**Instytut Informatyki  
Kolegium Nauk Przyrodniczych  
Uniwersytet Rzeszowski**

**Przedmiot:**

**Proseminarium 22/23**

**Dokumentacja techniczna projektu:**

***Cancer Data Manipulation***

**Wykonali:  
Adrian Wolan i Maciej Ziółkowski**

**Prowadzący: dr hab. Barbara Pękala**

**Rzeszów 2023**

# Temat projektu

Cancer Data Manipulation

# Cele projektu

Celem projektu jest stworzenie aplikacji Internetowej wykorzystującej nowoczesne rozwiązania branży IT, która umożliwiłaby użytkownikowi:

1. Wczytanie danych z pliku zawierającego dane na temat nowotworu
2. Edytowanie poszczególnych rekordów tabeli
3. Zerowanie losowych własności należących do wybranych przez użytkownika, bądź też losowych wierszy z tabeli
4. Resetowanie danych do stanu początkowego
5. Szczegółową analizę danych na podstawie grafów przedstawiających częstotliwość występowania konkretnych wartości z tabeli
6. Rekonstrukcję uprzednio wyzerowanych danych
7. Swobodne przechodzenie między widokami zawierającymi funkcjonalności aplikacji
8. Uruchomienie aplikacji z dowolnego urządzenia (zawierającego przeglądarkę z obsługą Javascript)

# Technologie

Przy tworzeniu aplikacji skorzystano z poniższych technologii:

1. Język Javascript
2. Framework Vue wraz z dodatkami:
   1. Pinia
   2. Vue Router
   3. Vuetify
3. Preprocesor CSS – SCSS
4. Vue Apexcharts
5. Webpack
6. Underscore
7. Github Pages – hosting: https://a1wol.github.io/Cancer-Data-Manipulation/

# Harmonogram realizacji projektu

Zadanie 1 - Wstępne ustalenie głównych założeń dotyczących projektu

Zadanie 2 - Zaprojektowanie GUI aplikacji

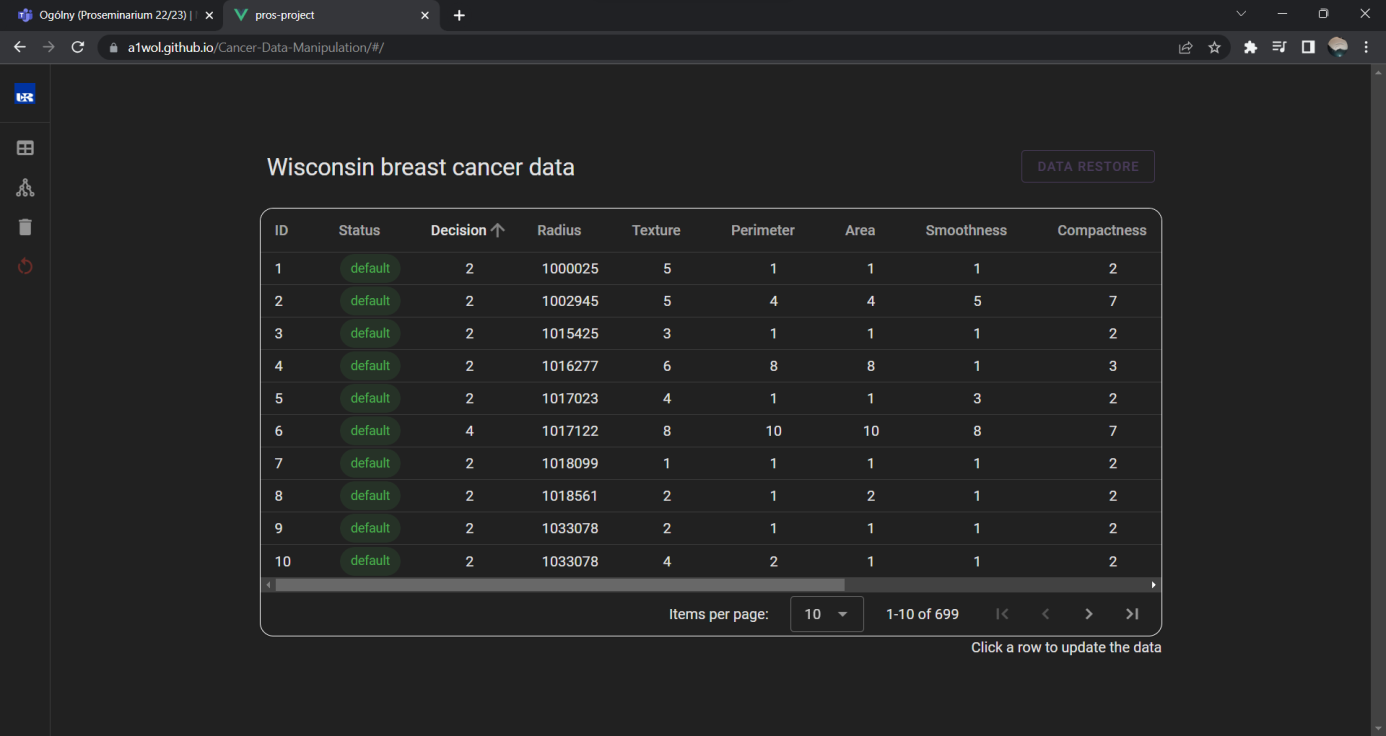
Zadanie 3 - Programowanie elementów działania aplikacji

