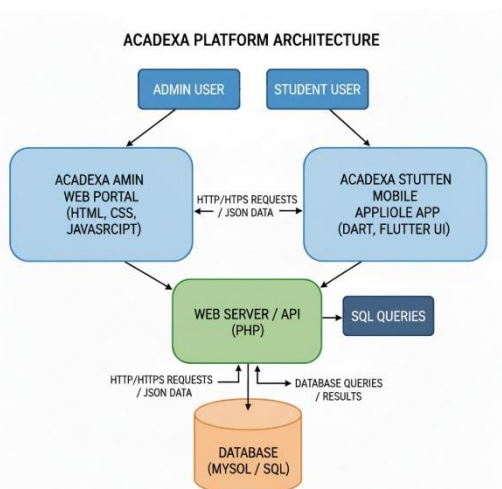| ![Marwadi University Logo] NAAC A+ Marwadi University Marwadi Chandarana Group	**Marwadi University** **Faculty of Engineering & Technology** **Department of Information and Communication Technology**	
Subject: CP (01CT0715)	**Aim: System Design and Architecture**	
	Date:	**Enrollment No: 92200133036**

1. Introduction

The proposed system consists of two main components: a Web Application for administrators and a Mobile Application for students. The admin portal enables class creation, attendance management, inquiry handling, and reporting, while the student-facing mobile app allows learners to access attendance records, view inquiries, and receive notifications.

A well-defined system design ensures the solution is modular, scalable, and maintainable, aligning with ICT engineering principles. This section elaborates on the modular breakdown, technology stack, and scalability plan.

2. Modular Design

The system is divided into distinct modules, each serving a specific function:

1. **Admin Web Application (Frontend)**
   - Used by administrators for managing classes, students, and attendance.
   - Provides dashboards and reporting features.
2. **Student Mobile Application (Frontend)**
   - Built using Flutter for cross-platform support (Android & iOS).
   - Students can log in, view attendance, and track inquiry updates.
3. **Backend Layer**
   - Handles business logic and API endpoints.
   - Processes admin requests and student queries.
4. **Database Layer**
   - Centralized SQL database storing class, student, inquiry, and attendance records.
   - Ensures data consistency across web and mobile platforms.
5. **Authentication & Authorization Module**
   - Role-based login (Admin and Student).
   - Secures system access and ensures data privacy.
6. **Reporting & Analytics Module**
   - Generates charts and reports for inquiries, attendance, and student progress.

3. Technology Stack

1. **Frontend (Admin Web Application):**
   - **Technologies:** HTML, CSS, JavaScript (or React if chosen).
   - **Reasoning:** Easy to maintain, widely supported, and efficient for dashboards.
2. **Frontend (Student Mobile App):**
   - **Technology:** Flutter.
   - **Reasoning:** Single codebase for Android and iOS, faster development, and strong UI capabilities.
3. **Backend Layer:**
   - **Technology:** Node.js with Express (or PHP, depending on your actual project).
   - **Reasoning:** Supports REST APIs, lightweight, handles concurrent requests efficiently.
4. **Database Layer:**
   - **Technology:** MySQL.
   - **Reasoning:** Relational structure fits student/class data, reliable for transactional queries, widely documented.
5. **Authentication:**
   - **Option 1:** Firebase Authentication.
   - **Option 2:** Custom SQL-based authentication.
   - **Reasoning:** Ensures secure login and scalability with multiple user roles.
6. **Hosting & Deployment (Future Scalability):**
   - **Options:** AWS, Azure, or DigitalOcean.
   - **Reasoning:** Enables cloud scaling, load balancing, and higher availability

4. Scalability Plan

To accommodate growth in users and data, the system includes scalability strategies:

- **Application Layer Scaling:**
  Deploy backend as microservices, use load balancers, and support containerization (Docker, Kubernetes).
- **Database Scaling:**
  - Use replication to handle read-heavy loads.
  - Sharding to split large datasets.
  - Indexing and caching (Redis) to reduce query response times.
- **API Performance:**
  REST APIs designed stateless for easy scaling across multiple servers.
- **Reliability & Fault Tolerance:**
  - Regular backups and replication for disaster recovery.
  - Failover servers to ensure uptime.
- **Cost Considerations:**
  Begin with on-premise/local hosting (XAMPP) for development, then migrate to cloud when usage increases.

5. Conclusion

The proposed architecture is **modular, robust, and scalable**. By separating the system into web, mobile, backend, and database layers, maintainability and reusability are ensured. The chosen technology stack (Flutter, Node.js/PHP, MySQL) is widely adopted and reliable for educational systems. Scalability planning with cloud deployment, caching, and database optimization addresses potential bottlenecks, ensuring the system can grow with future requirements.