

WPROWADZENIE DO PROGRAMOWANIA W PHP

SPIS TREŚCI

Spis treści	1
Cel zajęć.....	1
Uwaga	1
Pobranie i uruchomienie PHP	2
Instalacja Node.js i LESS	3
Pobranie i konfiguracja frameworka	4
XDEBUG.....	5
(Alternatywnie) VS Code	7
Zatrzymanie w pułapce	8
Omówienie frameworka	9
Rozszerzenie aplikacji	10
Commit projektu do GIT.....	14
Podsumowanie.....	14

CEL ZAJĘĆ

Celem głównym zajęć jest zdobycie następujących umiejętności:

- konfiguracji środowiska do programowania w PHP;
- połączenia z bazą danych;
- konfiguracji narzędzi do debugowania;
- analizy istniejących systemów webowych bez dokumentacji;
- uzupełniania istniejących systemów webowych o nowe funkcjonalności.

W praktycznym wymiarze uczestnicy zapoznają się z istniejącym naiwnym frameworkiem do tworzenia systemów webowych i uzupełnią go o nową funkcjonalność.

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**.

POBRANIE I URUCHOMIENIE PHP

Zaloguj się do systemu Windows / pulpitu zdalnego `rdp.wi.zut.edu.pl`:

- spoza sieci ZUT potrzebny VPN: <https://uci.zut.edu.pl/uslugi-uci/vpn.html>;
- nazwa użytkownika: `WIAD\ab12345`
- komputer: `rdp.wi.zut.edu.pl`

Odwiedź stronę <https://windows.php.net/download/>. Pobierz `PHP 8.2.* x64 NTS`.

Wypakuj pobrane repozytorium do `I:\php`.

Otwórz panel sterowania. W polu wyszukiwania wpisz `path`. Wybierz edycję zmiennych środowiskowych użytkownika. Znajdź zmienną `Path` i kliknij edycję. Dodaj ścieżkę `I:\php`.

UWAGA! W toku studiów być może do `Path` zostały dodane inne wersje PHP lub XAMPP. Za kilka kroków sprawdzana będzie wersja PHP – upewnij się, że faktycznie wiesz z której instancji PHP korzystasz przez resztę tego laboratorium 😊.

Skopiuj plik `I:\php\php.ini-development` jako `php.ini`, po czym edytuj jego zawartość – odkomentuj poniższe ustawienia:

```
extension_dir = "ext"
...
extension=curl
extension=gd
extension=intl
extension=mbstring
extension=openssl
extension=pdo_sqlite
```

Otwórz terminal PowerShell i wejdź do katalogu laboratoriów.

Wykonaj komendę

```
php -i | Select-String -Pattern '(PHP Version)|(extension_dir)|(OpenSSL support)|(PDO drivers)|(GD Support)|intl|(cURL support)|multibyte'
```

Oczekiwany wynik:

```
PS C:\Users\artur\workspace\AI2-lab\labA> php -i | Select-String -Pattern '(PHP Version)|(extension_dir)|(OpenSSL support)|(PDO drivers)|(GD Support)|intl|(cURL support)|multibyte'
```

```
PHP Version => 8.2.10
Zend Multibyte Support => provided by mbstring
PHP Version => 8.2.10
extension_dir => ext => ext
zend.multibyte => Off => Off
cURL support => enabled
GD Support => enabled
intl
intl.default_locale => no value => no value
intl.error_level => 0 => 0
intl.use_exceptions => Off => Off
Multibyte Support => enabled
Multibyte string engine => libmbfl
Multibyte (japanese) regex support => enabled
Multibyte regex (oniguruma) version => 6.9.8
OpenSSL support => enabled
PDO drivers => sqlite
OpenSSL support => enabled
```

Zastąp poniższy obrazek swoim zrzutem ekranu:

```
PS E:\labbyygit\Computer_Science\year3\AI-1\lab\lab6F> php -i | Select-String -Pattern '(PHP Version)|(extension_dir)|(OpenSSL support)|(PDO drivers)|(GD Support)|intl|(cURL support)|multibyte'

PHP Version => 8.3.12
Zend Multibyte Support => provided by mbstring
PHP Version => 8.3.12
extension_dir => ext => ext
zend.multibyte => Off => Off
cURL support => enabled
GD Support => enabled
intl
intl.default_locale => no value => no value
intl.error_level => 0 => 0
intl.use_exceptions => Off => Off
Multibyte Support => enabled
Multibyte string engine => libmbfl
Multibyte (japanese) regex support => enabled
Multibyte regex (oniguruma) version => 6.9.8
OpenSSL support => enabled
PDO drivers => mysql, sqlite
OpenSSL support => enabled
sqlite3.extension_dir => no value => no value
```

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

INSTALACJA NODE.JS I LESS

Ta sekcja częściowo mogła zostać zrealizowana podczas wcześniejszych laboratoriów. Upewnij się, że zainstalowana jest w miarę bieżąca wersja Node.js.

Wejdź na stronę <https://nodejs.org/en/download/current>. Pobierz wersję Current -> Windows Binary (.zip) -> 64-bit. Rozpakuj archiwum do I:\node. Dodaj ten folder do zmiennej środowiskowej Path użytkownika.

Uruchom nowe okno wiersza poleceń. Wykonaj polecenie `npm -v`. Powinno zadziałać – udało się zainstalować NODE i NPM lokalnie dla użytkownika.

