

POLITECHNIKA KRAKOWSKA  
IM. TADEUSZA KOŚCIUSZKI  
WYDZIAŁ FIZYKI MATEMATYKI I INFORMATYKI  
KIERUNEK INFORMATYKA

MIKOŁAJ RAŻNY

**Przygotowanie strony internetowej dla jednostki  
uniwersyteckiej, przy użyciu Systemu Zarządzania  
Treścią(CMS), na przykładzie strony Instytutu  
Fizyki Politechniki Krakowskiej**

**Preparing website for university unit using Content  
Management System(CMS), taking Cracow  
University of Technology Physics Institute website  
as an example**

PRACA INŻYNIERSKA  
STUDIA STACJONARNE

Ocena: .....  
Podpis promotora: .....  
Promotor: dr Radosław Kycia

Kraków 2016



## Spis treści

1. Wstęp.....	5
1.1 Cel pracy.....	5
1.2 Zakres pracy.....	6
2. Systemy zarządzania treścią (ang. Content Management System) oraz wizja tworzenia własnej strony internetowej.....	7
2.1 System Zarządzania Treścią.....	7
2.2 Różne systemy zarządzania treścią.....	9
2.2.1 Drupal.....	9
2.2.2 Joomla!.....	9
2.2.3 Wordpress.....	10
2.2.4 Porównanie.....	11
2.3 Samodzielne tworzenie strony internetowej.....	12
3. Technologie.....	14
3.1 Czemu Wordpress?.....	14
3.2 Przeznaczenie wykorzystanych technologii.....	14
3.3 Linux.....	15
4. Tworzenie strony.....	16
4.1 Wymagania.....	16
4.1.1 Wymagania funkcjonalne.....	16
4.1.2 Wymagania нефункционалне.....	19
4.2 Instalacja Linuxa.....	20
4.3 Instalacja komponentów i wordpressa.....	21
4.3.1 Przygotowanie Linuxa.....	21
4.3.2 Instalacja wordpressa.....	24
4.3.3 Wordpress Multisite.....	26
4.4 Wybór i instalacja motywu.....	27
4.5 Wyszukiwanie, instalacja oraz konfiguracja wtyczek. Wykorzystanie ich na stronie.....	28
4.5.1 Custom Facebook Feed.....	29
4.5.2 Customize Meta Widget.....	29
4.5.3 iThemes Security.....	30
4.5.4 Simple Link List Widget.....	31
4.5.5 Simple Share Buttons Adder.....	31
4.5.6 Simple Staff List.....	32
4.5.7 WP FAQ.....	33
4.5.8 WP Google Maps.....	34
4.5.9 WPGlobus.....	35
4.5.10 Photo Gallery.....	35
4.6 Tworzenie własnych wtyczek.....	36
4.6.1 Tworzenie i instalacja własnych wtyczek.....	36
4.6.2 Praca z bazą danych.....	37
4.6.3 Dodawanie danych na stronę [shortcode].....	38
4.6.4 Wtyczka do zarządzania listą habilitacji i doktoratów (thesis list).....	39
4.6.5 Wtyczka do zarządzania listą zajęć laboratoryjnych (class instruction).....	40
4.8 Dodawanie nowych stron.....	41

4.9 Edycja widgetów.....	41
4.10 Gotowa strona.....	41
4.11 Przeniesienie strony na serwery Politechniki Krakowskiej - wdrożenie.....	44
5. Podsumowanie.....	45
Bibliografia.....	46

# 1. Wstęp

## 1.1 Cel pracy

Celem pracy jest analiza popularnych Systemów Zarządzania Treścią(CMS) oraz, na podstawie zdobytych informacji, stworzenie nowej strony Instytutu Fizyki Wydziału Fizyki, Matematyki i Informatyki Politechniki Krakowskiej. Strona powinna zawierać informacje na temat Instytut Fizyki Politechniki Krakowskiej, a studentom oraz kandydatom na studia ma ułatwić dostęp do poszukiwanych przez nich informacji i dokumentów. Strona powinna również zawierać informacje na temat pracowników instytutu, które mogą oni samodzielnie edytować. Opiekun strony powinien mieć możliwość zarządzania danymi w prosty sposób. Celem jest również stworzenie dokumentacji technicznej i przygotowanie poradnika pomocnego w utrzymaniu strony.

Wykonanie strony ma być oparte o zagadnienia z inżynierii oprogramowania. Tworzenie jest podzielone na zebranie wymagań funkcjonalnych i нефункциональных, wybranie odpowiednich technologii mogących spełnić te wymagania, zaprojektowanie strony i ostatecznie jej wykonanie, przeprowadzane równoległe z testami (jako, że praca jest samodzielna, to testy przeprowadzane są równoległe z tworzeniem). Jest to standardowe podejście przy realizacji komercyjnych projektów.

