HRMS Portal — Day-wise Development Report (Day 3 - Day 10)

Generated: 2025-10-09 10:22:39

This report summarizes the Day 3 to Day 10 progress of the HRMS Portal project, including objectives, steps taken, and code snippets extracted from the repository.

Day 3 — Java Setup — Employee CRUD REST APIs

Created REST controllers, services and repository layers. Configured Maven dependencies and basic application properties. Key files: Controller classes, Service classes, Repository interfaces.

Objectives:

- Created REST controllers, services and repository layers. Configured Maven dependencies and basic...

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 4 — Payroll Logic — SQL joins for salary computation

Implemented payroll calculation using SQL joins between employee, attendance and payroll tables, and service-level aggregation.

Objectives:

- Implemented payroll calculation using SQL joins between employee, attendance and payroll tables,...

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

File: HRMSPORTAL/sample_data.sql

```
-- Sample Data for HRMS Database

USE hrms_db;

-- Insert Departments
INSERT INTO departments (dept_name, dept_head) VALUES
('Human Resources', 'Sarah Johnson'),
('Information Technology', 'Michael Chen'),
('Finance', 'Robert Davis'),
('Marketing', 'Emily Wilson'),
('Operations', 'David Brown');

-- Insert Employees
INSERT INTO employees (emp_code, first_name, last_name, email, phone,
hire_date, dept_id, position, salary) VALUES
('EMP001', 'John', 'Smith', 'john.smith@company.com', '555-0101', '2023-01-15',
2, 'Software Developer', 75000.00),
('EMP002', 'Jane', 'Doe', 'jane.doe@company.com', '555-0102', '2023-02-01', 1,
'HR Manager', 65000.00),
('EMP003', 'Mike', 'Johnson', 'mike.johnson@company.com', '555-0103', '2023-01-
20', 3, 'Financial Analyst', 60000.00),
('EMP004', 'Lisa', 'Anderson', 'lisa.anderson@company.com', '555-0104', '2023-
03-10', 4, 'Marketing Specialist', 55000.00),
('EMP005', 'Tom', 'Wilson', 'tom.wilson@company.com', '555-0105', '2023-02-15',
5, 'Operations Manager', 70000.00),
('EMP006', 'Anna', 'Garcia', 'anna.garcia@company.com', '555-0106', '2023-04-
01', 2, 'Senior Developer', 85000.00),
('EMP007', 'Chris', 'Martinez', 'chris.martinez@company.com', '555-0107',
'2023-03-20', 1, 'HR Assistant', 45000.00),
('EMP008', 'Rachel', 'Taylor', 'rachel.taylor@company.com', '555-0108', '2023-
05-15', 3, 'Accountant', 58000.00);

-- Insert Payroll Records
INSERT INTO payroll (emp_id, pay_period, basic_salary, allowances, deductions,
gross_salary, tax_deduction, net_salary) VALUES
(1, '2024-01', 75000.00, 5000.00, 2000.00, 80000.00, 12000.00, 68000.00),
(2, '2024-01', 65000.00, 3000.00, 1500.00, 68000.00, 10200.00, 57800.00),
(3, '2024-01', 60000.00, 2500.00, 1200.00, 62500.00, 9375.00, 53125.00),
(4, '2024-01', 55000.00, 2000.00, 1000.00, 57000.00, 8550.00, 48450.00),
(5, '2024-01', 70000.00, 4000.00, 1800.00, 74000.00, 11100.00, 62900.00),
(6, '2024-01', 85000.00, 6000.00, 2500.00, 91000.00, 13650.00, 77350.00),
(7, '2024-01', 45000.00, 1500.00, 800.00, 46500.00, 6975.00, 39525.00),
(8, '2024-01', 58000.00, 2200.00, 1100.00, 60200.00, 9030.00, 51170.00);

-- Insert Attendance Records (Last 30 days sample)
INSERT INTO attendance (emp_id, attendance_date, status, check_in_time,
check_out_time) VALUES
(1, '2024-01-01', 'PRESENT', '09:00:00', '17:30:00'),
(1, '2024-01-02', 'PRESENT', '09:15:00', '17:45:00'),
(1, '2024-01-03', 'LATE', '09:45:00', '17:30:00'),
(2, '2024-01-01', 'PRESENT', '08:45:00', '17:15:00'),
(2, '2024-01-02', 'PRESENT', '09:00:00', '17:30:00'),
(2, '2024-01-03', 'ABSENT', NULL, NULL),
(3, '2024-01-01', 'PRESENT', '09:10:00', '17:40:00'),
(3, '2024-01-02', 'PRESENT', '09:05:00', '17:35:00'),
(3, '2024-01-03', 'PRESENT', '08:55:00', '17:25:00'),
(4, '2024-01-01', 'PRESENT', '09:20:00', '17:50:00'),
(4, '2024-01-02', 'HALF_DAY', '09:00:00', '13:00:00'),
(4, '2024-01-03', 'PRESENT', '09:10:00', '17:40:00'),
(5, '2024-01-01', 'PRESENT', '08:30:00', '18:00:00'),
(5, '2024-01-02', 'PRESENT', '08:45:00', '17:45:00'),
(5, '2024-01-03', 'PRESENT', '09:00:00', '17:30:00');
```