Następnie zainstalujemy Less. Wykonaj polecenie:

```
npm install -g less less-plugin-clean-css
```

Sprawdź poprawność instalacji:

```
lessc -v
```

Wstaw zrzut ekranu wyniku działania poleceń `npm -v` i `lessc -v`

```
E:\labbyygit\Computer_Science\year3\AI-1\lab\lab6F>npm -v
10.9.0

E:\labbyygit\Computer_Science\year3\AI-1\lab\lab6F>lessc -v
lessc 4.2.1 (Less Compiler) [JavaScript]
```

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

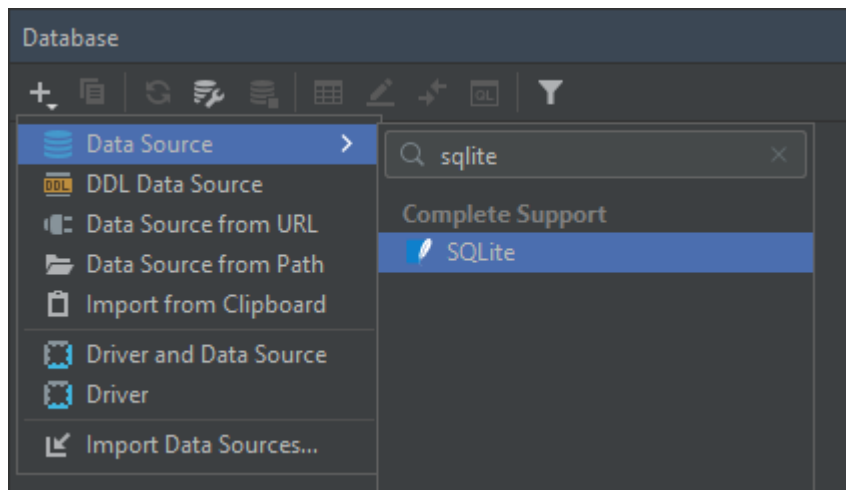
POBRANIE I KONFIGURACJA FRAMEWORKA

Sklonuj naiwny framework z Githuba: <https://github.com/IdeaSpotPL/custom-php-framework>.

Wejdź do pobranego katalogu i skompiluj style:

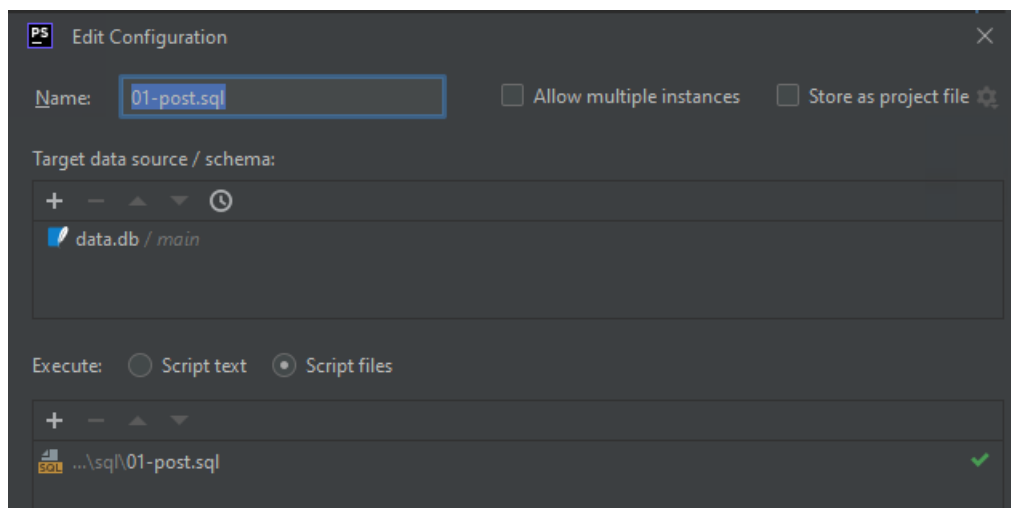
```
lessc public/assets/src/less/style.less public/assets/dist/style.min.css --clean-css  
--source-map
```

Otwórz projekt w PhpStorm. Rozwiń panel Database z prawej strony. Dodaj nowe źródło danych SQLite:



W sekcji File kliknij przycisk New + i utwórz plik data.db w katalogu głównym frameworka. Pobierz brakujące sterowniki, jeśli PhpStorm o to poprosi.

Kliknij prawym przyciskiem w drzewie projektu na sql/01-post.sql i wybierz z menu pozycję Run (...). Dostosuj konfigurację jak poniżej i kliknij Run.

