

CA Document Management System - Project Overview

Project Summary

A secure document management system designed specifically for Chartered Accountants (CAs) with enterprise-grade encryption, modern UI, and cloud storage integration. The system allows CAs to securely upload, store, organize, and manage client documents with military-grade encryption.

Technical Architecture

Frontend Stack

- **React 18** with **JSX** (NOT TypeScript)
- **ShadCN UI** components for consistent design
- **Tailwind CSS** for styling with custom CA theme
- **React Router** for navigation
- **Axios** for API communication
- **Lucide React** for icons
- **Crypto-JS** for client-side encryption

Backend Stack

- **Laravel 10** (PHP 8.1+)
- **Laravel Sanctum** for authentication
- **MySQL/PostgreSQL** database
- **Wasabi S3** for encrypted file storage
- **AWS S3 SDK** for Wasabi integration

Security Layer

- **Client-side AES-256 encryption** before upload
- **Server-side encryption** as backup layer
- **Unique encryption keys** per document
- **Encrypted storage** on Wasabi S3
- **Zero-knowledge architecture** - even Wasabi can't read documents

Design Theme & Colors

CA Professional Theme

CSS

```
:root {  
  /* Primary Theme Colors */  
  --ca-primary: #1e40af;    /* Professional Blue */  
  --ca-secondary: #059669; /* Success Green */  
  --ca-accent: #dc2626;     /* Important Red */  
  --ca-neutral: #64748b;    /* Slate Gray */  
  --ca-light: #f8fafc;      /* Light Background */  
  --ca-dark: #0f172a;       /* Dark Text */  
}
```

Design Principles

- **Professional & Trustworthy:** Blue-dominant color scheme
- **Clean & Minimal:** Focused on functionality
- **Accessible:** High contrast ratios
- **Responsive:** Mobile-first design
- **Consistent:** ShadCN component system

Project Structure

CAVault/

```
├── client/                # React Frontend
│   ├── src/
│   │   ├── components/
│   │   │   ├── ul/      # ShadCN components
│   │   │   ├── layout/  # Layout components
│   │   │   ├── auth/    # Authentication components
│   │   │   ├── documents/ # Document management
│   │   │   └── dashboard/ # Dashboard components
│   │   ├── pages/
│   │   │   ├── login.jsx
│   │   │   ├── register.jsx
│   │   │   ├── Dashboard.jsx
│   │   │   └── Documents.jsx
│   │   ├── hooks/
│   │   │   ├── useAuth.js
│   │   │   └── useDocuments.js
│   │   ├── services/
│   │   │   ├── api.js
│   │   │   └── encryption.js
│   │   └── utils/
│   ├── jsconfig.json     # Path resolution for @/ imports
│   ├── components.json   # ShadCN configuration
│   └── tailwind.config.js # Tailwind with CA theme
└── backend/              # Laravel Backend
    ├── app/
    │   ├── Http/Controllers/
    │   ├── Models/
    │   └── Services/
    └── config/
```



Security Implementation

Multi-Layer Encryption Strategy

Layer 1: Client-Side Encryption

javascript

// Before upload - encrypt with unique key per document

```
const encryptionKey = generateEncryptionKey();
```

```
const encryptedContent = encryptFile(fileContent, encryptionKey);
```

Layer 2: Server-Side Processing

php

```
// Laravel - Additional encryption before Wasabi storage
$encryptedContent = $this->encryptionService->encryptFile($content, $key);
```

Layer 3: Wasabi Storage

php

```
// Upload with server-side encryption
$this->s3Client->putObject([
    'Bucket' => $bucket,
    'Key' => $filePath,
    'Body' => $encryptedContent,
    'ServerSideEncryption' => 'AES256'
]);
```

Key Security Features

- **Zero-Knowledge Storage:** Even if Wasabi is breached, documents remain encrypted
- **Unique Keys:** Each document has its own encryption key
- **Key Rotation:** Regular key rotation for enhanced security
- **Audit Trail:** Complete access logging
- **Compliance Ready:** GDPR, HIPAA compliant

Key Components & Features

Authentication System

- **Multi-step Registration:** Personal info → Professional details → Confirmation
- **Secure Login:** Email/password with remember me
- **JWT Token Management:** Automatic token refresh
- **Protected Routes:** Role-based access control

Document Management

- **Drag & Drop Upload:** Modern file upload interface
- **Category Organization:** Tax Returns, Financial Statements, etc.
- **Advanced Search:** Full-text search with filters

- **Secure Download:** Decryption on-the-fly
- **Document Versioning:** Track document changes

Dashboard

- **Quick Stats:** Document count, storage usage, recent activity
- **Recent Documents:** Quick access to latest uploads
- **Storage Analytics:** Visual storage usage charts
- **Activity Feed:** Recent user activities

Development Setup

Frontend Configuration

Tailwind Config (tailwind.config.js)

javascript

```
module.exports = {
  darkMode: ["class"],
  content: ['./src/**/*.{js,jsx}'],
  theme: {
    extend: {
      colors: {
        ca: {
          primary: "#1e40af",
          secondary: "#059669",
          accent: "#dc2626",
          neutral: "#64748b",
          light: "#f8fafc",
          dark: "#0f172a",
        }
      }
    }
  }
}
```

ShadCN Components.json

json

```
{
  "$schema": "https://ui.shadcn.com/schema.json",
  "style": "default",
  "rsc": false,
  "tsx": false,
  "tailwind": {
    "config": "tailwind.config.js",
    "css": "src/index.css"
  },
  "aliases": {
    "components": "@components",
    "utils": "@lib/Utils"
  }
}
```

