

GROUP FOUR CapStone Project Documentation:

JobFinder Documentation

1. Conceptualization and Design

Overview

This is a full-stack job application platform built with modern web technologies, designed to connect job seekers with employers. The platform features a robust authentication system, user management, and job application functionality.

The System Uses the MVC Architecture

1. Model Layer (Backend)

Located in `/server/models/`, this layer handles data and business logic:

Database Models

- `UserModel.js`: Defines user schema and relationships
- `JobModel.js`: Manages job posting data structure
- `ApplicationModel.js`: Handles job application data
- `associations.js`: Defines relationships between models
- `index.js`: Centralizes model configurations

Key characteristics:

- Uses Sequelize ORM for database interactions
- Implements data validation and relationships

- Handles data persistence logic
- Manages business rules and constraints

2. View Layer (Frontend)

Located in `/client/app/` and `/client/Components/`, this layer handles the presentation:

Next.js App Router Structure

- `/app/page.tsx`: Main landing page
- `/app/layout.tsx`: Root layout component
- `/app/job/`: Job-related pages
- `/app/findwork/`: Job search interface
- `/app/myjobs/`: User's job management
- `/app/post/`: Job posting interface

Components

Located in `/client/Components/`:

- Reusable UI components
- Form components
- Layout components
- Feature-specific components

Key characteristics:

- Uses Next.js 13+ App Router
- Implements React components
- Handles UI rendering
- Manages user interactions
- Uses TailwindCSS for styling

GROUP FOUR CapStone Project Documentation:

3. Controller Layer (Backend)

Located in /server/controllers/, this layer handles request processing:

Controllers

- userController.js: Manages user-related operations
- jobController.js: Handles job-related operations

Key responsibilities:

- Processes HTTP requests
- Coordinates between Models and Views
- Implements business logic
- Handles data validation
- Manages response formatting

4. Additional Supporting Layers

Routes Layer (/server/routes/)

- Defines API endpoints
- Maps URLs to controllers
- Handles request routing
- Implements middleware

Middleware Layer (/server/middleware/)

- Authentication middleware
- Request validation
- Error handling
- Logging

Context Layer (/client/context/)

- Manages global state
- Handles authentication state
- Provides shared functionality
- Implements state management

Providers Layer (/client/providers/)

- Theme providers
- Authentication providers
- API providers
- Feature-specific providers

2. Core Functionality Implementation

User Authentication and Management

- Auth0 integration for secure login/signup
- User role management (jobseeker, employer, admin)
- Profile management with profile pictures
- Session handling with secure cookies

Key Features

1. User Management:

- User registration and profile creation
- Role-based access control
- Profile customization

2. Job Management:

- Job posting for employers
- Job search and filtering
- Application tracking

GROUP FOUR CapStone Project Documentation:

3. Application Process:

- Job application submission
- Application status tracking
- Communication system

3. Advanced Feature Development

Chapter President Functionalities

- Special access controls for chapter presidents
- Management dashboard
- Analytics and reporting capabilities

Data Analysis and Integration

- Job market trend analysis
- Application success metrics
- User engagement analytics

Module Integration

- Seamless integration between frontend and backend
- API-first architecture
- Real-time updates and notification

4. Evaluation and Optimization

System Review

1. Performance Metrics:

- Server response times
- Database query optimization
- Frontend rendering performance

2. Security Measures:

- Auth0 authentication
- CORS configuration

- Secure cookie handling
- Environment variable management

3. Code Quality:

- TypeScript for type safety
- ESLint for code quality
- Modular architecture

Key Files and Directories

Frontend (/client)

- /app: Next.js application routes
- /Components: Reusable UI components
- /context: React context providers
- /providers: Application providers
- /utils: Utility functions
- /types: TypeScript type definitions

Backend (/server)

- /routes: API route handlers
- /controllers: Business logic
- /models: Database models
- /middleware: Custom middleware
- /config: Configuration files
- /db: Database connection and setup

GROUP FOUR CapStone Project Documentation:

Dependencies and Technologies

Frontend Dependencies

- Next.js 15.0.3
- React 18
- TailwindCSS
- Radix UI components
- Axios for API calls
- React Hot Toast for notifications
- React Quill for rich text editing

Backend Dependencies

- Express.js
- Sequelize ORM
- MySQL2
- Auth0
- JWT
- Cookie Parser
- CORS