

Bootstrap 5 — szybki interfejs sklepu

Czas: 2 × 45 min

praca indywidualna

Cel: Zbudowanie responsywnej strony sklepu z gotowych komponentów Bootstrap 5 (siatka, navbar, karty, formularz z walidacją, modal) oraz porównanie tego podejścia z własnym CSS i z instalacją przez npm.

Po zajęciach potrafisz:

- dołącza Bootstrap z CDN z atrybutem integrity i jako zależność npm
- projektuje układ siatkę container/row/col z punktami przerwania (sm, md, lg)
- używa komponentów navbar z menu zwijanym, card, modal i formularza z klasami walidacji
- stosuje klasy narzędziowe (odstęp, flex, kolory, display) i nadpisuje wygląd zmiennymi CSS --bs-*
- ocenia zalety i wady frameworka CSS w porównaniu z własnym arkuszem

Sprzęt i materiały

- komputer z Windows 11, VS Code (Live Server) i Node.js LTS
- folder `r2_25_start` z Pliki_do_zadan (treść `tresc.txt`, folder `img`, folder `wlasny_css`)

Oprogramowanie

- **Dokumentacja Bootstrap 5.3** — przykłady komponentów do kopiowania i dostosowania
getbootstrap.com/docs/5.3/getting-started/introduction
- **Visual Studio Code** — rozszerzenie Live Server
code.visualstudio.com
- **Node.js LTS** — część 5: npm i Vite
nodejs.org

Przygotowanie przed zajęciami

- Przejrzyj w prezentacji 2.16 siatkę Bootstrapa (12 kolumn, punkty przerwania) i listę komponentów.

Bezpieczeństwo

- Dołączając pliki z CDN, używaj adresów z dokumentacji z atrybutami integrity i crossorigin — przeglądarka odrzuci zmieniony plik.
- Instaluj pakiety npm tylko z oficjalnego rejestru i o sprawdzonej nazwie (bootstrap, @popperjs/core).

Przebieg zadania

1 Szablon i pasek nawigacji

Utwórz folder `nazwisko_imie/cdn` i skopiuj do niego folder `img`. Treści wszystkich elementów weź z pliku `tresc.txt`.

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-sRIIL4kxILFvY47J16cr9ZwB07vP4J8+LH7qKQnuqkuIAvNWLzeN8tE5YBuJZqJLB"
crossorigin="anonymous">
...
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.8/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-FKyoEForCGlyvwx9Hj09JcYn3nv7wiPVLz7YYwJrWVcXK/BmnVDxM+D2scQbITxI"
crossorigin="anonymous"></script>
```

1. Utwórz `index.html` (`lang="pl"`, meta viewport) i dołącz CSS oraz JS Bootstrapa jak w ramce (skrypt na końcu body). Dołącz też własny plik `style.css` po CSS Bootstrapa.
2. Dodaj navbar z klasą `navbar-expand-lg`: logo i nazwa sklepu, 5 odnośników menu (pierwszy z klasą `active` i `aria-current="page"`), po prawej przycisk „Koszyk” z plakietką (badge) 0. Na wąskim ekranie menu ma się zwiijać pod przycisk-hamburger.
3. Ustaw kolor marki `#2f6b3a`: w `style.css` utwórz klasę `.bg-herbata` dla paska i stopki oraz klasę `.btn-herbata`, nadpisując zmienne `--bs-btn-bg`, `--bs-btn-hover-bg` i pokrewne (nie zmieniaj plików Bootstrapa).

2 Sekcja powitalna i siatka produktów

1. Dodaj sekcję powitalną: w `container` wiersz `row` z kolumną tekstu `col-lg-7` (nagłówek z klasą `display-5`, akapit `lead`, dwa przyciski) i kolumną grafiki `col-lg-5`, ukrytą poniżej lg klasami `d-none d-lg-block`.
2. Wyświetl 6 produktów w siatce `row row-cols-1 row-cols-sm-2 row-cols-lg-3 g-4` jako karty `card h-100`: grafika `card-img-top` z alt, plakietki kategorii i dostępności (zielona / żółta „ostatnie sztuki” / szara), nazwa, opis, parzenie, cena i przycisk.
3. Dopasuj karty klasami narzędziowymi: `d-flex flex-column` w treści karty i `mt-auto` w wierszu z ceną, tak aby ceny i przyciski były w jednej linii we wszystkich kartach. Dla produktu niedostępnego przycisk ma atrybut `disabled`.
4. Sprawdź układ w trybie urządzeń F12 (Ctrl+Shift+M) dla szerokości 390, 768 i 1280 px. Zapisz liczbę kolumn kart dla każdej szerokości i zrób zrzuty.

3 Formularz, modal i stopka

1. Dodaj formularz newslettera wg `tresc.txt` w kolumnie `col-md-10 col-lg-7` wyśrodkowanej `justify-content-center`: pola imię i e-mail obok siebie od sm (`col-sm-6`), lista, pole zgody `form-check`, przycisk na całą szerokość (`w-100`).
2. Włącz walidację Bootstrapa: formularz z klasą `needs-validation` i atrybutem `novalidate`, komunikaty `invalid-feedback`, a w pliku `app.js` przy wysyłce dodaj klasę `was-validated`, gdy `checkValidity()` zwraca `false`; przy poprawnych danych pokaż `alert alert-success`.
3. Dodaj modal „Koszyk” otwierany przyciskiem z paska (`data-bs-toggle="modal"`), z nagłówkiem, przyciskiem zamknięcia z `aria-label="Zamknij"` i stopką. Dodaj stopkę strony z danymi sklepu (`flex-column flex-md-row`).

4 Porównanie z własnym CSS

Folder `wlasny_css` zawiera ten sam fragment strony (menu i 3 karty) napisany bez frameworka.

1. Otwórz `wlasny_css/index.html` i swoją stronę. Porównaj: liczbę linii/bajtów własnego CSS, zachowanie menu na telefonie, rozmiar pobranych plików CSS i JS (F12 → Network, kolumna Transferred).
2. Zapisz w sprawozdaniu tabelę porównania i co najmniej 2 zalety i 2 wady Bootstrapa.

5 Wersja przez npm i Vite