Zadanie 4 - Dalsze rozwijanie funkcjonalności aplikacji wraz ze wstępnym testowaniem działania na urządzeniach mobilnych

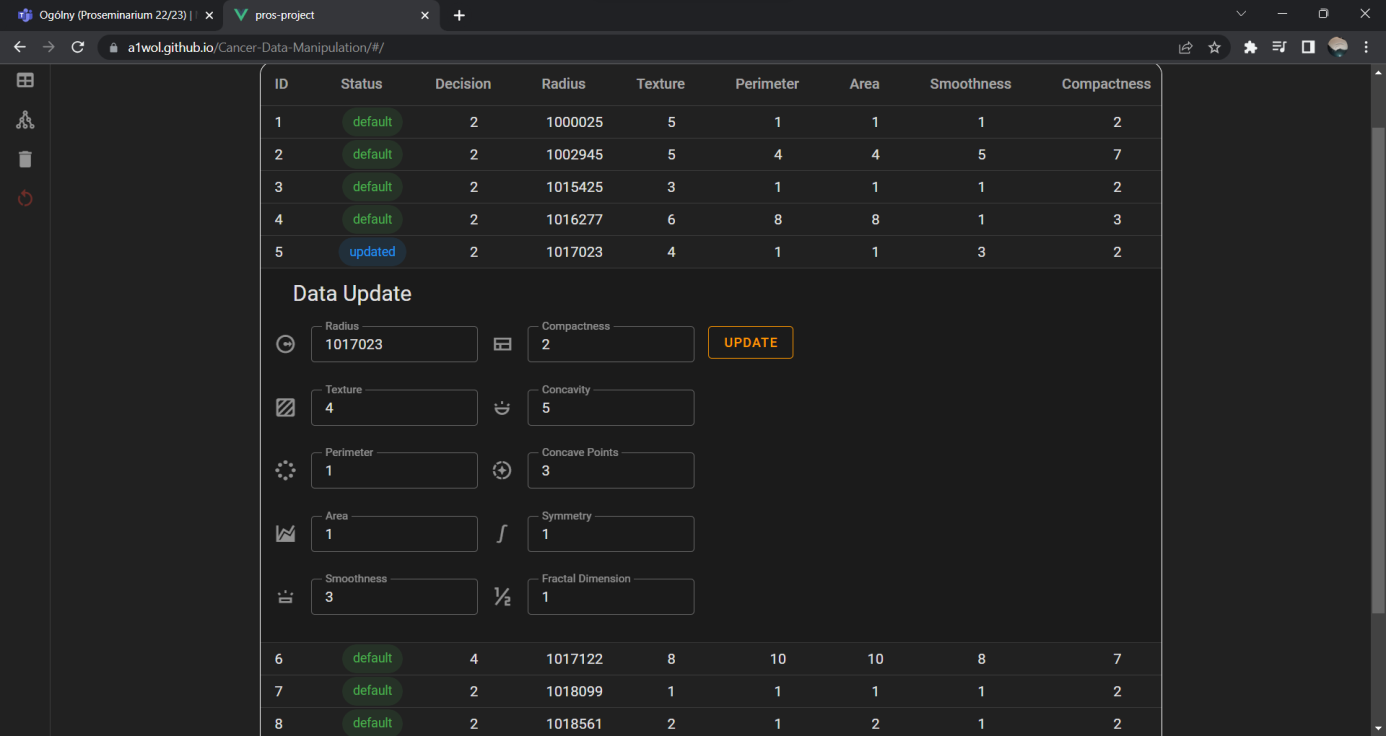
Zadanie 5 - Końcowe udoskonalanie oraz sprawdzanie aplikacji pod kątem funkcjonalności  
oraz wykrywania potencjalnych błędów.

