

Kurs PHP + Bootstrap: Generator ścieżek do plików

Cel kursu

Stworzenie prostej strony internetowej z formularzem, która będzie:

- Łączyć stringi w ścieżki do plików
- Wyświetlać aktualną datę po kliknięciu przycisku
- Generować proste skrypty hello world
- Walidować dane wejściowe
- Wykorzystywać Bootstrap do stylowania

Wymagania wstępne

- Ubuntu 20.04
- PHP zainstalowane (`sudo apt install php`)
- Przeglądarka internetowa
- Edytor kodu (VS Code, Vim, nano)

Środowisko pracy

PHP Built-in Server - najprostsza opcja na start

- Uruchomienie: `php -S localhost:8000`
- Brak dodatkowej konfiguracji
- Idealne do nauki i prototypowania

KROK 1: Przygotowanie środowiska (15 min)

Cele kroku:

- Sprawdzenie czy PHP działa
- Utworzenie struktury projektu
- Uruchomienie serwera developmentu

Zadania:

1. Sprawdzenie wersji PHP

- Komenda w terminalu do sprawdzenia instalacji
- Minimalna wymagana wersja: PHP 7.4+

2. Utworzenie struktury katalogów

```
php-bootstrap-kurs/
├── index.php    (główny plik)
├── test.php     (test środowiska)
├── assets/
│   ├── css/    (własne style)
│   └── js/     (własny JavaScript)
└── includes/   (pliki pomocnicze)
```

3. Test środowiska

- Utworzenie pliku testowego z phpinfo()
- Uruchomienie serwera
- Sprawdzenie w przeglądarce

Wskazówki:

- Używaj terminalu jak w Pythonie - wszystko widać w konsoli
 - Serwer zatrzymuje się Ctrl+C
 - Zmiany w plikach widać od razu po odświeżeniu przeglądarki
-

KROK 2: Podstawowa struktura HTML (20 min)

Cele kroku:

- Utworzenie szablonu HTML5
- Podłączenie Bootstrap CDN
- Przygotowanie struktury strony

Zadania:

1. Szablon HTML5

- DOCTYPE, html, head, body
- Meta tags dla responsive design
- Tytuł strony

2. Integracja Bootstrap

- Podłączenie CSS z CDN
- Podłączenie JavaScript bundle
- Test czy Bootstrap działa

3. Podstawowy layout

- Container Bootstrap
- Nagłówek strony (h1)
- Opis funkcjonalności
- Placeholder na formularz

Wskazówki:

- Używaj Bootstrap 5 (najnowsza wersja)
 - CDN to szybsze rozwiązanie niż lokalne pliki
 - Sprawdź czy Bootstrap działa dodając button z klasą btn-primary
-

KROK 3: Tworzenie formularza Bootstrap (25 min)

Cele kroku:

- Zbudowanie formularza z 4 polami
- Zastosowanie klas Bootstrap
- Dodanie przycisków akcji

Zadania:

1. Struktura formularza

- Form tag z method="POST"
- Form-group dla każdego pola
- Labels z opisami

2. Pola input

- Katalog bazowy (np. /home/user)
- Podkatalog (np. projects/scripts)
- Nazwa pliku (np. backup_script)
- Rozszerzenie (np. .sh)

3. Przyciski

- "Generuj ścieżkę" - główna akcja
- "Wyświetl datę" - dodatkowa funkcja
- Stylowanie Bootstrap

4. Placeholders i odpowiedzi

- Przykładowe wartości w placeholderach
- Tooltips z wyjaśnieniami
- Ikony Font Awesome (opcjonalnie)

Wskazówki:

- Używaj form-control dla inputów
 - btn-primary dla głównego przycisku
 - Zadbaj o accessibility (labels, aria-labels)
-

KROK 4: Podstawowa logika PHP (30 min)

Cele kroku:

- Obsługa wystania formularza
- Walidacja danych wejściowych
- Łączenie stringów w ścieżkę

Zadania:

1. Detekcja wystania formularza

- Sprawdzanie \$_POST
- Rozróżnienie między akcjami (generuj vs data)

2. Pobieranie danych

- Odczyt z \$_POST
- Podstawowe oczyszczanie danych

3. Walidacja

- Sprawdzanie czy pola nie są puste
- Weryfikacja dozwolonych znaków
- Lista błędów walidacji

4. Generowanie ścieżki

- Łączenie stringów
- Obsługa różnych separatorów (/ vs)
- Normalizacja ścieżki

Wskazówki:

- Zaczynij od prostego var_dump(\$_POST)
 - Waliduj każde pole osobno
 - Przechowuj błędy w tablicy
-

KROK 5: Obsługa błędów i komunikatów (20 min)

Cele kroku:

- System komunikatów z Bootstrap alerts
- Wyświetlanie błędów walidacji
- Success messages

Zadania:

1. Bootstrap alerts

- Alert-danger dla błędów
- Alert-success dla sukcesu

- Alert-info dla informacji

2. Walidacja szczegółowa

- Konkretnie komunikaty błędów
- Wskazanie problematycznego pola
- Sugestie naprawienia

