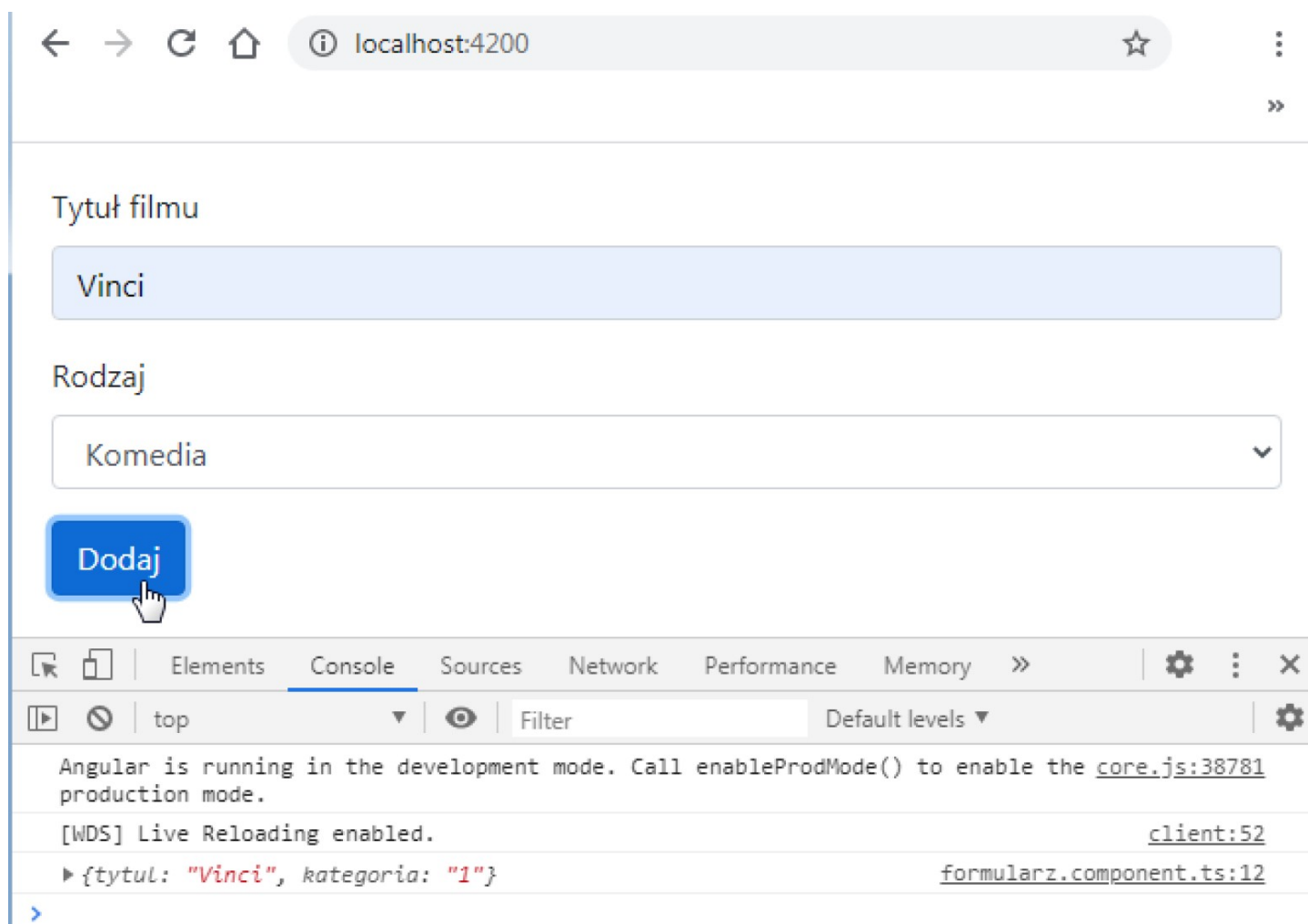


Część II. Aplikacja Web

Wykonaj aplikację internetową typu front-end obsługującą formularz z zastosowaniem dostępnego na stanowisku egzaminacyjnym frameworka Angular lub biblioteki React. Zastosuj bibliotekę Bootstrap do zdefiniowania stylu formularza.



Obraz 1a. Formularz wykonany we frameworku Angular. Wybrano przycisk Dodaj

Na obrazach 1a i 1b przedstawiono działanie formularza przygotowanego w środowisku Angular i React.js po wybraniu przycisku Dodaj. W konsoli widoczne wyświetlenie danych wprowadzonych do formularza

Założenia aplikacji

- Aplikacja składa się z jednego komponentu formularza.
- Formularz składa się z:
 - pola edycyjnego i jego etykiety o treści „Tytuł filmu”
 - listy rozwijalnej i jej etykiety o treści „Rodzaj”
 - przycisku „Dodaj”
- Lista rozwijalna ma 5 opcji:
 - pustą
 - Komedia o wartości 1
 - Obyczajowy o wartości 2
 - Sensacyjny o wartości 3
 - Horror o wartości 4
- Aplikacja w stanie początkowym wyświetla puste pola formularza
- Po wybraniu przycisku „Dodaj” jest generowane zdarzenie, które ma za zadanie wypisanie danych z formularza w konsoli przeglądarki (w miejscu trzech kropek wartości pól formularza)
 - w Angular postaci: {tytul: "...", kategoria: "..."} – obraz 1a

- Strona jest formatowana stylem: `body { padding: 20px; }`
- Elementy formularza są formatowane stylami biblioteki Bootstrap. Szablon HTML formularza należy zbudować sugerując się pomocą zamieszczoną w Tabeli 1. Należy zastosować znaczące nazwy dla identyfikatorów pól formularza
- Aplikacja powinna być zapisana 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 egzaminacyjnego.

Kod aplikacji przygotuj do nagrania na płytę. W podfolderze *web* powinno znaleźć się archiwum całego projektu o nazwie *web.zip* oraz pliki z kodem źródłowym które były modyfikowane w czasie egzaminu.

Angular

To use `ngModel` i `ngForm` add: `import { FormsModule } from '@angular/forms';` in `app.module.ts` file. Add `FormsModule` to imports table

To use Bootstrap add to `styles.css`: `@import "~bootstrap/dist/css/bootstrap.css";`

React.js

To use Bootstrap add: `import 'bootstrap/dist/css/bootstrap.css';`

Bootstrap Forms (Źródło <https://getbootstrap.com/docs/4.0/components/forms/>)

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.

Textual form controls—like `<input>`s, `<select>`s, and `<textarea>`s—are styled with the `.form-control` class. Included are styles for general appearance, focus state, sizing, and more.

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" />
  </div>
  <div class="form-group">
    ...
  </div>
</form>
```

Important! In React render method use `className` instead of `class`; `htmlFor` instead of `for`.

Bootstrap buttons (Źródło <https://getbootstrap.com/docs/4.0/components/buttons/>)

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 `<button>` element. Add modifier classes as: `btn-primary`, `btn-secondary`, `btn-success` and more to `btn` class in order to add background colors. e.g. `<button type="button" class="btn btn-success">Success</button>`