|  |  |  |
| --- | --- | --- |
| **AI2** | Bancewicz Aleksandra  Album ba34753  Grupa 1 | **Wersja 1** |
|  |
| **LAB B** |

Inicjalizacja projektu Symfony

# Spis treści

[Spis treści 1](#_Toc179535620)

[Cel zajęć 1](#_Toc179535621)

[Rozpoczęcie 1](#_Toc179535622)

[Uwaga 1](#_Toc179535623)

[Założenia projektu 2](#_Toc179535624)

[Opracowanie projektu bazy danych 2](#_Toc179535625)

[Repozytorium GIT 3](#_Toc179535626)

[Instalacja Symfony 3](#_Toc179535627)

[Omówienie pakietów 5](#_Toc179535628)

[Commit projektu do GIT 5](#_Toc179535629)

[Podsumowanie 6](#_Toc179535630)

# Cel zajęć

Celem głównym zajęć jest zdobycie umiejętności inicjalizacji frameworka Symfony oraz inicjalizacji projektów w repozytorium GIT. Ponadto utrwalone zostaną umiejętności projektowania struktury bazy danych z wykorzystaniem modeli ERD, jak również zbierania wymagań odnośnie nowych projektów.

# Rozpoczęcie

Rozpoczęcie zajęć. Powtórzenie zasad działania repozytorium GIT: clone, pull, push, commit, fetch, merge, init. Powtórzenie informacji o aktualnych wersjach Symfony. Powtórzenie sposobów instalacji Symfony (skrypt symfony lub composer). Powtórzenie wersji frameworka – skeleton i website-skeleton.

Wejściówka?

# Uwaga

Ten dokument aktywnie wykorzystuje niestandardowe właściwości. Podobnie jak w LAB A wejdź do Plik -> Informacje -> Właściwości -> Właściwości zaawansowane -> Niestandardowe i zaktualizuj pola. Następnie uruchom ten dokument ponownie lub Ctrl+A -> F9.

# Założenia projektu

Prowadzący omówi ogólne założenia projektu przyrostowo budowanego przez kolejne laboratoria do końca modułu Symfony. Zadawaj pytania, aby rozwiać wszelkie wątpliwości.

Tutaj możesz umieścić swoje notatki:

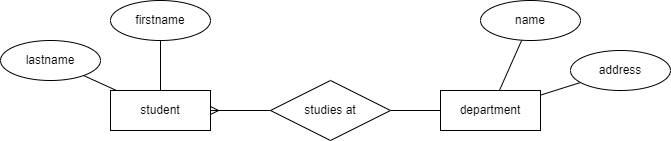
…notatki…  
**„PRZYROSTOWO BUDOWANEGO PRZEZ KOLEJNE LABORATORIA” – WAŻNE!!!** (zrób porządnie 😊)

# Opracowanie projektu bazy danych

Na tym etapie laboratorium założenia projektu powinny być już znane. Częściowo w grupie, a częściowo samodzielnie, wykorzystaj platformę draw.io do opracowania diagramu ERD dla projektu pogodynki. Kilka wytycznych:

* Osobna encja na miejscowość oraz na poszczególne wpisy o danych meteorologicznych.
* Temperatura musi być przechowywana w stopniach Celsjusza.
* Rozważyć w jaki sposób przechowywane będą dane – flat czy EAV.

Zupełnie niezwiązany przykład diagramu ERD:



Umieść zrzut ekranu swojego diagramu ERD:

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

Omów najważniejsze założenia swojego diagramu ERD:

1. **Integracja danych**: Diagram zakłada, że każdy wpis o danych meteorologicznych jest przypisany do konkretnej miejscowości. Dzięki temu możemy łatwo wyszukiwać i analizować dane dla konkretnej lokalizacji.
2. **Przechowywanie danych**: Temperaturę przechowujemy w stopniach Celsjusza, co jest standardem w Polsce oraz wielu innych krajach europejskich. Dzięki temu możemy łatwiej porównać dane z różnych miejscowości.
3. **Skalowalność**: Struktura tego diagramu pozwala na łatwe dodawanie nowych wpisów meteorologicznych i rozbudowę projektu, np. o dodatkowe parametry pogodowe, np. wilgotność, nasłonecznienie itp. W każdej chwili możemy rozbudować tabelę **Dane meteorologiczne** o kolejne kolumny, bez wpływu na integralność danych.

