

BIOINFORMATYKA ORŁY ZSŁ

Do stworzenia naszej aplikacji wybraliśmy język TypeScript – nadzbiór języka JavaScript. Jako bibliotekę do tworzenia interfejsu graficznego wybraliśmy React.js

Nad projektem pracowało łącznie 5 osób:

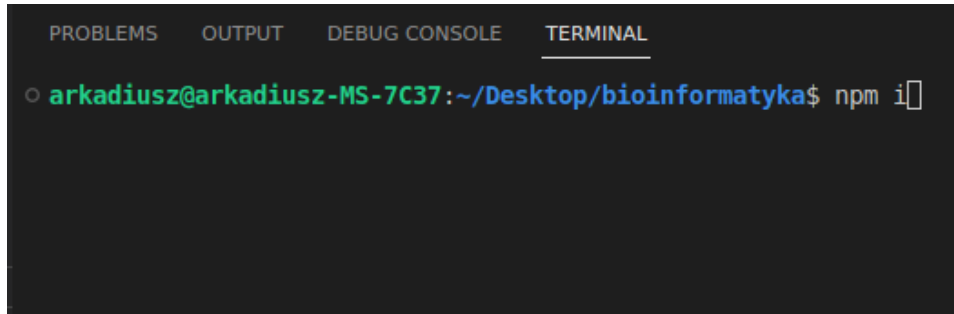
Karol Marek, Arkadiusz Skupień, Maciej Matuszczyk, Adrian Zięba oraz Sebastian Rogóż.

1. Instalacja aplikacji

Do pobrania aplikacji potrzebny jest Node.js. Można go pobrać ze strony: <https://nodejs.org/>

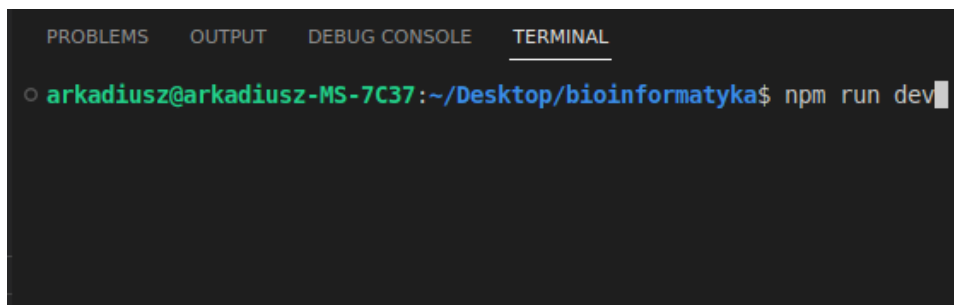
Otworzenie katalogu w terminalu

`npm install` - komenda pozwalająca na pobranie `node_modules`



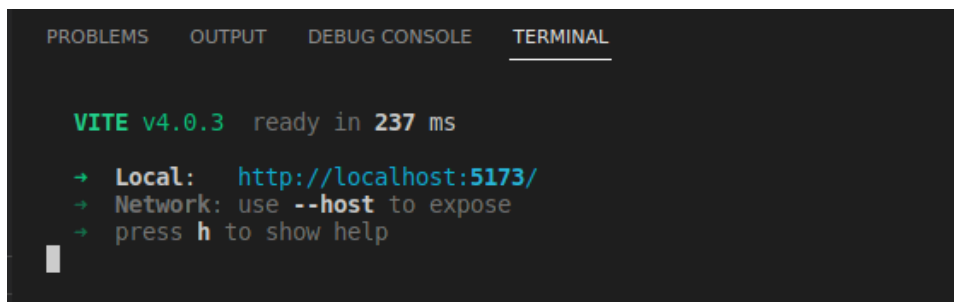
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
o arkadiusz@arkadiusz-MS-7C37:~/Desktop/bioinformatyka$ npm i
```

`npm run dev` - komenda tworząca serwer (domyślnie na localhost:5173)



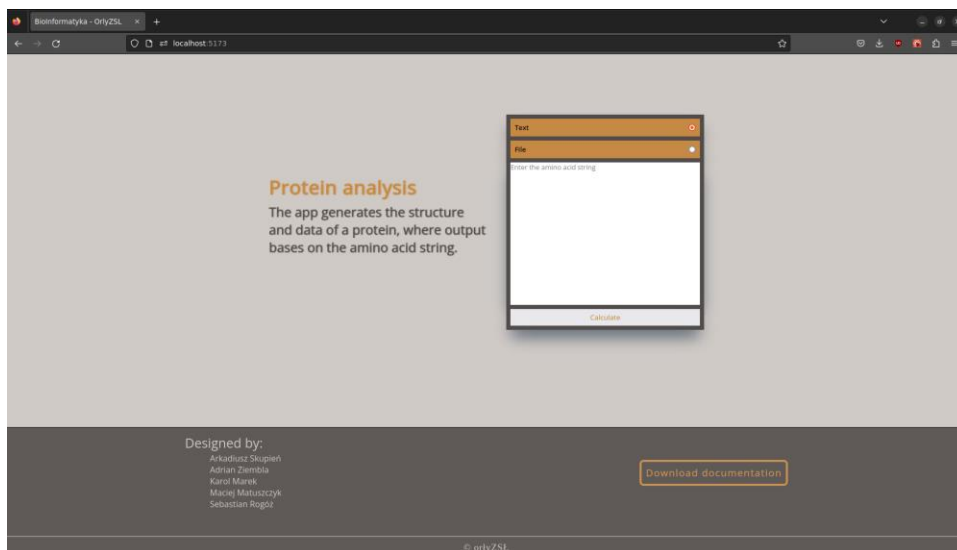
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
o arkadiusz@arkadiusz-MS-7C37:~/Desktop/bioinformatyka$ npm run dev
```

