SafeDocx Frontend Setup Guide

🚀 Quick Start (5 Steps)

Step 1: Initialize Vite + React + TypeScript

```bash
cd safedocx-dms/frontend

# Create Vite project
npm create vite@latest . -- --template react-ts

# Install dependencies
npm install

# Install additional packages
npm install react-router-dom axios zustand react-hook-form zod @hookform/resolvers lucide-react date-fns rea

# Install dev dependencies
npm install -D tailwindcss postcss autoprefixer @types/node
npx tailwindcss init -p
```

Step 2: Configure Tailwind CSS

Replace `tailwind.config.js` with the provided configuration (see artifact).

Update `src/index.css`:

```css
@tailwind base;
@tailwind components;
@tailwind utilities;

@layer base {
  * {
    @apply border-border;
  }
  body {
    @apply bg-background text-foreground;
    font-feature-settings: "rlig" 1, "calt" 1;
  }
}
```

Step 3: Create Project Structure

```bash
cd src

# Create directories
mkdir -p components/{ui,documents,folders,common}
mkdir -p pages/{Auth,Dashboard,Documents,Settings}
mkdir -p layouts
mkdir -p store
mkdir -p services
mkdir -p utils
mkdir -p hooks
mkdir -p types

# Create files
touch components/ui/{Button,Input,Card,Modal,Toast}.tsx
touch pages/Auth/{LoginPage,RegisterPage}.tsx
touch pages/Dashboard/DashboardPage.tsx
touch layouts/DashboardLayout.tsx
touch store/{authStore,documentStore}.ts
touch services/{api,authService,documentService}.ts
touch utils/{cn,formatters,fileTypes}.ts
touch types/index.ts
```

Step 4: Environment Variables

Create `.env` file:

```bash
VITE_API_URL=http://localhost:5000/api/v1
VITE_APP_NAME=SafeDocx Rwanda
```

Step 5: Set Up Routing

Create `src/App.tsx`:

```tsx
```

```tsx
import { BrowserRouter, Routes, Route, Navigate } from 'react-router-dom';
import { Toaster } from 'react-hot-toast';
import LoginPage from './pages/Auth/LoginPage';
import RegisterPage from './pages/Auth/RegisterPage';
import DashboardLayout from './layouts/DashboardLayout';
import DashboardPage from './pages/Dashboard/DashboardPage';
import DocumentsPage from './pages/Documents/DocumentsPage';
import { useAuthStore } from './store/authStore';

// Protected Route Component
const ProtectedRoute = ({ children }: { children: React.ReactNode }) => {
 const { isAuthenticated } = useAuthStore();
 return isAuthenticated ? <>{children}</> : <Navigate to="/login" />;
};

function App() {
 return (
  <BrowserRouter>
   <Toaster position="top-right" />
   <Routes>
    {/* Public Routes */}
    <Route path="/login" element={<LoginPage />} />
    <Route path="/register" element={<RegisterPage />} />

    {/* Protected Routes */}
    <Route
     path="/"
     element={
      <ProtectedRoute>
       <DashboardLayout />
      </ProtectedRoute>
     }
    >
     <Route index element={<Navigate to="/dashboard" />} />
     <Route path="dashboard" element={<DashboardPage />} />
     <Route path="documents" element={<DocumentsPage />} />
     <Route path="favorites" element={<div>Favorites Page</div>} />
     <Route path="recent" element={<div>Recent Page</div>} />
     <Route path="trash" element={<div>Trash Page</div>} />
     <Route path="settings" element={<div>Settings Page</div>} />
     <Route path="profile" element={<div>Profile Page</div>} />
    </Route>

    {/* 404 */}
    <Route path="*" element={<div>404 Not Found</div>} />
   </Routes>
```

```
    </BrowserRouter>
  );
}

export default App;
```

