EduManage

Abstract:

The **Institute Management System** is an all-in-one platform designed to streamline the management of educational institutions. It includes features for managing trainers, students, attendance, batch allocation, and task assignments. This system provides an admin interface for managing users and operations, while trainers can allocate tasks to students, track their attendance, and manage their batches. The project is developed using **Spring Boot** for backend services and **Hibernate** for database management, ensuring a scalable and efficient platform.

Objective:

1. **Admin Login and User Management**:
    - Provide secure login for administrators to manage users, view reports, and monitor system activities.
    - Admins can create, modify, and delete trainer, student, and batch records.
2. **Trainer Dashboard**:
    - Trainers will have their own dashboards to view their assigned batches, track student attendance, and allocate tasks.
    - Trainers can mark attendance and assign tasks to students based on the course structure.
3. **Student Attendance Management**:
    - Record and track attendance of students for each batch session.
    - Generate reports on student attendance, with options for monthly, weekly, or individual session tracking.
4. **Batch Allocation**:
    - Admins can allocate students to different batches based on their course selection, availability, and trainer assignments.
    - View all batch details, including the trainer, scheduled timings, and student roster.
5. **Task Assignment**:
    - Trainers can assign specific tasks (such as assignments, projects, or exams) to students within their batches.
    - Students can submit their tasks, and trainers can grade them within the system.

Operating System Requirements:

- **Server**: Linux (Ubuntu, CentOS) or Windows Server.
- **Client**: Windows 10/11 or macOS for accessing the system through a web interface.

Hardware Requirements:

- **Server**:
  - Processor: Quad-core or higher (Intel i5/i7 or AMD Ryzen).
  - RAM: Minimum 8 GB.
  - Storage: SSD with 100 GB or more.
  - Network: Stable internet connection (1 Gbps recommended).
- **Client**:
  - Processor: Dual-core or higher.
  - RAM: Minimum 4 GB.
  - Storage: 500 MB of free space.

Software Requirements:

- **Backend**:
  - Spring Boot 3.0 or higher.
  - Hibernate ORM 6.0 or higher.
  - JDK 17 or later.
- **Frontend (Optional)**:
  - Angular/React for building the user interface.
- **Database**:
  - PostgreSQL/MySQL for storing data related to students, trainers, attendance, and tasks.
- **Tools**:
  - IDE: IntelliJ IDEA or Eclipse.
  - Build Tool: Maven or Gradle.
  - API Documentation: Swagger/OpenAPI.
  - Testing: JUnit, Mockito.

Modules:

1. **Admin Module**:
    - Admin login, dashboard, user management (trainer, student, batch).
    - Overview of attendance, tasks, and batches.
2. **Trainer Module**:
    - Trainer login, dashboard with batch and student details.
    - Attendance tracking, task allocation, and grading.
3. **Student Module**:
    - Student login, viewing assigned batches, attendance status, and assigned tasks.
    - Task submission and feedback system.
4. **Batch Allocation Module**:
    - Admin assigns students to batches, manages batch schedules and trainer assignments.
    - Trainers and students can view batch details.
5. **Attendance and Task Management Module**:
    - Real-time attendance recording for students.
    - Task assignment and grading system.

Conclusion:

The **Institute Management System** is designed to automate and simplify the management of educational institutions, improving efficiency in student-trainer interactions. By leveraging **Spring Boot** and **Hibernate**, this platform offers a secure, scalable, and user-friendly solution for managing trainers, students, batches, attendance, and task allocation. It aims to reduce administrative overhead, enhance transparency, and provide both trainers and students with a seamless experience in managing tasks and attendance. The system is highly customizable, allowing future integrations and enhancements.