

# Report of the technical project

ARIZZI Leny

## Table of content

I. Finally, what do you think of this algorithm? Does it have drawbacks or edge cases? Is it relevant to implement in everyday projects?.....	3
II. You SHOULD explain us every tool (IDE, AI, rubber duck, actual duck, etc...) used in the creation of this project, if you think it's relevant.....	3
III. You SHOULD explain us everything you've learned during the creation of this project (new terms, architecture, concept, etc...)....	4
IV. Anything you say can and will be used to understand your way of working and thinking.....	4

## I. Finally, what do you think of this algorithm? Does it have drawbacks or edge cases? Is it relevant to implement in everyday projects?

Quicksort is an excellent general-purpose sorting algorithm. It is very fast in practice, elegant (because of recursion) and relatively simple to implement. Moreover, quicksort is efficient in memory if we use the in-place version (my 2nd version of the quicksort).

I think quicksort has some drawbacks. For example if we always use the smallest or largest element as pivot or when the array is already sorted we do not benefit from the speed of the algorithm. We can solve that by choosing the average element of the array as a pivot.

Actually, I think that quicksort is relevant to implement when you have special or specific cases, but all languages offer their own sorting method by default.

## II. You *SHOULD* explain us every tool (IDE, AI, rubber duck, *actual* duck, etc...) used in the creation of this project, if you think it's relevant.

For this project (Currency converter), I used Visual Studio Code because I'm familiar with Visual Studio Code. I use it a lot for my personal projects or university projects, therefore I know many of its functions.

Moreover, for part 1 which is about the algorithm, I made these functions in Java. For this, I used Eclipse IDE. Because with Java, I use Eclipse for its modularity. I'm very familiar with its features and advantages, which some other IDEs don't offer. With this IDE, I can also easily install SonarQube, which allows me to measure the code quality of my projects.

Afterwards, I used an API called Exchangerate-api (<https://www.exchangerate-api.com>) to get real-time exchange rate data against a currency. I used this API because it has a high quota and a returned data format that is what I wanted. Unfortunately, the exchange rate isn't entirely accurate due to the monthly quota (even though it's high), so I had to add a TTL to the fetched data to avoid constant API queries. However, I could obviously get real-time data by removing this TTL.

Finally, I used AI to get some information. For example, my API queries provide the exchange rates for currencies, not their symbols. The API calls return the exchange rates for over 160 currencies, but they don't share their symbols. To save myself a lot of internet research, I asked the AI for their symbol. Thereafter, I used AI to give me information I couldn't find on the internet about React, such as a file/folder structure for a project and methods I didn't understand like specific parameters only present in React (React.ChangeEvent<HTMLSelectElement>) and how to use them.

### III. You **SHOULD** explain us everything you've learned during the creation of this project (new terms, architecture, concept, etc...)

First, this project allowed me to learn some React syntax. I was already familiar with React and had completed personal projects with React (storyTelling website, Portofolio), but this project strengthened my skills in the framework. It taught me about docJS, for example, and that some of the syntax I was using in React wasn't actually the correct way to do things.

I learned how to properly structure a project and manage the different components together. Furthermore, I also learned what i18n is and how to use it on a website, and I learned about the need for a loader file and how to integrate it with React. This discovery will be a necessary addition to my future projects (I really like i18n now).

Then, I also saw how to properly call an API and integrate it into my website. Finally, I was able to learn the usefulness of a . env (for me before that had no interest)

### IV. Anything you say can and will be used to understand your way of working and thinking

I approach tasks methodically and logically, breaking problems down into smaller steps. I like to plan before acting, verify my assumptions carefully (for example I made mock data and use them before to make real API queries), and make sure my work is structured and maintainable. I value clarity (made comments and docJS), efficiency, and I adapt my approach based on feedback or changing requirements (I asked my friends, my family to know if my website is intuitive and their advices) .

I really liked doing this project because it taught me a lot of things and that's exactly what I'm looking for. I really like React and I like doing projects, so for me it was a really good moment.