

INICJALIZACJA PROJEKTU SYMFONY

SPIS TREŚCI

Spis treści	1
Cel zajęć.....	1
Rozpoczęcie	1
Uwaga	1
Założenia projektu.....	2
Opracowanie projektu bazy danych.....	2
Repozytorium GIT.....	3
Instalacja Symfony.....	3
Omówienie pakietów	4
Commit projektu do GIT.....	5
Podsumowanie.....	6

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 rozwiązać wszelkie wątpliwości.

Tutaj możesz umieścić swoje notatki:

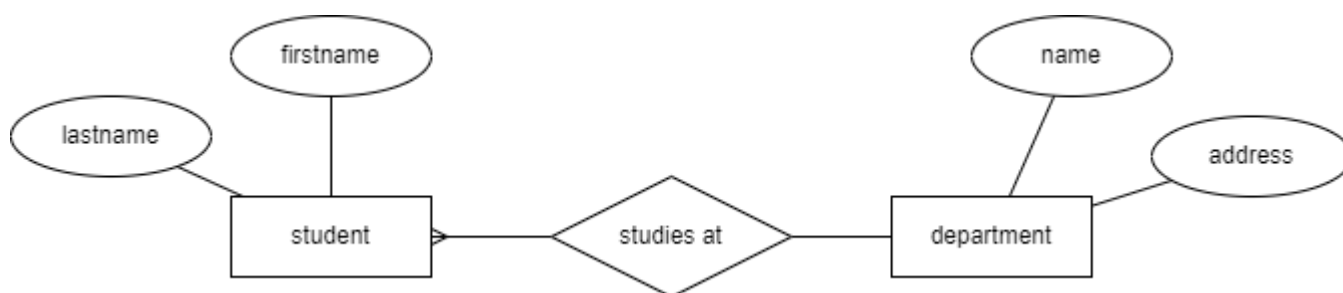
Notatki uuuu

OPRACOWANIE PROJEKTU BAZY DANYCH

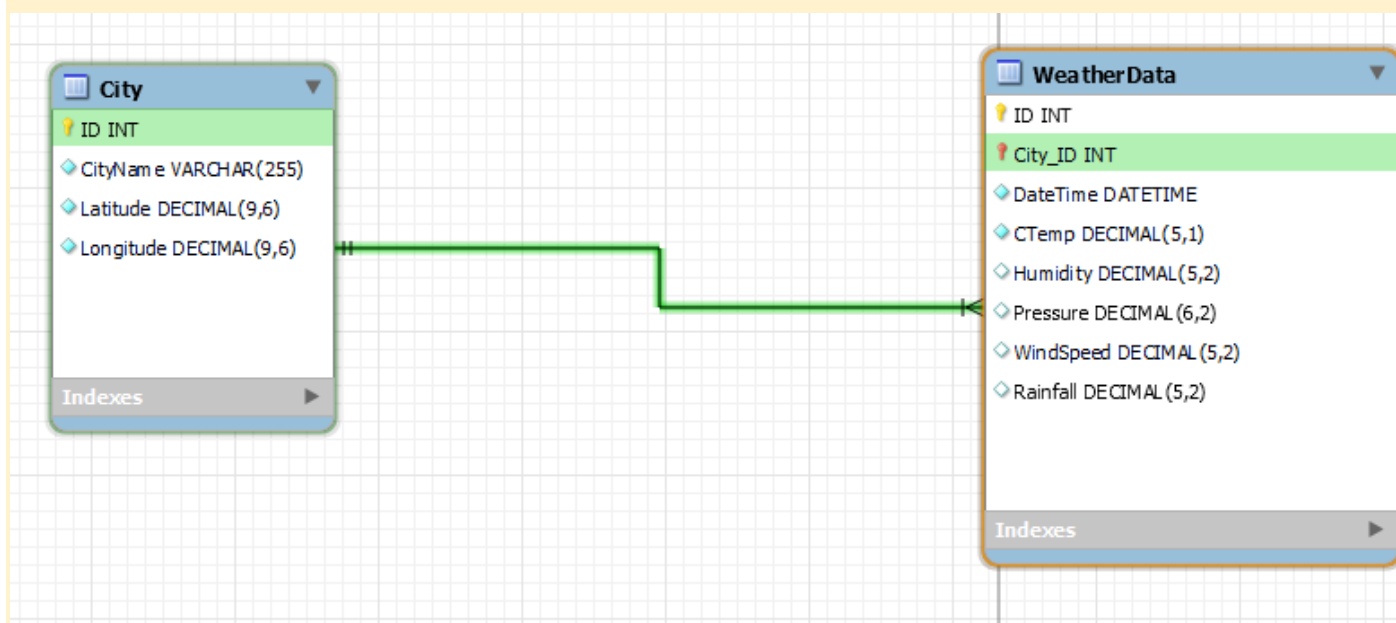
Na tym etapie laboratorium założenia projektu powinny być już znane. Częściowo w grupie, a częściowo samodzielnie, wykorzystaj platformę draw.io do opracowania diagramu ERD dla projektu pogodyнки. Kilka wytycznych:

- 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:



Miejscowość (City) przechowuje informacje o miejscowościach, takie jak nazwa, szerokość geograficzna i długość geograficzna.

Dane meteorologiczne (Weather Data) zawierają informacje o pomiarach meteorologicznych, takie jak temperatura, wilgotność, ciśnienie atmosferyczne itp., powiązane z konkretną miejscowością i datą-czasem pomiaru.

Pomiędzy encją Miejscowość a Danymi meteorologicznymi istnieje relacja (wiele danych do jednego miasta), która umożliwia powiązanie danych meteorologicznych z określonym miejscem.

Temperatura jest przechowywana jako DECIMAL z precyzją (np. DECIMAL(5, 1)), co umożliwia dokładne przechowywanie temperatury w stopniach Celsjusza. Inne parametry meteorologiczne również są przechowywane jako DECIMAL z odpowiednią precyzją, dostosowaną do dokładności pomiarów.

Ten diagram erd został wykonany w MySQL Workbench co pozwala od razu wygenerować polecenia sql w razie potrzeby.

Użyłem sposobu flat, bo łatwiej przechowywać dane.

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

REPOZYTORIUM GIT

Załącz 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://github.com/Michaldariusznowakowski/ai2>

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:

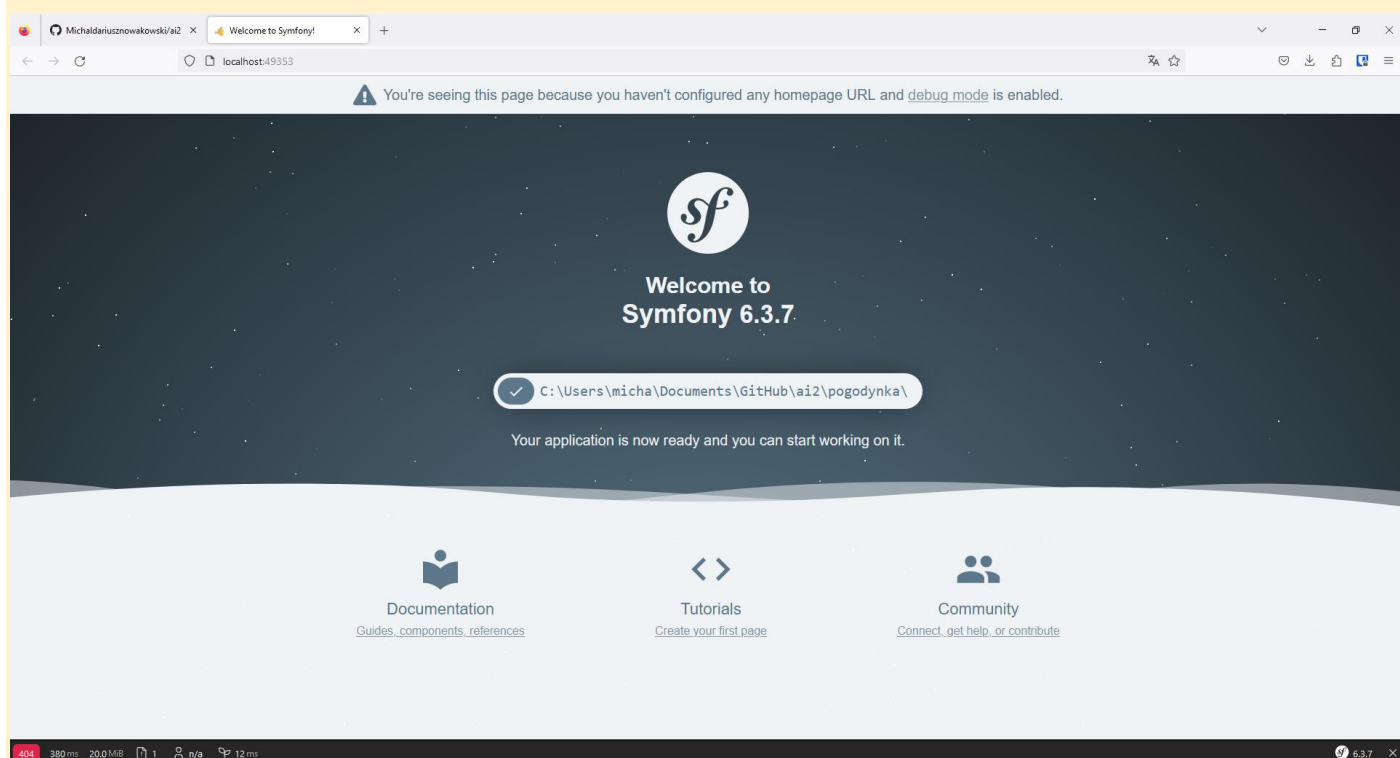
```
cd I:\AI2-lab\pogodynka\public  
php -S localhost:00000
```