📂 Complete File Structure

```
frontend/
├── public/
├── src/
│   ├── components/
│   │   ├── ui/
│   │   │   ├── Button.tsx          ✅ Created
│   │   │   ├── Input.tsx           ✅ Created
│   │   │   ├── Card.tsx
│   │   │   ├── Modal.tsx
│   │   │   ├── Dropdown.tsx
│   │   │   ├── Table.tsx
│   │   │   └── Toast.tsx
│   │   ├── documents/
│   │   │   ├── DocumentCard.tsx
│   │   │   ├── DocumentList.tsx
│   │   │   ├── DocumentPreview.tsx
│   │   │   ├── UploadZone.tsx
│   │   │   └── DocumentFilters.tsx
│   │   ├── folders/
│   │   │   ├── FolderTree.tsx
│   │   │   └── FolderBreadcrumb.tsx
│   │   └── common/
│   │       ├── SearchBar.tsx
│   │       ├── UserMenu.tsx
│   │       └── Sidebar.tsx
│   ├── pages/
│   │   ├── Auth/
│   │   │   ├── LoginPage.tsx       ✅ Created
│   │   │   ├── RegisterPage.tsx
│   │   │   └── ForgotPasswordPage.tsx
│   │   ├── Dashboard/
│   │   │   └── DashboardPage.tsx
│   │   ├── Documents/
│   │   │   ├── DocumentsPage.tsx
│   │   │   └── DocumentDetailPage.tsx
│   │   ├── Workflows/
```

```
│   │   │   └── WorkflowsPage.tsx
│   │   └── Settings/
│   │       └── SettingsPage.tsx
│   ├── layouts/
│   │   ├── DashboardLayout.tsx    ✅ Created
│   │   └── AuthLayout.tsx
│   ├── store/
│   │   ├── authStore.ts           ✅ Created
│   │   ├── documentStore.ts
│   │   └── uiStore.ts
│   ├── services/
│   │   ├── api.ts
│   │   ├── authService.ts
│   │   ├── documentService.ts
│   │   └── uploadService.ts
│   ├── hooks/
│   │   ├── useDocuments.ts
│   │   ├── useUpload.ts
│   │   └── useSearch.ts
│   ├── utils/
│   │   ├── cn.ts                  ✅ Created
│   │   ├── formatters.ts          ✅ Created
│   │   └── fileTypes.ts           ✅ Created
│   ├── types/
│   │   └── index.ts
│   ├── App.tsx
│   ├── main.tsx
│   └── index.css
├── .env
├── .env.example
├── vite.config.ts
├── tsconfig.json
├── tailwind.config.js             ✅ Created
├── postcss.config.js
└── package.json                   ✅ Created
```

🎨 Component Implementation Priority

Phase 1: Core UI Components (Week 1)

1. ✅ Button
2. ✅ Input
3. Card

4. Modal/Dialog
5. Dropdown
6. Table
7. Toast/Notification

Phase 2: Auth & Layout (Week 2)

1. ✅ LoginPage
2. RegisterPage
3. ✅ DashboardLayout
4. Sidebar Navigation
5. TopBar with Search
6. User Menu

Phase 3: Document Components (Week 3)

1. DocumentCard
2. DocumentList
3. UploadZone
4. DocumentFilters
5. DocumentPreview
6. FolderTree

Phase 4: Advanced Features (Week 4+)

1. Workflow Dashboard
2. Analytics Dashboard
3. Settings Pages
4. Mobile Responsive
5. Dark Mode

🔧 Additional Components to Create

Card Component

```tsx
```

```tsx
// src/components/ui/Card.tsx
import React from 'react';
import { cn } from '../../utils/cn';

interface CardProps extends React.HTMLAttributes<HTMLDivElement> {
  variant?: 'default' | 'outlined';
}

export const Card = ({ className, variant = 'default', ...props }: CardProps) => {
  return (
    <div
      className={cn(
        'rounded-lg',
        variant === 'default' && 'bg-white shadow-md',
        variant === 'outlined' && 'border-2 border-gray-200',
        className
      )}
      {...props}
    />
  );
};

export const CardHeader = ({ className, ...props }: React.HTMLAttributes<HTMLDivElement>) => (
  <div className={cn('px-6 py-4 border-b border-gray-200', className)} {...props} />
);

export const CardBody = ({ className, ...props }: React.HTMLAttributes<HTMLDivElement>) => (
  <div className={cn('px-6 py-4', className)} {...props} />
);

export const CardFooter = ({ className, ...props }: React.HTMLAttributes<HTMLDivElement>) => (
  <div className={cn('px-6 py-4 border-t border-gray-200', className)} {...props} />
);
```

