Employee Management System (EMS) Using MVC Design Pattern

The Employee Management System (EMS) is designed to automate key HR functions such as employee data management, salary adjustments, promotions. The system is built using the MVC (Model-View-Controller) design pattern, which separates concerns by organizing the system into three distinct components: Model, View, and Controller. This helps in maintaining a clean, modular, and scalable system.

Overview of EMS Using MVC Design

1. Model

The Model is the core component of the system that deals with data logic, including interacting with the database:

- **Core Classes**:
  - **Employee**: Base class representing an employee, with attributes like id, name, salary, address, and role.
  - **Address**: This class representing an employee address, with attributes like street, city, state and zipcode.
  - **EmployeeManager**: Handles storing, retrieving, and deleting employee records.
  - **FullTimeEmployee** and **PartTimeEmployee**: Extend the Employee class to represent specialized employee types.
  - **Payroll**: Calculates employee salaries.
  - **PayrollReport**: Generates payroll summaries.
- **Responsibilities**:
  - Managing and processing employee data.
  - Implementing promotion and salary update logic.
  - Storing and retrieving data from the database.

---

2. View

The **View** presents data to the user and provides a dynamic interface to interact with the system. It listens to updates from the Controller and displays changes in real-time.

- **Core Classes**:
  - **EmployeeView**: Displays employee details and promotion updates.
  - **FullTimeEmployeeView** and **PartTimeEmployeeView**: Show details specific to full-time or part-time employees.

    - **AuthenticationView:** Display a massage if employee exist or not in database.
    - **PayrollView**: Presents salary details.
    - **ReportView**: Displays reports on employees and payroll.
- **Responsibilities**:
    - Presenting data in a user-friendly format.
    - Updating dynamically to reflect changes in the system.

---

3. Controller

The **Controller** acts as an intermediary between the Model and the View. It processes user input, updates the Model, and ensures the View reflects the changes.

- **Core Classes**:
    - **EmployeeController**: Handles employee operations such as adding, promoting, or updating employees.
    - **FullTimeEmployeeController** and **PartTimeEmployeeController**: Manage operations for specific employee types.
    - **PayrollController**: Calculates and displays employee salaries.
    - **ReportController**: Generates and displays employee and payroll reports.
- **Responsibilities**:
    - Receiving and processing user input.
    - Invoking appropriate methods on the Model for data changes.
    - Updating the View to display changes.