[Document title]

[Document subtitle]

​​Table of Contents

​

​

​

​

​

​

​

​

​

​

​​

# Requirements Document

## 1. Business Requirements

**Problem Statement:**

Students and tutors currently face challenges in managing educational content, communication, and resource sharing. There is no centralized platform where learners can easily access categorized materials, submit tasks, or interact with tutors in real-time.

**Solution Statement:**

Develop a mobile and web application that allows students to view categorized learning materials (e.g., Art, Mathematics, Technology), while tutors can upload, edit, and manage these resources. An admin panel allows complete control over user management and content moderation.

## 2. Solution Requirements

**Functional Requirements**

| **Iteration** | **Feature/User Story** | **Description** |
| --- | --- | --- |
| **1** | User Authentication | - Users can register and log in using Firebase Authentication (email/password).  - Roles include Student, Tutor, and Admin. |
|  | Browse Learning Content | - Students can view content grouped by subject (e.g., Art, Math, Technology).  - Users can tap to view more details. |
|  | Navigation Setup | - Stack navigation bar with options for Home, Subjects etc. |
| **2** | Role-Based Access | - Tutors can add or edit content. - Admins can add/edit/delete content. - Students can only view content. |
|  | Add/Edit/Delete Content | - Forms for creating and updating content.  - Supports title, description, type, etc. |
|  | View Details | - Clicking an item opens a detailed view with structured layout and fields. |
| **3** | Search and filter | - Users can search content by keyword or filter by subject. |
|  | Postman/API Integration | - Backend endpoints tested using Postman for CRUD (Create, Read, Update, Delete) operations. |
|  | Final Push & Deployment | - GitHub repo cleanup and final deployment preparation. |

## Non-Functional Requirements

| **Category** | **Description** |
| --- | --- |
| **Performance** | The app should load the home screen within 2 seconds and detailed pages within 1 second. |
| **Security** | Use Firebase Auth for user verification. Admin and tutor access to be restricted with role-based permissions. |
| **Scalability** | MongoDB allows future expansion of data structure and volume. |
| **Maintainability** | Use of modular code, reusable components, and proper folder structure. |
| **Usability** | Mobile-first responsive design, simple navigation, and clearly labeled buttons and forms. |
| **Testing** | Manual and automated tests using Postman, with feedback loops for each iteration. |