

Zadanie: Tworzenie komponentu Angular do sortowania liczb z własną implementacją algorytmu sortowania

Opis zadania:

Twoim zadaniem jest:

1. **Stworzenie komponentu Angular**, który umożliwi użytkownikowi:
 - Wprowadzanie liczb do listy (np. poprzez pole tekstowe i przycisk “Dodaj”).
 - Wyświetlanie aktualnej listy wprowadzonych liczb.
 - Dodaj możliwość wyboru algorytmu sortowania (np. poprzez listę rozwijaną).
 2. **Przechowywanie wprowadzonych liczb** w tablicy w komponencie.
 3. **Zaimplementowanie własnej wersji algorytmów sortowania**
 - Sortowanie bąbelkowe.
 - Sortowanie przez wstawianie.
 - Sortowanie przez wybieranie.
 4. **Dodanie funkcjonalności sortowania**, która pozwoli użytkownikowi wybrać metodę sortowania
 5. **Aktualizacja wyświetlanej listy** po sortowaniu.
- **Nie korzystaj z wbudowanej metody** `Array.sort()`. Zaimplementuj algorytm sortowania samodzielnie.
 - Upewnij się, że interfejs użytkownika jest czytelny i przyjazny.

Kryteria oceny:

- **Funkcjonalność:**
- Poprawne dodawanie liczb do listy.
- Poprawne wyświetlanie listy liczb.
- Poprawne działanie funkcji sortowania.
- **Implementacja:**
- Własna implementacja wybranego algorytmu sortowania.
- Kod jest czytelny i dobrze zorganizowany.
- Stosowanie dobrych praktyk programowania w Angularze.
- **Interfejs użytkownika:**
- Intuicyjność i estetyka formularza oraz listy liczb.
- Responsywność i użyteczność aplikacji.

Dodatkowe wyzwania:

- **Walidacja danych:**
- Upewnij się, że użytkownik wprowadza tylko liczby.
- Dodaj komunikaty błędów dla nieprawidłowych danych.
- **Usuwanie liczb z listy:**
- Pozwól użytkownikowi usuwać wybrane liczby z listy

Tabela 1. Wybrane elementy frameworka Angular, biblioteki React.js i biblioteki Bootstrap

<p>Angular To use ngModel i ngForm add: <code>import { FormsModule } from '@angular/forms';</code> in <code>app.module.ts</code>. imports table To use Bootstrap add to <code>styles.css</code>: <code>@import '~bootstrap/dist/css/bootstrap.css';</code></p>
<p>React.js To use Bootstrap add: <code>import 'bootstrap/dist/css/bootstrap.css';</code></p>
<p>Bootstrap Forms (Źródło https://getbootstrap.com/docs/4.0/components/forms/) Be sure to use an appropriate type (e.g., email for email address or number for numerical information) to take advantage of newer input controls like <code><input></code>s, <code><select></code>s, and <code><textarea></code>s—are styled with the <code>form-control</code> class. Textual form controls—like <code><input></code>s, <code><select></code>s, and <code><textarea></code>s—are styled with the <code>form-control</code> class. <code><form></code> <code><div class="form-group"></code> <code><label for="exampleInputEmail1"></code> <code>address</label></code> <code><input type="email" class="form-control" id="exampleInputEmail1" /></code> <code></div></code> <code></form></code> Important! In React render method use <code>className</code> instead of <code>class</code>; <code>htmlFor</code> instead of <code>for</code>.</p>
<p>Bootstrap buttons (Źródło https://getbootstrap.com/docs/4.0/components/buttons/) Bootstrap includes several button classes, each serving its own semantic purpose, with a few extras thrown in for more control. The <code>btn</code> classes are designed to be used on the <code><button></code> element. Add modifier classes as: <code>btn-primary</code>, <code>btn-secondary</code>, <code>btn-success</code> and more to <code>btn</code> class in order to style the button. <code><button type="button" class="btn btn-success">Success</button></code></p>