

# dotitude-new-front - Complete Analysis Report

## dotitude-new-front - Complete Analysis Report

Generated on 2025-09-02 23:23:24

---

## Project Overview

None

### ■ Project Overview

- **Primary Language:** TypeScript
- **Repository Size:** 18689 KB
- **Stars:** 0
- **Forks:** 0
- **Last Updated:** 2025-09-02T17:26:06Z

### ■ Technologies Used

- **Angular** (frameworks)
- **Jest** (testing)
- **Mocha** (testing)

See [Technology Analysis](#) for detailed information.

### ■ Requirements

### System Requirements

- TypeScript runtime environment
- Package managers as specified in dependency files

## Dependencies

- **Total Dependencies:** 50
- **Production:** 26
- **Development:** 24

See [Dependency Analysis](#) for detailed information.

## ■ ■ Setup Instructions

### Prerequisites

```
# Install TypeScript runtime # Install package managers
```

### Installation

```
# Clone the repository git clone https://github.com/DinalieLiyanage/dotitude-new-front cd dotitude-new-front # Install dependencies (adjust based on project type) # npm install # for Node.js projects # pip install -r requirements.txt # for Python projects # cargo build # for Rust projects
```

### Running the Application

```
# Start the application (adjust based on project type) # npm start # for Node.js projects # python main.py # for Python projects # cargo run # for Rust projects
```

## ■ Code Metrics

- **Total Files:** 50
- **Total Lines of Code:** 1,855
- **Functions:** 23
- **Classes:** 119

## ■ ■ Architecture

This project follows a modular architecture. See [Architecture Documentation](#) for detailed information.

## UML Diagrams

- [Use Case Diagram](#)
- [Class Diagram](#)
- [Component Diagram](#)

## ■ API Documentation

See [API Documentation](#) for endpoint information.

## ■ Contributing

1. Fork the repository 2. Create a feature branch 3. Make your changes 4. Submit a pull request

## ■ License

See LICENSE file for details.

---

*Analysis generated on 2025-09-02 23:23:18 by GitHub Codebase Analyzer*

---

# Technology Stack Analysis

## Overview

This document provides a comprehensive analysis of the technologies, frameworks, and libraries used in dotitude-new-front.

## Detected Technologies

## Frameworks

- **Angular** ■ (Confidence: 1.00)

## Testing

- **Jest** ■ (Confidence: 1.00)
- **Mocha** ■ (Confidence: 1.00)

## Technology Distribution

Total technologies detected: 3

- **Frameworks:** 1 technologies
- **Testing:** 2 technologies

## Recommendations

Based on the detected technology stack:

1. **Primary Stack:** The project appears to use TypeScript 2. **Development Environment:** Ensure all detected technologies are properly configured 3. **Dependencies:** Review the dependency analysis for security and updates

---

*Generated on 2025-09-02 23:23:18*

---

# Dependency Analysis

## Overview

This document provides a comprehensive analysis of the project dependencies for dotitude-new-front.

## Summary

- **Total Dependencies:** 50
- **Production Dependencies:** 26
- **Development Dependencies:** 24

## Dependencies by Package Manager

### NPM

#### Production Dependencies

- **@angular/animations** ~9.1.3
- **@angular/cdk** ^9.2.4
- **@angular/common** ~9.1.3
- **@angular/compiler** ~9.1.3
- **@angular/core** ~9.1.3
- **@angular/fire** ^6.0.0
- **@angular/flex-layout** ^9.0.0-beta.31
- **@angular/forms** ~9.1.3
- **@angular/material** ^9.2.4
- **@angular/platform-browser** ~9.1.3
- **@angular/platform-browser-dynamic** ~9.1.3
- **@angular/router** ~9.1.3
- **@swimlane/ngx-charts** ^14.0.0
- **angular-material-rail-drawer** 0.0.6
- **angular-notifier** ^6.0.1
- **apexcharts** ^3.20.1
- **firebase** ^7.13.2
- **moment** ^2.29.1
- **ng-apexcharts** ^1.5.1
- **ng2-completer** ^9.0.1
- **ng2-google-charts** 3.4.0
- **ng2-smart-table** ^1.7.0
- **ngx-image-cropper** ^3.2.1
- **rxjs** ~6.5.4
- **tslib** ^1.10.0
- **zone.js** ~0.10.2