# Działanie aplikacji

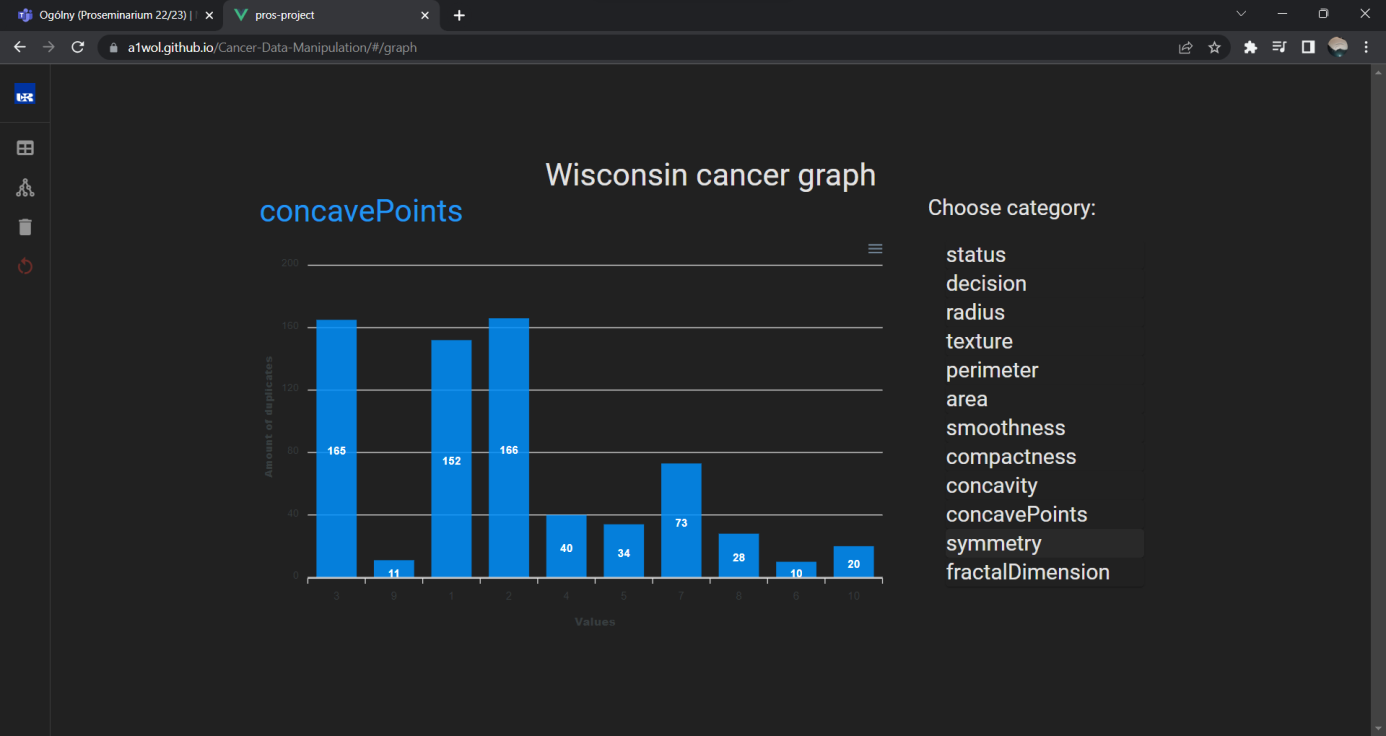
1. Tablica z danymi (strona startowa)



