

Frontend

Recruitment Exercise

RECIPIENT:

Hernâni Silva

Index

1. Introduction	2
2. Exercise	3
2.1. Functional Requirements	3
2.1.1. Authentication Flow	3
2.1.2. Dashboard	4
2.1.3. Logout Functionality	4
2.2. Deliverable	5
2.2.1. Time Expectation	5
2.2.2. Code delivery	5

1. Introduction

This exercise is designed to help us evaluate your frontend development skills, problem-solving ability, code quality, and attention to detail. It reflects the type of challenges you might encounter while working with us. The task is intentionally open-ended to give you the flexibility to make architectural and design choices. We're looking for clean, maintainable code. Please treat this as a real-world task—something you'd be comfortable sharing with your peers or deploying to production.

This exercise dives into one of the core responsibilities of a Ubiwhere developer main challenge - displaying a data set on a map and giving the users appropriate context about what is being displayed. We also care about user privacy, so a good architecture for private and public routes is one of our most valued features.

The next chapter lists a detailed description of this challenge objective.

2. Exercise

Build a simple frontend application that displays recent earthquake events retrieved from our api on an interactive map. Users must log in to access the main interface, which includes a list of earthquake events plotted on a map. Clicking on any event should display its details.

2.1. Functional Requirements

Focus on clarity, functionality, and clean code. Please use some of the following technologies and patterns in your implementation:

- **React** with **TypeScript**
- A CSS framework of your choice (preferably **CSS Modules** or **Tailwind CSS**)
- Use either **OpenLayers** or **Leaflet** opensource libraries for map display or generate an API Key for **MapBox**
- Implement a routing structure that supports both **public and private routes**
- Project setup using a modern bundler (e.g., **Vite**, or similar)
- A state management solution that doesn't include the observer pattern (ex: mobX)

API Documentation URL - <https://recruitment.ubiwhere.com/api/docs>

2.1.1. Authentication Flow

LOGIN CREDENTIALS : (user: hcsilvamatos5@gmail.com / password: Ubiwhere123)

After a successful login, the API will return a token. This token must be included in all subsequent API requests using the Authorization header in the following format:

```
Authorization: Bearer <token>
```

- Create a Login page - the only public route of this exercise
- After logging in, users should be redirected to the Dashboard. The authentication state should be persisted across page refreshes, so users remain logged in until they manually log out or the token expires without refreshing.

2.1.2. Dashboard

- Display an interactive map using Leaflet or OpenLayers
- Fetch recent earthquake data from the provided API, using authenticated requests. The data should be paginated. No need to have more than 5 pages.
- Plot each earthquake as a clickable marker on the map, with relevant tooltip associated
- When a user clicks a marker, show a details panel with a specific earthquake information that is available on the API via endpoint `/earthquakes/{id}`

2.1.3. Logout Functionality

- Add a button to log out and return to the login screen
- Ensure all private routes are inaccessible after logout

2.2. Deliverable

Feel free to share your solution in a way that you're comfortable with. We're simply looking to understand your approach, how you structure your code, and how you think through a problem.

2.2.1. Time Expectation

This exercise is designed to be completed in about **4 days** with a clear functional and production ready solution. It doesn't need to be pixel-perfect.

2.2.2. Code delivery

The minimal solution should include:

- A link to a repository (e.g., GitHub, GitLab)
- It must include a short README.MD with:
 - Setup instructions (how to install and run the project) and other specifications that you might consider important, like codebase structure or dependencies used

When you're ready, please email us with the deliverables listed above. We'll review your submission and get back to you as soon as possible. Feel free to get creative—if you'd like to take the project a step further or add something beyond the core requirements (like extra features, tests, or UI polish), **we truly value that kind of initiative and personal touch.**

Good luck - and if you have any questions, don't hesitate to reach out. We're happy to help!

ubiwhere

www.ubiwhere.com

hr@ubiwhere.com