

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.....	6
Zatrzymanie w pułapce.....	8
Omówienie frameworka.....	8
Rozszerzenie aplikacji.....	8
Commit projektu do GIT.....	11
Podsumowanie.....	11

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 8.2.10 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.

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)|in
tl|(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 I:\API\LabF> php -i | Select-String -Pattern '(PHP Version)|(extension_dir)|(OpenSSL support)|(PDO drivers)|(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

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

INSTALACJA NODE.JS I LESS

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`

```
PS I:\> npm -v
10.2.3
```

```
PS I:\> lessc -v
lessc 4.2.0 (Less Compiler) [JavaScript]
```

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

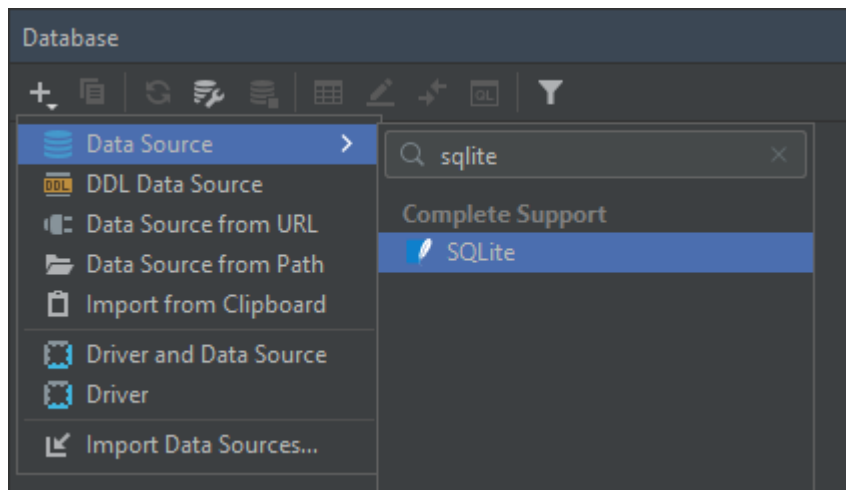
POBRANIE I KONFIGURACJA FRAMEWORKA

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

Wejdź do pobranego katalogu i skompiluj style:

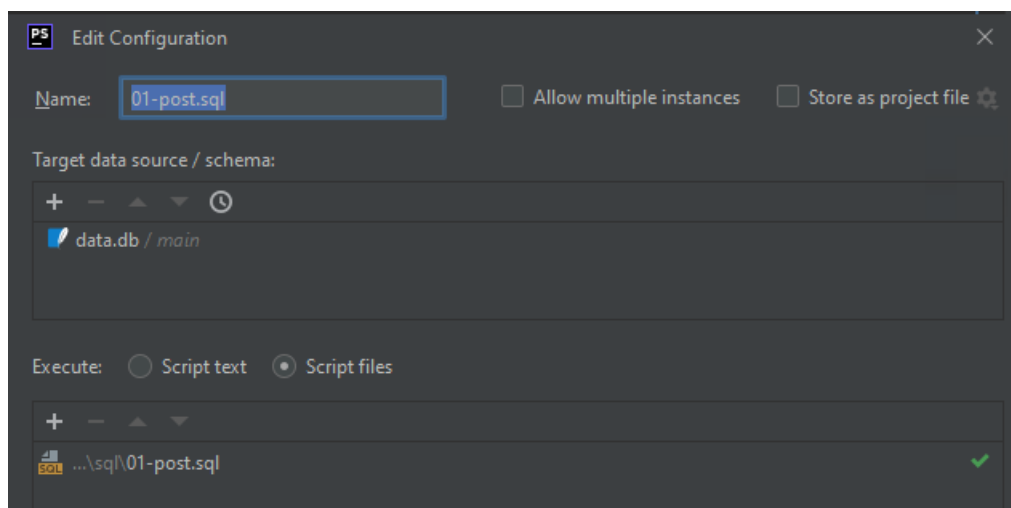
```
lessc public\assets\src\less\style.less public\assets\dist\style.min.css --clean-css  
--source-map
```

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.