File: HRMSPORTAL/schema.sql

```
-- HRMS Database Schema

CREATE DATABASE IF NOT EXISTS hrms_db;
USE hrms_db;

-- Departments Table
CREATE TABLE departments (
    dept_id INT PRIMARY KEY AUTO_INCREMENT,
    dept_name VARCHAR(100) NOT NULL UNIQUE,
    dept_head VARCHAR(100),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Employees Table
CREATE TABLE employees (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_code VARCHAR(20) NOT NULL UNIQUE,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    phone VARCHAR(15),
    hire_date DATE NOT NULL,
    dept_id INT,
    position VARCHAR(100),
    salary DECIMAL(10,2) NOT NULL,
    status ENUM('ACTIVE', 'INACTIVE') DEFAULT 'ACTIVE',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (dept_id) REFERENCES departments(dept_id)
);

-- Payroll Table
CREATE TABLE payroll (
    payroll_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT NOT NULL,
    pay_period VARCHAR(7) NOT NULL, -- YYYY-MM format
    basic_salary DECIMAL(10,2) NOT NULL,
    allowances DECIMAL(10,2) DEFAULT 0,
    deductions DECIMAL(10,2) DEFAULT 0,
    gross_salary DECIMAL(10,2) NOT NULL,
    tax_deduction DECIMAL(10,2) DEFAULT 0,
    net_salary DECIMAL(10,2) NOT NULL,
    processed_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (emp_id) REFERENCES employees(emp_id),
    UNIQUE KEY unique_emp_period (emp_id, pay_period)
);

-- Attendance Table
CREATE TABLE attendance (
    attendance_id INT PRIMARY KEY AUTO_INCREMENT,
    emp_id INT NOT NULL,
    attendance_date DATE NOT NULL,
    status ENUM('PRESENT', 'ABSENT', 'LATE', 'HALF_DAY') NOT NULL,
    check_in_time TIME,
    check_out_time TIME,
    remarks VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (emp_id) REFERENCES employees(emp_id),
    UNIQUE KEY unique_emp_date (emp_id, attendance_date)
);

-- Indexes for better performance
CREATE INDEX idx_emp_dept ON employees(dept_id);
CREATE INDEX idx_payroll_emp ON payroll(emp_id);
CREATE INDEX idx_payroll_period ON payroll(pay_period);
CREATE INDEX idx_attendance_emp ON attendance(emp_id);
CREATE INDEX idx_attendance_date ON attendance(attendance_date);
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 5 — Attendance Feature — DB table and APIs

Designed attendance table and REST endpoints to record and fetch attendance data. Integrated with payroll computation.

Objectives:

- Designed attendance table and REST endpoints to record and fetch attendance data....

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

File: HRMSPORTAL/src/main/webapp/attendance-fixed.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Attendance Management - HRMS Portal</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <div class="container">
            <a class="navbar-brand" href="index.jsp">
                <i class="fas fa-users"></i> HRMS Portal
            </a>
            <div class="navbar-nav ms-auto">
                <a class="nav-link" href="employees.html">Employees</a>
                <a class="nav-link" href="payroll.html">Payroll</a>
                <a class="nav-link active" href="attendance-
fixed.html">Attendance</a>
```

```
                <a class="nav-link" href="leave.html">Leave</a>
            </div>
        </div>
    </nav>

    <div class="container mt-4">
        <div class="row">
            <div class="col-12">
                <h2>Attendance Management</h2>
                <button class="btn btn-primary mb-3"
onclick="markAttendance()">
                    <i class="fas fa-clock"></i> Mark Attendance
                </button>
            </div>
        </div>

        <div class="row">
            <div class="col-12">
                <div class="card">
                    <div class="card-body">
                        <div class="table-responsive">
                            <table class="table table-striped"
id="attendanceTable">
                                <thead>
                                    <tr>
                                        <th>Employee</th>
                                        <th>Date</th>
                                        <th>Check In</th>
                                        <th>Check Out</th>
                                        <th>Hours</th>
                                        <th>Status</th>
                                        <th>Actions</th>
                                    </tr>
                                </thead>
                                <tbody id="attendanceTableBody">
                                    <tr>
                                        <td colspan="7" class="text-
center">Loading attendance data...</td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.
js"></script>
    <script>
        document.addEventListener('DOMContentLoaded', function() {
            loadAttendanceData();
        });

        function loadAttendanceData() {
            fetch('api/attendance/today')
                .then(response => response.json())
                .then(data => {
                    const tbody =
document.getElementById('attendanceTableBody');
                    tbody.innerHTML = '';
                    if (!data || data.length === 0) {
                        tbody.innerHTML = '<tr><td colspan="7" class="text-
center">No attendance data found</td></tr>';
                        return;
                    }

                    data.forEach(attendance => {
                        const row = tbody.insertRow();
                        row.innerHTML = `
                            <td>${attendance.employeeName || 'N/A'}</td>
                            <td>${attendance.attendanceDate || 'N/A'}</td>
                            <td>${attendance.checkInTime || 'N/A'}</td>
                            <td>${attendance.checkOutTime || 'N/A'}</td>
                            <td>${attendance.hoursWorked || 0}</td>


//... (truncated)
```