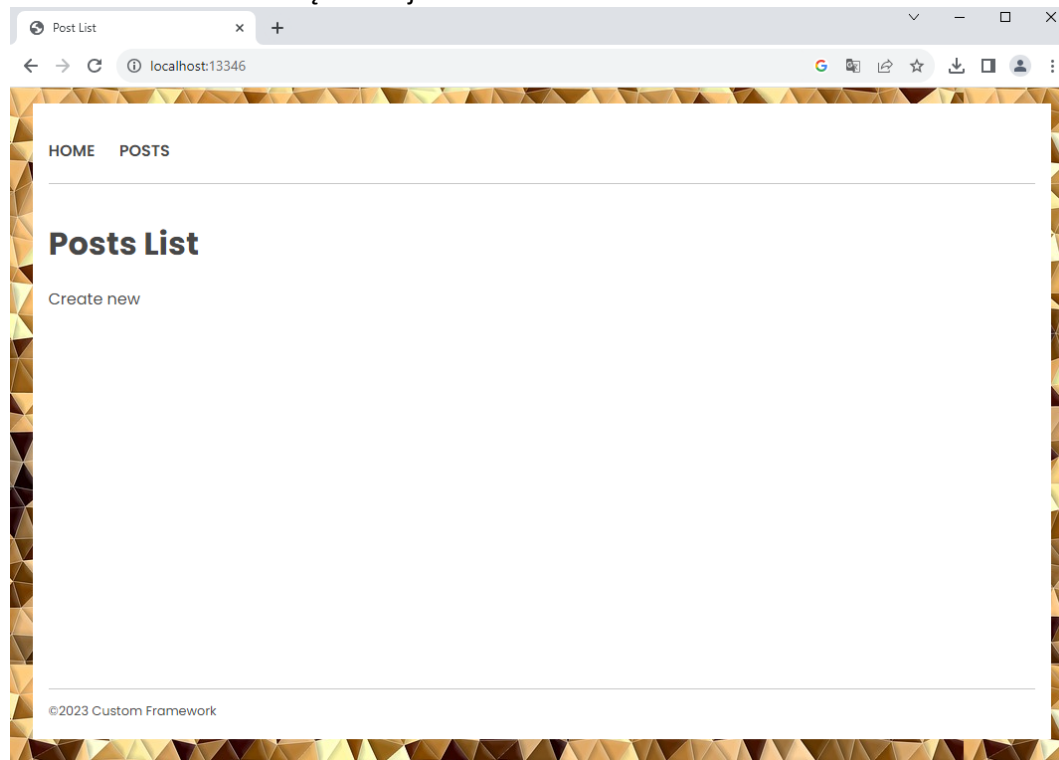


Skopiuj plik config/config.dist.php na config/config.php i dostosuj ustawienia bazy danych.

Terminalem wejdź do katalogu public we frameworku i uruchom wbudowany serwer PHP:

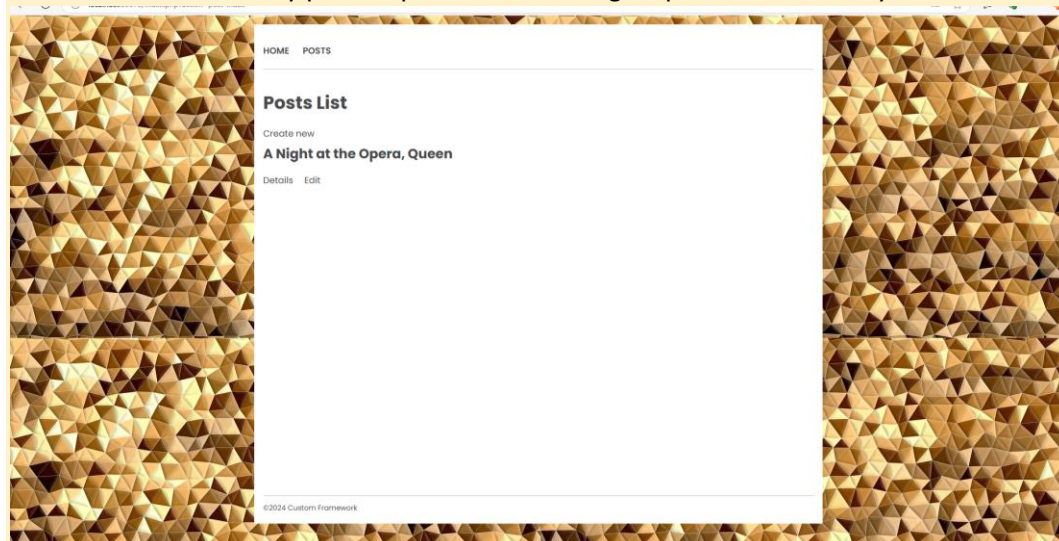
```
php -S localhost:00000
```

Wejdź przeglądarką na powyższy adres. Framework powinien się wyświetlić:



Utwórz nowy wpis poprzez kliknięcie na **Create new**, wypełnienie przykładowym wpisem z Twoimi danymi i wysyłkę formularza.

Wstaw zrzut ekranu listy postów po dodaniu nowego wpisu z Twoimi danymi.



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

XDEBUG

Wejdź na stronę <https://xdebug.org/wizard>. Postępuj zgodnie z instrukcjami, tj:

- skopiuj zawartość strony <http://localhost:53882/?action=info> do okienka kreatora
- pobierz wskazany plik DLL do odpowiedniego katalogu

- zmodyfikuj zawartość php.ini ze wskazanej ścieżki
- zrestartuj serwer WWW.