**Flat Model**: W tym przypadku przechowywanie wszystkich atrybutów w jednej tabeli jest efektywne przy mniejszych zbiorach danych. Gdyby dane miały być bardziej rozbudowane, rozważenie modelu EAV (Entity-Attribute-Value) mogłoby pozwolić na większą elastyczność w przyszłości, ale obecnie struktura jest prosta i przejrzysta.

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Repozytorium GIT

Załóż repozytorium GIT pod swój projekt. Skorzystaj z Github, Gitlab, Bitbucket itp. Repozytorium musi mieć dostęp publiczny. Główny branch musi nazywać się main.

**Nie inicjalizuj repozytorium żadnym plikiem README bądź licencją.**

Podaj link do założonego repozytorium:  
  
**https://git.wi.zut.edu.pl/ba34753/AI2.git**

…link, np. https://github.com/ideaspot-pl/ai2-pogodynka-202310…

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Instalacja Symfony

Zweryfikuj dostępne wersje Symfony na stronie <https://symfony.com/releases>. Wejdź terminalem do katalogu I:\AI2-lab. Upewnij się, że w tym katalogu znajduje się composer.phar (jeśli nie ma, pobierz). Zainstaluj Symfony w trybie website-skeleton z wykorzystaniem composera do katalogu I:\AI2-lab\pogodynka.

php composer.phar create-project symfony/skeleton:"6.3.x" pogodynka

cd pogodynka

php ..\composer.phar require webapp

Na pytanie o instalację receptury kontenerów Docker odpowiedz przecząco.

Dokumentacja tego kroku dostępna tutaj: <https://symfony.com/doc/current/setup.html>.

Po zakończeniu instalacji, wejdź do katalogu public projektu i uruchom wbudowany serwer PHP:

