Companies Management Application (TASK 2), Assignment Version Number: FE-2024-12-i

A modern, feature-rich React/Next.js application for managing company information. Built with React, Next.js, TypeScript, Apollo GraphQL, and tailwind css. It offers a seamless user experience with fast loading times, server rendering capabilites, and a modern design.

🌟 FEATURES

Features

- 🔐 **Authentication**: Secure login/logout and session management with OAuth (Github and Google)
- 🏢 **Company Management**: CRUD operations for company data
- 🔍 **URL-based Company Details Pre-filling**: Auto-populate company details page via `companyID` query parameter
- 📋 **Company Listing**: Comprehensive view of all created companies
- 🛡️ **Access Control**: Route protection for authenticated users only

Technical Highlights

- ⚡ **Server-Side Rendering**: Optimized performance with Next.js 15
- 🔗 **GraphQL Integration**: Seamless API communication with Apollo Client
- 📱 **Responsive Design**: Mobile-first approach with Tailwind CSS
- ✅ **Form Validation**: Robust validation using React Hook Form + Zod
- 🧪 **Comprehensive Testing**: Unit and integration tests with Jest
- 🎨 **Modern UI**: Clean, professional interface with modern feel

🏗️ ARCHITECTURE

Tech Stack

- **Frontend Framework**: Next.js 15.2.3 (React 19) with App Router
- **Design Pattern**: Atomic Design (Atoms → Molecules → Organisms → Views)
- **Styling**: Tailwind CSS 4 with custom utility variants
- **Form Handling**: React Hook Form + Zod validation schemas
- **Authentication**: NextAuth.js 5.0
- **Testing**: Jest + React Testing Library with comprehensive coverage
- **TypeScript**: 100% type safety

Project Structure

The application follows a sophisticated **Atomic Design Pattern** with Next.js App Router architecture:

```
src/
├── app/                           # Next.js 15 App Router
│   ├── (auth)/                    # Authentication route group
│   │   ├── sign-in/             # Login page
│   │   └── sign-in-error/       # Error handling
│   ├── (dashboard)/             # Protected dashboard routes
│   │   └── companies/           # Company management
│   │       └── [companyId]/     # Dynamic company details
│   ├── actions/                 # Server Actions
│   │   ├── auth.actions.ts      # Authentication logic
│   │   ├── companies.actions.ts # Company CRUD operations
│   │   ├── countries.actions.ts # Geography data for countries and states input fields
│   │   └── upload.actions.ts    # File upload handling
│   ├── api/                     # API Routes
│   │   ├── auth/[...nextauth]/  # NextAuth.js integration
│   │   └── s3-image/[s3Key]/    # Image serving endpoint ( with performance-optimizing cache implementation)
│   └── layout.tsx               # Root layout with context providers
│
├── components/                   # Atomic Design Architecture
│   ├── atoms/                   # Basic building blocks, A zero-dependency full suite of re-usable components library build from the scratch!
│   │   ├── Alert/              # Notification components
│   │   ├── Button/             # Interactive buttons
│   │   ├── Card/               # Content containers
│   │   ├── Dialog/             # Modal dialogs
│   │   ├── Input/              # Form inputs
│   │   ├── Select/             # Dropdown selects
│   │   ├── Toast/              # Toast notifications
│   │   └── [12+ more atoms]    # Comprehensive UI library
│   ├── molecules/              # Combined atom components
│   │   ├── CompanyLogo/        # Logo display component
│   │   ├── FormField/          # Form field compositions
│   │   └── layout/             # Header/Footer components
│   ├── organisms/              # Complex UI sections
│   │   ├── CreateCompanyDialog/ # Company creation modal
│   │   ├── EditLogoDialog/     # Logo editing interface
│   │   └── Forms/              # Multi-step form system
│   └── views/                  # Page-level components
│       ├── Companies/          # Company listing view
│       ├── CompanyDetails/     # Detailed company view
│       ├── CompanyPreview/     # Company preview
│       └── Custom404/          # Error page
│
├── lib/                        # Core utilities and configurations
│   ├── constants/              # Application constants
│   │
│   ├── graphql/                # GraphQL operations and types
│   │
│   ├── hooks/                  # Custom React hooks
│   │
│   └── utils/                  # Helper functions
│
└── tests/
    ├── components/
    │   └── atoms/            # Atomic component tests
    │
    └── test-utils.tsx         # Testing utilities
```

🚀 DEVELOPMENT

Prerequisites

- Node.js 18+
- npm or yarn package manager

Installation

1. Extract the ZIP Archive Extract the ZIP file:

Open the extracted folder in a terminal:

```
cd companies-app
```

2. **Install dependencies**

```
npm install
```

3. **Run the development server**

```
npm run dev
```

4. **Open your browser** Navigate to http://localhost:3000

Available Scripts

```
npm run dev              # Start development server with hot reload

# Production builds
npm run build            # Optimized production build
npm start               # Start production server

# Testing suite
npm run test            # Run complete test suite
npm run test:watch      # Watch mode for TDD workflow
npm run test:coverage   # Generate comprehensive coverage reports
npm run test:ci         # CI/CD optimized test execution
npm run test:debug      # Debug failing tests with Node inspector
```

```
# Code quality
npm run lint            # ESLint with Next.js configuration
```