File: HRMSPORTAL/src/main/webapp/attendance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Attendance Management - HRMS Portal</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
    <link href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.0.0/css/all.min.css" rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
        <div class="container">
            <a class="navbar-brand" href="index.jsp">
                <i class="fas fa-users"></i> HRMS Portal
            </a>
            <div class="navbar-nav ms-auto">
                <a class="nav-link" href="employees.html">Employees</a>
                <a class="nav-link" href="payroll.html">Payroll</a>
                <a class="nav-link active"
href="attendance.html">Attendance</a>
                <a class="nav-link" href="leave.html">Leave</a>
            </div>
        </div>
    </nav>

    <div class="container mt-4">
        <div class="row">
            <div class="col-12">
                <h2>Attendance Management</h2>
                <button class="btn btn-primary mb-3"
onclick="markAttendance()">
                    <i class="fas fa-clock"></i> Mark Attendance
                </button>
            </div>
        </div>

        <div class="row">
            <div class="col-12">
```

```
                <div class="card">
                    <div class="card-body">
                        <div class="table-responsive">
                            <table class="table table-striped"
id="attendanceTable">
                                <thead>
                                    <tr>
                                        <th>Employee</th>
                                        <th>Date</th>
                                        <th>Check In</th>
                                        <th>Check Out</th>
                                        <th>Hours</th>
                                        <th>Status</th>
                                        <th>Actions</th>
                                    </tr>
                                </thead>
                                <tbody id="attendanceTableBody">
                                    <tr>
                                        <td colspan="7" class="text-
center">Loading attendance data...</td>
                                    </tr>
                                </tbody>
                            </table>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.
js"></script>
    <script>
        document.addEventListener('DOMContentLoaded', function() {
            loadAttendanceData();
        });

