

Wdrożenie aplikacji PHP + MySQL

Czas: 2 × 45 min

praca indywidualna

Cel: Przygotowanie aplikacji PHP + MySQL do wdrożenia (konfiguracja poza kodem, katalog public, bezpieczne błędy, konto bazy z minimalnymi uprawnieniami) i przeniesienie jej na serwer przez SFTP z testem po wdrożeniu; Docker Compose jako alternatywa lokalna.

Po zajęciach potrafisz:

- przenosi dane dostępowe do pliku `.env` i zmiennych środowiskowych, a do repozytorium dodaje `.env.example`
- rozdziela katalog publiczny od kodu i konfiguracji oraz blokuje dostęp do plików wrażliwych
- konfiguruje obsługę błędów dla produkcji (`display_errors Off`, dziennik błędów)
- eksportuje i importuje bazę, tworzy konto aplikacji z minimalnymi uprawnieniami
- wysyła pliki przez SFTP, sprawdza HTTPS i wykonuje test po wdrożeniu wg listy kontrolnej

Sprzęt i materiały

- komputer z XAMPP i VS Code
- konto na serwerze szkolnym lub darmowym hostingu z PHP 8, MySQL/MariaDB i SFTP (dane od nauczyciela)
- folder `r2_29_start` z Pliki_do_zadan (aplikacja i plik `usterki.sql`)

Oprogramowanie

- **FileZilla Client** — protokół SFTP; alternatywa: WinSCP <https://winscp.net/filezilla-project.org/download.php?type=client>
- **XAMPP** — wersja deweloperska i eksport bazy apachefriends.org/download.html
- **Docker Desktop (opcjonalnie)** — krok 5 — jeśli jest zainstalowany w pracowni docker.com/products/docker-desktop

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 2.18 środowiska dev/test/prod oraz z zadania 29 roku 1 pracę z FileZillą.
- Odbierz od nauczyciela dane: adres serwera, login SFTP, nazwę bazy i hasło.

Bezpieczeństwo

- Hasła do serwera i bazy trzymaj tylko w pliku `.env` na serwerze i w menedżerze haseł — nigdy w repozytorium ani w sprawozdaniu (pisz „***”).
- Używaj SFTP (szyfrowane), nie zwykłego FTP. Nie zapisuj hasła w FileZilli na komputerze szkolnym.
- Plik z `phpinfo()` usuń zaraz po sprawdzeniu ustawień — ujawnia konfigurację serwera.

Przebieg zadania

1 Uruchomienie lokalne i przegląd problemów

1. Zaimportuj `usterki.sql` w phpMyAdmin, skopiuj aplikację do `htdocs\usterki` i sprawdź działanie (lista 5 zgłoszeń, dodanie nowego).
2. Zmień w `db.php` hasło na błędne i zrób zrzut komunikatu. Wypisz, jakie informacje o serwerze zdradza ten komunikat, oraz 4 inne problemy aplikacji przed wdrożeniem (np. konto root, hasło w kodzie, dostęp do pliku `.sql` z przeglądarki).

2 Konfiguracja przez zmienne środowiskowe

Docelowa struktura: `public/` (`index.php`, `style.css` — jedyny katalog widoczny z WWW), `src/` (`config.php`, `db.php`), `sql/`, `logs/`, plik `.env`, `.env.example`, `.gitignore`.

```
# .env.example
APP_ENV=development
DB_HOST=localhost
DB_PORT=3306
DB_NAME=usterki
DB_USER=usterki_app
DB_PASS=
```

1. Przebuduj strukturę katalogów jak w opisie i popraw ścieżkę w `public/index.php` (`require __DIR__ . '/../src/db.php'`).
2. Napisz `src/config.php`: funkcja wczytująca plik `.env` (wiersze KLUCZ=wartość, pomijanie komentarzy #) przez `putenv`, funkcja `env($klucz, $domyslna)` oparta na `getenv`. Zmienne ustawione przez serwer mają pierwszeństwo przed plikiem.
3. W `config.php` ustaw obsługę błędów zależnie od `APP_ENV`: w production `display_errors 0`, zawsze `log_errors 1` i `error_log` do `logs/php-error.log`.
4. W `src/db.php` zbuduj DSN z wartości env, a wyjątek połączenia złap: zapisz szczegóły `error_log`, zwróć kod 503 i ogólny komunikat. Sprawdź oba tryby (development/production) z błędnym hasłem.
5. Utwórz konto bazy `usterki_app` z uprawnieniami tylko SELECT, INSERT, UPDATE, DELETE do bazy `usterki`. Dodaj `.gitignore` z `.env` i `logs/*.log`.

