|  |  |  |
| --- | --- | --- |
| **AI2** | Skorupka Hubert  Album 49497  Grupa 1 | **Wersja 1** |
|  |
| **LAB B** |

Inicjalizacja projektu Symfony

# Spis treści

[Spis treści 1](#_Toc146283463)

[Cel zajęć 1](#_Toc146283464)

[Rozpoczęcie 1](#_Toc146283465)

[Uwaga 1](#_Toc146283466)

[Założenia projektu 2](#_Toc146283467)

[Opracowanie projektu bazy danych 2](#_Toc146283468)

[Repozytorium GIT 3](#_Toc146283469)

[Instalacja Symfony 3](#_Toc146283470)

[Omówienie pakietów 4](#_Toc146283471)

[Commit projektu do GIT 4](#_Toc146283472)

[Podsumowanie 6](#_Toc146283473)

# 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:

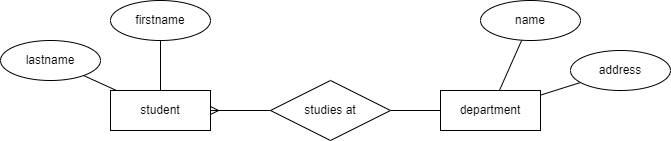
…notatki…

# Opracowanie projektu bazy danych

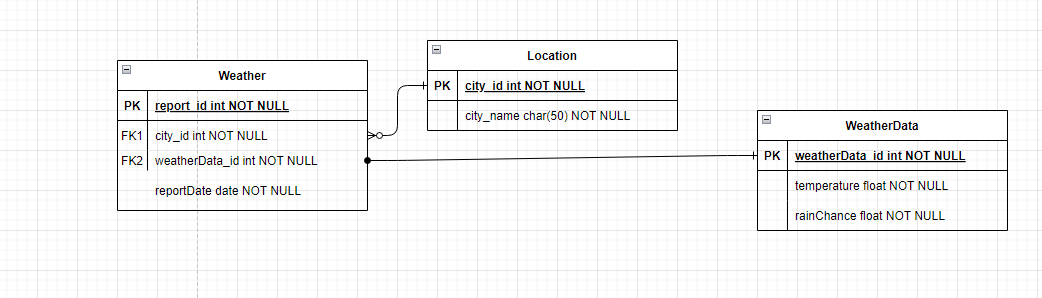
Na tym etapie laboratorium założenia projektu powinny być już znane. Częściowo w grupie, a częściowo samodzielnie, wykorzystaj platformę draw.io do opracowania diagramu ERD dla projektu pogodynki. Kilka wytycznych:

* 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:



Omów najważniejsze założenia swojego diagramu ERD:

Główną encją jest encja **Weather,** która zespala wszystkie inne encje obecne w diagramie> posiada ona dwa klucze obce które odpowiadają kolejno **Location** i **WeatherData**, gdzie Location przechowuje nazwy miejscowości obsługiwanych przez aplikacje pogodynki a WeatherData posiadada w sobie wszystkie zebrane przez stacje pogodowe informacje.

|  |  |  |
| --- | --- | --- |
| 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://github.com/huvcio/ai2-pogodynka-49397

|  |  |  |
| --- | --- | --- |
| 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