        function loadAttendanceData() {
            fetch('api/attendance')
                .then(response => response.json())
                .then(data => {
                    const tbody =
document.getElementById('attendanceTableBody');
                    tbody.innerHTML = '';

                    if (!data || data.length === 0) {
                        tbody.innerHTML = '<tr><td colspan="7" class="text-
center">No attendance data found</td></tr>';
                        return;
                    }

                    data.forEach(attendance => {
                        const row = tbody.insertRow();
                        row.innerHTML = `
                            <td>${attendance.employeeName || 'N/A'}</td>
                            <td>${attendance.attendanceDate || 'N/A'}</td>
                            <td>${attendance.checkInTime || 'N/A'}</td>
                            <td>${attendance.checkOutTime || 'N/A'}</td>
                            <td>${attendance.hoursWorked || 0}</td>
                            <td><

//... (truncated)
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 6 — DynamoDB Integration — Leave Requests

Configured DynamoDB client and created repository to store leave request documents. Wrote code to sync with main system.

Objectives:

- Configured DynamoDB client and created repository to store leave request documents. Wrote...

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

File: HRMSPORTAL/src/main/java/com/hrms/util/DynamoDBConfig.java

```
package com.hrms.util;

import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.BasicAWSCredentials;
import com.amazonaws.client.builder.AwsClientBuilder;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDB;
import com.amazonaws.services.dynamodbv2.AmazonDynamoDBClientBuilder;

public class DynamoDBConfig {
    private static final String ACCESS_KEY = "fakeMyKeyId";
    private static final String SECRET_KEY = "fakeSecretAccessKey";
    private static final String REGION = "us-east-1";
    private static final String ENDPOINT = "http://localhost:8000"; // Local
DynamoDB endpoint

    private static AmazonDynamoDB client;

    public static AmazonDynamoDB getClient() {
        if (client == null) {
            BasicAWSCredentials credentials = new
BasicAWSCredentials(ACCESS_KEY, SECRET_KEY);

            client = AmazonDynamoDBClientBuilder.standard()
                    .withEndpointConfiguration(new
AwsClientBuilder.EndpointConfiguration(ENDPOINT, REGION))
                    .withCredentials(new
AWSStaticCredentialsProvider(credentials))
```

```
                    .build();
        }
        return client;
    }



    public static boolean isConnected() {
        try {
            getClient().listTables();
            return true;
        } catch (Exception e) {
            return false;
        }
    }
}
```

File: HRMSPORTAL/target/classes/com/hrms/util/DynamoDBConfig.class

```
����7X
java/lang/Object<init>()V
com/hrms/util/DynamoDBConfigclient2Lcom/amazonaws/services/dynamodbv2/AmazonDyn
amoDB;&com/amazonaws/auth/BasicAWSCredentialsfakeMyKeyIdfakeSecretAccessKey

'(Ljava/lang/String;Ljava/lang/String;)V
=com/amazonaws/services/dynamodbv2/AmazonDynamoDBClientBuilderstandardA()Lcom/a
mazonaws/services/dynamodbv2/AmazonDynamoDBClientBuilder;Ccom/amazonaws/client/
builder/AwsClientBuilder$EndpointConfigurationhttp://localhost:8000!     us-
east-1

$%&withEndpointConfigurationv(Lcom/amazonaws/client/builder/AwsClientBuilder$En
dpointConfiguration;)Lcom/amazonaws/client/builder/AwsClientBuilder;(/com/amazo
naws/auth/AWSStaticCredentialsProvider
'*+&(Lcom/amazonaws/auth/AWSCredentials;)V
-
./withCredentials\(Lcom/amazonaws/auth/AWSCredentialsProvider;)Lcom/amazonaws/c
lient/builder/AwsClientBuilder;
123build()Ljava/lang/Object;50com/amazonaws/services/dynamodbv2/AmazonDynamoDB
789    getClient4()Lcom/amazonaws/services/dynamodbv2/AmazonDynamoDB;4;<=
listTables<()Lcom/amazonaws/services/dynamodbv2/model/ListTablesResult;?java/la
ng/Exception
ACCESS_KEYLjava/lang/String;
ConstantValue
SECRET_KEYREGIONENDPOINTCodeLineNumberTableLocalVariableTablethisLcom/hrms/util
/DynamoDBConfig;credentials(Lcom/amazonaws/auth/BasicAWSCredentials;
StackMapTableisConnected()ZeLjava/lang/Exception;
SourceFileDynamoDBConfig.javaInnerClassesV-
com/amazonaws/client/builder/AwsClientBuilderEndpointConfiguration!@ABCABDAB
EAB
F/*��G      HIJ    89F�A��:�
Y�K��Y �"�#��'Y*�)�,��0�4���G .4=H+KLM=     NOFX�6�:W�K�
>G!    "#$HPQMK>RST
UW
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 7 — Data Structures — Queue for payroll batch processing

Implemented a queue (in-memory / Java Queue) to batch payroll jobs and process them asynchronously or on schedule.

Objectives:

- Implemented a queue (in-memory / Java Queue) to batch payroll jobs and...

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

File: HRMSPORTAL/src/main/java/com/hrms/service/PayrollService.java