Modal Component

tsx

```tsx
// src/components/ui/Modal.tsx
import React, { useEffect } from 'react';
import { X } from 'lucide-react';
import { cn } from '../../utils/cn';
import Button from './Button';

interface ModalProps {
  isOpen: boolean;
  onClose: () => void;
  title?: string;
  children: React.ReactNode;
  size?: 'sm' | 'md' | 'lg' | 'xl';
  showCloseButton?: boolean;
}

const Modal: React.FC<ModalProps> = ({
  isOpen,
  onClose,
  title,
  children,
  size = 'md',
  showCloseButton = true,
}) => {
  useEffect(() => {
    if (isOpen) {
      document.body.style.overflow = 'hidden';
    } else {
      document.body.style.overflow = 'unset';
    }
    return () => {
      document.body.style.overflow = 'unset';
    };
  }, [isOpen]);

  if (!isOpen) return null;

  const sizeClasses = {
    sm: 'max-w-md',
    md: 'max-w-2xl',
    lg: 'max-w-4xl',
    xl: 'max-w-6xl',
  };

  return (
    <div className="fixed inset-0 z-50 overflow-y-auto">
      {/* Backdrop */}
```

```tsx
      <div
        className="fixed inset-0 bg-black bg-opacity-50 transition-opacity"
        onClick={onClose}
      />

      {/* Modal */}
      <div className="flex min-h-full items-center justify-center p-4">
        <div
          className={cn(
            'relative bg-white rounded-lg shadow-xl w-full',
            sizeClasses[size],
            'transform transition-all animate-scale-in'
          )}
          onClick={(e) => e.stopPropagation()}
        >
          {/* Header */}
          {title && (
            <div className="flex items-center justify-between px-6 py-4 border-b border-gray-200">
              <h2 className="text-xl font-semibold text-gray-900">{title}</h2>
              {showCloseButton && (
                <button
                  onClick={onClose}
                  className="p-2 hover:bg-gray-100 rounded-lg transition-colors"
                >
                  <X size={20} />
                </button>
              )}
            </div>
          )}

          {/* Body */}
          <div className="px-6 py-4">{children}</div>
        </div>
      </div>
    </div>
  );
};

export default Modal;
```

