Layered Architecture Design for a School Management System

When dividing the requirements of a school management system into a layered architecture pattern using JavaScript, PHP, and MySQL, we follow the classic **three-tier architecture**: **Presentation Layer**, **Business Logic Layer**, and **Data Access Layer**.

Layered Architecture Design for a School Management System

---

1. Presentation Layer

- **Role**: This is the front-end layer responsible for interacting with the user (students, teachers, administrators, and parents).
- **Technologies**:
  - **JavaScript** (for dynamic behavior, client-side validation, etc.)
  - **HTML/CSS** (for UI design)
  - **AJAX** (for asynchronous communication between client and server)
- **Functionalities**:
  - **Login/Authentication**: Forms for teachers, students, and admin to log in.
  - **Dashboard**: Display different dashboards for teachers, students, and admins based on their roles.
  - **Class Schedule**: Display class timetables for students and teachers.
  - **Assignment Submission**: Students submit assignments through the interface.
  - **Attendance**: View and mark attendance.
  - **Grade Viewing**: Students view their grades, and teachers can input grades.

---

2. Business Logic Layer (Application Layer)

- **Role**: This is the server-side layer responsible for processing business logic, interacting with the database, and sending responses to the Presentation Layer.
- **Technologies**:
  - **PHP** (for backend logic and API creation)
  - **JSON** (for data interchange between client-side JavaScript and server-side PHP)
- **Functionalities**:
  - **User Authentication & Authorization**: Handle login and session management, ensuring the right user accesses the right information.
  - **Role-Based Access Control**: Provide different functionality for students, teachers, and administrators.
  - **Assignment Management**:
    - **Teacher Side**: Create, update, and delete assignments.
    - **Student Side**: Submit assignments and check feedback.

- **Attendance Management**: Logic for adding, updating, and viewing attendance records.
- **Class Scheduling**: Create and update class schedules.
- **Grade Management**:
  - **Teacher Side**: Enter and update grades for students.
  - **Student Side**: View grades.
- **Communication Module**: Sending notifications/emails to students and parents.
- **Data Validation & Sanitization**: Validate input data from the front-end and ensure data integrity before interacting with the database.

---

3. Data Access Layer (Database Layer)

- **Role**: This layer is responsible for communicating with the database, fetching, inserting, updating, or deleting records.
- **Technologies**:
  - **MySQL** (Relational Database Management System)
  - **PHP** (to interact with the MySQL database through queries)
- **Database Tables** (Examples):
  - **Users**: (user_id, username, password, role, email, etc.)
  - **Students**: (student_id, name, class, contact, address, etc.)
  - **Teachers**: (teacher_id, name, subject, contact, etc.)
  - **Classes**: (class_id, subject_id, teacher_id, schedule)
  - **Subjects**: (subject_id, subject_name, description)
  - **Assignments**: (assignment_id, teacher_id, student_id, assignment_file, grade)
  - **Attendance**: (attendance_id, student_id, date, status)
  - **Grades**: (grade_id, student_id, subject_id, grade)
  - **Notifications**: (notification_id, user_id, message, date)
- **Data Access Logic**:
  - Queries to fetch user data for login.
  - Queries for CRUD operations on student records, teacher records, assignments, and grades.
  - Complex queries for generating reports like attendance reports, grade sheets, etc.

Layered Communication Flow

1. **Front-end (JavaScript + HTML)** sends a request (e.g., user login) to the **PHP server**.
2. **PHP (Business Logic)** verifies the user data and checks roles, then requests the user's data from **MySQL (Data Layer)**.
3. If valid, PHP returns the user dashboard information to the **Front-end**, which is rendered using JavaScript.
4. For operations like attendance marking or assignment submission, the flow involves:
   - **JavaScript** sending an AJAX request to **PHP**, which processes it, updates the database in **MySQL**, and sends a success or failure response.

Example Breakdown for School System Features

1. User Login (Authentication)

- **Presentation Layer**: Login form (HTML + JavaScript for form validation).
- **Business Logic Layer**: PHP processes the login request, validates user credentials, and creates a session.
- **Data Access Layer**: MySQL retrieves user details from the database.

2. Student Dashboard

- **Presentation Layer**: Displays student details, assignments, grades (JavaScript for dynamic content loading).
- **Business Logic Layer**: PHP fetches the student's assignments, grades, and schedule.
- **Data Access Layer**: MySQL queries fetch assignment data, class schedules, and grades.

3. Assignment Submission

- **Presentation Layer**: Form for uploading assignments (JavaScript for file handling).
- **Business Logic Layer**: PHP handles the file upload, checks file type and size, and stores the file path.
- **Data Access Layer**: MySQL stores the assignment metadata in the assignments table (e.g., assignment_id, student_id, file_path).

4. Grade Management

- **Presentation Layer**: Teachers input grades (JavaScript handles form validation).
- **Business Logic Layer**: PHP checks if the teacher is authorized and stores the grade.
- **Data Access Layer**: MySQL updates the grades table with the new records.

Tools and Frameworks

- **PHP Framework**: Use a PHP framework like Laravel or CodeIgniter to manage routing, middleware, and business logic efficiently.
- **JavaScript Framework**: Use Vue.js or React.js for building interactive user interfaces.
- **Database**: MySQL Workbench for database design and management.

This layered architecture allows for **clear separation of concerns**, with each layer responsible for a distinct part of the school system. It ensures that changes in one layer (e.g., UI or database) don't significantly impact others.