



PHASE 1 — Planning & Design (*DONE*)

- ✓ Problem statement
- ✓ Feature list
- ✓ ER Diagram
- ✓ Tech stack decision (.NET + EF + MSSQL)

 **Status:** Completed


PHASE 2 — Database Layer (EF Core) (*DONE*)

- ✓ Models (Entities)
- ✓ DbContext
- ✓ Relationships
- ✓ Migrations
- ✓ Database creation

 **Status:** Completed

 Do NOT touch unless schema changes

PHASE 3 — Architecture Setup

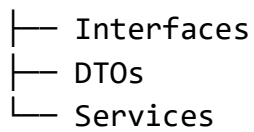
 Goal: Clean separation of responsibilities

What to create:

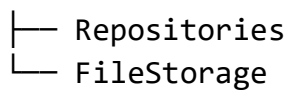
- ✓ Repositories
- ✓ Services
- ✓ DTOs
- ✓ Dependency Injection

Structure:

Core




Infrastructure



You are here now

PHASE 4 — Controllers (API Layer)

 Goal: Expose backend functionality

Controllers:

- ✓ UsersController
- ✓ DocumentsController
- ✓ CommentsController

Responsibilities:

- Input validation
- Call Services

- Return HTTP responses

🚫 No DB logic allowed here

PHASE 5 — File Upload & Storage

🎯 Goal: Real document handling

- ✓ Upload using IFormFile
- ✓ Store file in wwwroot/uploads
- ✓ Save metadata in DB
- ✓ Secure file access

Bonus:

- File size validation
- File type validation

PHASE 6 — Inline Comments System

🎯 Goal: Core project feature

- ✓ Add comment on document
- ✓ Store text + position
- ✓ Fetch comments per document
- ✓ Edit / delete comments

(Optional: highlight text ranges)

PHASE 7 — Authentication & Authorization


 Goal: Secure the app

- ✓ User registration
- ✓ Login / Logout
- ✓ Password hashing
- ✓ Role-based access

Options:

- ASP.NET Identity (recommended)
- JWT (if API-only)

PHASE 8 — Frontend Integration (MVC / Razor)


 Goal: Usable UI

- ✓ Login / Register pages
- ✓ Upload document page
- ✓ Document list
- ✓ Document viewer
- ✓ Comment panel

Use:

- Razor Pages / MVC
- AJAX / Fetch API

PHASE 9 — Advanced Features (Optional)

 Goal: Stand out in evaluation

- ✓ Search documents
- ✓ Version history
- ✓ Permissions (owner/collaborator)
- ✓ Notifications
- ✓ Activity logs