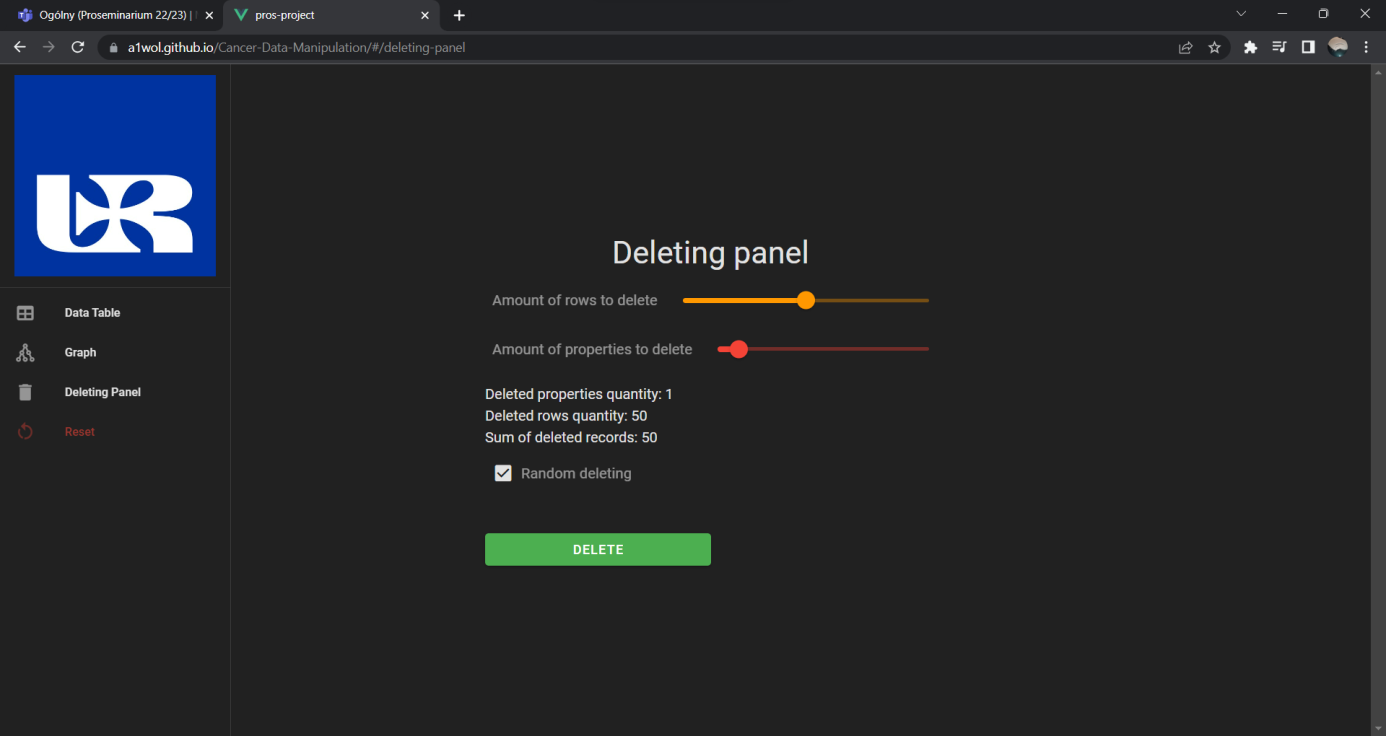
1. Edycja danych. Status wiersza zmienia się na „edited”



