

## Full-Stack Technical Challenge — Forest BD Viewer

### Goal

Your mission is to build a full-stack geospatial application that allows authenticated users to visualize and interact with French forest data. The application should expose layers from the **BD Forêt®** database and integrate basic spatial analysis.

---

### Core Features

You are free to choose your preferred stack (frontend/backend) and architecture for this exercise. For context, here's what we use at Symbiose:

Frontend : Typescript, Next.js,

Backend : Nest.js, TypeORM

Architecture : GraphQL, microservices architecture, Turborepo, gRPC, Kafka

DB : Postgresql & PostGis, Redis

#### 1. Authentication

- Implement user registration, login, and logout, and the requirements you think necessary.

#### 2. Interactive Mapping Interface

- Map display using **Mapbox GL JS**.
- Show forest layers from **BD Forêt**.
- Let users click (on the map) for the **region → department -> commune -> lieuudit** to display BD Forêt data only relevant zone (avoid full load). (Interactive map please)
- Visualize a layer of BD forêt data (=tree species mapping) and a layer of parcelle cadastre (if only when sufficiently zoomed in).
- Add relevant filters, buttons, or controls to optimize user experience.

#### 3. Backend

- Set up APIs to manage geospatial data and communicate with the frontend.
- Use TypeORM and GraphQL (or another approach if justified) to implement CRUD operations for the users table.
- For each user, store the last map view state (position, zoom, filters). When the user logs in again, the map should return to this saved state.

## 4. Deployment (optional but appreciated)

If you can, please deploy. If deployment is not feasible in time, a working local setup via Docker is also fine.

Please include deployment URLs and credentials (if needed) in your README.

---

### Required Datasets

You will work with official open French geospatial data:

#### ◆ BD Forêt® V2 – Forest Inventory (IGN)

-  <https://geoservices.ign.fr/bdforet#telechargementv2>
- You can see a concrete example here : <https://inventaire-forestier.ign.fr/cartoifn/carto/afficherCarto/V2/#zoom=6&lat=6256527.45826&lon=141332.38423&layers=B0FFFFFF>

#### ◆ French Cadastre – Parcel Data (Optional)

-  <https://cadastre.data.gouv.fr/data/etalab-cadastre/latest/>
- Here is an example : <https://cadastre.data.gouv.fr/map?style=ortho>
- Useful for overlaying land parcels or ownership boundaries

You're free to use a **subset** of data for prototyping (e.g., 1 or 2 departments), as long as your backend could scale.

---

### Bonus Features (Optional)

#### Bonus A — Polygon Drawing Tool

- Let users draw a polygon on the map.
- On submit, send it to the backend and return stats:
  - Size of hectares of the polygon
  - Parcel Name in the polygons
  - Types of tree species in the polygon and their respective surface areas (in hectares)

#### Bonus B — LiDAR Analysis (Advanced)

- From a drawn polygon, compute the Canopy Height Model derived from LiDAR HD data points relevant to this polygon.

- Canopy Height Model = MNS (Modèle Numérique de Surface) - MNT (Modèle Numérique de Terrain)
- You can retrieve the MNS and MNT data from:
  - <https://diffusion-lidarhd.ign.fr/> and select look at “Produits Dérivés” for MNS

Nuages de points      Produits dérivés

Sélectionnez les dalles qui vous intéressent puis téléchargez la liste des liens pour les récupérer.

Choix du mode de sélection

Clic  Polygone  Rectangle

▼ Liste des MNS

LHD\_FXX\_0599\_6329\_MNS\_O\_0M50\_LAMB93...

> Liste des MNH

▼ Liste des MNT

LHD\_FXX\_0599\_6329\_MNT\_O\_0M50\_LAMB93...

and MNT :

- OR <https://forge.inrae.fr/lidar/lidarHD> ( you can access the tutorial [here](#) for MNS & MNS)
  - Generate a **Canopy Height Model** = MNS - MNT
  - Return:
    - Canopy Height Model overlay the Map Displayed + Basic stats (min/max/mean height)
- 

## 📦 Deliverables

Please share the following:

1. **Git repository** with:
  - Source code (frontend + backend)
  - Docker + Docker Compose config
  - SQL files or scripts to import and index the datasets
  - Bonus: scripts for LiDAR processing (if Bonus B implemented)

2. **README** including:

- Setup instructions to run the app locally
- Description of APIs / GraphQL schema
- Any assumptions or simplifications made

3. Optionally: a short screen recording (max 5 min) or a few screenshots demonstrating:

- Auth + map view Authentication flow
  - Map view with region/department filtering
  - Polygon query results
  - Bonus features if implemented
- 

 **Time Estimate**

- You should plan for 2 to 3 days of focused work for the core features.
- Bonus features (polygon queries, LiDAR) are appreciated, but optional.

If you're short on time, feel free to focus on clean code and architecture over completeness. Please count the number of hours you have spent on it, so you can put also in the README.

---

 **Questions?**

If anything is unclear, feel free to reach out. We're looking forward to seeing how you approach the challenge!