Przykładowe ustawienia do wpisania do php.ini:

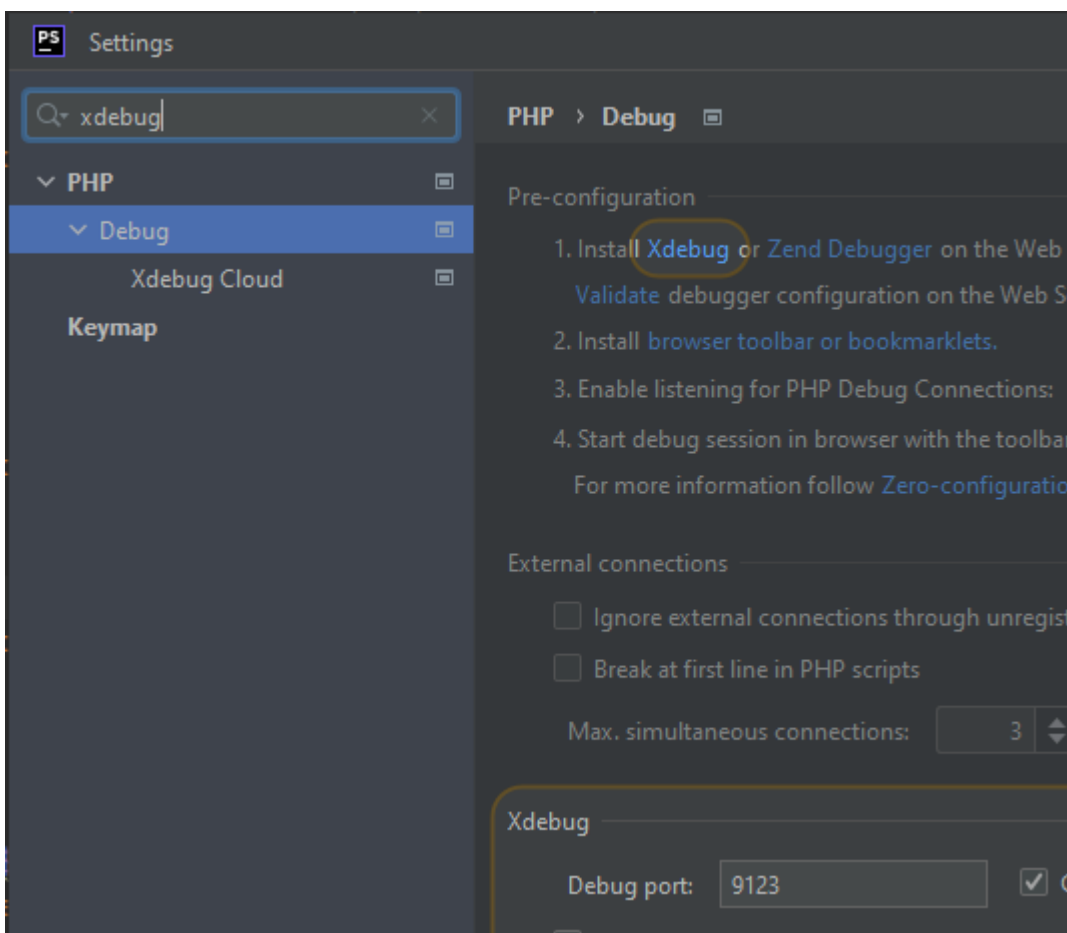
```
zend_extension = xdebug
xdebug.mode = debug
xdebug.start_with_request = yes
xdebug.client_port = 53882
```

Wyróżniony powyżej port 9123 należy zmienić na inny, losowy, unikalny. W stronie PHP INFO powinien pojawić się wpis o XDEBUG:

This program makes use of the Zend Scripting Language Engine:
Zend Engine v4.2.11, Copyright (c) Zend Technologies
with Xdebug v3.2.2, Copyright (c) 2002-2023, by Derick Rethans

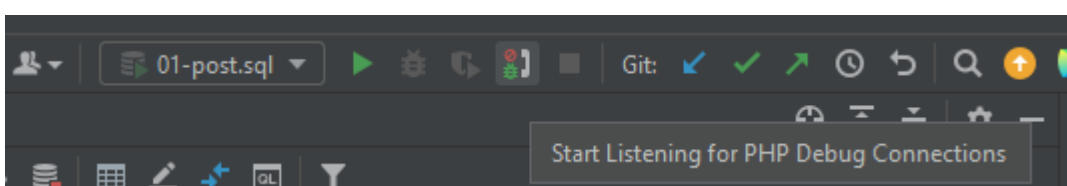
zendengine

W PhpStorm kliknij File -> Settings (Ctrl + Alt + S). Wyszukaj xdebug. Ustaw port zgodnie z powyżej wylosowanym:



Zatwierdź zmiany.

Otwórz plik public/index.php projektu i kliknij obok jednej z pierwszych instrukcji kodu, żeby utworzyć pułapkę (breakpoint). W górnym pasku PhpStorm kliknij ikonkę słuchawki (Start Listening (...)):

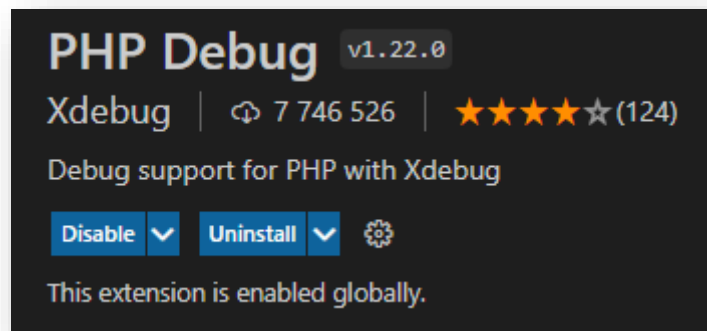


Odśwież stronę w przeglądarce. Wykonywanie powinno się zatrzymać, a PhpStorm powinien wyświetlać informacje o stanie zmiennych w pamięci i umożliwiać przejście przez aplikację krok po kroku.

