

Projekt końcowy cz. 4 — wdrożenie, dokumentacja i prezentacja

Czas: 2 × 45 min

praca w zespołach 2–3-osobowych, prezentacje przed klasą

Cel: Wdrożenie aplikacji zespołowej na serwer, przygotowanie dokumentacji (README, instrukcja użytkownika, opis API i bazy), audyt Lighthouse oraz prezentacja projektu z demonstracją na żywo.

Po zajęciach potrafisz:

- wdraża aplikację PHP + MySQL zgodnie z listą kontrolną i wykonuje test po wdrożeniu
- pisze README, instrukcję użytkownika i opis API dla różnych odbiorców
- mierzy jakość aplikacji (Lighthouse) i poprawia najważniejsze uwagi
- przygotowuje i przeprowadza 5–7-minutową prezentację z demonstracją
- ocenia projekt własny i innych zespołów według kryteriów

Sprzęt i materiały

- komputer z XAMPP, Git i FileZillą; repozytorium projektu (tag v0.3-frontend)
- konto na serwerze szkolnym lub hostingu z PHP 8, MySQL/MariaDB, SFTP i HTTPS
- projektor; folder `r2_31_projekt` (szablon_README.md, kryteria_prezentacji.txt)

Oprogramowanie

- **FileZilla Client** — SFTP; alternatywa WinSCP
filezilla-project.org/download.php?type=client
- **Lighthouse w Chrome/Edge** — F12 → Lighthouse; PageSpeed Insights dla adresu publicznego: <https://pagespeed.web.dev/developer.chrome.com/docs/lighthouse/overview>
- **Markdown w VS Code** — podgląd README: Ctrl+Shift+V
code.visualstudio.com/docs/languages/markdown

Przygotowanie przed zajęciami

- Zamknijcie Issues z kamienia „cz. 3 frontend” lub przenieście je do „Could”. Ustalcie scenariusz demo i podział mówienia.
- Odbierz dane dostępowe do serwera (SFTP, baza) — przypomnij sobie zadanie 29.

Bezpieczeństwo

- Przed wdrożeniem zmień lub usuń konta demonstracyjne; w dokumentacji publicznej nie podawaj haseł produkcyjnych.
- Plik `.env` z danymi produkcyjnymi tworzysz tylko na serwerze — nie trafia do repozytorium ani na slajdy.
- Podczas demo nie pokazuj panelu hostingu ani pliku `.env` na projektorze.

Przebieg zadania

1 Wdrożenie

Postępuj według listy kontrolnej z zadania 29 (docs/WDROZENIE.md w projekcie wzorcowym jest przykładem).

1. Utwórz bazę i konto aplikacji z minimalnymi uprawnieniami, zaimportuj skrypt SQL. Wyślij pliki przez SFTP (bez .env, logów i .git); katalog główny = public/ lub .htaccess z blokadą katalogów prywatnych.
2. Utwórz na serwerze .env z `APP_ENV=production`, sprawdź HTTPS i przekierowanie z http. Zmień hasła kont demonstracyjnych.
3. Wykonaj test po wdrożeniu: obie role, główny CRUD, reguła biznesowa, endpoint JSON (200 i błąd), dostęp do `/.env` i `/src/` (403/404), polskie znaki. Wyniki zapisz w tabeli; błędy zgłoś jako Issues i popraw.
4. Opcjonalnie: przygotuj Dockerfile i docker-compose.yml (PHP + MariaDB) jako sposób uruchomienia lokalnego i opisz go w README.

2 Audyt jakości

1. Uruchom Lighthouse (Mobile, okno InPrivate) dla strony głównej i jednego formularza na wersji wdrożonej. Zapisz 4 wyniki i LCP/CLS.
2. Popraw min. 2 uwagi (np. brak favicon → błąd 404 w konsoli, rozmiar obrazów, kontrast, etykiety, meta description) i zmierz ponownie. Wyniki przed i po wpisz do README.

3 Dokumentacja

Każdy dokument ma innego odbiorcę: README — programista, instrukcja — użytkownik, API — autor klienta (np. aplikacji mobilnej).

1. Uzupełnij `README.md` wg szablonu: opis, adres wdrożenia, funkcje, technologie, uruchomienie lokalne krok po kroku, konta demo (tylko lokalnie), struktura katalogów, bezpieczeństwo, wyniki audytu, zespół.
2. Napisz `docs/INSTRUKCJA.md` dla każdej roli (kroki głównych czynności i tabela najczęstszych komunikatów) oraz `docs/API.md` (metoda, adres, parametry z typami, przykładowa odpowiedź, kody błędów, przykład curl).
3. Dodaj do `docs/` ERD, specyfikację z zadania 31 (zaktualizowaną: co zrealizowano, co przeniesiono do Could/Won't) i zrzuty ekranu. Oznacz wersję tagiem `v1.0` i utwórz Release z opisem zmian.

4 Prezentacja i ocena

Czas: 5–7 min + 2–3 min pytań. Przebieg i arkusz oceny koleżeńskiej — `kryteria_prezentacji.txt`.

1. Przygotuj scenariusz demo (kto, co klika, jakie dane) i maks. 6 slajdów: problem, architektura (ERD, endpoint), bezpieczeństwo i Lighthouse, podział pracy i wnioski. Przygotuj plan B (nagranie ekranu lub wersja na XAMPP).
2. Przeprowadź prezentację: każda osoba mówi, demo na wersji wdrożonej. Odpowiedz na pytania klasy i nauczyciela.
3. Wypełnij arkusze oceny koleżeńskiej dla 2 innych zespołów, a po zajęciach napisz indywidualną refleksję (co zrobisz, czego się nauczyłeś, co poprawiłbyś).

W sprawozdaniu umieść

- ☐ adres wdrożonej aplikacji, tag v1.0 i Release
- ☐ tabela testu po wdrożeniu i wyniki Lighthouse przed/po
- ☐ README, docs/INSTRUKCJA.md, docs/API.md w repozytorium
- ☐ slajdy lub scenariusz demo, arkusze oceny koleżeńskiej

☐ indywidualna refleksja i odpowiedzi na pytania

Pytania kontrolne

1. Czym różnią się odbiorcy README, instrukcji użytkownika i opisu API? Podaj po jednej informacji, która pasuje tylko do jednego dokumentu.
2. Co sprawdziliście po wdrożeniu, czego nie da się sprawdzić na XAMPP?
3. Która reguła bezpieczeństwa była najtrudniejsza do wdrożenia w waszym projekcie i dlaczego?
4. Co zrobilibyście inaczej, zaczynając projekt od nowa?

Kryteria oceny

Kryterium	Punkty
Wdrożenie: działa przez HTTPS, .env na serwerze, zablokowane pliki prywatne, zmienione konta demo	4
Test po wdrożeniu i audyt Lighthouse przed/po z min. 2 poprawkami	3
README wg szablonu, instrukcja użytkownika, opis API, aktualna specyfikacja; tag v1.0 i Release	5
Prezentacja: czas, demo bez błędów, udział każdej osoby, odpowiedzi na pytania	4
Ocena koleżeńska 2 zespołów i indywidualna refleksja	2
Odpowiedzi na pytania kontrolne (indywidualnie)	2
Razem	20

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