

# PORTOFOLIU TEHNIC

Procedura nr. 21663739 · Front End & Python Developer (junior)

Proiectele de mai jos sunt demonstrații personale/open-source, cu date sintetice. Nu reprezintă produse MAIA și nu conțin date cadastrale, personale sau integrări guvernamentale reale.

## Matricea competențelor tehnice P1.1

| Criteriu                     | Proiect                    | Dovadă directă                                                                |
|------------------------------|----------------------------|-------------------------------------------------------------------------------|
| React.js / Next.js           | Farm Registry Web          | React în src/App.tsx și src/main.tsx; README și cod public.                   |
| JavaScript ES6+ / TypeScript | Web + Mobile               | Fișiere .tsx, tsconfig și typecheck publice.                                  |
| React Native                 | Farm Registry Mobile       | App.tsx, Expo, TypeScript; instrucțiuni în README.                            |
| Python                       | Farm Registry Python Tools | app/main.py: FastAPI, Pydantic și Shapely.                                    |
| REST / OpenAPI               | Web + Python Tools         | GET /parcels, POST /validate/parcel, client REST și /docs.                    |
| State management             | Farm Registry Web          | React Query/useQuery și stare UI locală.                                      |
| Testare automată             | Web + Python Tools         | Test React și tests/test_main.py; CI public verificabil pentru proiectul web. |
| Git / colaborare             | Toate repo-urile           | Istoric Git public, README, CI și structură pregătită pentru review.          |

## 1. Farm Registry Web

Link: <https://github.com/luciandanileico94-dev/farm-registry-web>

Rol: dezvoltator al demonstrației personale/open-source.

Stack: React, TypeScript, Vite, React Query, Leaflet, GeoJSON, REST, testare React, GitHub Actions.

- Interfață responsive în limba română pentru căutarea și vizualizarea parcelor sintetice.
- Hartă interactivă Leaflet cu poligoane GeoJSON, selecție și afișarea stării de validare.
- Consum al endpoint-ului GET /parcels, cu fallback explicit la fixtures sintetice.
- Test automat și instrucțiuni reproductibile pentru build și rulare.

## 2. Farm Registry Python Tools

Link: <https://github.com/luciandanileico94-dev/farm-registry-python-tools>

Rol: dezvoltator al demonstrației personale/open-source.

Stack: Python, FastAPI, Pydantic, Shapely, GeoJSON, REST/OpenAPI, pytest.

- Endpoint-uri GET /health, GET /parcels și POST /validate/parcel.
- Validarea geometriilor GeoJSON, calcul de suprafață și raportarea problemelor topologice.
- Modele tipizate și documentație API generată prin OpenAPI.
- Teste automate pentru logica principală.

**Demonstrație front-end + Python:** Farm Registry Web și Farm Registry Python Tools formează împreună o demonstrație funcțională: clientul React consumă endpoint-ul GET /parcels expus de serviciul FastAPI; codul este separat în două repository-uri pentru claritate și rulare independentă.

## 3. Farm Registry Mobile

Link: <https://github.com/luciandanileico94-dev/farm-registry-mobile>

Rol: dezvoltator al demonstrației personale/open-source.

Stack: React Native, Expo, TypeScript, state local offline-friendly, GitHub Actions.

- Interfață mobile-first pentru operatori de teren, cu căutare după parcelă și starea sincronizării.
- Scenariu UI orientat spre utilizare offline și contract tipizat pentru integrarea ulterioară cu API.
- Instrucțiuni de rulare prin Expo Go sau simulator și verificare TypeScript.

## Componentă GIS / date spațiale

Componenta GIS este demonstrată prin Leaflet și GeoJSON în proiectul web și prin validarea geometriilor cu Shapely în proiectul Python. Datele sunt sintetice; nu se revendică experiență într-un proiect GIS comercial sau guvernamental livrat.

## AI engineering — competență complementară

Utilizare practică a Claude, Codex și Gemini; proiectarea și operarea de agent harnesses; configurarea de skills și plugins; prompt/context engineering; analizarea volumelor mari de informație; bucle iterative de implementare, review, testare și corectare. Rezultatele generate sunt validate manual și prin verificări executabile. AI este utilizat ca instrument de accelerare, nu ca substitut pentru responsabilitatea tehnică.

## Limitări declarate

Nu se revendică proiecte guvernamentale anterioare, contracte MAIA, integrări MPass/MConnect/MSign, beneficiari publici sau recomandări inexistente. Soluțiile comerciale interne din experiența e-commerce nu sunt publicate din motive de confidențialitate; punctajul tehnic solicitat este susținut prin codul public indicat mai sus.

Index public suplimentar: <https://github.com/luciandanileico94-dev/tender-portfolio>

Data: 12.08.2026

Ofertant: Lucian Danileico (persoană fizică)

Semnătură electronică: \_\_\_\_\_