# Software Requirements Specification (SRS)

## Multi-Community Admission & Enrollment System

## 1. Introduction

### 1.1 Purpose

The Multi-Community Admission & Enrollment System is a SaaS-based platform designed to allow a super-admin to sell admission and enrollment software to multiple schools (communities). The system facilitates online admission, student enrollment, automated payment processing, structured data storage, and comprehensive dashboards for parents and school administrators.

### 1.2 Scope

The system will be developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) and will include features such as:  
- Online admission form for parents to enroll their students.  
- Parent dashboard for enrollment, course selection, and payment management.  
- Admin dashboard for managing admissions, approvals, payments, and reporting.  
- Automated recurring payments (monthly/quarterly) via Stripe or PayPal.  
- User roles & access control for Parents, Admins, and Directors.  
- Automatic email confirmations for applications and admissions.  
- Schools can customize their UI, including logos, themes, etc.

### 1.3 Users

- Super Admin: Manages software sales and school accounts.  
- School Admin: Manages admissions, payments, and course offerings.  
- Parents: Submit applications, enroll students, and make payments.

## 2. Functional Requirements

### 2.1 Super Admin Features

- Manage multiple schools (communities).  
- Create and sell subscription plans for schools.  
- Track system usage and school subscriptions.  
- View revenue and system reports.

### 2.2 School Admin Features

- View and manage parent & student details.  
- Admission approval system.  
- Add, update, and delete courses.  
- Validate payments using Stripe or PayPal.  
- Filter student data and generate reports.  
- Customize school UI (logo, themes, colors, etc.).

### 2.3 Parent Features

- Fill out an online admission form.  
- View dashboard with enrolled students and course details.  
- Select and enroll students in available courses.  
- Set up automated monthly or quarterly payments.  
- Cancel or modify enrollment based on course policies.

### 2.4 Structured Data Tables

- Parent Table: Full Name, Address (Street, City, State, ZIP Code), Primary Contact Number, Secondary Contact Number, Email, Employment Information, Notes, Approval Status.  
- Student Table: Application Number, Full Name, Date of Birth, Grade Level, Gender, Allergies/Medical Conditions, Special Needs (if applicable), Parent/Guardian Association, Approval Status.  
- Payment Table: Transaction ID, Amount, Date, Payment Method (Credit Card, Debit Card, PayPal, ACH Transfer), Reference Number, Description, Payment Status.  
- Auto-Fill Feature: ZIP Code → Auto-fill City & State based on postal data.

## 3. Non-Functional Requirements

### 3.1 Performance & Scalability

- The system should support multiple schools with thousands of users concurrently.  
- Optimized database queries for high performance.

### 3.2 Security

- Role-based access control for different user types.  
- Data encryption for sensitive information using bcrypt.  
- Secure authentication using JWT.

### 3.3 Usability

- Responsive UI for seamless use across devices.  
- Intuitive dashboards and navigation for all user roles.

### 3.4 Availability & Reliability

- 99.9% uptime guarantee for schools using the platform.  
- Automated backups and data recovery mechanisms.

## 4. Testing Requirements

### 4.1 Unit Testing

- Form validations.  
- Payment processing.  
- Dashboard functionalities.

### 4.2 User Testing

- Parents and admins test the system for usability.

### 4.3 Security Testing

- Protect sensitive data.  
- Prevent breaches and unauthorized access.

## 5. Technology Stack

- Frontend: React.js (with Material UI or Tailwind CSS).  
- Backend: Node.js, Express.js.  
- Database: MongoDB.  
- Authentication: JWT-based authentication.  
- Payment Integration: Stripe & PayPal.  
- Hosting & Deployment: AWS, or similar cloud hosting (VPS).

## 6. Conclusion

This Multi-Community Admission & Enrollment System is designed to simplify and automate the admission and payment process for schools. By leveraging the MERN stack, the system ensures scalability, security, and ease of use for super admins, school admins, and parents. The structured database, automated recurring payment feature, and seamless user roles ensure a smooth experience for all users. Schools will also have the ability to customize their UI, including logos, themes, and branding preferences.