

✓ CORE FUNCTIONALITY (What You Must Build)

1. Project Management

 Role: **Admin**


- **Add/Remove Projects**
 - Create a new project (give: name, description, deadline)
 - Option to delete it or mark it as "Completed"
- **View Projects**
 - All users (Admins, Leads, Devs) should see a list of **only active (not completed)** projects

2. Team Assignment

 Role: **Project Lead**

- **Assign Developers to a Project**
 - Leads can choose from existing users with **Developer role** and assign them to a project they lead
- **View Assigned Projects**
 - Developers log in and can see only the projects they are part of

3. Asset & Resource Management

 Role: Admin + Project Lead

- **Upload Documents**
 - For a project, Admin or Lead can upload PDFs/docs (e.g. design notes, meetings)

- **View Documents**
→ Only users assigned to that project (Developers) can **view** those docs



PRIVILEGE SEPARATION (Role-wise Power)

Role	What They Can Do
Admin	Create/Delete projects, Mark as completed, Manage users (create, change roles), Upload docs to any project
Lead	Assign Devs to only their projects, Upload docs to only their projects
Developer	View only assigned projects, view documents of those projects



LOGIN SECURITY

1. Secure Login

- Use **BCrypt** to store passwords hashed in DB (no plain text)
- Use **JWT tokens** to handle authentication

2. MFA (Optional – but you're doing it)

- After entering password, user gets an **OTP on email** before login completes (can be basic 6-digit)



PAGES YOU MUST BUILD

1. Sign In/Register Page

- **Login page** for all users

- **Only Admin** can register new users (name, email, role)
-

2. Dashboard (Dynamic per Role)

One dashboard route (`/dashboard`) but content **changes based on role**:

- **Admin:**
 - Add/Remove Projects
 - Mark project as completed
 - Create/Edit Users (assign role)
 - Upload docs to any project
 - **Project Lead:**
 - See projects they lead
 - Assign developers to those projects
 - Upload docs to their projects
 - **Developer:**
 - See list of assigned projects
 - View uploaded documents
-

3. Account Settings

- Change password
 - Enable/disable MFA (if implemented)
-

4. Project Details Page

When user clicks on a project:

- Show project name, description, deadline
- List of team members
- List of uploaded documents



Summary (Your Deliverables)

Item	What to Build
Backend	Java + Spring Boot (Auth, Roles, APIs)
Frontend	React + Tailwind (Pages above)
DB	MySQL or PostgreSQL (tables for users, projects, docs, roles, assignments)
Security	JWT, BCrypt, Optional MFA (via email)
Docs	2000-word report (design, dev, testing, security)
Video	8-min walk-through with voiceover

You're now **crystal clear** on what they expect.

Want to move on to **backend schema + tables design next?**

```
pixelforge-backend/
├── src/
│   ├── main/
│   │   ├── java/com/pixelforge/
│   │   │   ├── config/
│   │   │   │   ├── SecurityConfig.java
│   │   │   │   ├── JwtFilter.java
│   │   │   │   ├── JwtUtil.java
│   │   │   │   └── MailConfig.java (for MFA email OTP)
│   │   │   ├── controller/
│   │   │   │   ├── AuthController.java
│   │   │   │   ├── UserController.java
│   │   │   │   ├── ProjectController.java
│   │   │   │   └── DocumentController.java
│   │   │   ├── dto/
│   │   │   │   ├── AuthRequest.java
│   │   │   │   ├── AuthResponse.java
│   │   │   │   ├── ProjectDTO.java
│   │   │   │   └── DocumentDTO.java
│   │   │   ├── entity/
│   │   │   │   ├── User.java
│   │   │   │   ├── Role.java
│   │   │   │   ├── Project.java
│   │   │   │   ├── Document.java
│   │   │   │   └── MFAOtp.java
│   │   │   ├── repository/
│   │   │   │   ├── UserRepository.java
│   │   │   │   ├── ProjectRepository.java
│   │   │   │   ├── DocumentRepository.java
│   │   │   │   └── MFAOtpRepository.java
│   │   │   ├── service/
│   │   │   │   ├── AuthService.java
│   │   │   │   ├── ProjectService.java
│   │   │   │   ├── UserService.java
│   │   │   │   ├── DocumentService.java
│   │   │   │   └── MFAService.java
│   │   │   ├── util/
│   │   │   │   ├── RoleEnum.java
│   │   │   │   └── PixelforgeApplication.java
│   │   └── resources/
│   │       ├── application.properties
│   │       └── templates/ (for email OTP)
└── pom.xml
```

