



Product Requirements & Specification Document

Project Name

EduNotes - Student Notes Management

Description

EduNotes is a backend API enabling students to upload, organize, and retrieve study notes. The system supports file uploads and user authentication, built using Node.js, Express, and MongoDB.

1. Goals & Objectives

Goal	Description
Secure Notes Management	Allow authenticated users to manage their study notes
File Upload Support	Enable uploading of various file types (PDF, DOCX, etc.)
Organization & Retrieval	Provide endpoints for organizing and retrieving notes

2. Core Features

Feature	Description
User Authentication	Register, login, JWT-based authentication
Notes CRUD	Create, read, update, delete notes
File Upload	Upload and associate files with notes
Notes Organization	Tagging and categorization of notes
Search & Retrieval	Search notes by title, tags, or content

3. User Stories

ID	As a...	I want to...	So that...
1	Student	Register and log in	My notes are secure and private
2	Student	Upload and organize notes	I can manage my study materials efficiently
3	Student	Retrieve and search my notes	I can quickly find what I need
4	Student	Delete or update my notes	I can keep my notes up to date

4. API Endpoints



Method	Endpoint	Auth	Description
POST	/api/auth/register	No	Register new user
POST	/api/auth/login	No	User login, returns JWT
GET	/api/notes	Yes	List all user notes
POST	/api/notes	Yes	Create a new note
GET	/api/notes/:id	Yes	Retrieve a specific note
PUT	/api/notes/:id	Yes	Update a note
DELETE	/api/notes/:id	Yes	Delete a note
POST	/api/notes/:id/upload	Yes	Upload file to a note
GET	/api/notes/search	Yes	Search notes by query

5. Data Model

User

```
{
  "_id": "ObjectId",
  "name": "String",
  "email": "String",
  "passwordHash": "String"
}
```

Note

```
{
  "_id": "ObjectId",
  "userId": "ObjectId",
  "title": "String",
  "content": "String",
  "tags": ["String"],
  "fileUrl": "String",
  "createdAt": "Date",
  "updatedAt": "Date"
}
```

6. Non-Functional Requirements

Requirement	Specification
Security	JWT auth, password hashing, input validation
File Storage	Local or cloud (configurable)



Performance	< 500ms response time for standard operations
Scalability	Support for 1,000+ concurrent users
Documentation	OpenAPI/Swagger for all endpoints

7. Technology Stack

Layer	Technology
Language	Node.js
Framework	Express
Database	MongoDB
Auth	JWT
File Upload	Multer (Express)
Docs	Swagger/OpenAPI

8. Acceptance Criteria

- Users can register, log in, and manage their notes securely
- Notes can be created, updated, deleted, and retrieved
- Files can be uploaded and linked to notes
- Notes can be organized with tags and searched
- All endpoints require authentication except registration/login
- API documentation is available and up to date

9. Out of Scope

- Frontend/UI
- Real-time collaboration
- Advanced analytics

10. Milestones

Milestone	Description	Target Date
API Skeleton	Project setup, basic endpoints	Week 1
Auth & User Management	Registration, login, JWT	Week 2
Notes CRUD	Notes endpoints, MongoDB models	Week 3
File Upload	File storage integration	Week 4
Search & Tagging	Search, tags endpoints	Week 5



Documentation & Testing	Swagger, unit tests	Week 6
-------------------------	---------------------	--------

11. Example API Usage

```
POST /api/notes
Authorization: Bearer <token>
Content-Type: application/json

{
  "title": "Calculus Notes",
  "content": "Limits, derivatives, integrals...",
  "tags": ["math", "calculus"]
}
```

12. Risks & Mitigations

Risk	Mitigation
File upload abuse	Limit file size/type, scan files
Data loss	Regular backups
Unauthorized access	Strict JWT validation

End of Document