

Projekt roczny cz. 3 — JavaScript, testy i publikacja

Czas: 2 × 45 min

praca w zespołach 2–3 osobowych

Cel: Dodanie do projektu zespołowego interaktywnych elementów JavaScript, przetestowanie witryny (scenariusze, W3C, Lighthouse), opublikowanie jej i zaprezentowanie efektów.

Po zajęciach potrafisz:

- implementuje zaplanowane funkcje JavaScript w osobnym pliku, uruchamiane tylko na podstronach, które ich potrzebują
- przygotowuje i wykonuje scenariusze testów oraz dokumentuje wyniki w protokole
- poprawia błędy wskazane przez walidator W3C i audyt Lighthouse
- publikuje witrynę (GitHub Pages lub hosting) i sprawdza działanie wersji online
- prezentuje projekt w 5 minut, uzasadniając decyzje projektowe

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu; smartfon do testu (opcjonalnie)
- repozytorium projektu po zadaniu 32

Oprogramowanie

- **Visual Studio Code** — Live Server
code.visualstudio.com
- **Przeglądarka Chrome lub Edge** — F12: Console, Lighthouse, tryb urządzeń
- **Walidator W3C** — HTML; CSS:
<https://jigsaw.w3.org/css-validator/validator.w3.org>
- **GitHub Pages** — lub hosting szkolny przez SFTP (zadania 28–29)
pages.github.com

Przygotowanie przed zajęciami

- `git pull` u każdego członka zespołu; sprawdźcie w karcie projektu, które elementy JS zaplanowaliście i kto je wykonuje.
- Powtórzcie zadania 25–27: DOM, zdarzenia, walidacja formularza, localStorage.

Bezpieczeństwo

- Dane wpisane przez użytkownika wstawiajcie przez `textContent`; nie używajcie `innerHTML` z danymi z formularza.
- Formularz na stronie statycznej nigdzie nie wysyła danych — nie zbierajcie prawdziwych danych osobowych, a w testach używajcie danych zmyślonych (`example.edu.pl`).
- Przed publikacją usuńcie z repozytorium pliki prywatne i dane dostępne (zadanie 28).

Przebieg zadania

1 JavaScript

Cały kod w pliku `js/main.js` dołączonym z `defer`. Kod dla jednej podstrony uruchamiajcie warunkowo, np. `const katalog = document.querySelector("#katalog"); if (katalog) { ... } —` —
inaczej na pozostałych stronach pojawi się błąd.

1. J1. Zaimplementujcie co najmniej 2 elementy JS z karty projektu, w tym co najmniej jeden reagujący na dane wpisane przez użytkownika (filtr, wyszukiwarka, walidacja formularza, kalkulator, quiz).
2. J2. Jeśli projekt ma formularz: własna walidacja przy `submit` z `preventDefault`, komunikaty pod polami (`textContent`), ustawienie focusu w pierwszym błędnym polu, potwierdzenie po poprawnym wypełnieniu.
3. J3. Strona ma pozostać używalna bez JavaScript: treść (np. karty katalogu) jest w HTML, a skrypt ją tylko filtruje lub uzupełnia. Sprawdźcie to, wyłączając JS (F12 → Ctrl+Shift+P → „Disable JavaScript”).
4. J4. Każdy członek zespołu zatwierdza swoją część kodu osobnym commitem z opisowym komunikatem.

2 Testy i poprawki

Test zapisujcie w tabeli: numer, funkcja, dane lub czynność, oczekiwany wynik, wynik (OK / błąd i poprawka).

1. T1. Przygotujcie i wykonajcie co najmniej 8 scenariuszy testów (w tym: dane poprawne, puste pole, wartość graniczna, brak wyników, wielkość liter, odświeżenie strony, wąski ekran 360 px, nawigacja klawiaturą). Sprawdźcie konsolę na każdej podstronie.
2. T2. Sprawdźcie wszystkie podstrony walidatorem W3C HTML i arkusz walidatorem CSS — cel: 0 błędów.
3. T3. Uruchomcie Lighthouse (Mobile) na stronie głównej i jeszcze jednej podstronie. Cel z karty: dostępność i SEO ≥ 90 . Poprawcie co najmniej dwa zgłoszone problemy i zapiszcie wyniki przed i po.

3 Publikacja i prezentacja

1. U1. Opublikujcie witrynę przez GitHub Pages (Settings → Pages) lub na serwerze szkolnym. Sprawdźcie wersję online: wszystkie podstrony, grafiki, skrypty (wielkość liter w nazwach plików!), test na telefonie.
2. U2. Uzupełnijcie `PROJEKT.txt` lub `README.md`: adres strony, autorzy i ich wkład, lista wymagań z karty z oznaczeniem spełnione / niespełnione.
3. U3. Przygotujcie prezentację na 5 minut: cel i odbiorcy, pokaz na żywo (wersja online), jedna ciekawa decyzja techniczna każdej osoby, wynik Lighthouse, czego się nauczyliście. Każdy członek zespołu mówi.
4. U4. Po prezentacjach oceńcie jeden inny projekt według kryteriów z tego zadania (ocena koleżeńska: 2 mocne strony, 1 propozycja zmiany).

W sprawozdaniu umieść

- ☐ opis elementów JS z fragmentami kodu i autorem
- ☐ protokół testów (min. 8 scenariuszy)
- ☐ wyniki W3C oraz Lighthouse przed i po poprawkach
- ☐ adres opublikowanej strony i repozytorium, zestawienie wymagań spełnionych / niespełnionych
- ☐ ocena koleżeńska innego zespołu
- ☐ odpowiedzi na pytania kontrolne (każdy osobno)

Pytania kontrolne

1. Dlaczego kod JavaScript dla jednej podstrony uruchamiamy warunkowo (`if (element)`)? Jaki błąd pojawi się bez tego?
2. Co oznacza, że strona działa bez JavaScript (progressive enhancement), i jak to sprawdziliście?
3. Który problem wskazany przez Lighthouse poprawiliście i dlaczego był ważny dla użytkowników?
4. Które wymaganie z karty projektu nie zostało spełnione i co należałoby zrobić inaczej?

Kryteria oceny

Kryterium	Punkty
J1–J2 — min. 2 działające elementy JS, w tym obsługa danych użytkownika; walidacja z komunikatami i focusem	6
J3–J4 — strona używalna bez JS, warunkowe uruchamianie, brak błędów w konsoli, commity każdej osoby	2
T1 — protokół min. 8 scenariuszy z wynikami i poprawkami	3
T2–T3 — W3C bez błędów, Lighthouse dostępność i SEO ≥ 90 lub opisane poprawki	3
U1–U2 — działająca wersja online, uzupełniona dokumentacja projektu	3
U3–U4 — prezentacja 5 min z udziałem wszystkich, ocena koleżeńska	3
Razem	20

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