|  |  |  |
| --- | --- | --- |
| **AI1** | Świątek Mateusz  Album 53712  Grupa 1-8 | **Wersja 1** |
|  |
| **LAB F** |

Wprowadzenie do programowania w PHP

# Spis treści

[Spis treści 1](#_Toc147315526)

[Cel zajęć 1](#_Toc147315527)

[Uwaga 1](#_Toc147315528)

[Pobranie i uruchomienie PHP 2](#_Toc147315529)

[Instalacja Node.js i LESS 3](#_Toc147315530)

[Pobranie i konfiguracja frameworka 4](#_Toc147315531)

[XDEBUG 5](#_Toc147315532)

[(Alternatywnie) VS Code 6](#_Toc147315533)

[Zatrzymanie w pułapce 8](#_Toc147315534)

[Omówienie frameworka 8](#_Toc147315535)

[Rozszerzenie aplikacji 8](#_Toc147315536)

[Commit projektu do GIT 11](#_Toc147315537)

[Podsumowanie 11](#_Toc147315538)

# 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](https://windows.php.net/download/);
* nazwa użytkownika: WIAD\ab12345
* komputer: rdp.wi.zut.edu.pl

Odwiedź stronę [https://windows.php.net/download/](https://github.com/inazwisko/ai1-lab/tree/lab-f…). 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:



Zastąp poniższy obrazek swoim zrzutem ekranu:

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

|  |  |  |
| --- | --- | --- |
| 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](https://github.com/IdeaSpotPL/custom-php-framework). 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

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

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

# Pobranie i konfiguracja frameworka

Sklonuj naiwny framework z Githuba: [https://github.com/IdeaSpotPL/custom-php-framework](https://uci.zut.edu.pl/uslugi-uci/vpn.html).

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:

Wejdź przeglądarką na powyższy adres. Framework powinien się wyświetlić:



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

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

Obraz zawierający zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

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

# XDEBUG

Wejdź na stronę [https://xdebug.org/wizard](https://nodejs.org/en/download/current). 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:



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:



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).



|  |  |  |
| --- | --- | --- |
| 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 styli 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):



Wstaw zrzut ekranu listy (z danymi):



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

Wstaw zrzut ekranu kodu podglądu (showAction):



Wstaw zrzut ekranu podglądu:



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

Wstaw zrzut ekranu kodu tworzenia (createAction):



Wstaw zrzut ekranu tworzenia:



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

Wstaw zrzut ekranu kodu edycji (editAction):



Wstaw zrzut ekranu edycji:



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

Wstaw zrzut ekranu kodu kasowania (deleteAction):



|  |  |  |
| --- | --- | --- |
| 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…](https://xdebug.org/wizard)

# Podsumowanie

W kilku zdaniach podsumuj zdobyte podczas tego laboratorium umiejętności.

…podsumowanie…

Zweryfikuj kompletność sprawozdania. Utwórz PDF i wyślij w terminie.