Cloud Drive System

1. Overview

This document outlines the design and implementation of a **Cloud Drive System** that supports user-based folder structures, role-based permissions, secure sharing mechanisms, and compliance requirements. The system is designed for **personal, business, and client use cases**, ensuring security, scalability, and ease of use.

2. User Folder Structure

Each user has **three primary folders**:

1. **Personal Folder**
    - Private by default.
    - Can be shared with specific users if needed.
2. **Business/Team Folder**
    - Shared based on employee roles.
    - Team-based access control with predefined permissions.
    - Departmental subfolders (e.g., HR, Finance, Marketing, Sales).
3. **Client Folder**
    - **Personal Clients Folder** (for individual clients managed by the user).
    - **Business Clients Folder** (company-wide client management with team access).
    - Each client gets a dedicated subfolder with restricted access.

3. Permissions & Access Control

Each folder/file follows **role-based access control (RBAC):**

- **Owner**: Full control (edit, delete, share, change permissions).
- **Manager**: Can edit and share but cannot delete top-level folders.
- **Editor**: Can modify content but cannot change permissions.
- **Viewer**: Read-only access.

Parent-Child Inheritance Rules:

- Sharing a parent folder **automatically applies permissions to subfolders and files** unless manually overridden.
- Sensitive folders (e.g., Legal, Finance) can have **restricted editing and download permissions**.

4. Sharing & Security Mechanisms

4.1 Time-Based Sharing

- Temporary access (e.g., 1 week, 1 month) for external users.
- Auto-expiration removes access without manual intervention.

4.2 Direct & Group Sharing

- **Internal Teams**: Share with predefined company groups (e.g., "Sales Team").
- **External Clients**: Secure sharing via links with optional password protection.
- **Public Links**: Optional view-only mode with expiration settings.

4.3 Secure Client Data Management

- Clients only see their **own** folders, preventing data leaks.
- **Watermarking & View-Only Mode** for sensitive documents.
- **Restricted downloads** for compliance-heavy industries.

4.4 Logging & Tracking

- Audit logs for file access, modifications, and sharing activities.
- Alerts for **suspicious login attempts or unauthorized file access**.
- **Geolocation & IP Restrictions** to block access from unauthorized locations.

5. Compliance & Governance

- **GDPR, HIPAA, SOC 2 Compliance**: Enforce secure data storage and sharing.
- **Legal Hold & eDiscovery**: Lock files from deletion for legal investigations.
- **Data Retention Policies**: Auto-archive and delete old files as per company rules.

6. Usability & Performance Considerations

6.1 User Experience (UX)

- **Drag & Drop Uploads** for intuitive file management.
- **Smart Search & Filters** for quick access.
- **Bulk Sharing & Permissions Management** to improve efficiency.
- **Offline Mode** for remote work scenarios.

6.2 System Optimizations

- **SSO & MFA Integration** for enhanced authentication security.
- **Custom Branding** for business clients (logo, theme, domain).
- **Auto-Generated Team Folders** based on user roles.
- **API Integration** with collaboration tools (e.g., Slack, Microsoft Teams).

7. Implementation Adjustments

✅ Key Additions

1. **Approval Workflow for Sensitive Sharing** (Admin review before external sharing).
2. **Data Classification** (Public, Internal, Confidential, Restricted labels).
3. **AI-Powered Auto-Suggestions** for smarter sharing recommendations.
4. **Audit Dashboard** for real-time compliance tracking.

❌ Simplifications & Removals

1. **Reduce Overcomplicated Sharing Rules** to enhance usability.
2. **Limit Custom Role Creation** to predefined levels (Owner, Manager, Editor, Viewer).
3. **Automate Permissions Based on User Role** to prevent manual errors.

8. Next Steps

Now that the system architecture is defined, we will proceed with **technical implementation**, focusing on:

1. **Database Design & Schema**
2. **API Structure & Backend Security**
3. **Frontend UI/UX Workflows**
4. **Integration with Existing Systems**

Next Phase: Technical Implementation

Database Design

- **Users Table**: Stores user details, roles, authentication tokens.
- **Folders Table**: Defines folder hierarchy, ownership, and permissions.
- **Files Table**: Tracks uploaded files, metadata, and version history.
- **Sharing Table**: Manages shared access with expiration settings.
- **Audit Logs Table**: Logs all actions for security and compliance.

API Architecture

- **Authentication & Authorization**
  - JWT-based authentication with refresh tokens.
  - Role-based access enforcement at the API level.
- **File & Folder Management**
  - Create, update, delete, and move files/folders.
  - Versioning & rollback support.
- **Sharing & Access Control**
  - Grant/revoke permissions dynamically.
  - Time-based link generation for secure sharing.
- **Audit & Compliance**
  - Real-time logging & security alerts.
  - Data retention policies enforcement.