Document Card Component

```tsx
```

```tsx
// src/components/documents/DocumentCard.tsx
import React from 'react';
import { FileText, MoreVertical, Download, Share2, Trash2 } from 'lucide-react';
import { formatFileSize, formatDate } from '../../utils/formatters';

interface DocumentCardProps {
  document: {
    id: string;
    name: string;
    size: number;
    updatedAt: string;
    type: string;
  };
  onSelect?: (id: string) => void;
  onDownload?: (id: string) => void;
  onShare?: (id: string) => void;
  onDelete?: (id: string) => void;
}

const DocumentCard: React.FC<DocumentCardProps> = ({
  document,
  onSelect,
  onDownload,
  onShare,
  onDelete,
}) => {
  const [showMenu, setShowMenu] = React.useState(false);

  return (
    <div
      className="group relative bg-white rounded-lg border-2 border-gray-200 hover:border-primary-500 hover:s
      onClick={() => onSelect?.(document.id)}
    >
      {/* Document Icon */}
      <div className="aspect-square flex items-center justify-center bg-gradient-to-br from-primary-50 to-primar
        <FileText className="w-16 h-16 text-primary-600" />
      </div>

      {/* Document Info */}
      <div className="p-4">
        <h3 className="font-medium text-gray-900 truncate mb-1">{document.name}</h3>
        <div className="flex items-center justify-between text-sm text-gray-500">
          <span>{formatFileSize(document.size)}</span>
          <span>{formatDate(document.updatedAt)}</span>
        </div>
      </div>
```

```
{/* Actions Menu */}
<div className="absolute top-2 right-2 opacity-0 group-hover:opacity-100 transition-opacity">
 <button
  onClick={(e) => {
   e.stopPropagation();
   setShowMenu(!showMenu);
  }}
  className="p-2 bg-white rounded-lg shadow-md hover:bg-gray-100"
 >
  <MoreVertical size={16} />
 </button>

 {showMenu && (
  <div className="absolute right-0 mt-2 w-48 bg-white rounded-lg shadow-lg border border-gray-200 py-1 z
   <button
    onClick={(e) => {
     e.stopPropagation();
     onDownload?.(document.id);
     setShowMenu(false);
    }}
    className="flex items-center w-full px-4 py-2 text-sm text-gray-700 hover:bg-gray-100"
   >
    <Download size={16} className="mr-3" />
    Download
   </button>
   <button
    onClick={(e) => {
     e.stopPropagation();
     onShare?.(document.id);
     setShowMenu(false);
    }}
    className="flex items-center w-full px-4 py-2 text-sm text-gray-700 hover:bg-gray-100"
   >
    <Share2 size={16} className="mr-3" />
    Share
   </button>
   <div className="border-t border-gray-200 my-1" />
   <button
    onClick={(e) => {
     e.stopPropagation();
     onDelete?.(document.id);
     setShowMenu(false);
    }}
    className="flex items-center w-full px-4 py-2 text-sm text-red-600 hover:bg-gray-100"
   >
    <Trash2 size={16} className="mr-3" />
```

```
          Delete
        </button>
      </div>
    ))}
  </div>
 </div>
 );
};

export default DocumentCard;
```

🚀 Running the Frontend

```bash
# Development
npm run dev

# Build for production
npm run build

# Preview production build
npm run preview
```

The app will run on `http://localhost:5173`

📱 Testing Your Setup

1. **Test Login Flow:**
   - Navigate to http://localhost:5173/login
   - Enter credentials you created in backend
   - Should redirect to /dashboard
2. **Test Protected Routes:**
   - Try accessing /dashboard without login
   - Should redirect to /login
3. **Test Sidebar Navigation:**
   - Click different menu items
   - Observe active states

4. **Test Responsive Design:**
   - Resize browser window
   - Mobile menu should appear < 1024px

🎯 Next Steps

Immediate (This Week)

1. Create RegisterPage component
2. Implement DocumentsPage with grid/list view
3. Create UploadZone component
4. Add document API service

Short-term (2-3 Weeks)

1. Document preview modal
2. Search functionality
3. Folder navigation
4. File upload with progress
5. Document filters

Medium-term (1-2 Months)

1. Workflow dashboard
2. Analytics page
3. Settings page
4. Mobile responsive polish
5. Dark mode

🐛 Common Issues & Solutions

Issue: Tailwind classes not working

Solution: Make sure `index.css` imports Tailwind directives

Issue: API calls failing (CORS)

Solution: Check backend CORS configuration allows http://localhost:5173

Issue: Authentication not persisting

Solution: Check Zustand persist middleware configuration

Issue: Icons not showing

Solution: Ensure `lucide-react` is installed: `npm install lucide-react`

📚 Helpful Resources

- **Tailwind CSS Docs:** https://tailwindcss.com/docs
- **Lucide Icons:** https://lucide.dev/icons
- **React Hook Form:** https://react-hook-form.com
- **Zustand:** https://github.com/pmndrs/zustand
- **Vite:** https://vitejs.dev

You now have everything to build a modern, production-ready frontend! Start with the MVP features and iterate based on feedback. 🥕