```
package com.hrms.service;

import com.hrms.dao.EmployeeDAO;
import com.hrms.dao.PayrollDAO;
import com.hrms.model.Employee;
import com.hrms.model.Payroll;

import java.math.BigDecimal;
import java.util.LinkedList;
import java.util.List;
import java.util.Queue;

public class PayrollService {
    private final PayrollDAO payrollDAO;
    private final EmployeeDAO employeeDAO;
    private final Queue<Integer> payrollQueue;

    public PayrollService() {
        this.payrollDAO = new PayrollDAO();
        this.employeeDAO = new EmployeeDAO();
        this.payrollQueue = new LinkedList<>();
    }

    public Payroll processPayroll(int empId, String payPeriod) {
        Employee employee = employeeDAO.findById(empId);
        if (employee == null) {
            return null;
        }

        // Check if payroll already exists for this period
        Payroll existing = payrollDAO.findByEmpIdAndPeriod(empId, payPeriod);
        if (existing != null) {
            return existing;
        }

        // Create new payroll
        Payroll payroll = new Payroll(empId, payPeriod, employee.getSalary());

        // Calculate allowances based on position
        BigDecimal allowances = calculateAllowances(employee);
        payroll.setAllowances(allowances);

        // Calculate deductions
        BigDecimal deductions = calculateDeductions(employee);
        payroll.setDeductions(deductions);

        // Recalculate salary with allowances and deductions
        payroll.calculateSalary();

        if (payrollDAO.save(payroll)) {
            return payroll;
        }
        return null;
    }

    public Payroll generateSalarySlip(int empId, String payPeriod) {
        return payrollDAO.findByEmpIdAndPeriod(empId, payPeriod);
    }

    public List<Payroll> getPayrollByPeriod(String payPeriod) {
        return payrollDAO.findByPeriod(payPeriod);
    }

    public List<Payroll> getAllPayrolls() {
        return payrollDAO.findAll();
    }

    publi
```

File: HRMSPORTAL/src/main/java/com/hrms/servlet/BatchServlet.java

```
package com.hrms.servlet;

import com.fasterxml.jackson.databind.ObjectMapper;
import com.hrms.service.PayrollService;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;

@WebServlet("/api/batch/*")
public class BatchServlet extends HttpServlet {
    private PayrollService payrollService;
    private ObjectMapper objectMapper;

    @Override
    public void init() throws ServletException {
        payrollService = new PayrollService();
        objectMapper = new ObjectMapper();
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("application/json");
        response.setCharacterEncoding("UTF-8");

