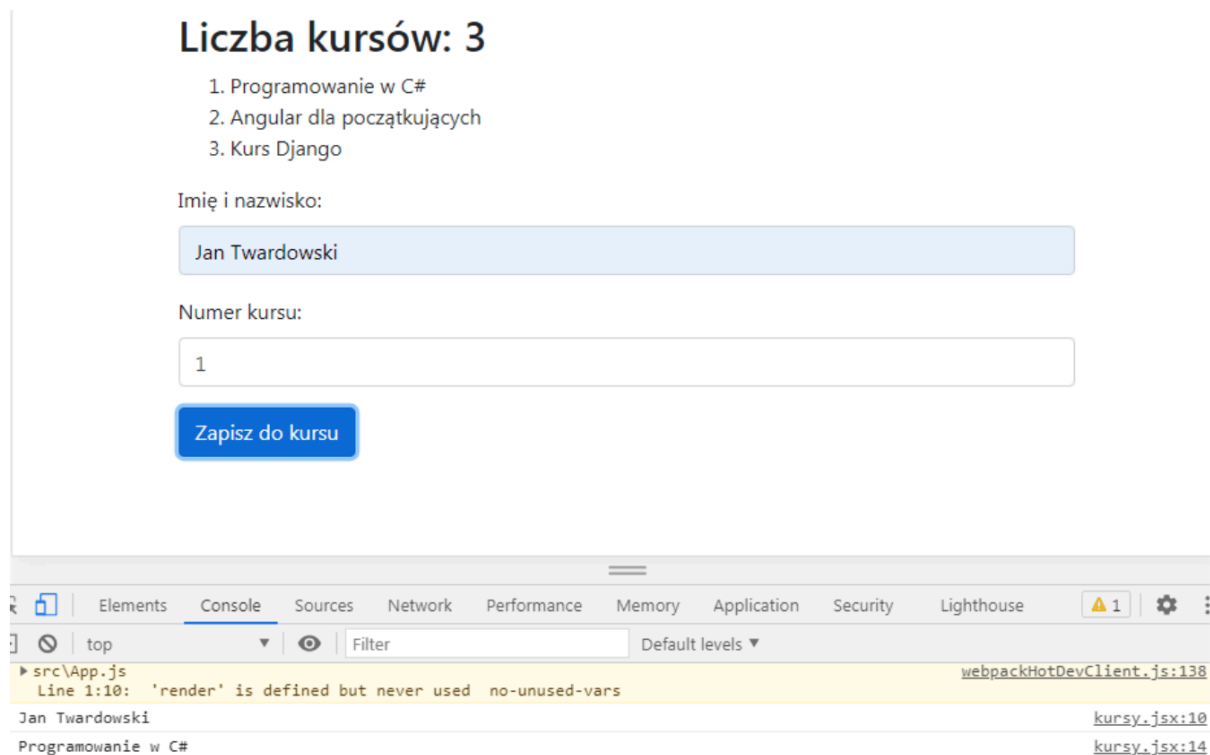


Wykonaj aplikację internetową typu `front-end` obsługującą zapisy na kursy z zastosowaniem biblioteki `React`. Zastosuj bibliotekę `Bootstrap` do zdefiniowania stylu formularza.



**Obraz 1. Aplikacja React.js**

Na obrazie 1 przedstawiono działanie aplikacji przygotowanej w środowisku `React.js`, stan po wybraniu przycisku „Zapisz do kursu”. W konsoli widoczne wyświetlanie `imienia` i `nazwiska` oraz `nazwy kursu` na podstawie danych wprowadzonych do formularza.

Założenia aplikacji:

- Aplikacja składa się z jednego komponentu
- Danymi komponentu jest `tablica` z `nazwami kursów`, o elementach: "Programowanie w C#", "Angular dla początkujących", "Kurs Django". Dla uproszczenia tablica może być zdefiniowana jako `pole komponentu`. Należy założyć, że tablica w przyszłości może się zmienić, co będzie miało wpływ na zachowanie i wygląd aplikacji.

- Komponent wyświetla:
  - Nagłówek **drugiego** stopnia o treści: „Liczba kursów: <liczba>”, gdzie <liczba> oznacza **wielkość** tablicy z nazwami kursów
  - Listę **numerowaną** generowaną automatycznie dla wszystkich elementów tablicy, niezależnie od jej wymiaru
  - Formularz składający się z:
    - pola **edycyjnego** i jego etykiety o treści „Imię i nazwisko:”
    - edycyjnego pola **numerycznego** i jego etykiety o treści „Numer kursu:”
    - **przycisku** „Zapisz do kursu”
- Aplikacja w stanie początkowym wyświetla puste pola formularza
- Elementy formularza są formatowane zgodnie z obrazem 1 za pomocą stylów biblioteki **Bootstrap**. Do budowy szablonu HTML należy wykorzystać pomoc zamieszczoną w **Tabeli 1**. Należy zastosować znaczące nazwy dla identyfikatorów pól formularza
- Po wybraniu przycisku formularza jest generowane zdarzenie zatwierdzenia formularza, które wyświetla w konsoli przeglądarki:
  - **Wartość** wpisaną w pierwszym polu formularza
  - **Nazwę kursu** odpowiadającą numerowi wpisanemu w drugie pole formularza, gdy kurs o takim numerze istnieje. W przeciwnym wypadku wyświetla komunikat "Nieprawidłowy numer kursu"
- Aplikacja powinna być napisana czytelnie, z zachowaniem zasad czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji
- Dokumentacja do programu wykonana zgodnie z wytycznymi z **części III** zadania

W podfolderze web powinno znaleźć się archiwum całego projektu o nazwie **web.zip** oraz pliki z kodem źródłowym, które były modyfikowane.

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

**Tabela 1**

## Część III. Testy utworzonych aplikacji

Wykonaj testy oraz dokumentację do aplikacji.

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji. Zrzuty powinny obejmować cały obszar ekranu monitora z widocznym paskiem zadań.

Jeżeli aplikacja **uruchamia się**, na zrzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem lub okno terminala z kompilacją projektu.

Jeżeli aplikacja `nie uruchamia się` z powodu błędów kompilacji, należy na zrzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym.

Wykonać należy tyle zrzutów ile interakcji podejmuje aplikacja. Wymagane zrzuty ekranu:

Aplikacja `web` – dowolna liczba zrzutów nazwanych `web1`, `web2` ... (np. stan początkowy, po wypełnieniu pól, po zatwierdzeniu konsola przeglądarki, gdy w drugim polu formularza jest prawidłowy numer oraz gdy jest nieprawidłowy numer kursu)

W edytorze tekstu pakietu biurowego utwórz plik z dokumentacją i nazwij go egzamin. Dokument powinien zawierać informacje:

Nazwę `systemu operacyjnego`, na którym pracował zdający

Nazwy `środowisk programistycznych`, z których zdający korzystał na egzaminie

Nazwy `języków programowania` / `frameworków` / `bibliotek` użytych podczas tworzenia aplikacji

Zrzuty ekranu i dokument umieścić w folderze o nazwie `testy`.