

English version

Table of Contents

- [Main features](#)
 - [Architecture](#)
 - [Technical requirements](#)
 - [Installation & Getting Started](#)
 - [Using AI Search](#)
 - [Environment & config](#)
 - [Security](#)
 - [Testing](#)
 - [Documentation](#)
 - [Credits](#)
 - [Versión en castellano](#)
-

AIWeek Search Engine

Fullstack platform for advanced contact management and search, featuring AI-powered natural language queries. Includes a .NET Core backend and a React frontend, JWT authentication, roles, advanced filters, CSV export, and a modern robust UX.

Official repository: <https://github.com/TINAlbert/AIWeek-SearchEngine>

Main features

- Simple and advanced contact search (name, email, company, profiles, etc.)
 - **AI Search:** natural language queries, SQL generation and execution via local LLM (Ollama)
 - Export contacts to CSV
 - User, role, and avatar management
 - Modern, responsive, accessible UI
 - JWT security and refresh token
 - Reusable advanced filter history
-

Architecture

- [/backend/](#) — .NET Core 7+ REST API, Entity Framework Core, Identity, JWT
- [/frontend/](#) — React 19+, Tailwind CSS, Axios, Context API

See detailed docs:

- [Backend README](#)
 - [Frontend README](#)
-

Technical requirements

- **Backend:** .NET Core 7+, Entity Framework Core, Microsoft Identity, JWT, AutoMapper, Scalar (OpenAPI), xUnit
 - **Frontend:** React 19+, Vite, Tailwind CSS, Axios, React Context, React Router DOM, React Hook Form, Yup
 - **AI:** Ollama ([llama3](#) or [sqlcoder](#) model)
-

Installation & Getting Started

1. Backend

```
cd backend/SearchServiceEngine
dotnet restore
dotnet ef database update
dotnet run
```

Default API: <http://localhost:5252/api>

2. Frontend

```
cd frontend
npm install
# Set API URL in .env
# VITE_API_BASE_URL=http://localhost:5252/api
npm run dev
```

Default app: <http://localhost:5173>

3. Ollama (AI) setup

1. Install Ollama: <https://ollama.com/download>
2. Download model:

```
ollama pull llama3
# or
ollama pull sqlcoder
```

3. Start Ollama:

```
ollama serve
ollama run llama3
```

Backend expects Ollama at <http://localhost:11434>.

4. Check connectivity with </api/ai/ping>.

Using AI Search

- Access from sidebar: **AI Search** ("Sparkles" icon)
 - Describe your query in natural language (e.g., "Active contacts in Madrid")
 - The AI generates and executes SQL, showing results and the generated query
 - Only authenticated users can access
 - Backend only executes SELECT SQL generated by AI
 - Extended timeout for AI requests (up to 5 minutes)
-

Environment & config

- **Frontend:** `.env`:

```
VITE_API_BASE_URL=http://localhost:5252/api
```

- **Backend:** `appsettings.json`:

```
{
  "Jwt": {
    "Key": "SuperSecretKey12345678901234567890123456789012",
    "Issuer": "AIWeekIssuer",
    "Audience": "AIWeekAudience"
  },
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=aiweek.db"
  },
  "AvatarsPath": "wwwroot/avatars",
  "SeedInitialData": true
}
```

Security

- JWT authentication (Bearer Token)
 - Secure, revocable refresh token
 - Roles and claims in backend and frontend
 - Private route protection and role validation in UI
 - Only authenticated users can access AI endpoints
-

Testing

- Backend: unit tests with xUnit and Moq

- Frontend: ready for React Testing Library
-

Documentation

- [Backend README](#)
 - [Frontend README](#)
-

Credits

Developed by Albert G.M. ([GitHub: TINAlbert](#)). Based on best practices in architecture, security, and UX for modern web applications. Repository: <https://github.com/TINAlbert/AIWeek-SearchEngine>

Versión en castellano

Índice

1. [English version](#)
 1. [Table of Contents](#)
 2. [AIWeek Search Engine](#)
 1. [Main features](#)
 2. [Architecture](#)
 3. [Technical requirements](#)
 4. [Installation & Getting Started](#)
 1. [1. Backend](#)
 2. [2. Frontend](#)
 3. [3. Ollama \(AI\) setup](#)
 5. [Using AI Search](#)
 6. [Environment & config](#)
 7. [Security](#)
 8. [Testing](#)
 9. [Documentation](#)
 10. [Credits](#)
2. [Versión en castellano](#)
 1. [Índice](#)
 2. [AIWeek Search Engine](#)
 3. [Características principales](#)
 4. [Arquitectura y estructura](#)
 5. [Requisitos técnicos](#)
 6. [Instalación y puesta en marcha](#)
 1. [1. Backend](#)
 2. [2. Frontend](#)
 3. [3. Configuración y uso de Ollama \(IA\)](#)
 7. [Uso de la Búsqueda IA](#)
 8. [Variables de entorno y configuración](#)

- 9. Seguridad
 - 10. Testing
 - 11. Documentación específica
 - 12. Créditos y agradecimientos
-

AIWeek Search Engine

Plataforma fullstack para la gestión y búsqueda avanzada de contactos personales, con integración de IA para consultas en lenguaje natural. Incluye backend en .NET Core y frontend en React, autenticación JWT, roles, filtros avanzados, exportación CSV y una experiencia moderna y robusta.

