



**creative.**

# Symulacja epidemii

Zadanie dla Kandydatów

(sytuacja opisana w zadaniu jest hipotetyczna, w tej chwili nie budujemy takiego systemu...)

Zostaliśmy poproszeni o stworzenie bardzo, bardzo uproszczonej aplikacji do symulacji rozwoju epidemii (np. COVID-19, ale aplikacja będzie umożliwiała również symulowanie rozprzestrzeniania się grypy czy stworzenie na jej podstawie gry o zombie, wszystko zależy od doboru parametrów i od wyobraźni).

## Uwagi

Zadanie przeznaczone jest zarówno dla FullStacków jak i dla Frontendowców. FullStacków prosimy o wykonanie całego zadania, Frontendowców - tylko frontendu, wówczas obliczenia związane z symulacją muszą być przeprowadzone we frontendzie

**Gorąco zachęcamy do przygotowania pełnej aplikacji - czyli frontend + backend, takie rozwiązania będą preferowane.**

## Symulacje

Użytkownik systemu będzie tworzył symulacje -- dla każdej z nich będzie zadawał początkowe parametry:

- N - Nazwa symulacji (string)
- P - wielkość populacji
- I - początkowa liczba osób zarażonych
- R - wskaźnik określający ile osób zaraża jedna zarażona osoba, czyli znany z newsów covidowych wskaźnik R
- M - wskaźnik śmiertelności, określający ilu spośród zarażonych umiera
- $T_i$  - ilość dni, która upływa od momentu zarażenia do wyzdrowienia chorego
- $T_m$  - ilość dni, która upływa od momentu zarażenia do śmierci chorego
- $T_s$  - Ilość dni, dla których ma być przeprowadzona symulacja

Symulacja będzie więc obiektem zawierającym następujące dane (N, P, I, R, M,  $T_i$ ,  $T_m$ ,  $T_s$ )

Po zdefiniowaniu symulacji system będzie generował początkową populację, to znaczy rekord zawierający informację o:

Pi - liczbie osób zarażonych

Pv - liczbie osób zdrowych, podatnych na infekcję

Pm - liczbie osób zmarłych

Pr - liczbie osób, które wyzdrowiały i nabyły odporność

Następnie iteracyjnie tworzył będzie kolejne obiekty zawierające zbiór danych (Pi, Pv, Pm, Pr), dla kolejnych dni symulacji -- w każdym następnym obiekcie liczba osób w poszczególnych grupach będzie się zmieniać zgodnie z regułami:

- M spośród zarażonych Tm dni wcześniej przenosi się niestety z grupy osób zarażonych do grupy osób zmarłych
- Wszystkie osoby zarażone Ti dni wcześniej (i żywe...) przenoszą się z grupy osób zarażonych do grupy osób, które wyzdrowiały i nabyły odporność
- Każda zarażona osoba zaraża R zdrowych podatnych na infekcję osób

W wyniku przeprowadzonej np. dla 100 dni symulacji otrzymamy więc 100 obiektów (Pi, Pv, Pm, Pr), gdzie  $P_i + P_v + P_m + P_r = P$

Ponieważ chcemy maksymalnie uprościć zadanie, w przyjętym modelu pomijamy takie aspekty jak to, że R może zmieniać się w czasie, bo zależy od zastosowanych środków (maseczki, izolacja, etc.) i od mutacji wirusa czy bakterii, że choroba może przebiegać bezobjawowo, więc nie wszyscy chorzy są poddawani izolacji, że osoba chora nie zaraża innych od momentu zarażenia, a dopiero po kilku dniach (w przypadku COVIDu po ok. 5), że odporność nabyta po wyzdrowieniu może być nietrwała -- słowem pomijamy te i jeszcze bardzo wiele innych elementów, które niepotrzebnie skomplikowałyby proste zadanie rekrutacyjne, ale jednocześnie gdyby wziąć je pod uwagę nasz system stałby się bardziej interesujący, więc oczywiście jeżeli będziecie Państwo mieli ochotę je włączyć do symulacji, to prosimy :) Zależy nam jednak o wiele bardziej na tym, żeby zobaczyć poprawnie działający, dobrze zaprojektowany i ładny system, radzimy więc **najpierw wypełnić warunki zadania, a potem ewentualnie dodawać nadprogramowe funkcjonalności.**

## Baza danych

System będzie przechowywał dane - obiekty symulacji (N, P, I, R, M, Ti, Tm, Ts) i zasymulowane dla każdego dnia symulacji dane (Pi, Pv, Pm, Pr) w dowolnie wybranej bazie danych.

## Backend

Backend będzie stworzony z użyciem dowolnego frameworku i udostępnił interfejs REST lub GraphQL dla frontendu. Wszystkie obliczenia związane z symulacją będą odbywać się w backendzie.

## Frontend

Frontend będzie stworzony z użyciem Angulara (lub ewentualnie Reacta, preferujemy Angular) i będzie komunikował się z Backendem za pośrednictwem interfejsu REST lub GraphQL.

Frontend będzie aplikacją webową - i tak powinien wyglądać.

Frontend będzie składał się z trzech widoków:

- Widok listy, prezentujący listę symulacji - lista symulacji zawierająca kolumny (N, P, I, R, M, Ti, Tm, Ts) wraz z funkcjonalnością tworzenia nowej symulacji, usunięcia symulacji oraz przejścia do widoku szczegółów symulacji
- Widok szczegółów symulacji - będzie umożliwiał przejście do edycji parametrów symulacji; po dokonaniu edycji nastąpi ponowne przeliczenie całego zestawu zasymulowanych danych; widok ten będzie prezentował w sposób estetyczny i czytelne zasymulowane dane, będzie składał się z listy obiektów (Pi, Pv, Pm, Pr) dla każdego dnia symulacji oraz wykresów
- Widok edycji parametrów symulacji - edycja zbioru parametrów (N, P, I, R, M, Ti, Tm, Ts) z odpowiednimi walidacjami

Nie będzie uwierzytelniania ani autoryzacji.

Mnóstwo bardzo ładnych przykładów, na których można się wzorować tworząc widok szczegółów symulacji znajduje się tutaj:

<https://towardsdatascience.com/top-5-r-resources-on-covid-19-coronavirus-1d4c8df6d85f>

Oczywiście nie oczekujemy, że Państwa frontend będzie równie złożony jak podlinkowane w artykule przykłady - w końcu to tylko zadanie rekrutacyjne, a i w naszej symulacji mamy do dyspozycji tylko wartości (Pi, Pv, Pm, Pr), a np. nie mamy danych geograficznych - ale zależałoby nam, żeby przygotowany przez Państwa frontend był funkcjonalny, przedstawiał dane w czytelny sposób i z różnych perspektyw (różne wykresy), a przede wszystkim żeby robił wrażenie :)

Powodzenia! :)