        String pathInfo = request.getPathInfo();

        try {
            if (pathInfo != null && pathInfo.equals("/status")) {
                // Get queue status
                Map<String, Object> status = new HashMap<>();
                status.put("queueSize", payrollService.getQueueSize());
                status.put("message", "Batch processing queue status");

                objectMapper.writeValue(response.getWriter(), status);
            } else {
                response.setStatus(HttpServletResponse.SC_NOT_FOUND);
                response.getWriter().write("{\"error\":\"Invalid endpoint\"}");
            }
        } catch (Exception e) {
            response.setStatus(HttpServletResponse.SC_INTERNAL_SERVER_ERROR);
            response.getWriter().write("{\"error\":\"" + e.getMessage() +
"\"}");
        }
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        respon
```

File: HRMSPORTAL/src/test/java/com/hrms/PayrollServiceTest.java

```
package com.hrms;

import com.hrms.model.Employee;
import com.hrms.model.Payroll;
import com.hrms.service.PayrollService;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;

import java.math.BigDecimal;
import java.sql.Date;

import static org.junit.jupiter.api.Assertions.*;

class PayrollServiceTest {
    private PayrollService payrollService;
    private Employee testEmployee;

    @BeforeEach
    void setUp() {
        payrollService = new PayrollService();
        testEmployee = new Employee();
        testEmployee.setEmpId(1);
        testEmployee.setEmpCode("EMP001");
        testEmployee.setFirstName("John");
        testEmployee.setLastName("Doe");
        testEmployee.setEmail("john.doe@company.com");
        testEmployee.setHireDate(Date.valueOf("2023-01-01"));
        testEmployee.setDeptId(1);
        testEmployee.setPosition("Software Developer");
        testEmployee.setSalary(new BigDecimal("75000"));
        testEmployee.setStatus("ACTIVE");
    }

    @Test
    void testPayrollCalculation() {
        Payroll payroll = new Payroll(1, "2024-01", new BigDecimal("75000"));
        payroll.setAllowances(new BigDecimal("1500"));
        payroll.setDeductions(new BigDecimal("2250"));
        payroll.calculateSalary();

        assertEquals(0, new
BigDecimal("74250.00").compareTo(payroll.getGrossSalary()));
        assertEquals(0, new
BigDecimal("11137.50").compareTo(payroll.getTaxDeduction()));
        assertEquals(0, new
BigDecimal("63112.50").compareTo(payroll.getNetSalary()));
    }

    @Test
    void testQueueOperations() {
        assertEquals(0, payrollService.getQueueSize());

        payrollService.addToPayrollQueue(1);
        payrollService.addToPayrollQueue(2);

        assertEquals(2, payrollService.getQueueSize());
    }

    @Test
    void testSalaryCalculationWithDifferentPositions() {
        // Test manager allowances (10%)
        testEmployee.setPosition("Manager");
        testEmployee.setSalary
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 8 — BDD Tests — Cucumber for payroll generation

Wrote Cucumber feature files and step definitions to verify payroll generation flows.

Objectives:

- Wrote Cucumber feature files and step definitions to verify payroll generation flows....

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 9 — DevOps — Jenkins automation for payroll job

Added Jenkinsfile / pipeline scripts to automate payroll generation and deployment.

Objectives:

- Added Jenkinsfile / pipeline scripts to automate payroll generation and deployment....