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

```
public>php -S localhost:13346  
[Fri Sep 22 12:53:59 2023] PHP 8.2.10 Development Server (http://localhost:13346) started
```

Otwórz ulubioną przeglądarkę internetową i wejdź pod adres <http://pogodynka.localhost:00000>.

Zamień poniższy zrzut ekranu na Twój odpowiednik. Upewnij się, że widoczne i poprawne są wszystkie zaznaczone fragmenty:



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>.

`require`

Interesujące pakiety to Symfony Framework w wersji 6.3, Doctrine do zarządzania bazą danych, oraz narzędzia do obsługi formatu YAML, Twig dla szablonów i PHPStan do statycznej analizy kodu.

Require-dev

PHPUnit do testów jednostkowych, Symfony Maker Bundle do generowania kodu, oraz Symfony Web Profiler Bundle, który pomaga w analizie wydajności i debugowaniu aplikacji. Te pakiety nie są niezbędne w produkcji, dlatego są umieszczone w require-dev.

Podpięcie pakietów do "require" jest konieczne do działania głównego projektu i jest to lista zależności produkcyjnych. Natomiast pakietów podpiętych do "require-dev" używa się wyłącznie w środowisku deweloperskim i testowym do celów rozwoju, debugowania i testowania aplikacji, co pozwala na oddzielenie narzędzi deweloperskich od produkcji.

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.

...or create a new repository on the command line

```
echo "# ai2-pogodynka-202310" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/ideaspot-pl/ai2-pogodynka-202310.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/ideaspot-pl/ai2-pogodynka-202310.git
git branch -M master
git push -u origin master
```

Postępuj zgodnie z instrukcjami aby wysłać swój projekt do repozytorium. Upewnij się, że projekt jest dodany w taki sposób, żeby composer.json znajdował się w głównym katalogu repozytorium, a nie podkatalogu pogodynka.

Przykładowo:

```
git init
git branch -M main
git add *
git commit -m "initial symfony"
git remote add origin https://github.com/ideaspot-pl/ai2-pogodynka-202310.git
git push -u origin main
```

