SmartVault — AI-Optimized Personal Cloud Storage (Updated Plan)

1. Project Overview
SmartVault is a personal cloud storage application with AI-powered features:
- Auto-categorization of files
- Semantic search
- OCR text extraction
- File summarization
- Duplicate detection
- Cleanup suggestions
This will be built as a full-stack portfolio project using a client/server folder structure and free storage options.

2. Tech Stack
Frontend:
- Next.js (TypeScript)
- TailwindCSS + shadcn/ui
- React Query (data fetching)

Backend:
- Node.js + Express (TypeScript)
- MongoDB Atlas (metadata storage)
- Multer for file uploads (stored locally in /uploads)
- MinIO (optional S3-compatible storage if needed)
- JWT Authentication

AI Services:
- OpenAI API (embeddings + summarization)
- Tesseract.js for OCR

Deployment:
- Frontend → Vercel
- Backend → Render/Heroku
- DB → MongoDB Atlas

3. Folder Structure
smartvault/
■■■ client/ # Next.js frontend
■ ■■■ app/
■ ■■■ components/
■ ■■■ lib/
■ ■■■ public/
■ ■■■ package.json
■■■ server/ # Express backend
■ ■■■ src/
■ ■ ■■■ controllers/
■ ■ ■■■ routes/
■ ■ ■■■ models/
■ ■ ■■■ services/ (ai, search, storage)
■ ■ ■■■ middleware/
■ ■ ■■■ uploads/ # Local storage for files
■ ■■■ package.json
■ ■■■ Dockerfile
■■■ docker-compose.yml # MongoDB local dev
■■■ README.md

4. Data Models
User:
{ "_id": "ObjectId", "email": "string", "passwordHash": "string", "name": "string", "createdAt": "Date" }

File:

{ "_id": "ObjectId", "ownerId": "ObjectId", "filename": "string", "path": "string", "mimeType": "string", "size": "number",
"tags": ["string"], "embeddings": ["number"], "summary": "string", "createdAt": "Date" }

5. Key APIs
Auth:
- POST /auth/register
- POST /auth/login

Files:
- POST /files/upload
- GET /files
- GET /files/:id/download
- DELETE /files/:id

AI Features:
- POST /files/:id/analyze
- GET /search?q=term

6. AI Workflows
On Upload:
- Extract text (PDF/DOCX → text, images → OCR)
- Create embedding with OpenAI API
- Generate summary & tags
- Save metadata to MongoDB

On Search:
- Convert query to embedding
- Compare with file embeddings (cosine similarity)
- Return sorted results

Duplicate Detection:
- Compare embeddings of new file to existing files
- Suggest deletion/archival

7. MVP Timeline
Week 1:
- Day 1: Project setup
- Day 2-3: Auth system
- Day 4-5: File upload API + local storage
- Day 6-7: File listing & download

Week 2:
- Day 8-9: AI analysis pipeline
- Day 10: Semantic search
- Day 11: Duplicate detection
- Day 12: Deployment
- Day 13-14: Documentation & demo video

8. Deliverables
- Full repo with client/ & server/
- Working auth + file upload/download
- AI tagging + semantic search
- Duplicate detection & cleanup suggestions
- Deployment ready for portfolio