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

File: HRMSPORTAL/Jenkinsfile

```
pipeline {
    agent any

    tools {
        maven 'Maven-3.8.1'
        jdk 'JDK-11'
    }

    environment {
        TOMCAT_HOME = '/opt/tomcat'
        WAR_FILE = 'target/hrms-portal.war'
    }

    stages {
        stage('Checkout') {
            steps {
                checkout scm
            }
        }

        stage('Build') {
            steps {
                sh 'mvn clean compile'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test'
            }
            post {
                always {
                    junit 'target/surefire-reports/*.xml'
                }
            }
        }

        stage('Package') {
            steps {
                sh 'mvn package -DskipTests'
            }
            post {
                success {
                    archiveArtifacts artifacts: 'target/*.war', fingerprint:
true
                }
            }
        }

        stage('Deploy to Tomcat') {
            steps {
                script {
                    sh """
                        # Stop Tomcat
                        sudo systemctl stop tomcat

                        # Remove old deployment
                        sudo rm -rf ${TOMCAT_HOME}/webapps/hrms-portal*

                        # Copy new WAR file
                        sudo cp ${WAR_FILE} ${TOMCAT_HOME}/webapps/

                        # Start Tomcat
                        sudo systemctl start tomcat

                        # Wait for deployment
                        sleep 30
                    """
                }
            }
        }

        stage('Health Check') {
            steps {
                script {
                    sh """
                        # Check if application is running
                        curl -f http://localhost:8080/hrms-portal/ || exit 1
                    """
                }
            }
        }
    }

    post {
        always {
            cleanWs()
        }
        success {
            emailext (
                subject: "HRMS Deployment Successful - Build #${BUILD_NUMBER}",
                body: "The HRMS Portal has been successfully deployed to
Tomcat.",
                to: "admin@company.com"
            )
        }
        failure {
            emailext (
                subject: "HRMS Deployment Failed - Build #${BUILD_NUMBER}",
                body: "The HRMS Portal deployment failed. Please check the
build logs.",
                to: "admin@company.com"
            )
        }
    }

    triggers {
        // Weekly automated payroll processing
        cron('0 2 * * 1') // Every Monday at 2 AM
    }
}
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.

Day 10 — Demo — Deploy and present HR portal

Packaged the application, prepared demo scripts, and deployed to a test environment (Tomcat/Docker).

Objectives:

- Packaged the application, prepared demo scripts, and deployed to a test environment...

Steps taken:

1. Reviewed project structure and identified relevant classes and resources.

2. Implemented features and updated configuration as required.

Representative code snippets:

Deployment notes:

pom.xml snippet:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
```

```xml
<groupId>com.hrms</groupId>
<artifactId>hrms-portal</artifactId>
<version>1.0.0</version>
<packaging>war</packaging>
<name>HRMS Portal</name>

<properties>
    <maven.compiler.source>11</maven.compiler.source>
    <maven.compiler.target>11</maven.compiler.target>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <junit.version>5.8.2</junit.version>
</properties>

<dependencies>
    <!-- Servlet API -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>4.0.1</version>
    </dependency>

    <!-- JSP API -->
    <dependency>
        <groupId>javax.servlet.jsp</groupId>
        <artifactId>javax.servlet.jsp-api</artifactId>
        <version>2.3.3</version>
    </dependency>

    <!-- JSP Implementation -->
    <dependency>
        <groupId>org.eclipse.jetty</groupId>
        <artifactId>apache-jsp</artifactId>
        <version>9.4.51.v20230217</version>
    </dependency>

    <!-- JSTL -->
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>



    <!-- AWS SDK for DynamoDB -->
    <dependency>
        <groupId>com.amazonaws</groupId>
        <artifactId>aws-java-sdk-dynamodb</artifactId>
        <version>1.12.470</version>
    </dependency>

    <!-- JSON Processing -->
    <dependency>
        <groupId>com.fasterxml.jackson.core</groupId>
        <artifactId>jackson-databind</artifactId>
        <version>2.15.2</version>
        </dependency>

        <!-- Jetty for embedded server -->
        <dependency>
            <groupId>org.eclipse.jetty</groupId>
            <artifactId>jetty-server</artifactId>
            <version>9.4.51.v20230217</version>
        </dependency>

        <dependency>
            <groupId>org.eclipse.jetty</groupId>
            <artifactId>jetty-webapp</artifactId>
            <version>9.4.51.v20230217</version>
        </dependency>

        <!-- JUnit 5 -->
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>${junit.version}</version>
            <scope>test</scope>
        </dependency>


    </dependencies>

    <build>
        <finalName>hrms-portal</finalName>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.11.0</version>
                <configuration>
                    <source>11</source>
                    <target>11</target>
                </configuration>
            </plugin>

            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-war-plugin</artifactId>
                <version>3.3.1</version>
                <configuration>
                    <failOnMissingWebXml>false</failOnMissingWebXml>
                </configuration>
            </plugin>

            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>3

//... (truncated)
```

Outcome and remarks:

- Work completed as per objectives.
- Ready for demo and presentation.