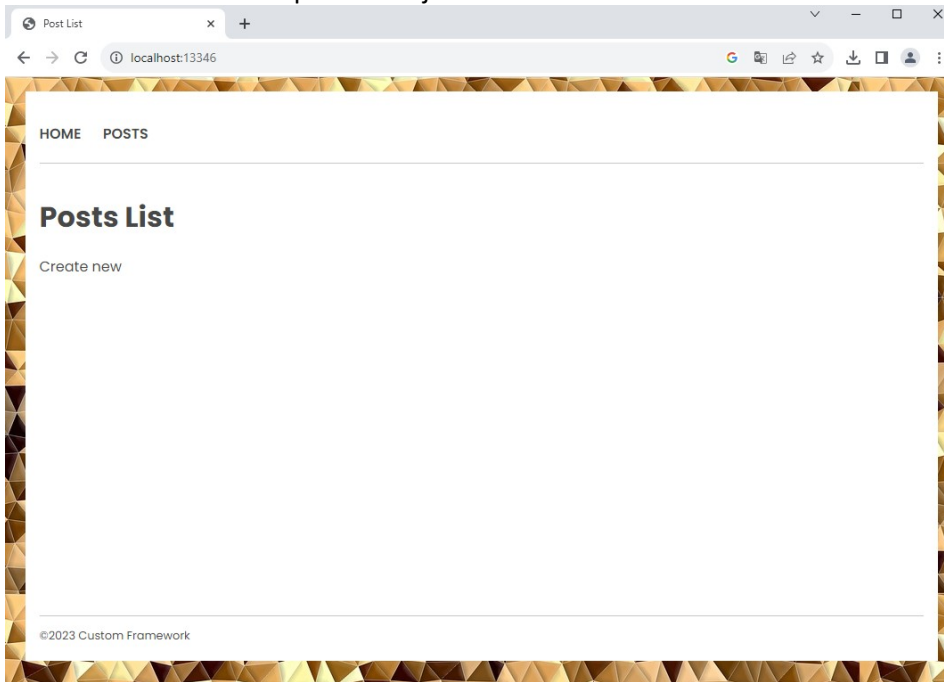


Skopiuj plik config\config.dist.php na config\config.php i dostosuj ustawienia bazy danych.