JSConfig for Path Resolution

```
json
{
  "compilerOptions": {
    "baseUrl": ".",
    "paths": {
      "@/*": ["/src/*"]
    }
  }
}
```

Required Dependencies

```
bash
```

Core React packages

`npm install react react-dom react-router-dom`

ShadCN and UI

`npm install @radix-ui/react-dialog @radix-ui/react-dropdown-menu`

`npm install lucide-react clsx tailwind-merge`

`npm install class-variance-authority @radix-ui/react-slot`

API and utilities

`npm install axios crypto-js`

`npm install tailwindcss-animate`

ShadCN Components Used

bash

`npx shadcn@latest add button input label card`

`npx shadcn@latest add avatar dropdown-menu sheet`

`npx shadcn@latest add select alert badge toast`

`npx shadcn@latest add tabs table progress tooltip`

UI/UX Design Patterns

Layout Structure

- **Fixed Header:** Navigation, user menu, notifications
- **Collapsible Sidebar:** Main navigation with CA branding
- **Main Content:** Dashboard/documents area
- **Mobile Responsive:** Hamburger menu, touch-friendly

Component Design

- **Cards:** Clean white cards with subtle shadows
- **Buttons:** Primary blue, secondary outlined
- **Forms:** Multi-step with progress indicators
- **Icons:** Lucide React for consistency
- **Typography:** Clean, professional fonts

Animations & Interactions

- **Smooth Transitions:** 300ms ease-out

- **Hover Effects:** Subtle color changes
- **Loading States:** Spinners and skeleton screens
- **Progress Indicators:** Step-by-step flows

API Integration

Authentication Endpoints

javascript

// Login

POST /api/login { email, password }

// Register

POST /api/register { name, email, password, ca_license_number, ... }

// Get User

GET /api/user

// Logout

POST /api/logout

Document Endpoints

javascript

// Upload Document

POST /api/documents (FormData with file)

// Get Documents

GET /api/documents?category=&search=

// Download Document

GET /api/documents/:id/download

// Delete Document

DELETE /api/documents/:id

Deployment Strategy

Production Build

bash


```
# Build React frontend
```

```
npm run build
```

```
# Copy to Laravel public directory
```

```
cp -r build/* ../backend/public/app/
```

```
# Deploy Laravel backend
```

```
composer install --optimize-autoloader --no-dev
```

```
php artisan config:cache
```

```
php artisan route:cache
```

Environment Configuration

```
env
```

```
# Backend (.env)
```

```
WASABI_ACCESS_KEY_ID=your_key
```

```
WASABI_SECRET_ACCESS_KEY=your_secret
```

```
WASABI_BUCKET=ca-documents-secure
```

```
WASABI_ENDPOINT=https://s3.wasablsys.com
```

```
# Frontend
```

```
REACT_APP_API_URL=https://yourdomain.com/api
```

Sample Component Structure

Example: Document Upload Component

```
javascript
```

```
// src/components/documents/DocumentUpload.jsx
import React, { useState } from 'react';
import { Button } from '@components/ui/button';
import { Card, CardContent, CardHeader, CardTitle } from '@components/ui/card';
import { Upload, FileText } from 'lucide-react';

const DocumentUpload = ({ onUploadSuccess }) => {
  const [selectedFile, setSelectedFile] = useState(null);
  const [uploading, setUploading] = useState(false);

  const handleUpload = async () => {
    // Implementation here
  };

  return (
    <Card className="w-full max-w-md mx-auto ca-shadow">
      <CardHeader>
        <CardTitle className="flex items-center gap-2">
          <Upload className="w-5 h-5" />
          Upload Document
        </CardTitle>
      </CardHeader>
      <CardContent>
        { /* Upload UI */ }
      </CardContent>
    </Card>
  );
};

export default DocumentUpload;
```

Development Guidelines

Code Standards

- **JSX Only:** No TypeScript, use regular JavaScript
- **ShadCN Components:** Always use ShadCN for UI consistency
- **CA Theme:** Use ca-primary, ca-secondary colors
- **Responsive:** Mobile-first design approach
- **Accessibility:** ARIA labels, keyboard navigation

Naming Conventions

- **Components:** PascalCase (DocumentUpload.jsx)
- **Hooks:** camelCase starting with 'use' (useAuth.js)
- **Services:** camelCase (api.js, authService.js)
- **CSS Classes:** Tailwind utilities + custom ca-* classes

File Organization

- **Pages:** Top-level route components
- **Components:** Reusable UI components
- **Hooks:** Custom React hooks
- **Services:** API and business logic
- **Utils:** Helper functions

Testing Strategy

- **Component Testing:** Jest + React Testing Library
- **E2E Testing:** Cypress for user flows
- **API Testing:** Laravel Feature Tests
- **Security Testing:** Penetration testing for encryption

Success Metrics

- **Security:** 100% encrypted documents
- **Performance:** <3s page load times
- **Reliability:** 99.9% uptime
- **User Experience:** <5 clicks for common tasks
- **Scalability:** Support 1000+ concurrent users

Future Enhancements

- **Mobile App:** React Native version
 - **Advanced Search:** AI-powered document search
 - **Collaboration:** Multi-user document sharing
 - **Integration:** Accounting software connections
 - **Analytics:** Document usage analytics
-

Quick Reference for AI

When working on this project, remember:

- Use **JSX**, not **TypeScript**
- Always use **ShadCN components** for UI
- Follow **CA professional theme** colors
- Implement **security-first** approach
- Create **responsive, accessible** designs
- Use **@/ imports** for components
- Follow **multi-layer encryption** strategy
- Design for **Chartered Accountants** specifically

This is a **production-ready, enterprise-grade document management system** focused on security, professionalism, and user experience for CA firms.