```
npm create vite@latest lisc-npm -- --template vanilla
cd lisc-npm
npm install
npm install bootstrap @popperjs/core

// src/main.js
import 'bootstrap/dist/css/bootstrap.min.css'
import './style.css'
import * as bootstrap from 'bootstrap'
```

1. Utwórz projekt Vite poleceniami z ramki, przenieś do `index.html` treść strony bez znaczników CDN, grafiki umieść w folderze `public/img`, a `style.css` i kod walidacji w `src`.
2. Sprawdź w `package.json` wpis bootstrap i jego zakres wersji (`^5.3.8`). Wyjaśnij, co oznacza znak `^`.
3. Uruchom `npm run dev`, a potem `npm run build` i zapisz rozmiary plików CSS i JS w dist. Spakuj obie wersje (bez node_modules) w folderze `nazwisko_imie`.

W sprawozdaniu umieść

- ☐ zrzuty strony dla 390, 768 i 1280 px (w tym rozwinięte menu na telefonie), formularza z błędami walidacji i otwartego modala
- ☐ tabela porównania z wersją z własnym CSS oraz zalety i wady Bootstrapa
- ☐ rozmiary plików z `npm run build` i wyjaśnienie znaku `^`
- ☐ foldery cdn i npm bez node_modules, odpowiedzi na pytania

Pytania kontrolne

1. Co oznacza zapis `row-cols-1 row-cols-sm-2 row-cols-lg-3` i od jakich szerokości działa każda klasa?
2. Dlaczego własny plik CSS dołączasz po CSS Bootstrapa i dlaczego lepiej nadpisywać zmienne `--bs-*` niż edytować `bootstrap.min.css`?
3. Do czego służy atrybut `integrity` przy plikach z CDN?
4. Kiedy wybraćbyś wersję z CDN, a kiedy instalację przez npm?

Kryteria oceny

Kryterium	Punkty
Bootstrap z CDN z integrity, navbar zwijany z aktywnym odnośnikiem i koszykiem	3
Sekcja powitalna i siatka 6 kart: 1/2/3 kolumny, wyrównane ceny, plakietki, disabled	4
Formularz z walidacją (was-validated, komunikaty, alert sukcesu)	3
Modal koszyka i stopka; kolor marki przez własne klasy i zmienne --bs-btn-*	3
Porównanie z własnym CSS (tabela, zalety, wady)	2
Wersja npm + Vite: działa, build, rozmiary	2
Odpowiedzi na pytania kontrolne	3
Razem	20

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).