3. Wyświetlanie wyników

- Formatowanie wygenerowanej ścieżki
- Podgląd w czytelnej formie
- Opcje kopiowania (przygotowanie)

Wskazówki:

- Używaj is-invalid klasy Bootstrap dla błędnych pól
 - Grupuj komunikaty według typu
 - Dodaj możliwość zamykania alertów
-

KROK 6: Ulepszenie UX (15 min)

Cele kroku:

- Zachowanie danych po błędzie
- Stylowanie wyników
- Responsive design

Zadania:

1. Persistence danych

- Zachowanie wartości w polach input
- Zapobieganie utracie danych po błędzie

2. Stylowanie wyników

- Card Bootstrap dla wyniku
- Formatowanie ścieżki (monospace font)
- Ikony i wizualne elementy

3. Responsive design

- Test na różnych rozdzielczościach
- Optymalizacja dla mobile
- Czytelność na małych ekranach

Wskazówki:

- Używaj value="<?php echo htmlspecialchars(\$dane); ?>"

- Card-header, card-body dla struktury
 - Testuj w trybie mobile przeglądarki
-

KROK 7: Dodanie przycisku "Wyświetl datę" (15 min)

Cele kroku:

- Druga funkcjonalność w tym samym formularzu
- Obsługa różnych akcji
- Formatowanie daty w PHP

Zadania:

1. Rozróżnienie akcji

- Różne wartości name dla przycisków
- Logika if/else dla akcji

2. Funkcjonalność daty

- Aktualna data i czas
- Różne formaty (ISO, polski, timestamp)
- Ładne wyświetlanie

3. Integracja z UI

- Osobna sekcja na datę
- Możliwość wyboru formatu
- Historia wyświetlanych dat

Wskazówki:

- Używaj date() i DateTime w PHP
 - Sprawdzaj isset(\$_POST['action_name'])
 - Formatuj datę według polskich standardów
-

KROK 8: Podstawowy JavaScript (20 min)

Cele kroku:

- Walidacja client-side
- Dynamiczne interakcje
- Podstawowe animacje

Zadania:

1. Walidacja formularza

- Sprawdzanie pustych pól przed wysłaniem

- Komunikaty błędów po stronie klienta
- Disable przycisku przy nieprawidłowych danych

2. Interaktywność

- Auto-focus na następne pole
- Licznik znaków w polach
- Preview ścieżki podczas pisania

3. Animacje Bootstrap

- Fade in/out dla komunikatów
- Smooth scroll do wyników
- Loading spinner podczas przetwarzania

Wskazówki:

- Zaczynaj od vanilla JavaScript, potem jQuery
 - Używaj Bootstrap JS komponentów
 - Graceful degradation - strona musi działać bez JS
-

KROK 9: Testowanie i debugging (15 min)

Cele kroku:

- Kompleksowe testowanie funkcjonalności
- Wykrywanie i naprawa błędów
- Optymalizacja wydajności

Zadania:

1. Test cases

- Różne kombinacje danych wejściowych
- Graniczne przypadki (puste pola, długie stringi)
- Błędne dane (znaki specjalne, spacje)

2. Cross-browser testing

- Test w Chrome, Firefox, Safari
- Sprawdzenie na mobile
- Różne rozdzielczości ekranu

3. Performance check

- Szybkość ładowania strony
- Responsywność formularza
- Optymalizacja Bootstrap (tylko potrzebne komponenty)

Wskazówki:

- Używaj Developer Tools w przeglądarce
 - Testuj systematycznie każdą funkcję
 - Dokumentuj znalezione błędy
-

KROK 10: Dokumentacja i podsumowanie (10 min)

Cele kroku:

- Dokumentacja kodu
- Podsumowanie zdobytej wiedzy
- Plan dalszego rozwoju

Zadania:

1. Komentarze w kodzie

- Opis funkcji PHP
- Komentarze HTML dla struktury
- Dokumentacja JavaScript

2. README projektu

- Instrukcja instalacji i uruchomienia
- Lista funkcjonalności
- Screenshots działania

3. Plan rozszerzenia

- Pomysły na nowe funkcje
- Technologie do nauki
- Przygotowanie do kursu z Docker Compose

Wskazówki:

- Pisz komentarze w języku polskim
 - Używaj DocBlock dla funkcji PHP
 - Przygotuj listę kolejnych kroków nauki
-

Podsumowanie kursu

Czego się nauczysz:

- Podstawy PHP (zmienne, tablice, walidacja)
- Bootstrap components i grid system
- Obsługa formularzy HTML

- Client-side JavaScript
- Debugging i testowanie
- Struktura projektu webowego

Czas realizacji: ~3 godziny

Następne kroki:

- Refaktoryzacja z Docker Compose
- Dodanie bazy danych (MySQL)
- Zaawansowane funkcjonalności JavaScript
- Bezpieczeństwo aplikacji webowych

Przydatne linki:

- PHP Documentation: [php.net](https://www.php.net)
- Bootstrap Documentation: getbootstrap.com
- MDN Web Docs: developer.mozilla.org