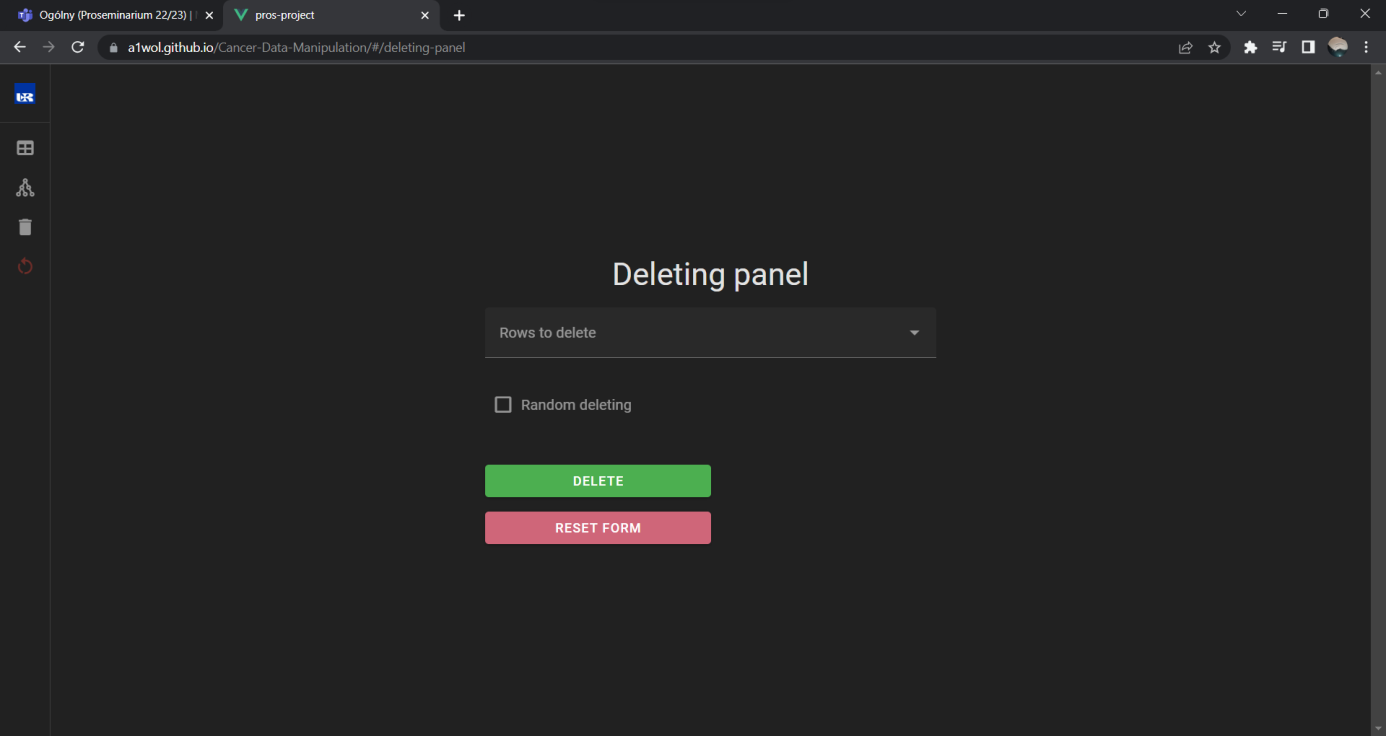
1. Analiza grafów



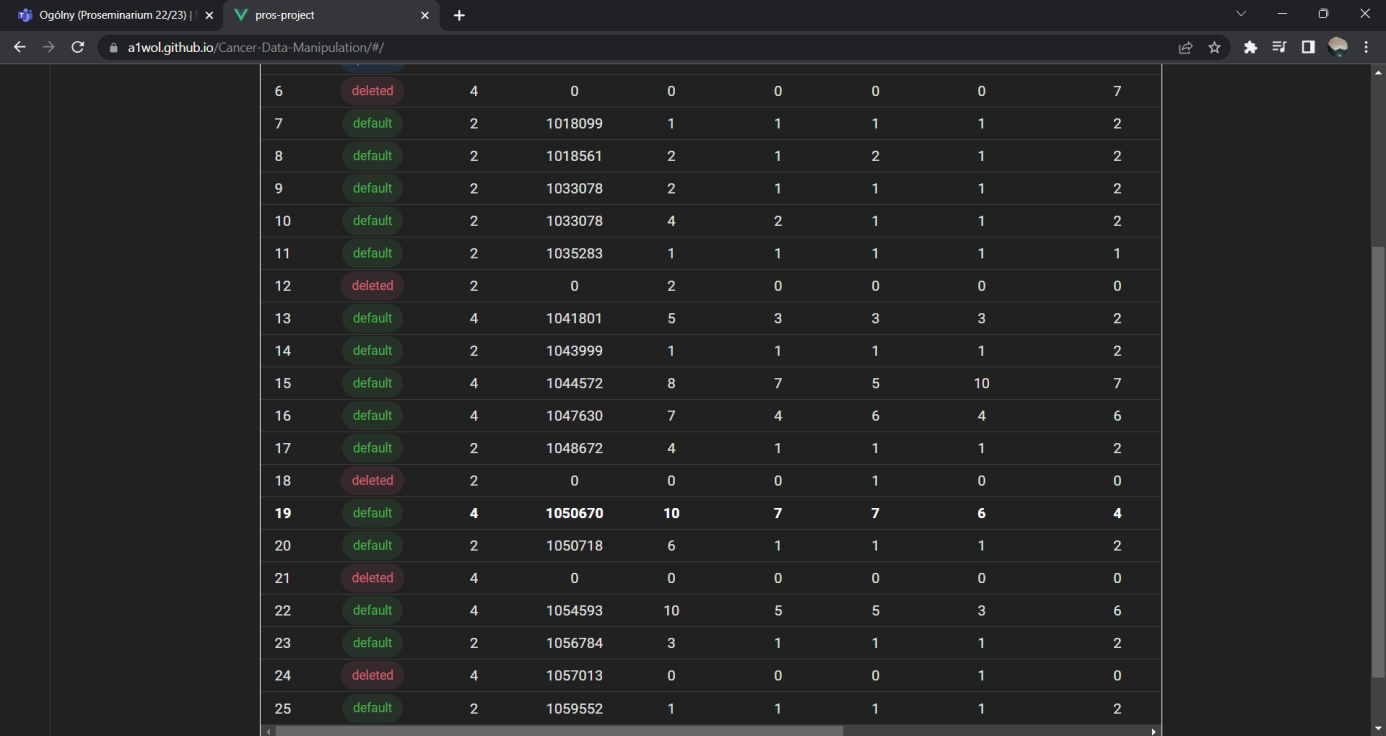
1. Panel usuwania danych wraz z widocznym rozwiniętym paskiem nawigacyjnym:  
   Należy zwrócić uwagę na chwilowy brak możliwości ponownego wczytania danych.  
   Pojawi się ona po ich edycji.



