin can view payroll records per employee

4. Employee Self-Service Portal

- Login for employees to:

View their profile
See attendance and task status
View payroll summary
Request leave or update personal details

Technical Requirements

Component	Technology Used
Frontend	JSP, HTML, CSS, JS
Backend	Java Servlets, JDBC
Design Pattern	MVC, DAO, Singleton, Factory
Database	MySQL
Security	Session management & role restriction
Authentication	Simple login system (Servlet + JDBC)

Java Concepts to Be Tested

Object-Oriented Programming (OOP)
Collections Framework
Servlets & JSP
JDBC and DAO pattern
MVC Architecture
Software Design Patterns (DAO, Singleton, Factory)
Error Handling & Logging (try-catch, log to file)

Architecture & Design

Follows MVC pattern
Use JDBC + DAO layer for all database interactions
Servlets as controllers, JSP for views
Use Session for authentication

Reflective Thinking (Optional)

- How to scale this to support more users?
- How to integrate with third-party systems (e.g., email)?
- Adding performance reviews & project KPIs
- JWT or token-based auth (future upgrade)

Practical Considerations

- Cross-browser compatibility
- Use Git/GitHub for version control
- Host using Apache Tomcat (local or cloud)
- Environment-based config: local vs production
- Keep code modular for reusability and testing