🌐Scientific Twins

An elegant and powerful service for analyzing scientific texts, offering two intelligent modes:

🔬**Scientific Plagiarism Detection** — identifies conceptual overlap with published academic papers. 🧪**Scientific Doppelganger Search** — finds interdisciplinary works with similar logical structure.

The project consists of: - **Backend**: FastAPI + OpenAI + Crossref - **Frontend**: Vite + React + TypeScript + TailwindCSS

🚯Features

⚠️*Plagiarism* Mode

- Summarizes the text using an LLM
- Searches scientific papers via Crossref using dynamically generated queries
- Cleans abstracts (HTML → plain text)
- Dual similarity check: LLM + TF-IDF
- Final probability evaluation of potential plagiarism

™️*Doppelganger* Mode

- Generates interdisciplinary search queries
- Compares texts by logical and structural similarity
- Classifies scientific domains
- Produces a top-3 list of conceptual analogs

🔯Technologies

🕉️Backend

- Python 3.13
- FastAPI
- OpenAI SDK
- Crossref API
- PyMuPDF
- BeautifulSoup4
- Scikit-learn

🛅Frontend

- Vite
- React + TypeScript
- TailwindCSS
- ESLint
- PostCSS

Run:

```
npm run dev
```

♒Backend Setup

1. Clone the repository

```
git clone <url>
cd scientific-analyzer
```

2. Create a virtual environment

```
python -m venv venv
source venv/bin/activate      # macOS/Linux
venv\Scripts\activate        # Windows
```

3. Install dependencies

```
pip install -r requirements.txt
```

4. Environment variables

Create a `.env` file:

```
OPENAI_API_KEY=your_key
```

5. Start the API

```
uvicorn main:app --host 0.0.0.0 --port 8000
```

▶️Frontend

Project directory:

```
front/science-twins-ui
```

Install and run:

```
cd front/science-twins-ui
npm install
npm run dev
```

Typically available at: `http://localhost:5173`

✔️Frontend Recovery

1️⃣Remove dependencies

```
Remove-Item -Recurse -Force node_modules
Remove-Item package-lock.json
```

2️⃣Clear npm cache

```
npm cache clean --force
```

3️⃣Reinstall

```
npm install
```

4️⃣Run

```
npm run dev
```