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

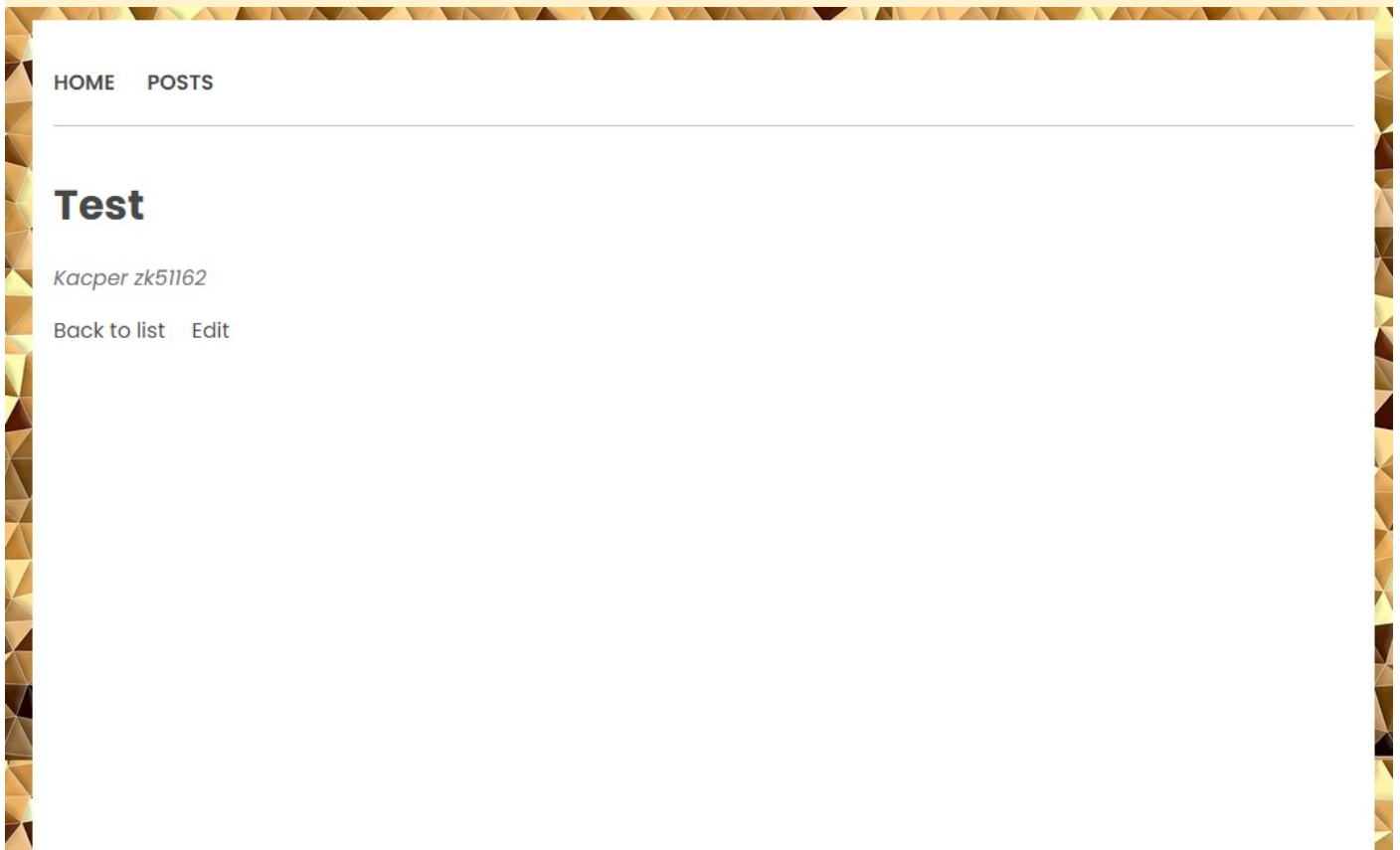
```
php -S localhost:51162
```

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:<port>/?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`:

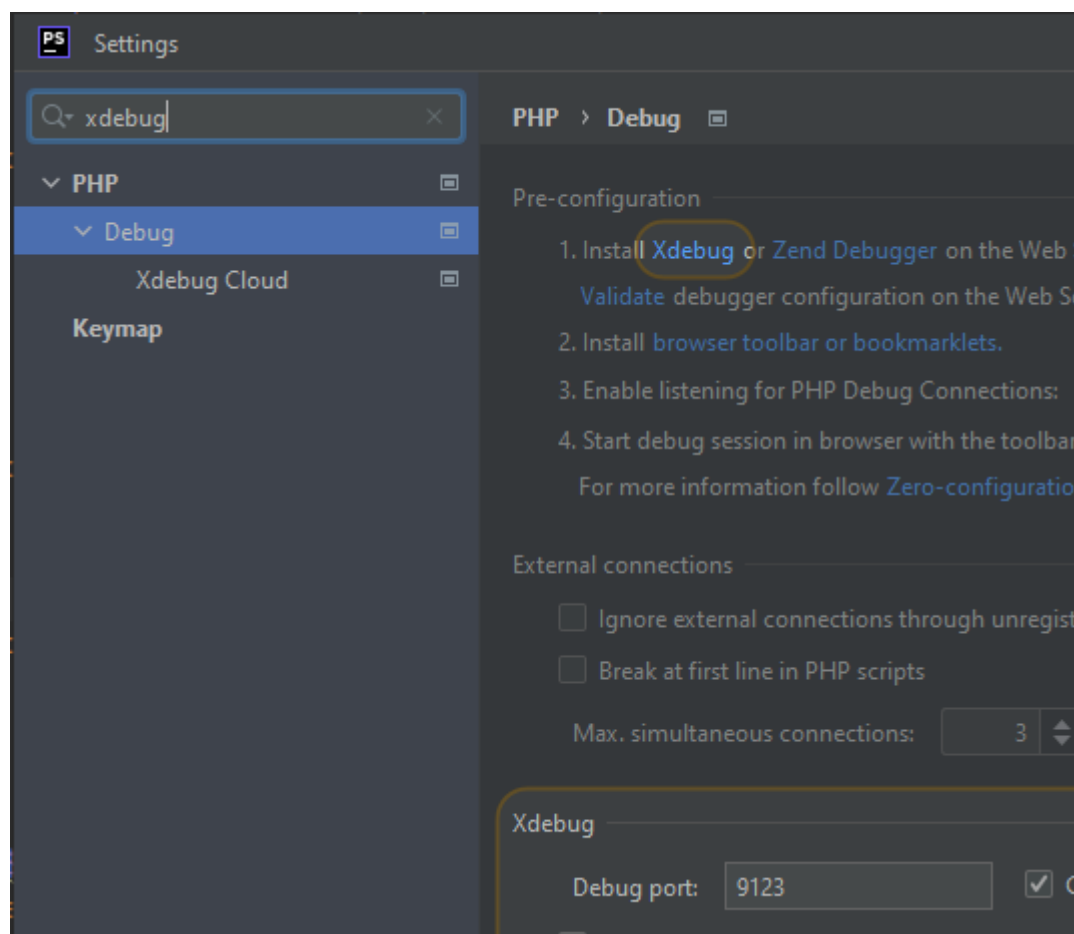
```
zend_extension = xdebug
xdebug.mode = debug
xdebug.start_with_request = yes
xdebug.client_port = 9123
```