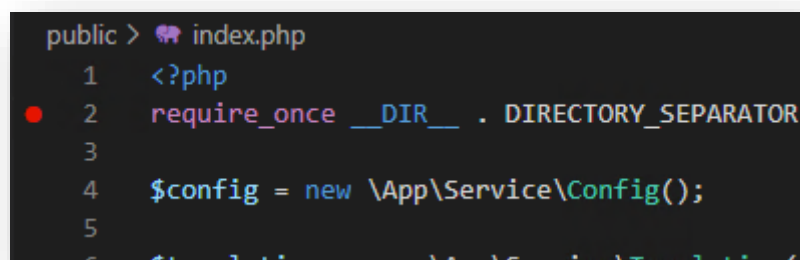
(ALTERNATYWNIE) VS CODE

Jeśli z jakiegoś powodu nie możesz lub nie chcesz korzystać z PhpStorm – spróbuj jeszcze raz. Jeśli dalej nie możesz, poniżej instrukcja na VS Code – ale przez resztę semestru będzie Ci o wiele trudniej!

W Visual Studio Code zainstaluj rozszerzenie PHP Debug:



Otwórz plik index.php projektu i kliknij obok numeru linii przy pierwszych instrukcjach, utworzony zostanie breakpoint:



Kliknij menu Run -> Add Configuration... . Utworzony zostanie plik launch.json. Edytuj pozycję port, na port wskazany w konfiguracji XDEBUG w php.ini:

```
.vscode > {} launch.json > Launch Targets > {} Listen for Xdebug
1  {
2      // Use IntelliSense to learn about possible attributes
3      // Hover to view descriptions of existing attributes
4      // For more information, visit: https://go.microsoft.com/fwlink/?linkid=829397
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "name": "Listen for Xdebug",
9              "type": "php",
10             "request": "launch",
11             "port": 9123
12         },
13     ]
14 }
```

Kliknij menu Run -> Start Debugging (F5). Rozpocznie się oczekiwanie na połączenie:



Odśwież stronę frameworka, rozpocznie się proces debugowania.

ZATRZYMANIE W PUŁAPCE

Wstaw zrzut ekranu potwierdzający skuteczne zatrzymanie aplikacji na pułapce (breakpoint) w IDE (PhpStorm lub VSCode).


```

1 <?php
2 require_once __DIR__ . DIRECTORY_SEPARATOR . '..' . DIRECTORY_SEPARATOR .
3
4 $config = new \App\Service\Config(); $config = uninitialized
5
6 $templating = new \App\Service\Templating(); $templating = uninitialized
7 $router = new \App\Service Router(); $router = uninitialized
8
9 $action = $_REQUEST['action'] ?? null; $action = uninitialized, $_REQUEST
10 switch ($action) { $action = uninitialized
11     case 'post-index':
12     case null:
13         $controller = new \App\Controller\PostController(); $controller =
14         $view = $controller->indexAction($templating, $router); $view = un
15         break;
16     case 'post-create':
17         $controller = new \App\Controller\PostController(); $controller =
18         $view = $controller->createAction($_REQUEST['post'] ?? null, $temp
19         break;
20     case 'post-edit':
21         if (!$_REQUEST['id']) { $_REQUEST = array(0)
22         break;
23     }
24     $controller = new \App\Controller\PostController(); $controller =
25     $view = $controller->editAction($_REQUEST['id'], $_REQUEST['post'])
26     break;
27     case 'post-show':
28         if (!$_REQUEST['id']) { $_REQUEST = array(0)
29         break;
30     }
31     $controller = new \App\Controller\PostController(); $controller =
32     $view = $controller->showAction($_REQUEST['id'], $templating, $rou
33     break;
34     case 'post-delete':
35         if (!$_REQUEST['id']) { $_REQUEST = array(0)
36         break;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

[Sat Dec 7 03:08:58 2024] PHP 8.3.12 Development Server (http://localhost:53882) started
[Sat Dec 7 03:09:03 2024] [::1]:65096 Accepted
[Sat Dec 7 03:09:03 2024] [::1]:65097 Accepted
[Sat Dec 7 03:09:03 2024] Xdebug: [Step Debug] Time-out connecting to debugging client, waited: 200
ms. Tried: 127.0.0.1:53882 (through xdebug.client_host/xdebug.client_port).
[Sat Dec 7 03:09:03 2024] [::1]:65097 [200]: GET /
[Sat Dec 7 03:09:03 2024] [::1]:65097 Closing
[Sat Dec 7 03:09:03 2024] [::1]:65096 [200]: GET /assets/dist/style.min.css
[Sat Dec 7 03:09:03 2024] [::1]:65096 Closing
[Sat Dec 7 03:09:20 2024] [::1]:65099 Accepted
[Sat Dec 7 03:09:20 2024] [::1]:65100 Accepted

```

Punkty:

0

1

OMÓWIENIE FRAMEWORKA

Połącz się z prowadzącym na spotkaniu TEAMS. Za pomocą udostępniania ekranu, prowadzący omówi strukturę frameworka, obejmującą:

- plik konfiguracyjny config.php i config.dist.php
- organizację assetów – public/assets/src i public/assets/dist
- propozycja organizacji style.less i załączonych stylów oraz katalogu public/assets/src/vendor
- znaczenie i propozycja wykorzystania katalogu sql na migracje
- plik index.php – front-controller
- katalog Controller na przykładzie PostController – akcje związane z postami
- przykładowy model w Post.php
- serwisy w katalogu src/Service – czytnik konfiguracji, router, szablony
- szablony i sposób ich dziedziczenia w katalogu templates

Uważnie analizuj pokazywany kod i zadawaj pytania, aby zrozumieć.