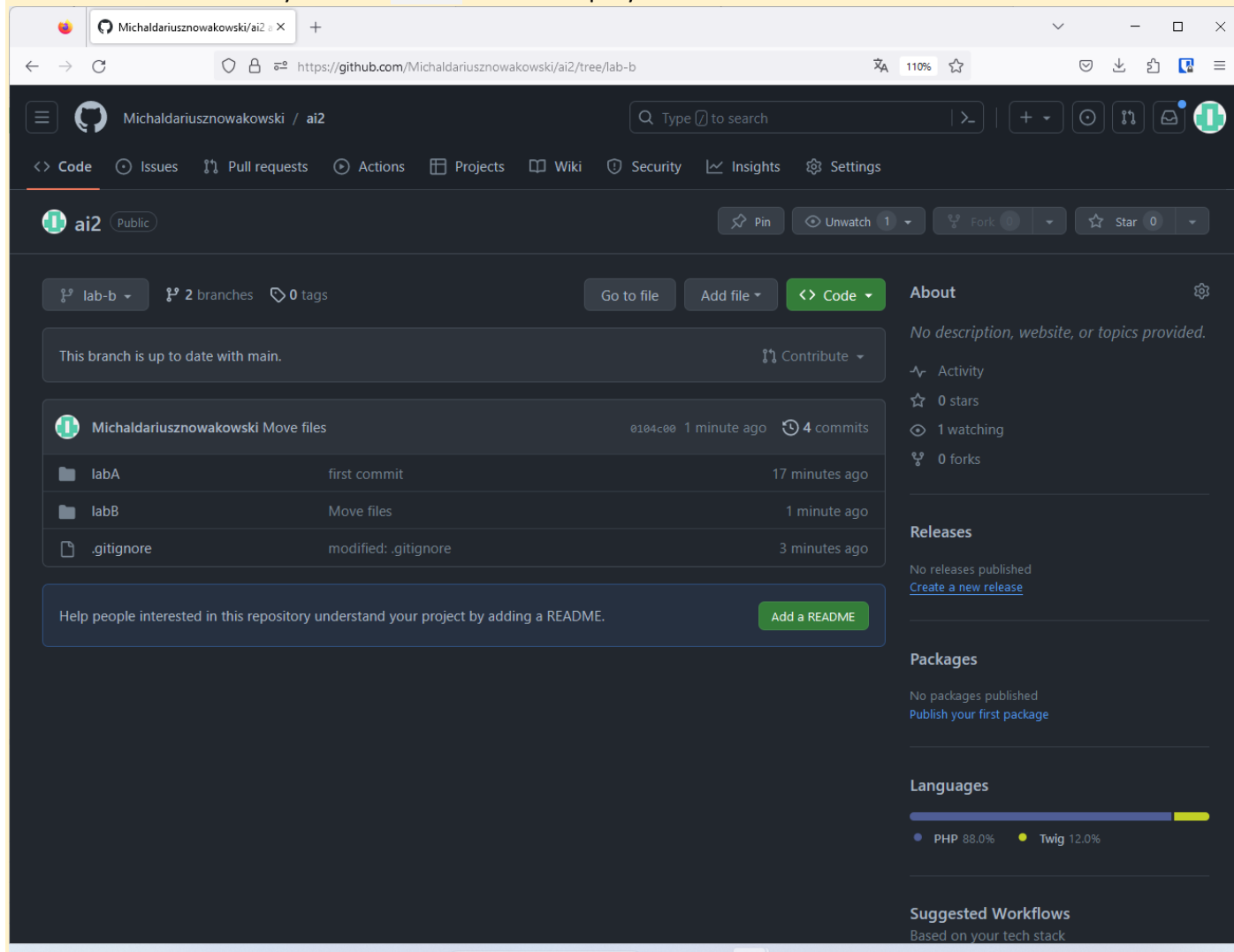
Upewnij się, czy wszystko dobrze się wysłało. Jeśli tak, to z poziomu przeglądarki utwórz branch o nazwie lab-b na podstawie głównej gałęzi kodu.

Podaj link do brancha lab-b w swoim repozytorium:

...link, np. <https://github.com/ideaspot-pl/ai2-pogodynka-202310/tree/lab-b...>

<https://github.com/Michaldariusznawakowski/ai2/tree/lab-b>

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



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

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

W laboratorium miałem okazję nauczyć się różnicy między require a require-dev w narzędziu Composer. require służy do definiowania zależności niezbędnych do działania aplikacji na produkcji, podczas gdy require-dev dotyczy zależności potrzebnych tylko podczas procesu rozwoju i testowania.

Ponadto, stworzyłem własne repozytorium w systemie kontroli wersji Git i wykonałem pierwszy commit, co pozwoliło mi zapisywać zmiany w projekcie. Dodatkowo, utworzyłem nową gałąź (branch) w repozytorium, co umożliwiło pracę

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.