# Payroll System – Project Report

**Project Title:** Payroll System

**Description:**
A system that automates the complex calculation of employee salaries, taxes, and benefits to eliminate manual errors and ensure compliance with financial regulations. The system allows multiple user roles, provides CRUD operations for employee management, and persists data using file storage.

**1. Problem Statement**
Manual salary calculation is prone to errors, especially when employees have different types (full-time, part-time), benefits, and deductions. Organizations require a reliable and automated solution that ensures accuracy and proper record-keeping.

**Solution:**
The Payroll System calculates salaries dynamically, handles benefits, allows CRUD operations for employees, and supports admin/viewer roles to secure sensitive operations.

**2. System Features**
**User Roles:**
Admin: Can add, update, delete, and view employees.
Viewer: Can only view employee records; cannot modify data.

**Employee Management:**
- Add full-time and part-time employees.
- Update employee information.
- Delete employee records.
- Display employee list in a tabular format with calculated salary.

**Salary Calculation:**
Full-time: Base salary + benefits – tax.
Part-time: Hours worked × hourly rate – tax.

**File Storage:**
Employee data stored in employees.txt.
Automatically reads and writes employee data to the file.

**Exception Handling:**
Invalid salary, negative hours, invalid input handled via custom exceptions and input validation.

**3. Packages and Classes**
**Model:**
Employee (abstract), FullTimeEmployee, PartTimeEmployee, UserRole (Enum).

**Repository:**
EmployeeRepository (interface), FileEmployeeRepository.

**Service:**
PayrollService, EmployeeFactory.

**UI:**
PayrollApp (JavaFX GUI with role-based access).

**4. OOP & SOLID Principles Used**

Encapsulation, Abstraction, Inheritance, Polymorphism.

**SOLID:**
SRP, OCP, LSP, ISP, DIP.

### 5. Screenshots

(Admin view, Viewer view, Employee salary list – insert screenshots here)

### 6. Class Diagram (UML)

Employee (abstract) → FullTimeEmployee, PartTimeEmployee
PayrollService → EmployeeRepository
FileEmployeeRepository implements EmployeeRepository
PayrollApp → PayrollService
EmployeeFactory → Employee

### 7. System Execution Steps

1. Run PayrollApp.java in a JavaFX-compatible IDE.
2. Login as Admin or Viewer.
3. Manage or view employees based on role.
4. Salaries are automatically calculated.
5. Data persists in employees.txt.

### 8. Conclusion

The Payroll System provides a reliable, secure, and efficient solution for managing employee salaries while applying OOP and SOLID principles.