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

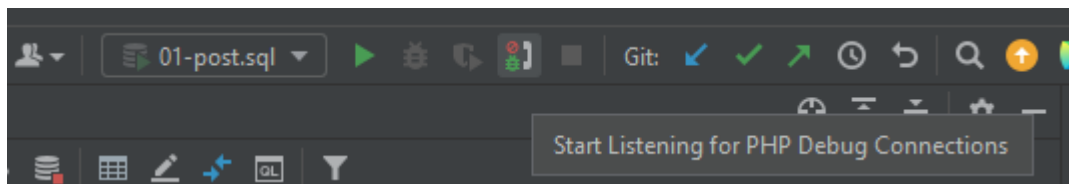
zendengine

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



Zatwierdź zmiany.

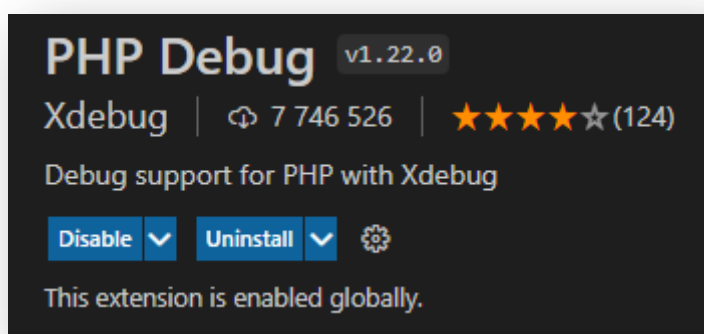
Otwórz plik `public/index.php` projektu i kliknij obok jednej z pierwszych instrukcji kodu, żeby utworzyć pułapkę (breakpoint). W górnym pasku PhpStorm kliknij ikonkę słuchawki (Start Listening (...)):



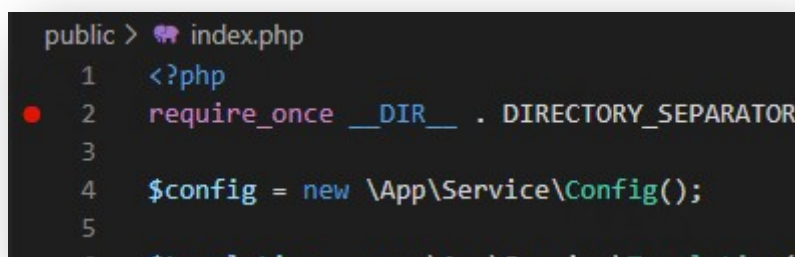
Odśwież stronę w przeglądarce. Wykonywanie powinno się zatrzymać, a PhpStorm powinien wyświetlać informacje o stanie zmiennych w pamięci i umożliwiać przejście przez aplikację krok po kroku.

(ALTERNATYWNIE) VS CODE

W Visual Studio Code zainstaluj rozszerzenie PHP Debug:



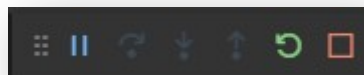
Otwórz plik `index.php` projektu i kliknij obok numeru linii przy pierwszych instrukcjach, utworzony zostanie breakpoint:



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=830383
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 }
```

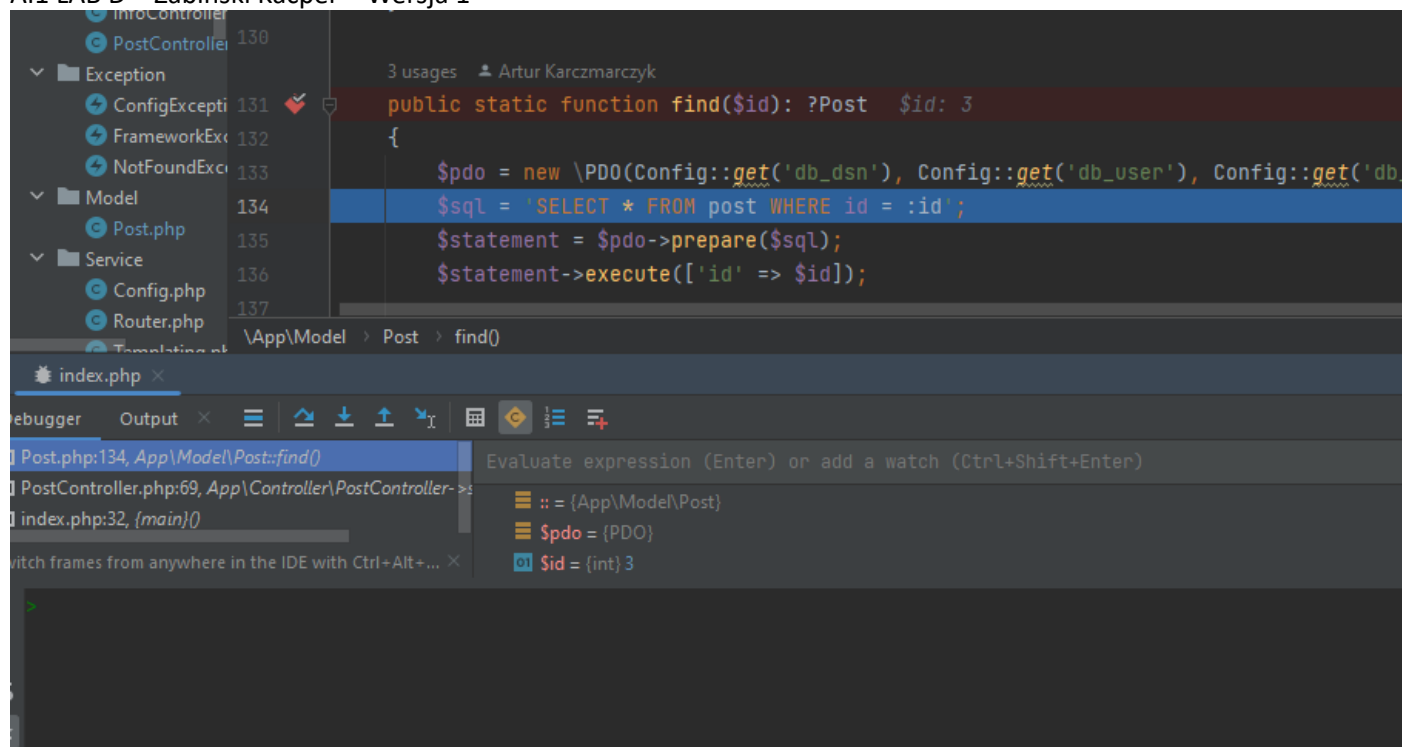
Kliknij menu Run -> Start Debugging (F5). Rozpocznie się oczekiwanie na połączenie:



Odśwież stronę frameworka, rozpocznie się proces debugowania.

ZATRZYMANIE W PUŁAPCE

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



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

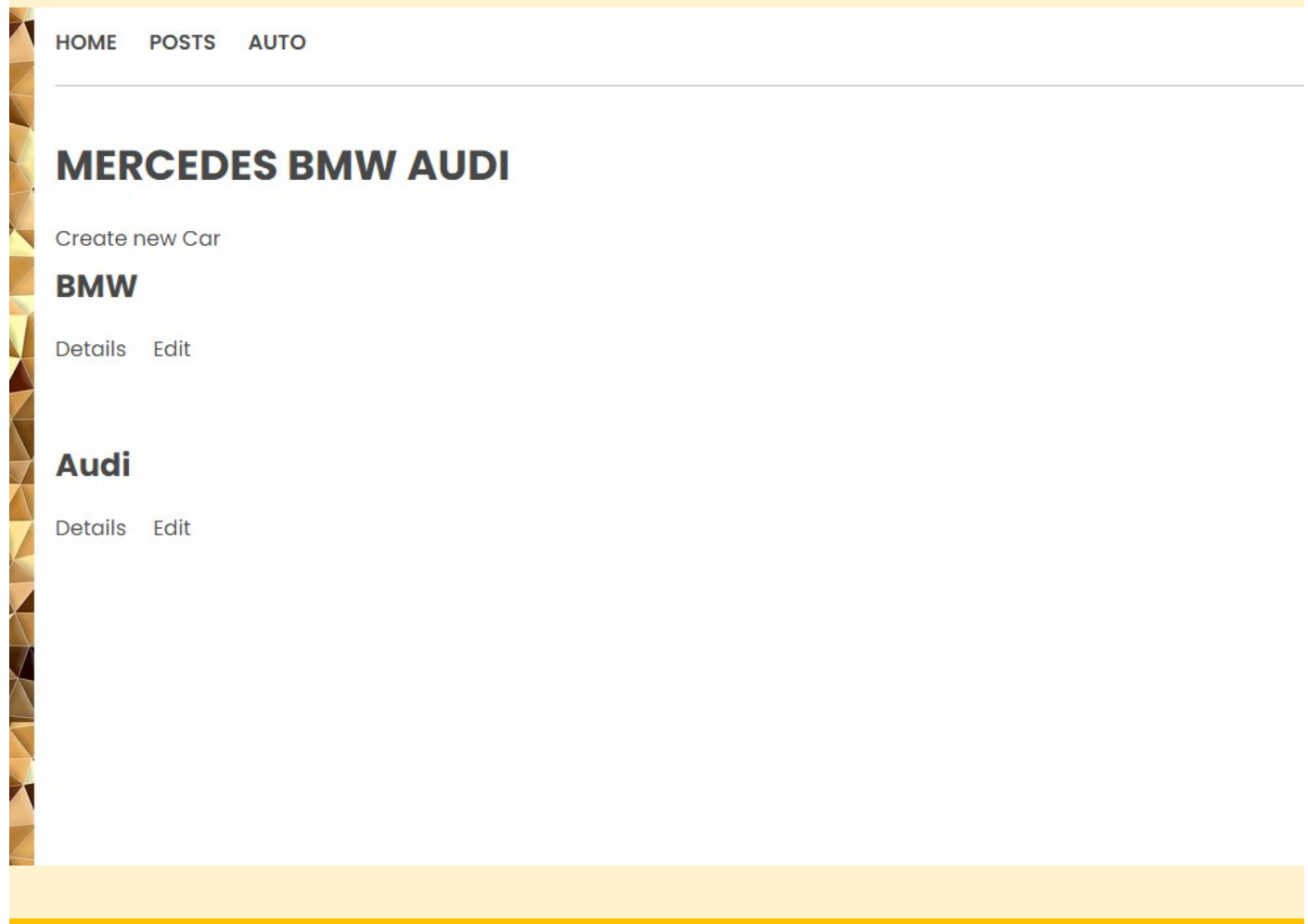
Wymyśl inny model podobny do prezentowanego `Post`. Rozszerz aplikację o implementację swojego modelu oraz związane z nim akcje – lista, prezentacja, tworzenie, edycja, kasowanie.

