# ECHIPA FRONT-END:

Teșu Cristina - ScrumMaster

Teodorescu Alexandru

Chipăruș Denis

Bîrzu Alexandra

Cîrstean Vlad

Păduraru Andra - Elena

Iordache Viorel

Corban Cristian

Țocu Andrei

Păduraru Teodor

Dăscălița Gheorghiță

Zbanț Radu

Hugeanu Felix

In procesul de slelectare al tehnologiilor ne-am concentrat pe doua criterii importante:

* Limitarea tehnologiilor folosite la un numar cat mai mic: deoarece orice tehnologie noua ar insemna invatarea ei de catre toti membrii echipei intr-un timp limitat
* Avantajele pe care le ofera respectivele tehnologii

Astfel procesul de alegere a insemnat balansarea celor doua pentru a obtine cel mai bun raport intre functionalitate si ease of use.

In cele din urma, datorita dificultatii de a gestiona aplicatii separate pentru frontend si backend, ne-am decis sa nu folosim niciun framework si sa utilizam **HTML, CSS, vanilla Javascript si Bootstrap.** Pentru templating folosim solutia oferita de Django (framework-ul de la backend). Pagina principala, necesita Knockout.Js, Gridstack si JQuery.

# TEHNOLOGII PE CARE LE-AM LUAT ÎN CONSIDERARE :

## ANGULAR

|  |  |
| --- | --- |
| **PROS** | **CONS** |
| Two-way data binding (View-ul și Model-ul se sincronizează continuu) | Greu de gestionat si configurat |
| Directives (permit crearea de conținut dinamic și bogat cu HTML) | Dificil de invatat |
| Dependency injection (dependențele se separă de componente, făcându-le mai ușor de reutilizat, de organizat și de testat) | Multa functionalitate de care nu avem nevoie |
| Community (comunitate mare, mai multe materiale și trainning-uri) |  |
| Sustinut de Google |  |

## REACT

|  |  |
| --- | --- |
| **PROS** | **CONS** |
| Virtual DOM (îmbunătățește experiența utilizatorului și crește viteza de dezvoltare a aplicației) | Mediul de lucru se schimbă constant și dezvoltatorii trebuie să învețe noi metode de lucru |
| One-direction data flow (crește stabilitatea codului - modificarea elementelor copil nu afectează părintele) | JSX’s complexity (extensia JSX ce permite combinarea codului HTML cu cod JavaScript este destul de greu de învățat) |
| Open-source (o comunitate în continuă creștere) |  |

## VUE

|  |  |
| --- | --- |
| **PROS** | **CONS** |
| Dimensiune (ocupă puțin spațiu) | Ecosistem de plugin-uri limitat |
| Usor de folosit si integrat | In permanenta dezvoltare |
| Versatilitate |  |
| Cod foarte ușor de citit, modificat și întreținut |  |
| Rapid pentru proiecte de dimensiuni mai mici |  |

CSS:

* CSS-preprocessors (Sass, Less etc): deoarece ar incetini intreaga echipa, am decis sa nu folosim niciun preprocessor. Am reusit sa rezolvam problema customizabilitatii folosind CSS Variables.

Css-Frameworks:

Alegerea noastra este Bootstrap: este un framework popular cu multa documentatie si are foarte multe componente care pot fi folosite. De mentionat faptul ca layout-ul este realizat cu Flex si Grid.

Alte optiuni de Css-Frameworks:

* Bulma: este light si css-only, dar in acelasi timp nu are la fel de multe optiuni ca bootstrap
* Materialize: stilul acestui framework nu e pe gustul tuturor
* Nothing: desi ne ofera cea mai multa customizabilitate, am considerat ca e mai eifcient dar si mai sigur sa folosim un framework care deja a fost testat de mii de utilizatori.