cd \public

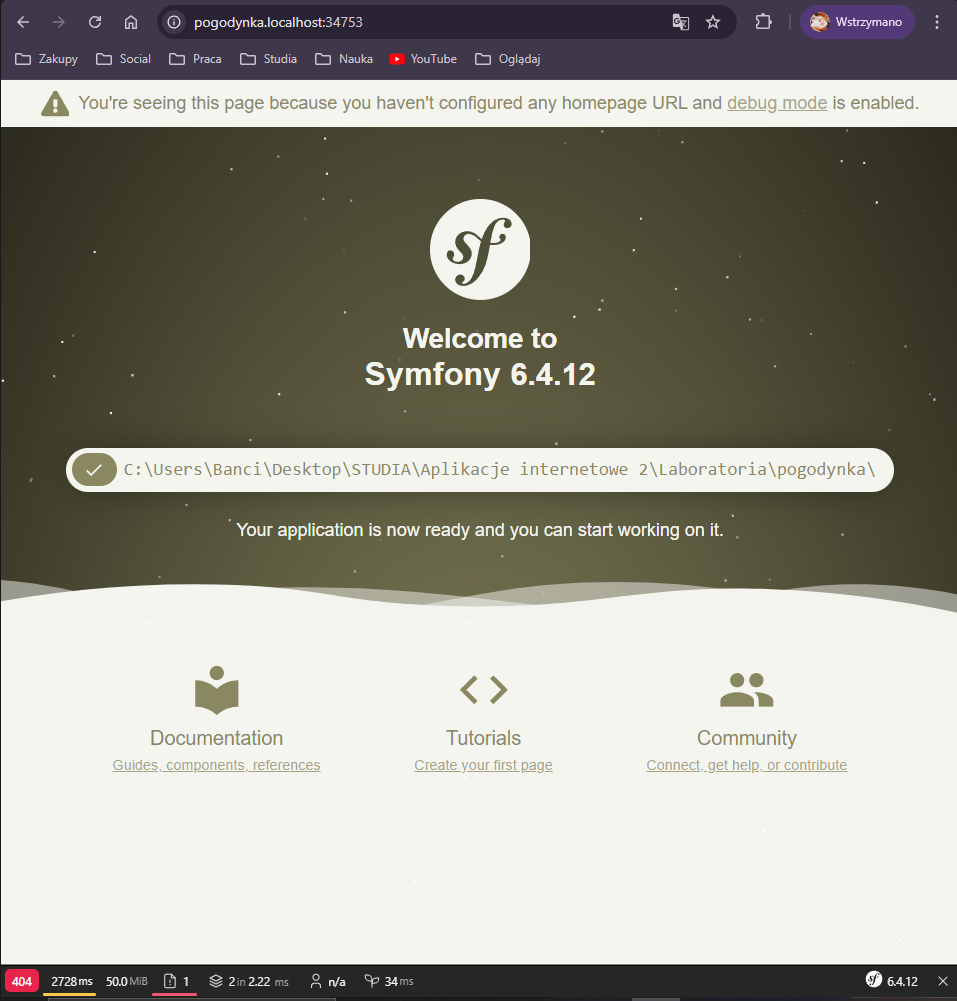
php -S localhost:ba34753

Powyżej wyróżniony został numer indeksu. W przypadku powodzenia efekt powinien być zbliżony do poniższego:



Otwórz ulubioną przeglądarkę internetową i wejdź pod adres http://pogodynka.localhost:ba34753.

Zamień poniższy zrzut ekranu na Twój odpowiednik. Upewnij się, że widoczne i poprawne są wszystkie zaznaczone fragmenty – adres URL i port, wersja Symfony 6.x, ścieżka na dysku:



|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Omówienie pakietów

Otwórz plik composer.json. Zweryfikuj jakie pakiety zainstalowane są w require i require-dev. Poszukaj informacji o nich w dokumentacji Symfony. Dobry punkt startowy: <http://ispot.link/require-dev>.

W maksymalnie 150 słowach opisz najciekawsze pakiety z require i require-dev:

Opis najciekawszych pakietów:

symfony/framework-bundle (require) – kluczowy pakiet integrujący komponenty Symfony, odpowiedzialny za konfigurację frameworka i podstawowe funkcjonalności, takie jak routing czy kontrolery.

symfony/console (require) – umożliwia tworzenie aplikacji CLI. Jest przydatny do zarządzania aplikacją za pomocą konsoli, np. poprzez komendy Symfony.

symfony/dotenv (require) – pozwala na łatwe zarządzanie zmiennymi środowiskowymi z plików .env, co upraszcza konfigurację aplikacji.

phpunit/phpunit (require-dev) – popularne narzędzie do pisania i uruchamiania testów jednostkowych w PHP, wspierane przez Symfony.

symfony/test-pack (require-dev) – zestaw narzędzi do testowania aplikacji, zawierający PHPUnit oraz inne pakiety wspierające pisanie testów.

W maksymalnie 2 zdaniach opisz czym różni się podpięcie pakietów do require a require-dev:

Pakiety w sekcji require są niezbędne do uruchomienia aplikacji, natomiast require-dev zawiera pakiety potrzebne tylko do rozwoju i testowania aplikacji, np. narzędzia do testów.

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Commit projektu do GIT

Otwórz projekt w PhpStorm lub VS Code. Edytuj plik .gitignore i dodaj do listy ignorowanych plików wszystkie pliki/katalogi konfiguracyjne Twojego IDE.

W przeglądarce wejdź na stronę założonego wcześniej repozytorium. Znajdź instrukcję wysyłania do repozytorium nowego projektu.



Postępuj zgodnie z instrukcjami aby wysłać swój projekt do repozytorium. Upewnij się, że projekt jest dodany w taki sposób, żeby jego cała zawartość znajdowała się w głównym katalogu repozytorium, a nie podkatalogu ~~pogodynka~~.

Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie lab-b na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-b w swoim repozytorium:

https://git.wi.zut.edu.pl/ba34753/AI2/src/branch/lab-b

Umieść zrzut ekranu strony brancha lab-b w swoim repozytorium:

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie  
Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| Punkty: | 0 | 1 |

# Podsumowanie

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Podczas tego laboratorium nauczyłam się, jak prawidłowo zainstalować i skonfigurować środowisko Symfony oraz Composer, w tym rozwiązywać problemy związane z brakującymi rozszerzeniami PHP. Poznałam również, jak tworzyć i zarządzać plikiem .gitignore, dodając wpisy specyficzne dla IDE, aby wykluczyć niepotrzebne pliki z repozytorium Git. Zyskałam praktyczną wiedzę na temat zarządzania repozytorium Git, w tym jak inicjować repozytorium, dodawać pliki, tworzyć commit, synchronizować lokalne i zdalne zmiany

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.