

## Instructions

Read carefully all the requirements and create a product that fits the expectations. **You have to create a public GitHub repository** to upload the exercise and **share the link** in the answer field.

## Requirements

Create a Pokemon app with the following features:

- Login screen:

The creds consist of a username/password form. You should validate the user creds locally. (admin as username and admin as password, anything different should be considered as incorrect creds). Shows all the validation that you think makes sense.

- The user should remain logged against a storage instance that fits your preference (Local db, Local storage, cookies), so if the user tries to log in already logged, it should be redirected to the main page, and if it is not logged and it tries to go to the main page it should be redirected to the login page.

- Main page:

- The home screen will have a search bar with a list of Pokemon. You'll have to use this API for it: <https://pokeapi.co/>. The API is paginated so you should create a solution for it.
- Each Pokemon should be shown with its photo and name.

- Detail view:

- If the user clicks on a Pokemon from the list, a modal should be shown with detailed information about the Pokemon (Abilities, moves, and forms).

## Notes:

- Feel free to use any library for local state management and UI.
- You can define any UI/UX definition you need to show the information. This is an example: <https://www.behance.net/gallery/146710797/Pokedex-Ui-Study> but feel free to do it in a way that fits better your preferences
- We will evaluate the architecture you think is better for the app. Please take into account that the app could have more features in the future.
- Be prepared to discuss your solution.

## Presentation and Code Review

Once you have completed the exercise, you will be required to present your project to the technical interview panel. During the presentation, you should explain your user story, design choices, the technical architecture, and demonstrate the functionality of the application. This will be done over Google meet and you will screen share either your GitHub repository or IDE.

After the presentation, the interview panel will conduct a code review of your project. You will be asked to explain your coding decisions and answer any questions related to the code. The interview panel will evaluate your project based on the following criteria:

- Clean Architecture: Your architecture should adhere to Clean Architecture principles, including separation of concerns and independence of components.
- Test-Driven Development: Your project should follow TDD methodologies and include unit tests.
- Code quality: Your code should be well-organized, readable, and adhere to best practices.
- Functionality: Your application should perform as expected in the requirements without errors or bugs. Optional but desired: no warnings in the browser console.
- Presentation: Your presentation should be clear, concise, and demonstrate a good understanding of the project.

## Scoring

You will be scored based on your adherence to Clean Architecture principles, TDD methodologies, code quality, user story, and code review remarks. Each criterion will be weighted equally. The maximum score for the exercise is 100 points.

We hope this exercise challenges your technical abilities and helps us assess your suitability for the intermediate full-stack software developer position. Good luck!