Repositorio oficial: <https://github.com/TINAlbert/AIWeek-SearchEngine>

Características principales

- Búsqueda simple y avanzada de contactos (nombre, email, empresa, perfiles, etc.)
 - **Búsqueda IA:** consulta en lenguaje natural, generación y ejecución de SQL vía LLM local (Ollama)
 - Exportación de contactos a CSV
 - Gestión de usuarios, roles y avatares
 - UI moderna, responsiva y accesible
 - Seguridad JWT y refresh token
 - Historial reutilizable de filtros avanzados
-

Arquitectura y estructura

- `/backend/` — API REST en .NET Core 7+, Entity Framework Core, Identity, JWT
- `/frontend/` — React 19+, Tailwind CSS, Axios, Context API

Ver documentación detallada en:

- [Documentación Backend \(SearchServiceEngine\)](#)
 - [Documentación Frontend](#)
-

Requisitos técnicos

- **Backend:** .NET Core 7+, Entity Framework Core, Microsoft Identity, JWT, AutoMapper, Scalar (OpenAPI), xUnit
 - **Frontend:** React 19+, Vite, Tailwind CSS, Axios, React Context, React Router DOM, React Hook Form, Yup
 - **IA:** Ollama (modelo `llama3` o `sqlcoder`)
-

Instalación y puesta en marcha

1. Backend

```
cd backend/SearchServiceEngine
# Restaurar paquetes y aplicar migraciones
dotnet restore
dotnet ef database update
# Ejecutar backend
dotnet run
```

Por defecto, la API estará en <http://localhost:5252/api>.

2. Frontend

```
cd frontend
npm install
# Configura la URL de la API en .env
# VITE_API_BASE_URL=http://localhost:5252/api
npm run dev
```

La app estará en <http://localhost:5173>.

3. Configuración y uso de Ollama (IA)

1. Instala Ollama: <https://ollama.com/download>
2. Descarga el modelo LLM:

```
ollama pull llama3
# o
ollama pull sqlcoder
```

3. Inicia el servicio:

```
ollama serve
ollama run llama3
```

El backend espera Ollama en <http://localhost:11434>.

4. Comprueba la conectividad con </api/ai/ping>.

Uso de la Búsqueda IA

- Accede desde el menú lateral: **Búsqueda IA** (icono "Sparkles")
- Describe la consulta en lenguaje natural (ej: "Contactos activos de Madrid")
- La IA genera y ejecuta la SQL, mostrando resultados y la consulta generada
- Solo usuarios autenticados pueden acceder
- El backend solo ejecuta SQL generada que sea un SELECT

- Timeout extendido para peticiones de IA (hasta 5 minutos)

Variables de entorno y configuración

- **Frontend:** `.env` con:

```
VITE_API_BASE_URL=http://localhost:5252/api
```

- **Backend:** `appsettings.json` con:

```
{
  "Jwt": {
    "Key": "SuperSecretKey12345678901234567890123456789012",
    "Issuer": "AIWeekIssuer",
    "Audience": "AIWeekAudience"
  },
  "ConnectionStrings": {
    "DefaultConnection": "Data Source=aiweek.db"
  },
  "AvatarsPath": "wwwroot/avatars",
  "SeedInitialData": true
}
```

Seguridad

- Autenticación JWT (Bearer Token)
- Refresh token seguro y revocable
- Roles y claims en backend y frontend
- Protección de rutas privadas y validación de roles en la UI
- Solo usuarios autenticados pueden acceder a la IA

Testing

- Backend: pruebas unitarias con xUnit y Moq
- Frontend: preparado para React Testing Library

Documentación específica

- [Documentación Backend \(SearchServiceEngine\)](#)
- [Documentación Frontend](#)

Créditos y agradecimientos

Desarrollado por el programador Albert G.M. ([GitHub: TINAlbert](#)). Basado en mejores prácticas de arquitectura, seguridad y experiencia de usuario para aplicaciones web modernas. Repositorio: <https://github.com/TINAlbert/AIWeek-SearchEngine>