📖 USAGE GUIDE

Authentication

1. Access the hosted app on https://companies-app.root.sx (or locally http://localhost:3000)
2. Login through Github or Google
3. Navigate through `/companies` list page and explore the app

Company Management

Creating a New Company

1. Navigate to `/companies` and click "ADD COMPANY" button
2. Complete the multi-step form process:
    - **Company Details**: Name, description, industry, size
    - **Contact Information**: Email, phone, website
    - **Address Details**: Complete address with country/state selection
    - **Logo Upload**: JPEG/PNG files up to 2MB
3. Preview your company information
4. Submit to create the company

Editing Existing Companies

1. Visit `/companies/[companyId]` to view company details
2. Click the edit icon to open the logo upload modal
3. Upload new logo for the company

Extra Features

- **Export Functionality**: CSV export of company data
- **Search & Filters**: Advanced company filtering and search

🎨 UI/UX DESIGN

Design Principles

- **Accessibility First**: WCAG compliant components
- **Mobile Responsive**: Seamless experience across devices
- **Loading States**: Proper feedback during async operations
- **Error Handling**: Comprehensive error handling and user-friendly error messages

Component Library

- Reusable UI components with consistent styling
- Tailwind CSS variants for theme consistency

- Icon integration with Lucide React
- Form components with built-in validation states

🛡️ SECURITY & PERFORMANCE

Security Measures

- **Route Protection**: Authentication guards for sensitive areas
- **XSS Prevention**: Sanitized user inputs
- **CSRF Protection**: NextAuth.js security features

Performance Optimizations

- **Server-Side Rendering**: Improved SEO and initial load times
- **Code Splitting**: Automatic route-based splitting
- **Image Optimization**: Next.js built-in image optimization
- **Bundle Analysis**: Optimized package sizes

🔍 KEY TECHNICAL ACHIEVEMENTS

1. Sophisticated Architecture Implementation

- **Atomic Design Pattern**: 20+ reusable atoms, strategic molecule compositions
- **Route Groups**: Clean separation with `(auth)` and `(dashboard)` organization
- **Server Actions**: Modern Next.js 15 server-side operations
- **Dynamic Routing**: Flexible `[companyId]` parameter handling

2. Production-Ready Features

- **Authentication System**: NextAuth.js with custom providers and middleware
- **File Management**: File upload/download optimizations
- **Export Functionality**: CSV generation with custom hooks (`useJsonToCsv`)
- **Responsive Design**: Mobile-first approach with Tailwind CSS 4

3. Developer Experience Excellence

- **Comprehensive Testing**: 15+ component test files with mocking utilities
- **TypeScript Integration**: 100% type coverage with shared type definitions
- **Development Tools**: Turbopack integration, hot reload, debug configurations
- **Code Quality**: ESLint configuration with Next.js best practices

4. Performance Optimizations

- **Server-Side Rendering**: Full SSR implementation meeting requirements
- **Code Splitting**: Automatic route-based splitting
- **Image Optimization**: Next.js built-in image optimization
- **Bundle Analysis**: Optimized package sizes

🎯 Architectural Highlights

Atomic Design System

- **Atoms** (20+ components): Button, Input, Card, Dialog, Toast, Select, etc.
- **Molecules**: FormField compositions, CompanyLogo, Header/Footer
- **Organisms**: Multi-step Forms, Create/Edit Dialogs, Complex UI sections
- **Views**: Complete page implementations with business logic

Advanced Next.js Implementation

- **Route Groups**: `(auth)` and `(dashboard)` for logical organization
- **Server Actions**: Direct server-side operations in `app/actions/`
- **Dynamic Routes**: `[companyId]` for company-specific pages
- **API Routes**: Custom endpoints for S3 integration and authentication
- **Middleware**: Route protection and request handling

📝 TASK 2 REQUIREMENTS FULFILLMENT

✅ **GraphQL Integration**: Full implementation with Apollo Client
✅ **Login Screen**: Custom authentication without backend dependency
✅ **Access Control**: Protected routes for unauthorized users
✅ **URL Pre-filling**: Company data population via companyID parameter
✅ **File Upload**: Logo upload with preview and format validation
✅ **File Restrictions**: JPEG/PNG only, 2MB maximum size
✅ **Complete Form Fields**: All input fields utilized as required
✅ **Logical Grouping**: User-friendly field organization
✅ **Company Listing**: Comprehensive company management page
✅ **Error Handling**: Graceful error management throughout
✅ **SSR Framework**: Next.js with server-side rendering capabilities