Wstaw zrzut ekranu kodu listy (`indexAction`):

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

1 usage Artur Karczmarczyk *

Wstaw zrzut ekranu listy (z danymi):



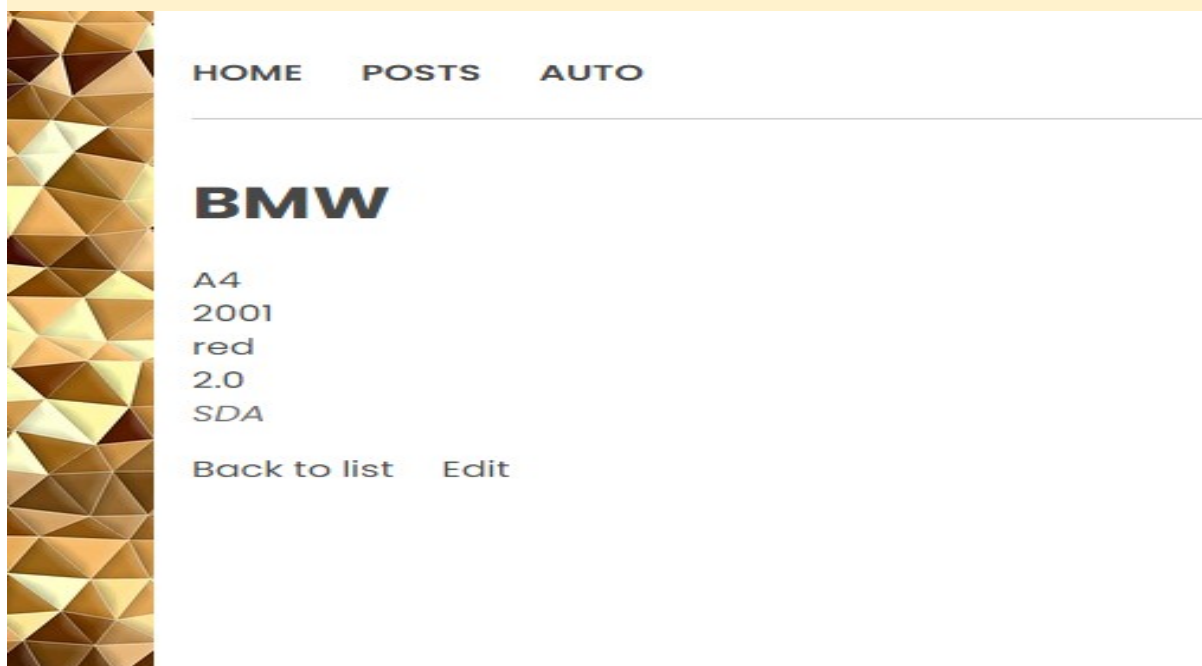
Punkty:	0	1
---------	---	---

Wstaw zrzut ekranu kodu podglądu (showAction):

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

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

Wstaw zrzut ekranu podglądu:



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

Wstaw zrzut ekranu kodu tworzenia (createAction):

