# High Level Design Document

## Introduction

This High Level Design (HLD) document outlines the architecture and core components for **EduNotes**, a note-taking application designed for students. EduNotes enables users to create, edit, and organize notes with support for media attachments and tagging, leveraging modern frontend technologies.

---

## Project Name and Purpose

**Project Name:** EduNotes
**Purpose:** To provide students with an intuitive, open-source platform for managing notes, supporting multimedia content and organizational tags.

---

## System Architecture Overview

EduNotes is a single-page application (SPA) built with React, utilizing a modular component-based architecture. The system is structured as follows:

| Module/Component | Role/Responsibility |
|---|---|
| UI Layer | Renders user interface, handles user interactions |
| Notes Manager | Manages CRUD operations for notes |
| Media Handler | Handles media (images, files) upload and display |
| Tag Manager | Manages creation, assignment, and filtering of tags |
| Storage Layer | Persists notes, media, and tags (local storage or API) |

---

## Component Interactions

| Interaction Sequence |
|---|
| 1. User interacts with UI to create/edit/delete notes. |
| 2. UI Layer invokes Notes Manager for note operations. |
| 3. Notes Manager updates Storage Layer and notifies UI Layer. |
| 4. Media Handler processes media uploads and links them to notes. |
| 5. Tag Manager allows users to assign/filter tags, updating UI and Notes Manager as needed. |

---

## Data Flow Overview

| Data Flow Description |
|---|

| |
|---|
| User actions (create/edit/delete notes) flow from UI Layer to Notes Manager. |
| Notes, media, and tags are persisted via the Storage Layer (local storage or backend API). |
| Media files are processed by Media Handler and linked to notes. |
| Tag assignments and filters are managed by Tag Manager, affecting note display in the UI. |

---

## Technology Stack

| Layer/Function | Technology/Framework |
|---|---|
| Frontend Framework | React |
| Styling | Tailwind CSS, CSS |
| Language | JavaScript, HTML |
| State Management | React Context/State |
| Storage | Browser Local Storage (MVP) |
| Media Handling | HTML5 APIs |

---

## Scalability & Reliability

- **Scalability:** Designed as a modular SPA; can integrate with backend APIs for cloud storage and multi-device sync in future iterations.
- **Reliability:** Utilizes browser local storage for offline access; error handling in UI for failed operations.
- **Security:** Media and note data are stored locally; no sensitive data is transmitted externally in MVP.

---

**End of Document**