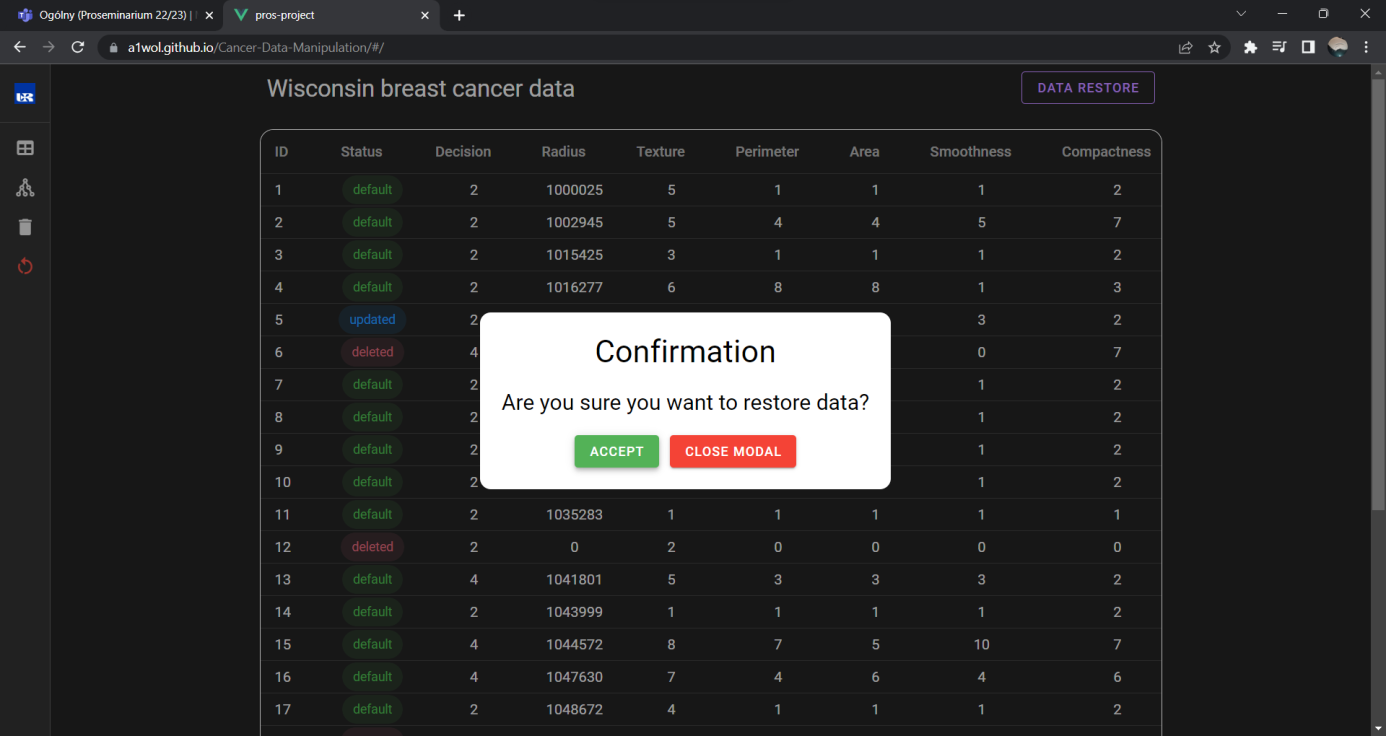
1. Panel usuwania danych – ręczne wprowadzanie wierszy