Komenda powinna zwrócić:

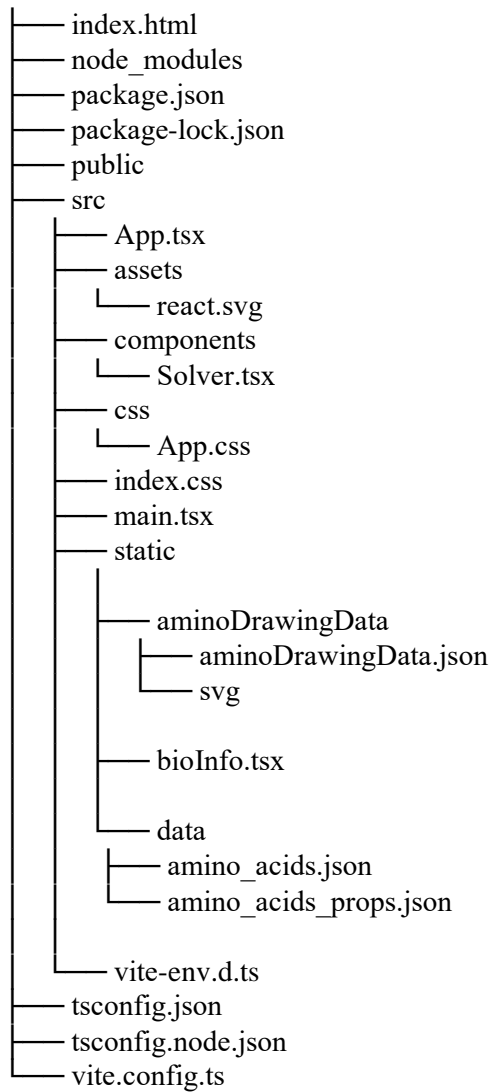


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
VITE v4.0.3 ready in 237 ms
→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h to show help
```

Strona jest dostępna w przeglądarce:



2. Opis struktury aplikacji:



3. Opis plików

index.html

Plik html odczytywany przez przeglądarkę. Do niego renderowane są obiekty.

node_modules

W nim znajdują się moduły niezbędne do poprawnego uruchomienia aplikacji (w tym React)

package.json

Zawiera podstawowe informacje o wersji node, samej aplikacji i o importowanych modułach.

<https://docs.npmjs.com/cli/v9/configuring-npm/package-json>

package-lock.json

Przechowuje szczegółowe informacje o wersji node, o importowanych modułach.

<https://docs.npmjs.com/cli/v9/configuring-npm/package-lock-json>

src

Główny katalog aplikacji

Pliki i podkatalogi:

assets

Ścieżka: src/assets

components

Ścieżka: src/components

Solver.tsx

Ścieżka: src/components/Solver.tsx

Odpowiada za wyświetlanie białek i ich właściwości

Props.tsx

Oblicza wykresy i dane dla aminokwasów

CSS

Ścieżka: src/css

static

Ścieżka: src/static

data

amino_acids_props.json

zawiera informacje o właściwościach aminokwasów

amino_acids.json

Zawiera informacje o podziale kodonów na aminokwasy. Umożliwia odczyt kodu genetycznego

Bioinfo.tsx

Skrypt dzielący kod genetyczny na aminokwasy oraz liczący właściwości białek: ich masę, wzór chemiczny, punkt izoelektryczny, hydrofobowość

App.tsx

Ścieżka: src/App.tsx

Importuje components/Solver.tsx i eksportuje moduł

index.css

Ścieżka: src/index.css

Główny plik css

main.tsx

Ścieżka: src/main.tsx

Renderuje aplikację do kodu html i 'wrzuca' ją do div#root

vite-env.d.ts

Ścieżka: src/vite-env.d.ts

tsconfig.json

Plik konfiguracyjny języka TypeScript

tsconfig.node.json

Plik konfiguracyjny języka TypeScript

vite.config.ts

Plik konfiguracyjny serwera vite

4. Opisy funkcji i klas

bioinfo.tsx

function ObjKey(str: string)

Konwertuje typ string na Object Key używany przy operacjach na obiektach

class BioInformatyka

constructor(_data: string = "")

Konstruktor

dnaToRna(_data: string)

Konwertuje DNA na RNA – zamiana tyminy na uracyl

setData(_data: string)

Ustawia dane wejściowe pod kalkulację

readRNA()

Za pomocą regex'a wyszukuje metioninę i ciąg nukleotydów dopóki nie znajdzie kodonu końca (lub innej metioniny)

codonsIntoAcids(_data: string){

Dzieli nukleotydy na kodony

codonIntoAcid(_data: string)

Dzieli 3 nukleotydy na kodon

getAcids()

Zwraca aminokwasy


```
getProperties(_aminoAcid: string){
```

Za pomocą funkcji liczy właściwości znalezionej białka: długość sekwencji, masę, hydrofobowość, punkt izoelektryczny oraz wzór chemiczny białka

```
calcMass(_aminoAcid: string)
```

Funkcja licząca masę białka

```
calcGravy( amounts: Map<string, number> ) {
```

Funkcja licząca hydrofobowość białka

```
calcIsoelectricPoint(amounts: Map<string,number>) {
```

Funkcja licząca punkt izoelektryczny

```
drawSVG(_aminoAcid : string) {
```

Funkcja zwraca div'a z wieloma obrazami aminokwasów układających się we wzór aminokwasu

Solver.tsx

Odpowiada za elementy wyświetlane na stronie (main, footer)

Props.tsx

Odpowiada za wykresy i dane generowane dla białek