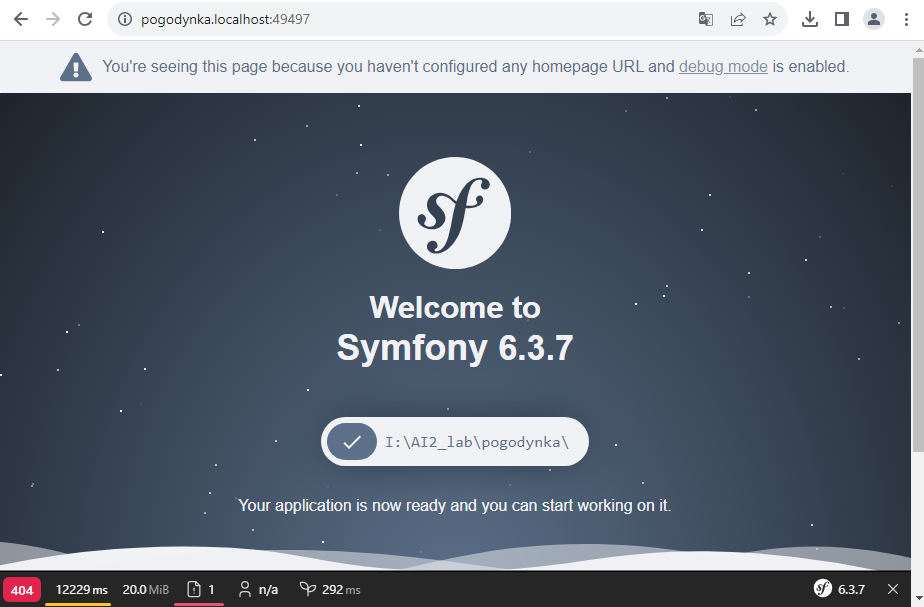
php -S localhost:49497

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:49497.

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>.

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

W sekcji "**require**" wyróżniają się pakiety takie jak doctrine/orm dla zaawansowanej obsługi baz danych, symfony/framework-bundle zapewniający podstawową strukturę aplikacji, oraz symfony/mailer i symfony/notifier ułatwiające pracę z pocztą i powiadamianiem. Dodatkowo, symfony/security-bundle zapewnia funkcje bezpieczeństwa, a symfony/twig-bundle ułatwia obsługę szablonów.

W sekcji "**require-dev**" wyróżniają się pakiety takie jak phpunit/phpunit umożliwiający jednostkowe testowanie, symfony/maker-bundle automatyzujący proces generowania kodu oraz symfony/web-profiler-bundle dostarczający rozbudowany profiler i narzędzia diagnostyczne.   
Pakiety te wspierają efektywny rozwój aplikacji, ułatwiając testowanie, generowanie kodu oraz analizę wydajności, co przekłada się na wyższą jakość oprogramowania.

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

W pliku ***composer.json*** różnica między sekcją **require** a **require-dev** polega na tym, że pakiety wymienione w sekcji **require** są niezbędne do działania aplikacji w produkcji, podczas gdy pakiety w sekcji **require-dev** są potrzebne jedynie podczas procesu developmentu, testowania i debugowania aplikacji, czyli nie są wymagane w środowisku produkcyjnym.

|  |  |  |
| --- | --- | --- |
| 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 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

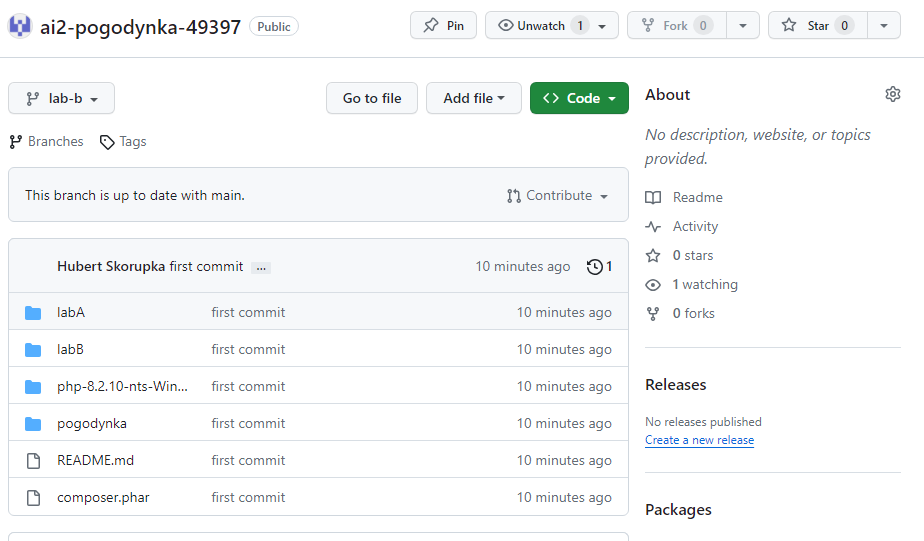
git push -u origin main

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:

https://github.com/huvcio/ai2-pogodynka-49397/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.

Przypomnieliśmy sobie jak poprawnie używać Gita i Symphony.

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.