OUC ACCESS CONTROL SYSTEM - REACT PROJECT STRUCTURE

=====================================================

PROJECT OVERVIEW

-----------------

This is a Next.js 13+ application using the App Router pattern with TypeScript,

Tailwind CSS, and a comprehensive API structure for the OUC Access Control System.

COMPLETE FILE STRUCTURE

------------------------

OUCAccess/

│

├── 📁 public/ # Static assets

│ ├── 🖼️ ouc-logo.png # OUC organization logo (PNG format)

│ ├── 🖼️ ouc-logo.svg # OUC organization logo (SVG format)

│ ├── 🖼️ google.svg # Google sign-in icon

│ ├── 🖼️ default-profile.png # Default user profile image

│ ├── 🖼️ PhotoID.jpeg # Sample photo ID image

│ ├── 🖼️ church-exterior.jpg # Church building image

│ ├── 🖼️ logo.png # Alternative logo file

│ └── 📁 uploads/ # User uploaded files directory

│

├── 📁 src/ # Source code directory

│ ├── 📁 app/ # Next.js App Router (Pages & API)

│ │ ├── 📁 api/ # Backend API routes

│ │ │ ├── 📁 auth/ # Authentication endpoints

│ │ │ │ ├── config.ts # Auth configuration

│ │ │ │ └── [...nextauth]/ # NextAuth.js routes

│ │ │ ├── 📁 church-members/ # Member management API

│ │ │ │ └── route.ts # CRUD operations for members

│ │ │ ├── 📁 send-email/ # Email notification API

│ │ │ │ └── route.ts # Email sending functionality

│ │ │ ├── 📁 access-request/ # Access request API

│ │ │ │ └── route.ts # Handle access requests

│ │ │ ├── 📁 health/ # Health check API

│ │ │ │ └── route.ts # Server health monitoring

│ │ │ └── 📁 upload/ # File upload API

│ │ │ └── route.ts # Handle file uploads

│ │ │

│ │ ├── 📁 login/ # Login page directory

│ │ │ └── page.tsx # Login form component

│ │ ├── 📁 access-request/ # Access request page directory

│ │ │ └── page.tsx # Main access request form page

│ │ ├── 📁 admin/ # Admin pages directory

│ │ │ └── page.tsx # Admin dashboard

│ │ ├── 📄 page.tsx # Home page (root route)

│ │ ├── 📄 layout.tsx # Root layout component

│ │ └── 🎨 globals.css # Global CSS styles

│ │

│ ├── 📁 components/ # Reusable React components

│ │ ├── 📄 AccessRequestForm.tsx # Main form component (26KB)

│ │ ├── 📄 Providers.tsx # Auth context providers

│ │ ├── 📄 PhotoFrame.tsx # Photo upload/display component

│ │ ├── 📄 GoogleSignInButton.tsx # Google authentication button

│ │ └── 📁 admin/ # Admin-specific components

│ │ └── EditMemberModal.tsx # Member editing modal

│ │

│ ├── 📁 lib/ # Utility libraries

│ │ ├── db.ts # Database connection utilities

│ │ └── services/ # Service layer functions

│ ├── 📁 hooks/ # Custom React hooks

│ ├── 📁 types/ # TypeScript type definitions

│ │ └── database.ts # Database type definitions

│ ├── 📁 db/ # Database configurations

│ │ ├── init.sql # Database initialization script

│ │ └── migrations/ # Database migration files

│ ├── 📁 scripts/ # Build/deployment scripts

│ │ └── run-migrations.js # Database migration runner

│ ├── 📁 styles/ # Additional stylesheets

│ └── 📄 middleware.ts # Next.js middleware

│

├── 📁 node\_modules/ # NPM dependencies (auto-generated)

├── 📁 .next/ # Next.js build output (auto-generated)

├── 📁 .git/ # Git repository data

├── 📁 .cursor/ # Cursor IDE configuration

│

├── 📄 package.json # Project dependencies & scripts

├── 📄 package-lock.json # Dependency version lock file

├── 📄 next.config.js # Next.js configuration

├── 📄 tailwind.config.js # Tailwind CSS configuration

├── 📄 tsconfig.json # TypeScript configuration

├── 📄 next-env.d.ts # Next.js TypeScript definitions

├── 📄 server.js # Custom server configuration

├── 📄 README.md # Project documentation

├── 📄 .gitignore # Git ignore rules

│

├── 📄 deploy.bat # Windows deployment script

├── 📄 deploy.sh # Unix/Linux deployment script

├── 📄 deploy-Github.bat # GitHub deployment script

└── 📄 DeployToStaging.txt # Staging deployment instructions

KEY COMPONENTS BREAKDOWN

------------------------

AUTHENTICATION SYSTEM:

- NextAuth.js configuration in src/app/api/auth/

- Google OAuth integration

- Credential-based email authentication

- Session management and middleware

MAIN APPLICATION FEATURES:

- Login page with dual authentication (Google + Email)

- Access request form with member management

- Photo upload and display functionality

- Admin dashboard for member administration

- Email notification system

DATABASE INTEGRATION:

- MySQL database with church members table

- CRUD operations for member records

- Database migration scripts

- Connection pooling and error handling

API STRUCTURE:

- RESTful API endpoints for all operations

- Separate routes for different functionalities

- Proper error handling and response formatting

- Email sending capabilities with nodemailer

STYLING & UI:

- Tailwind CSS for responsive design

- Custom CSS in globals.css

- Component-based styling approach

- Navy blue theme with white text

DEVELOPMENT TOOLS:

- TypeScript for type safety

- Next.js 13+ App Router for modern routing

- Custom hooks for reusable logic

- Deployment scripts for multiple environments

STATIC ASSETS:

- Organization logos in multiple formats

- Profile images and placeholders

- Google authentication icons

- Uploaded user files storage

CONFIGURATION FILES:

- Environment variables for different deployments

- Next.js configuration for image optimization

- Tailwind configuration for custom themes

- TypeScript configuration for strict typing

DEPLOYMENT & SCRIPTS:

- Automated deployment scripts for Windows and Unix

- GitHub integration scripts

- Staging server deployment instructions

- Database migration utilities

PROJECT STATISTICS:

-------------------

- Main form component: 26KB (792 lines)

- Total package size: ~264KB dependencies

- Multiple deployment environments supported

- Full-stack TypeScript application

- Responsive design implementation

IMPORTANT NOTES:

----------------

1. This is a production-ready application structure

2. Follows Next.js 13+ best practices with App Router

3. Implements proper separation of concerns

4. Includes comprehensive error handling

5. Supports multiple authentication methods

6. Designed for scalability and maintenance

END OF DOCUMENT