#### Development Dependencies

- **@angular-devkit/architect** ~0.900
- **@angular-devkit/build-angular** ~0.901.3
- **@angular/cli** ~9.1.3
- **@angular/compiler-cli** ~9.1.3
- **@angular/language-service** ~9.1.3
- **@types/jasmine** ~3.5.0
- **@types/jasminewd2** ~2.0.3
- **@types/node** ^12.11.1
- **codelyzer** ^5.1.2
- **firebase-tools** ^8.0.0
- **fuzzy** ^0.1.3
- **inquirer** ^6.2.2
- **inquirer-autocomplete-prompt** ^1.0.1
- **jasmine-core** ~3.5.0
- **jasmine-spec-reporter** ~4.2.1
- **karma** ~5.0.0
- **karma-chrome-launcher** ~3.1.0
- **karma-coverage-istanbul-reporter** ~2.1.0
- **karma-jasmine** ~3.0.1
- **karma-jasmine-html-reporter** ^1.4.2
- **protractor** ~5.4.3
- **ts-node** ~8.3.0
- **tslint** ~6.1.0
- **typescript** ~3.8.3

## Security Considerations

1. **Regular Updates:** Keep dependencies up to date to avoid security vulnerabilities
2. **Vulnerability Scanning:** Use tools like `npm audit`, `safety`, or `cargo audit`
3. **License Compliance:** Review licenses of all dependencies

## Recommendations

1. **Dependency Management:** Use lock files to ensure consistent builds
2. **Minimal Dependencies:** Remove unused dependencies to reduce attack surface
3. **Version Pinning:** Consider pinning critical dependencies to specific versions

---

Generated on 2025-09-02 23:23:18

---

# Architecture Documentation

## Overview

This document provides an architectural analysis of dotitude-new-front.

## Project Structure

The project follows a structured approach with the following main directories:

- **.firebase/**: 1 files
- **e2e/**: 4 files
- **src/**: 35 files

## Language Distribution

- **TypeScript**: 22 files, 843 lines
- **HTML**: 6 files, 695 lines
- **JSON**: 5 files, 251 lines
- **JavaScript**: 2 files, 65 lines
- **Unknown**: 15 files, 1 lines

## Architectural Patterns

Based on the codebase analysis, the following patterns are observed:

1. **Modular Structure**: Code is organized into logical modules 2. **Separation of Concerns**: Different functionalities are separated 3. **Scalable Design**: Structure supports future growth

## Key Components

The main components of the system include:

- **.editorconfig**: Unknown (0 lines)
- **app.e2e-spec.ts**: TypeScript (24 lines)
- **app.po.ts**: TypeScript (12 lines)
- **tsconfig.json**: JSON (14 lines)
- **proxy.config.json**: JSON (10 lines)

- **cvmgt-config.ts**: TypeScript (16 lines)
- **app-routing.module.ts**: TypeScript (22 lines)
- **app.component.html**: HTML (8 lines)
- **app.component.scss**: Unknown (0 lines)
- **app.component.spec.ts**: TypeScript (36 lines)

## Data Flow

1. **Input Processing**: Data enters through defined interfaces 2. **Business Logic**: Core functionality processes the data 3. **Output Generation**: Results are formatted and returned

## Scalability Considerations

1. **Modular Design**: Components can be scaled independently 2. **Technology Stack**: Selected technologies support horizontal scaling 3. **Performance**: Code structure allows for optimization

---

*Generated on 2025-09-02 23:23:18*

---

# API Documentation

## Overview

This document provides API documentation for dotitude-new-front.

## API Analysis

No clear API endpoints were detected in the codebase. This could mean:

1. The project doesn't expose APIs 2. APIs are dynamically generated 3. API definitions are in configuration files

## OpenAPI Specification



```
openapi: 3.0.0 info: title: dotitude-new-front API version: 1.0.0
description: API for dotitude-new-front servers: - url:
http://localhost:3000 description: Development server paths: # API
paths would be defined here based on detected endpoints
```

## Usage Examples

```
# Example API calls would be provided here curl -X GET
"http://localhost:3000/api/endpoint"
```

---

*Generated on 2025-09-02 23:23:24*

---