

# FRONTEND DEVELOPER VUE.JS

## @ FootballTeam

Mateusz Staniuk

# Repository

[https://github.com/mstaniuk/FT\\_front\\_end](https://github.com/mstaniuk/FT_front_end)

I've used the Github repository to host my work.

I've used the master branch for the finished product, development for large working chunks and feature branches to specify the field of current work.

README.md contains all info needed to get the project up and running.

# Technology

Since time to finish this project was specified to 8 hours I've decided to base my work on Vue CLI as a project template, I did change project structure for my needs though.

I've used Webpack and Babel for code transpilation, Axios as http client, the fuzzysearch package for simple search matching, Vue router for SPA.

I've created custom SVG handling for SVG icons using html-loader and SvgIcon vue component.

Dev env is working on a webpack dev server provided by vue cli.

# Working functionalities

## List page

The list page contains toolbar as a way of navigation and content filtering, fetched user list and pagination.

## User Page

The user page has two functionalities - user edit and creating a new user.

## RWD

Created pages look good on all tested resolutions.

# Not working functionalities

Unfortunately, since time was restricted to only 8 hours some functionalities are missing.

## No error handling

It's UX and architectural error to leave the user without notice when something went wrong and assume that async calls will always go good.

## No tests

It's a very big flaw to leave software without any way of the automatic test.

## No success prompts

same as with error handling user has no idea if anything did go well.

## Things that could be done better

### Loading indicators

For user, it's very unintuitive when things are happening behind the scenes and there are no indicators of that. There are a lot of unused fields in the store that could have been used to track what is happening.

### Structure of User Page

A lot of things are going on in one component - User. It could be broken down into multiple reusable components.

### Async image loading

Could be implemented using intersection observer as a separate component that loads an image only if are in view.

## Summary

This project is not even close to production application but in my opinion, it's a very good starter.