```
2 usages  Artur Karczmarczyk *
public function save(): void
{
    $pdo = new \PDO(Config::get('db_dsn'), Config::get('db_user'), Config::get('db_pass'));
    if (! $this->getId()) {
        $sql = "INSERT INTO post (subject, content,model,year,color,engine) VALUES (:subject, :content,:model,:year,:color,:engine)";
        $statement = $pdo->prepare($sql);
        $statement->execute([
            ':subject' => $this->getSubject(),
            ':model' => $this->getModel(),
            ':year' => $this->getYear(),
            ':color' => $this->getColor(),
            ':engine' => $this->getEngine(),
            ':content' => $this->getContent()
        ]);

        $this->setId($pdo->lastInsertId());
    } else {
        $sql = "UPDATE post SET subject = :subject, model = :model,year=:year,color=:color,engine=:engine, content = :content WHERE id = :id";
        $statement = $pdo->prepare($sql);
        $statement->execute([
            ':subject' => $this->getSubject(),
            ':model' => $this->getModel(),
            ':year' => $this->getYear(),
            ':color' => $this->getColor(),
            ':engine' => $this->getEngine(),
            ':content' => $this->getContent(),
            ':id' => $this->getId()
        ]);
    }
}
```

Wstaw zrzut ekranu tworzenia:

HOMEPOSTSAUTO

Create Auto

Mark

Model

Year

Color

Engine

Content

Submit

Back to list

©2023 Custom Framework

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

Wstaw zrzut ekranu kodu edycji (editAction):

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

    if ($requestPost) {
        $post->fill($requestPost);
        // @todo missing validation
        $post->save();

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

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

Wstaw zrzut ekranu edycji:

HOME POSTS AUTO

Edit Auto BMW (3)

Mark	BMW
Model	A4
Year	2001
Color	red
Engine	2.0
Content	SDA

Back to list [Delete](#) [Submit](#)

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

Wstaw zrzut ekranu kodu kasowania (deleteAction):

```
1 usage  Artur Karczmarczyk *
public function delete(): void
{
    $pdo = new \PDO(Config::get('db_dsn'), Config::get('db_user'), Config::get('db_pass'));
    $sql = "DELETE FROM post WHERE id = :id";
    $statement = $pdo->prepare($sql);
    $statement->execute([
        ':id' => $this->getId(),
    ]);

    $this->setId( id: null);
    $this->setSubject( subject: null);
    $this->setModel( model: null);
    $this->setYear( year: null);
    $this->setColor( color: null);
    $this->setEngine( engine: null);
    $this->setContent( content: null);
}
}
```

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

COMMIT PROJEKTU DO GIT

Zacommituj 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:

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

PODSUMOWANIE

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

...podsumowanie...

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.