# **📘 Software Requirements Specification (SRS)**

## **1. Introduction**

### **Purpose**

This document outlines the software requirements for the development, testing, deployment, and maintenance of the ACA Student Hub, a comprehensive Learning Management System (LMS) tailored for the Attueyi Coding Academy (ACA). It provides a detailed account of features, technical architecture, user roles, and functional and non-functional expectations.

### **Scope**

The ACA Student Hub is a platform that supports student onboarding, academic management, communication, and community-building across various learning tracks. It includes dashboards for students, tutors, and admins, facilitating a seamless learning experience from registration through graduation.

### **Intended Audience**

* Software Engineers
* Product Managers
* Quality Assurance Engineers
* System Architects
* ACA Admin and Operations Teams

## **2. Overall Description**

### **System Environment**

* Web-based application accessible on modern browsers
* Mobile responsiveness
* Hosted on cloud infrastructure (e.g., AWS, GCP)

### **Dependencies**

* Email service provider (e.g., Mailtrap)
* Payment gateway (e.g., Paystack, Flutterwave)
* Authentication system (e.g., Firebase Auth, OAuth)
* Video hosting (e.g., Google Drive)

### **System Overview**

* The system supports student lifecycle from registration to course completion.
* It provides communication channels, academic tracking, support, and gamification elements.

## **3. System Features**

### **Student Features (User Role: Student)**

* **Registration & Onboarding**
  + As a prospective student, I want to register my details, select a track, and receive email confirmations.
  + As a scholarship applicant, I want to take an entrance exam and get notified of my results.
  + As a paid student, I want to make payments and receive onboarding instructions.
* **Dashboard**
  + View announcements, events, assignments, leaderboard, and class resources.
  + Access track-specific and cohort-wide content.
  + Submit assignments, view grades, and track progress.
  + Join forums and discussion boards.
  + Get feedback and support.
* **Course Interaction**
  + View course outline, monitor progress, take quizzes, and view attendance.
  + Access recordings, notes, and reference projects.
* **Social & Community**
  + Participate in weekly challenges and view leaderboard standings.
  + Get notified about birthdays, hackathons, and social events.

### **Admin Features (User Role: Admin)**

* **User Management**
  + Approve/reject student applications.
  + Grant dashboard access to successful candidates.
* **Cohort Management**
  + Upload orientation content.
  + Create events, town halls, and announcements.
  + Monitor student progress and performance metrics.
* **Communication & Support**
  + Manage all cohort/town hall announcements.
  + Enable and oversee motivational messages.
* **Payments & Scholarships**
  + Review payment confirmations.
  + Update payment status manually if required.
  + Generate scholarship exam results and notify candidates.

### **Tutor Features (User Role: Tutor)**

* **Class Management**
  + Create assignments, quizzes, and projects.
  + Grade submissions and provide feedback.
  + Upload class notes and help documents.
  + Host virtual or recorded classes.
* **Student Monitoring**
  + Track student attendance and progress.
  + Send personalized messages or reminders.
* **Forum Interaction**
  + Moderate discussions, answer student queries.

### **Forum Moderator Features (User Role: Forum Moderator)**

* **Forum Oversight**
  + Review and approve posts.
  + Remove inappropriate content.
  + Mediate conflicts within discussions.
  + Highlight useful threads.

## **4. External Interface Requirements**

### **APIs**

* RESTful APIs for frontend-backend communication
* Webhooks for payment confirmations
* Email APIs (e.g., Mailtrap) for notifications

### **Database**

* Relational database (e.g., PostgreSQL) for structured data
* Document storage for profile pictures, notes, projects (e.g., Google Drive)

### **Hardware**

* Hosted on cloud servers (no specialized local hardware)
* Optimized for mobile, tablet, and desktop devices

### **Other Systems**

* Payment integrations (Paystack, Flutterwave)
* Authentication and authorization system (Firebase Auth, OAuth)

## **5. Non-functional Requirements**

### **Performance**

* Handle 10,000+ concurrent users with minimal latency

### **Scalability**

* Modular design allowing for track-level expansion

### **Security**

* Encrypted data storage
* Secure login using 2FA for sensitive roles
* Regular security audits and patch management

### **Usability**

* Intuitive UI/UX
* Responsive design across devices

### **Maintainability**

* Modular codebase with documentation
* Automated test suites for unit and integration testing

### **Availability**

* 99.9% uptime via cloud infrastructure
* Backup and disaster recovery mechanisms

## **6. Other Requirements**

### **Legal & Compliance**

* GDPR compliance for data privacy
* COPPA compliance for students under 13 (if applicable)

### **Licensing**

* Open-source dependencies to follow MIT/BSD or compatible licenses

## **7. Appendices**

### **A. Acronyms**

* LMS: Learning Management System
* ACA: Attueyi Coding Academy

### **B. Reference Documents**

* ACA Curriculum Outline
* Branding Guidelines
* Sample Timetables and Track Outlines

### **C. Definitions**

* **Track**: A specialized learning path (e.g., Frontend, Backend)
* **Cohort**: A batch of students learning together in a specific time frame