**3. Backend Service & API Development — Scalable, Secure Microservices**

**Objective:**  
Build a robust, modular backend architecture that serves the mobile and web clients through well-documented, secured, and versioned APIs.

**Detailed Components:**

* **Architecture Approach:**
  + **Microservices** or **modular monolith**, depending on scale and budget.
  + Each service handles a domain (e.g., user mgmt, payroll, leave, attendance).
* **API Layer:**
  + RESTful APIs (optionally GraphQL) with consistent naming and error codes.
  + API versioning strategy (e.g., /v1/attendance).
  + OpenAPI / Swagger documentation auto-generated and published.
* **Security & Authentication:**
  + OAuth 2.0 / JWT-based user and service authentication.
  + Role-Based Access Control (RBAC) per endpoint.
  + Rate limiting, CORS, and IP filtering for added protection.
* **Scalability & Performance:**
  + Load-balanced services, horizontal scaling support.
  + Caching (e.g., Redis) for frequently accessed resources.
  + Asynchronous task queue (e.g., Celery, Kafka) for non-blocking operations.
* **Database Design:**
  + PostgreSQL or MySQL as primary RDBMS; MongoDB if hybrid required.
  + Secure encrypted storage of sensitive data (e.g., PII, payroll).
  + Backup and rollback support for critical tables.
* **Monitoring & Logging:**
  + Centralized logging (e.g., Fluentd, ELK).
  + Metrics collection (e.g., Prometheus + Grafana).
  + Alerting on error rates, CPU spikes, unauthorized access attempts