Miejsce na Twoje notatki:

...notatki...

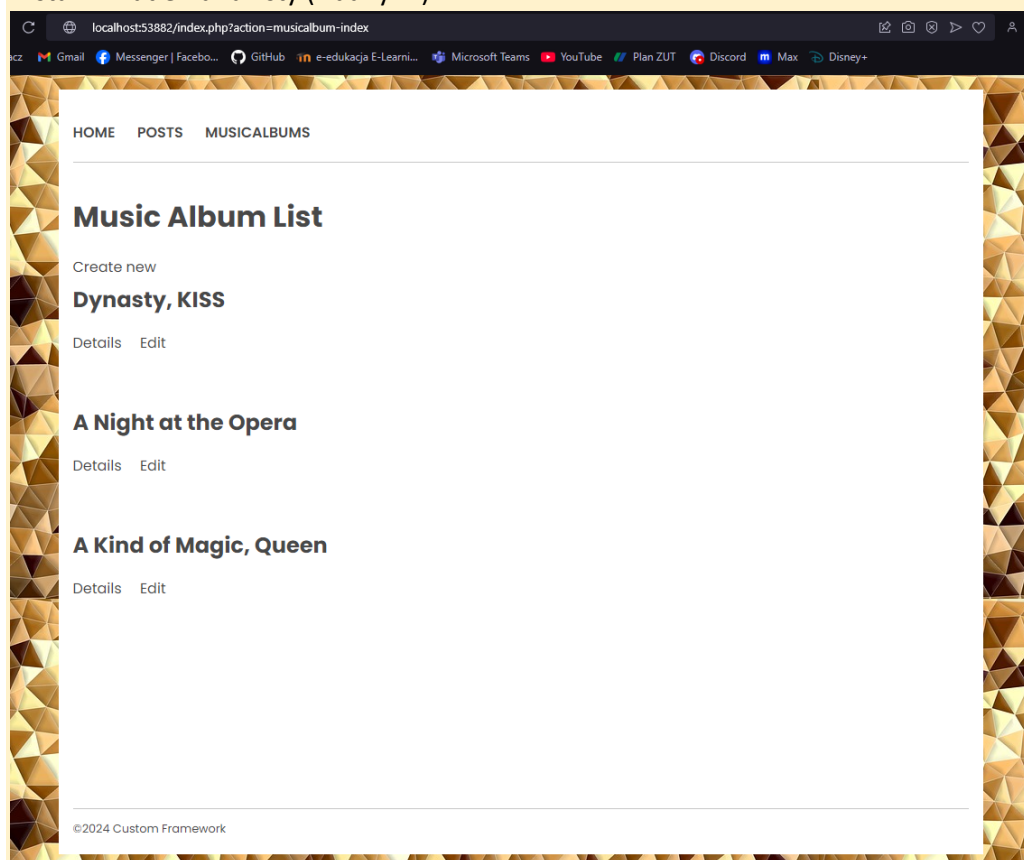
ROZSZERZENIE APLIKACJI

Wymyśl inny model podobny do prezentowanego `Post`. Rozszerz aplikację o implementację swojego modelu oraz związane z nim akcje – lista, prezentacja, tworzenie, edycja, kasowanie.

Wstaw zrzut ekranu kodu listy (`indexAction`):

```
public function indexAction(Templating $templating, Router $router): ?string
{
    $musicalbums = MusicAlbum::findAll();
    $html = $templating->render('musicalbum/index.html.php', [
        'musicalbums' => $musicalbums,
        'router' => $router,
    ]);
    return $html;
}
```

Wstaw zrzut ekranu listy (z danymi):



Punkty:

0

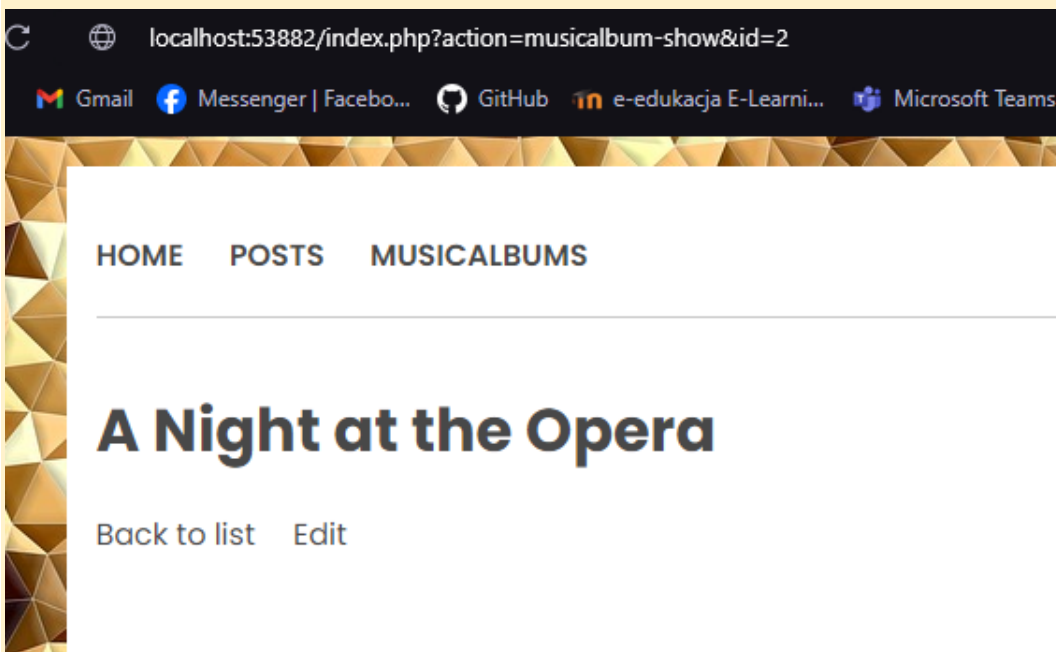
1

Wstaw zrzut ekranu kodu podglądu (showAction):

```
public function showAction(int $musicalbumId, Templating $templating, Router $router): ?string
{
    $musicalbum = Musicalbum::find($musicalbumId);
    if (!$musicalbum) {
        throw new NotFoundException("Missing musicalbum with id $musicalbumId");
    }

    $html = $templating->render('musicalbum/show.html.php', [
        'musicalbum' => $musicalbum,
        'router' => $router,
    ]);
    return $html;
}
```

Wstaw zrzut ekranu podglądu:



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

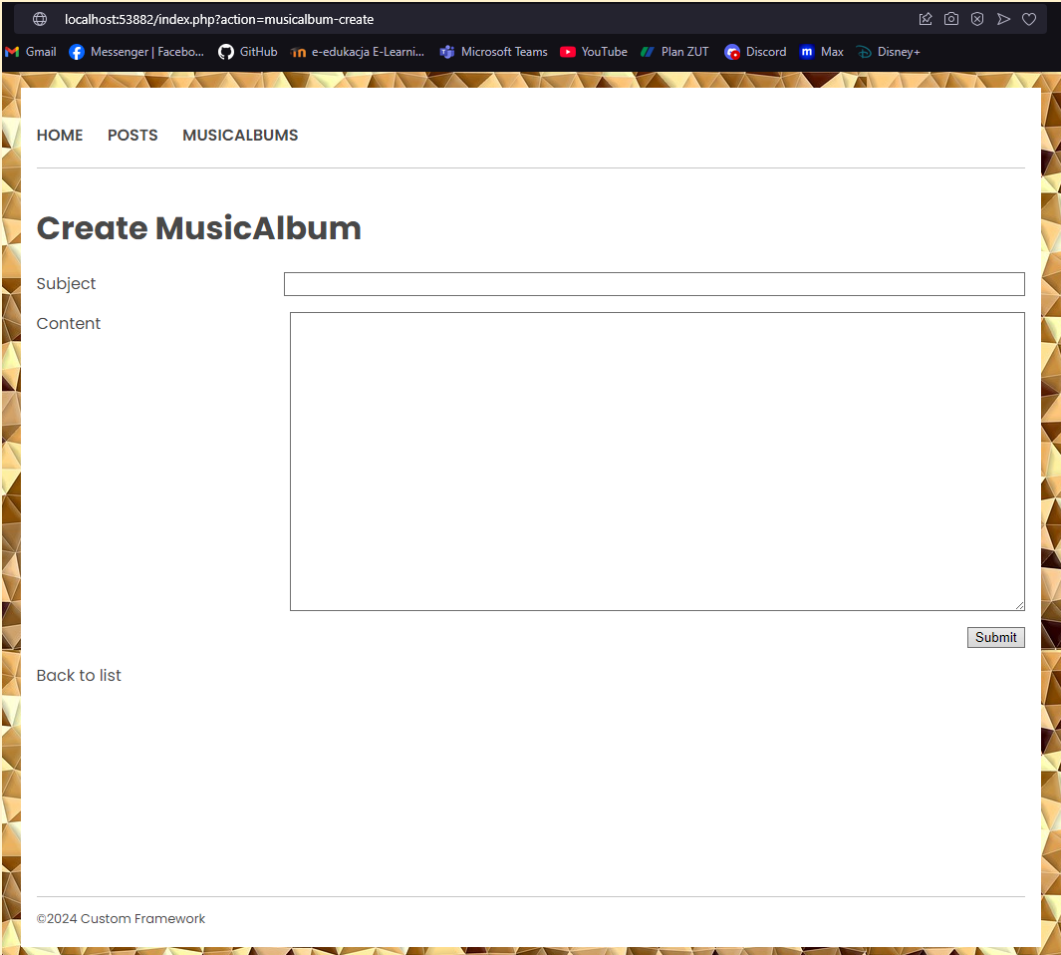
Wstaw zrzut ekranu kodu tworzenia (createAction):

```
public function createAction(?array $requestMusicAlbum, Templating $templating, Router $router): ?string
{
    if ($requestMusicAlbum) {
        $musicalbum = MusicAlbum::fromArray($requestMusicAlbum);
        // @todo missing validation
        $musicalbum->save();

        $path = $router->generatePath( action: 'musicalbum-index');
        $router->redirect($path);
        return null;
    } else {
        $musicalbum = new MusicAlbum();
    }

    $html = $templating->render( template: 'musicalbum/create.html.php', [
        'musicalbum' => $musicalbum,
        'router' => $router,
    ]);
    return $html;
}
```

