



# Product Requirements & Specification Document

---

## Project Name

EduNotes - Note Taking App

## Description

EduNotes is an open-source, web-based note-taking application designed for students. It enables users to create, edit, organize, and manage notes with support for media attachments and tagging. The app is built using React, JavaScript, HTML, CSS, and Tailwind CSS.

---

## 1. Goals & Objectives

| Goal                   | Description   |
|------------------------|---|
| Simple Note Management | Allow students to easily create, edit, and delete notes |
| Media Support          | Enable embedding images and links in notes              |
| Tagging & Organization | Support tagging and searching/filtering notes           |
| Open Source            | Codebase is public and community-contributable          |
| Responsive UI          | Accessible on desktop and mobile devices                |

---

## 2. User Stories

| ID  | As a... | I want to...                   | So that...                      |
|-----|---------|--------------------------------|---------------------------------|
| US1 | Student | Create, edit, and delete notes | I can manage my study materials |
| US2 | Student | Attach images/links to notes   | I can enrich my notes           |
| US3 | Student | Tag notes with keywords        | I can organize and filter notes |
| US4 | Student | Search notes by text or tag    | I can quickly find information  |
| US5 | Student | Use the app on any device      | I have access anywhere          |

---

## 3. Functional Requirements

| ID  | Requirement   |
|-----|---|
| FR1 | Users can create, edit, and delete text notes                       |
| FR2 | Users can attach images (via upload or URL) and hyperlinks to notes |
| FR3 | Users can assign, edit, and remove tags from notes                  |
| FR4 | Users can filter and search notes by text and tags                  |



|     |  |
|-----|--|
| FR5 | Notes are listed in a scrollable, responsive interface   |
| FR6 | All data is stored locally in the browser (localStorage) |
| FR7 | The app is fully responsive (desktop/mobile)             |
| FR8 | The codebase is open-source and documented               |

#### 4. Non-Functional Requirements

| ID   | Requirement                                |
|------|--|
| NFR1 | Fast load times and smooth UI interactions |
| NFR2 | Accessible (WCAG 2.1 AA compliance)        |
| NFR3 | Minimalist, distraction-free design        |
| NFR4 | No user authentication required            |
| NFR5 | All features work offline (PWA optional)   |

#### 5. Technical Specifications

| Area         | Specification                               |
|--------------|---|
| Frontend     | React (functional components, hooks)        |
| Styling      | Tailwind CSS, custom CSS as needed          |
| State Mgmt   | React Context or useState/useReducer        |
| Storage      | Browser localStorage                        |
| Media Upload | HTML5 File API, image preview               |
| Search/Tags  | Simple text search, tag-based filtering     |
| Build Tools  | Vite or Create React App                    |
| Testing      | Manual testing, basic unit tests (Jest/RTL) |
| Deployment   | GitHub Pages or Netlify                     |

#### 6. UI/UX Requirements

| Page/Component    | Requirements  |
|-------------------|---|
| Notes List        | List of notes, search bar, tag filter, create note button               |
| Note Editor       | Rich text area, image/link embed, tag input, save/cancel/delete actions |
| Tag Management    | Add/remove tags, display tag chips                                      |
| Responsive Layout | Mobile-friendly navigation, adaptive components                         |
| Accessibility     | Keyboard navigation, ARIA labels, color contrast                        |



---

## 7. Out of Scope

- User authentication or cloud sync
  - Real-time collaboration
  - Advanced rich text formatting (beyond bold/italic/lists)
  - Third-party integrations
- 

## 8. Acceptance Criteria

- Users can create, edit, and delete notes
  - Users can attach images and links to notes
  - Users can tag notes and filter/search by tag or text
  - All data persists in localStorage
  - App is responsive and accessible
  - Codebase is open-source and documented
- 

## 9. Milestones & Timeline

| Milestone               | Description                         | Target Date |
|-------------------------|-------------------------------------|-------------|
| Project Setup           | Repo, tooling, base structure       | Week 1      |
| Core Note Features      | CRUD notes, localStorage            | Week 2      |
| Media & Tag Support     | Image/link embed, tagging/filter    | Week 3      |
| Responsive UI & Testing | Mobile layout, accessibility, tests | Week 4      |
| Documentation & Release | README, open-source release         | Week 5      |

---

## 10. Open Questions

- Should PWA/offline support be included in MVP?
  - What image size limits should be enforced?
  - Should export/import of notes be supported in future versions?
- 

**End of Document**