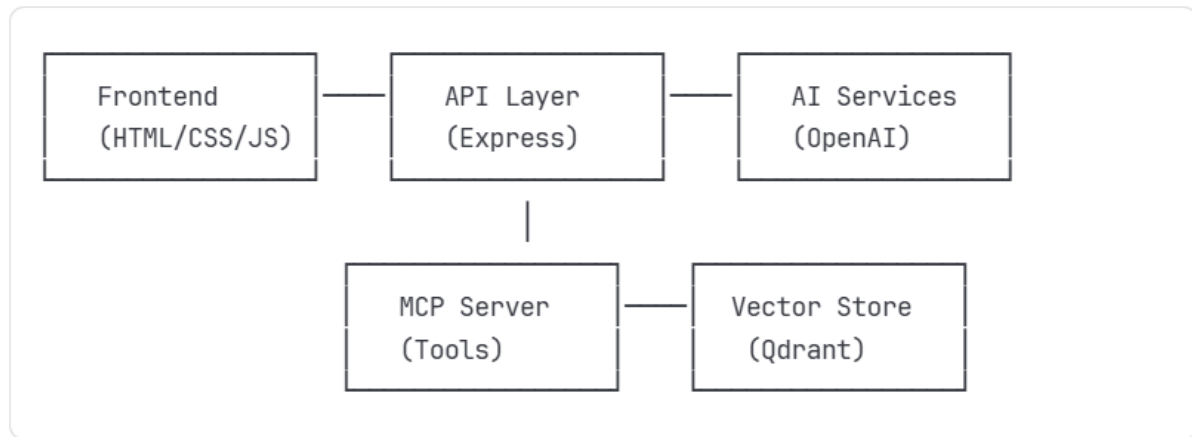


## # 🤖 Document Q&A System

system architecture



> AI-powered document question answering system with vector search, semantic understanding, and MCP integration.

[![MIT License](https://img.shields.io/badge/License-MIT-green.svg)](https://choosealicense.com/licenses/mit/)

[![Node.js Version](https://img.shields.io/badge/node-%3E%3D18.0.0-brightgreen)](https://nodejs.org/)

[![TypeScript](https://img.shields.io/badge/TypeScript-007ACC?logo=typescript&logoColor=white)](https://www.typescriptlang.org/)

[![OpenAI](https://img.shields.io/badge/OpenAI-412991?logo=openai&logoColor=white)](https://openai.com/)

## ## ✨ Features

- 📄 **Document Upload & Management** - Support for text documents with automatic indexing
- 🔍 **Semantic Search** - Vector-based similarity search using OpenAI embeddings
- 🤖 **AI-Powered Q&A** - Get intelligent answers with source citations
- ⚙️ **MCP Integration** - Model Context Protocol server for tool integration
- 🌐 **Modern Web Interface** - Clean, responsive UI with real-time interactions

- 🚀 **\*\*Production Ready\*\*** - Docker support, TypeScript, and comprehensive error handling
- 🗃️ **\*\*Vector Database\*\*** - Qdrant for efficient similarity search
- 🔒 **\*\*Secure\*\*** - Input validation, rate limiting, and security headers

## ## 🚀 Quick Start

### ### Prerequisites

- Node.js  $\geq 18.0.0$
- Docker Desktop
- OpenAI API key
- Git

### **install dependencies**

npm install

### **set up environment**

# Copy the example environment file

cp .env.example .env

# Edit .env and add your OpenAI API key

# OPENAI\_API\_KEY=sk-your-openai-api-key-here

### **start service**

# Start Qdrant database

npm run docker:up

### **# Build the project**

npm run build

# Start development server

npm run dev

## Access the Application

Open your browser and go to: <http://localhost:3000>

## Usage

Adding Documents

## Open the web interface

Enter a document name and content in the "Add Document" section

Click "Add Document" to upload and index it

## Asking Questions

Type your question in the chat interface

Get AI-generated answers with source citations

View confidence scores and relevant document excerpts

## Core Components

- **API Layer:** Express.js REST API with TypeScript
- **AI Services:** OpenAI integration for embeddings and chat completion
- **Vector Store:** Qdrant for semantic search and similarity matching
- **MCP Server:** Model Context Protocol for tool integration
- **Frontend:** Vanilla JavaScript with modern CSS

## Available Scripts

bash

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

```
npm run build      # Build TypeScript to JavaScript
npm run start      # Start production server
npm run test       # Run unit tests
npm run test:watch # Run tests in watch mode
npm run lint       # Run ESLint
npm run docker:up  # Start Docker services
npm run docker:down # Stop Docker services
npm run mcp        # Start MCP server
```

## **Deployment**

### **Local Production**

```
bash
```

```
# Build the project
```

```
npm run build
```

```
# Set environment to production
```

```
export NODE_ENV=production
```

```
# Start the server
```

```
npm start
```

### **Docker Deployment**

```
dockerfile
```

```
FROM node:18-alpine
```

```
WORKDIR /app
```

```
COPY package*.json ./
```

```
RUN npm ci --only=production
```

```
COPY dist ./dist
```

```
COPY public ./public
```

EXPOSE 3000

CMD ["npm", "start"]

## Testing

bash

*# Run all tests*

npm test

*# Run tests with coverage*

npm run test:coverage

*# Run tests in watch mode*

npm run test:watch

## Acknowledgments

- [OpenAI](#) for providing the AI models
- [Qdrant](#) for the vector database
- [Vercel AI SDK](#) for the AI integration
- [Model Context Protocol](#) for the MCP specification