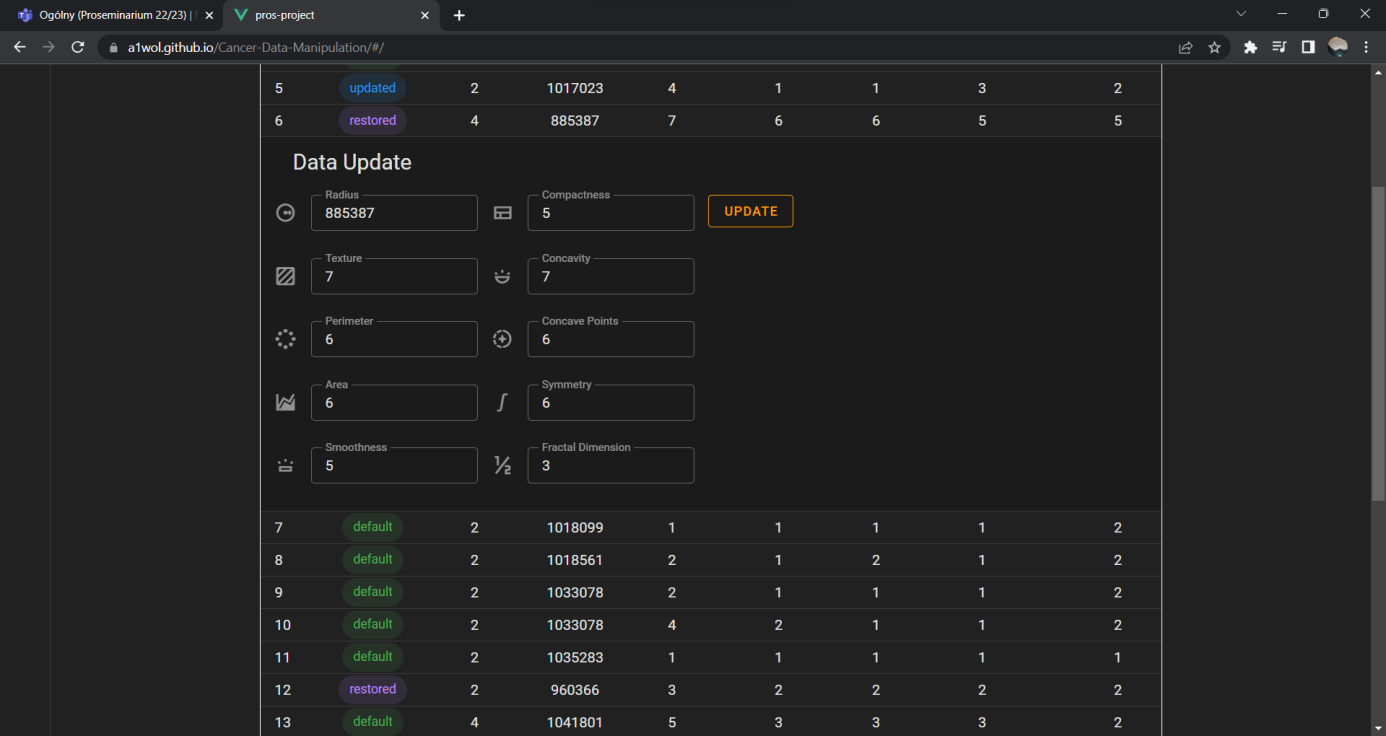
1. Widok tabeli z danymi po wyzerowaniu niektórych danych



1. Po kliknięciu na przycisk „Data Restore” uruchamiany jest popup potwierdzający chęć odrestaurowania danych. Podobne potwierdzenie wymagane jest przed wyzerowaniem danych



1. Po zaakceptowaniu, wyliczone dane są wprowadzone w miejsca wcześniej wyzerowane, na podstawie ich wartości decyzyjnych



# Struktura programu

|  |  |
| --- | --- |
|  | 1. Assets: Pliki związane z graficzną oprawą aplikacji, takie jak zdjęcia 2. Classes: Pliki Javascriptowe zawierające wyeksportowane klasy służące do tworzenia obiektów 3. Components: Komponenty Vue zbierane i obsługiwane w większych agregatorach - widokach. Przechowywane są tu także powtarzające się elementy interfejsu, czyli popup 4. Dist/data: Folder przechowujący dane na temat nowotworu. Dane te są następnie przetwarzane przez tzw. Raw-loader 5. Plugins: Pliki odpowiedzialne za prawidłową pracę dodatków do aplikacji 6. Router: Plik przechowujące ścieżki do widoków 7. Store: Przechowywanie i modyfikacja zmiennych dostępnych globalnie z każdego poziomu aplikacji 8. Views: widoki Vue 9. App.vue: tzw. Root aplikacji, czyli główny element całego projektu |

# Literatura

* <https://vuejs.org/>
* <https://pinia.vuejs.org/>
* <https://webpack.js.org/>
* <https://docs.github.com/en/pages>
* <https://sass-lang.com/>
* <https://underscorejs.org/>
* <https://apexcharts.com/docs/vue-charts/#>
* <https://vuetifyjs.com/en/>