3 Wdrożenie na serwer

Jeśli katalogu głównego domeny nie da się ustawić na `public/`, dodaj w katalogu aplikacji `.htaccess` z przekierowaniem do `public/` i blokadą `src/`, `sql/`, `logs/` i `.env`.

1. Wyeksportuj bazę z lokalnego phpMyAdmin (Eksportuj → SQL) i zaimportuj ją w phpMyAdmin serwera. Na serwerze konto bazy tworzy zwykle panel hostingu — zapisz, jakie uprawnienia ma.
2. Połącz się FileZillą przez SFTP (port 22) i wyślij pliki bez `.env`. Na serwerze utwórz `.env` z danymi produkcyjnymi i `APP_ENV=production`.
3. Sprawdź na chwilę plikiem `info.php` z `phpinfo()` wartości `display_errors`, `log_errors` i wersję PHP, po czym usuń plik.

4 Test po wdrożeniu

1. Wykonaj test wg listy kontrolnej: strona przez https (kłódka, przekierowanie z http), lista z bazy, dodanie zgłoszenia, polskie znaki, odświeżenie po dodaniu (PRG).
2. Sprawdź, że adresy `/.env`, `/src/db.php`, `/sql/usterki.sql` zwracają 403 lub 404. Wynik każdego punktu zapisz w tabeli (OK / błąd i poprawka).

5 Docker Compose — alternatywa lokalna (jeśli dostępny)

Bez Dockera opisz tylko rolę każdej sekcji pliku docker-compose.yml.

```
services:
  app:
    build: . # Dockerfile: php:8.3-apache + pdo_mysql
    ports: ["8080:80"]
    environment: { DB_HOST: db, DB_NAME: usterki, DB_USER: usterki_app, DB_PASS: ... }
  db:
    image: mariadb:10.11
    environment: { MARIADB_DATABASE: usterki, MARIADB_USER: usterki_app, ... }
    volumes:
      - dane_db:/var/lib/mysql
      - ./sql/usterki.sql:/docker-entrypoint-initdb.d/01-usterki.sql:ro
```

1. Napisz Dockerfile (obraz php:8.3-apache, `docker-php-ext-install pdo_mysql`, katalog główny Apache = public) i docker-compose.yml wg ramki. Uruchom `docker compose up -d` i otwórz `http://localhost:8080`.
2. Wyjaśnij, dlaczego DB_HOST ma wartość `db`, a nie `localhost`, i co daje wolumen `dane_db`.

W sprawozdaniu umieść

- ☐ lista problemów wersji startowej z uzasadnieniem
- ☐ kod `config.php`, `db.php`, `.env.example`, `.gitignore` (bez haseł)
- ☐ adres wdrożonej aplikacji, zrzut z kłódką HTTPS i tabela testu po wdrożeniu
- ☐ zrzuty: komunikat 503 w produkcji i wpis w dzienniku błędów
- ☐ docker-compose.yml lub jego opis, odpowiedzi na pytania

Pytania kontrolne

1. Dlaczego dane dostępne do bazy nie powinny być zapisane w kodzie w repozytorium?
2. Czym różni się konfiguracja błędów PHP w środowisku deweloperskim i produkcyjnym?
3. Po co aplikacji osobne konto bazy z ograniczonymi uprawnieniami?
4. Dlaczego katalogiem głównym serwera powinien być public/, a nie cały projekt?

Kryteria oceny

Kryterium	Punkty
Analiza wersji startowej: zrzut błędu i 5 problemów	2
Struktura public/src, config.php z .env i getenv, .env.example, .gitignore	4
Obsługa błędów: display_errors zależne od APP_ENV, log do pliku, 503 przy braku bazy	3
Konto bazy z minimalnymi uprawnieniami, eksport i import bazy	2
Wdrożenie przez SFTP, .env na serwerze, HTTPS, test wg listy kontrolnej, blokada plików wrażliwych	4
Docker Compose (uruchomienie lub poprawny opis)	2
Odpowiedzi na pytania kontrolne	3
Razem	20

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