Mini Project Documentation

Project Title: Shift Allocation and Overtime Forecasting System

Submitted by: Haniya Mumtaj T

1. Introduction

The Shift and Overtime Optimization System is designed to efficiently allocate employee shifts and forecast overtime using machine learning techniques. The system aims to minimize overtime costs, ensure fair shift rotations, and respect employee availability and leave constraints.

2. Module Overview

The system comprises the following modules:

1. Data Collection and Preprocessing

2. Rule-Based and AI-Based Shift Allocation

3. Overtime Forecasting using ML

4. Database Integration

5. API Endpoints for Shift and OT Management

6. Optional Frontend Dashboard

3. Architecture Overview

The backend is developed using Python with optional FastAPI for API endpoints. MySQL is used for storing employee and attendance data. Machine learning models are built using scikit-learn and pandas. Streamlit or JavaScript can be used for frontend dashboards if the project is converted to major project.

4. Module-Wise Design

4.1 Data Collection and Preprocessing

Historical data includes fields like Emp_ID, Date, Hours_Worked, Shift, Department, Role, Leaves, Task_Completed, OT_Hours. Categorical features are encoded, missing values filled, and numerical features normalized.

4.2 Shift Allocation

Rule-Based Allocation assigns shifts based on role, department, and leave status. AI-Based Allocation uses classification models to predict suitable shifts.

4.3 Overtime Forecasting

RandomForestRegressor is used to predict OT_Hours based on features like Hours_Worked, Day_of_Week, Department, Shift_Code, Leaves.

4.4 Database Integration

MySQL schema includes 'employees' and 'attendance' tables with foreign key relationships.

4.5 API Endpoints

Endpoints include /allocate_shift, /predict_ot, /update_attendance for managing shift and overtime operations.

4.6 Frontend Dashboard (Future Scope)

Streamlit can be used to visualize predicted overtime and shift allocations.

5. Deployment Strategy

The system can be deployed on a local server or cloud environment. A daily cron job schedules shift allocation and overtime forecasting tasks.

6. Database Design

```
CREATE TABLE employees (
  emp_id INT PRIMARY KEY,
  name VARCHAR(50),
  role VARCHAR(50),
  department VARCHAR(50),
  current_shift VARCHAR(20),
  leaves INT
);
CREATE TABLE attendance (
  emp_id INT,
  date DATE,
  hours_worked FLOAT,
  ot_hours FLOAT,
  FOREIGN KEY(emp_id) REFERENCES employees(emp_id)
);
```

7. User Interface Design

The optional frontend built with Streamlit allows administrators to view predicted overtime per employee and manage shift allocations interactively.

8. Non-Functional Requirements

1. Scalability to handle large employee datasets.

2. Security for employee data.

3. Maintainability through modular code structure.

4. Performance optimization for real-time predictions.

9. Assumptions and Constraints

1. Historical data is available and accurate.

2. Employees have fixed roles and departments.

3. Leave data is updated regularly.

4. Shift allocation respects employee availability.