Wstaw zrzut ekranu tworzenia:



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

Wstaw zrzut ekranu kodu edycji (editAction):

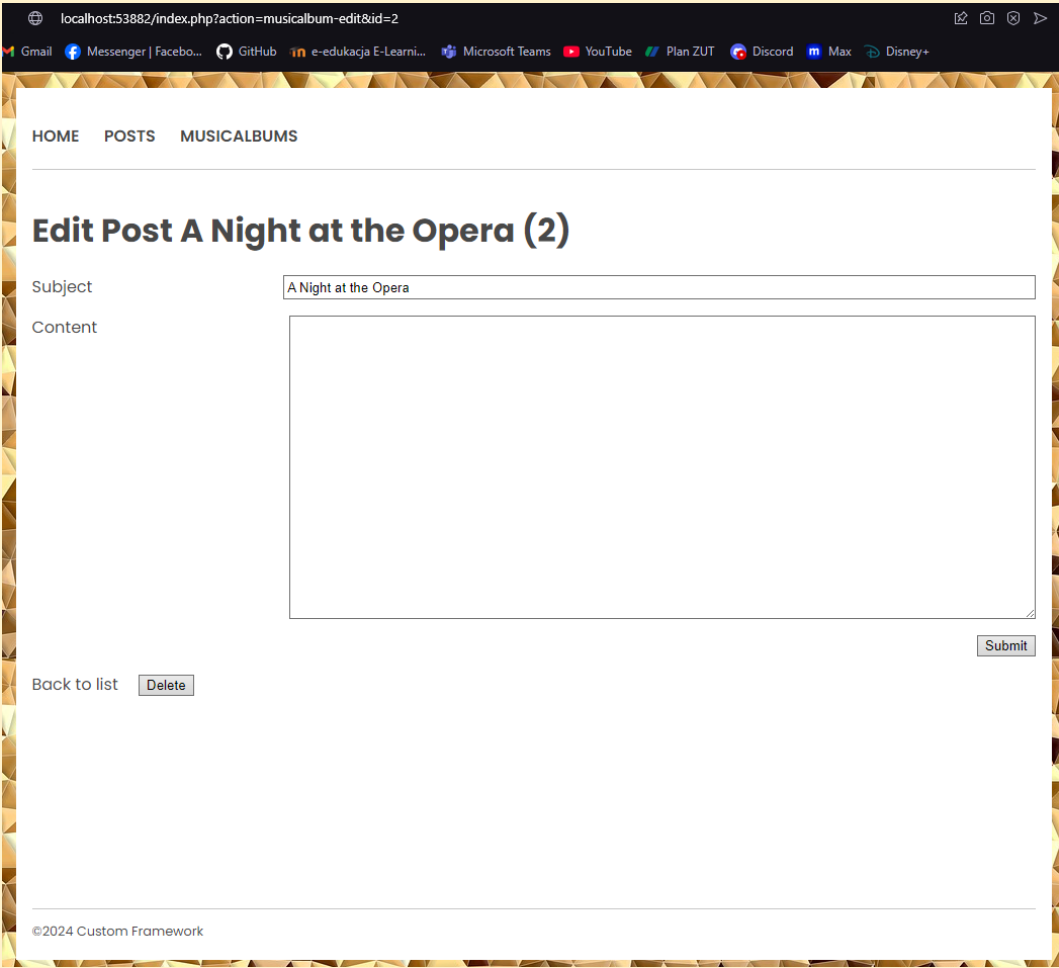
```
public function editAction(int $musicalbumId, ?array $requestMusicalbum, Templating $templating, Router $router): ?string
{
    $musicalbum = Musicalbum::find($musicalbumId);
    if (! $musicalbum) {
        throw new NotFoundException( message: "Missing post with id $musicalbumId");
    }

    if ($requestMusicalbum) {
        $musicalbum->fill($requestMusicalbum);
        // @todo missing validation
        $musicalbum->save();

        $path = $router->generatePath( action: 'musicalbum-index');
        $router->redirect($path);
        return null;
    }

    $html = $templating->render( template: 'musicalbum/edit.html.php', [
        '$musicalbum' => $musicalbum,
        'router' => $router,
    ]);
    return $html;
}
```

Wstaw rzrzut ekranu edycji:



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

Wstaw rzrzut ekranu kodu kasowania (deleteAction):

```

public function deleteAction(int $musicalbumId, Router $router): ?string
{
    $musicalbum = MusicAlbum::find($musicalbumId);
    if (!$musicalbum) {
        throw new NotFoundException(message: "Missing musicalbum with id $musicalbumId");
    }

    $musicalbum->delete();
    $path = $router->generatePath(action: 'musicalbum-index');
    $router->redirect($path);
    return null;
}

```

Punkty:

0

1

COMMIT PROJEKTU DO GIT

Zacommittuj i pushnij swoje rozwiązanie do **swojego** repozytorium GIT.

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

Podaj link do brancha `lab-f` w swoim repozytorium:

https://github.com/mkowiecka/Computer_Science/tree/master/year3/AI-1/lab/6F

...link, np. <https://github.com/inazwisko/ai1-lab/tree/lab-f...>

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

Nauczyłam się korzystać z PhpStorm i zapoznałam się z tworzeniem framework.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.