

SaaS Notes App Requirements (Simple Step-by-Step Explanation)

Step 1: Multi-Tenancy (Multiple Companies)

- Your app should support **two companies (tenants)**: Acme and Globex.
 - **Isolation**: Users from Acme cannot see Globex notes, and vice versa.
 - **Example**:
 - Alice logs in to Acme → sees only Acme notes.
 - Bob logs in to Globex → sees only Globex notes.
 - **How to separate data**:
 - Each note has a `company` or `tenantId` field.
 - Only fetch notes that match the user's company.
-

Step 2: Authentication & Roles

- Users log in with **email + password** using JWT.
 - **Roles**:
 - **Admin**: can invite users and upgrade the company plan.
 - **Member**: can only create, view, edit, delete notes.
 - **Test Accounts (password = 'password')**:
 - `admin@acme.test` → Admin, Acme
 - `user@acme.test` → Member, Acme
 - `admin@globex.test` → Admin, Globex
 - `user@globex.test` → Member, Globex
 - **Example**:
 - Admin logs in → sees upgrade button + invite options.
 - Member logs in → can only manage notes.
-

Step 3: Subscription Plans

- **Free Plan**: max 3 notes per company.
 - Example: Acme Free → all Acme users together can create 3 notes only.
 - **Pro Plan**: unlimited notes.
 - Example: Globex Pro → users can create unlimited notes.
 - **Upgrade**:
 - Only Admin can upgrade the company to Pro.
 - After upgrade → limit is removed immediately.
 - **Frontend Behavior**:
 - Free company reaches 3 notes → show **"Upgrade to Pro"** button.
 - Once Admin upgrades → unlimited notes for all users.
-

Step 4: Notes API (CRUD)

- Users can:
 - **Create note** → add a note for their company.

- **View notes** → see only company notes.
 - **View single note** → fetch one note from their company.
 - **Update note** → edit note from their company.
 - **Delete note** → remove note from their company.
 - **Example:**
 - Alice (Acme) adds a note → only Acme users can see it.
 - Bob (Globex) cannot see Alice's note.
-

Step 5: Deployment

- Host **backend + frontend** on **Vercel**.
 - Enable **CORS** so external scripts can access API.
 - Add **Health Check Endpoint:**
 - `/health` → returns `{ "status": "ok" }`.
-

Step 6: Frontend

- Minimal but functional UI.
 - Features:
 - Login using test accounts.
 - List, create, delete notes.
 - Show **"Upgrade to Pro"** when Free plan reaches 3 notes.
 - Respect roles: Members vs Admins.
 - **Example:**
 - Admin logs in → sees upgrade button.
 - Member logs in → cannot upgrade, can only manage notes.
-

Step 7: Evaluation / Testing

- Automated scripts will check:
 - Health endpoint works.
 - Login works for all test accounts.
 - Companies cannot see each other's notes.
 - Role restrictions are respected.
 - Free plan limit enforced; Pro removes limit.
 - CRUD endpoints work correctly.
 - Frontend works and shows correct behavior.
-

Step 8: Subscription Limit Example (Simple)

- Limit is **per company, not per user**.
- Example:
 - Acme Free → Alice, Bob, Carol create notes → total = 3 notes.
 - 4th note attempt → alert: **"Limit reached. Upgrade to Pro"**.
 - Globex Free → only 1 note → no alert.
 - Once Admin upgrades → unlimited notes for all users of that company.

Summary: - Multiple companies with private data. - Users have roles (Admin vs Member). - Free = 3 notes max per company; Pro = unlimited. - Only Admin can upgrade company. - Backend + frontend on Vercel, APIs + frontend fully working. - Frontend shows upgrade button when Free plan limit is reached.