Dodatkową motywacją tworzenia strony dla Instytutu Fizyki, jest potrzeba uaktualnienia i odświeżenia jej wyglądu. Do tego ważną cechą końcowego produktu ma być relatywnie łatwe zarządzanie stroną – chociażby łatwe dodawanie wpisów dotyczących aktualności związanych z instytutem.

Końcowym rezultatem pracy ma być nowa, oficjalna strona Instytutu Fizyki Politechniki Krakowskiej, dzięki której pracownik za nią odpowiedzialny będzie mógł bezproblemowo, intuicyjnie zmieniać treść dostosowując ją do bieżących potrzeb.

Praca ma charakter aplikacyjny, a jej wynik końcowy – strona Instytutu Fizyki Politechniki Krakowskiej, dostosowana do potrzeb i wymagań pracowników, znacząco usprawni pracę jednostki.

## **1.2 Zakres pracy**

Pracę rozpoczyna opisanie Systemów Zarządzania Treścią zawarte w rozdziale 2. Każdy z nich różni się i może mieć odrobinę inne przeznaczenie. Trzy CMS na których skupia się praca to: Drupal [19], Joomla! [18] oraz Wordpress [17], które obecnie mają największy udział w rynku. Przedstawiona została też wizja tworzenia własnej strony od podstaw – wymagane umiejętności, oraz różnica między taką stroną, a tworzoną dzięki CMS.

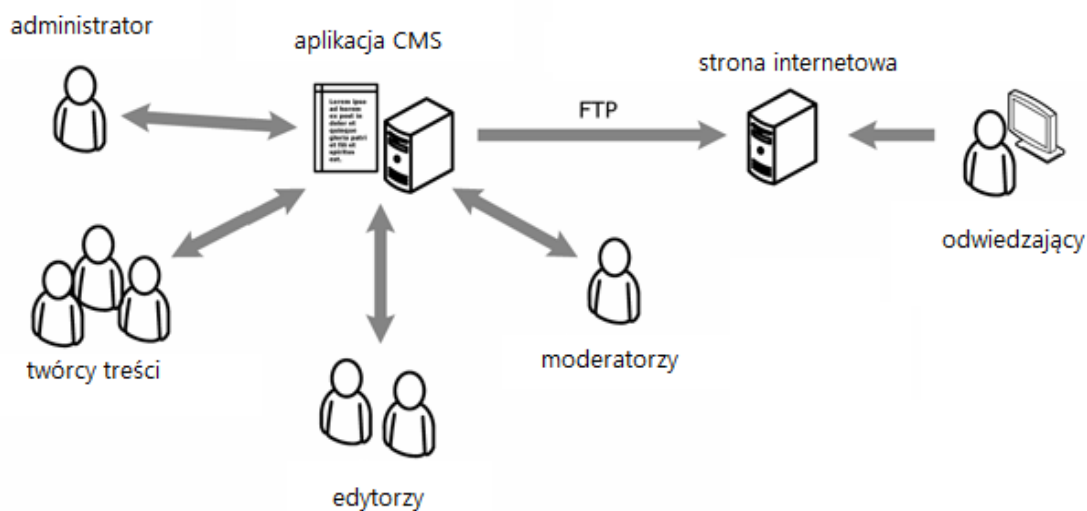
Kolejny rozdział opisuje technologie, z których korzystano przy tworzeniu strony. Krótko opisana została każda z nich. Do tego dodane zostało uzasadnienie wykorzystania właśnie tej technologii.

Rozdział czwarty skupia się na samym tworzeniu strony. Na początku opisane są wymagania, które strona ma w założeniu spełniać, a następnie sam proces. Od utworzenia serwera z systemem Linuxa, poprzez instalację i konfigurację wybranego CMS, wybranie odpowiednich motywów i grafiki, aż po selekcję odpowiednich wtyczek, pozwalających na spełnienie wymagań. Do tego, ze względu na brak odpowiednich wtyczek – stworzenie własnych.

## 2. Systemy zarządzania treścią (ang. Content Management System) oraz wizja tworzenia własnej strony internetowej

### 2.1 System Zarządzania Treścią [1, 2]

System Zarządzania Treścią (ang. Content Management System) jest systemem, który pozwala na uproszczony sposób tworzenia strony internetowej niewymagający znajomości programowania. Systemy CMS działają najczęściej jako skrypty po stronie serwera (obowiązkowo wyposażonego w obsługę współpracującej z wybranym CMS bazy danych), z którymi łączy się poprzez przeglądarkę internetową. Udostępniają one panel administracyjny, dzięki któremu możemy utworzyć stronę oraz skonfigurować wszelkie jej aspekty, dzięki dołączonym modułom oraz formularzom. Dzięki CMS nasza strona może być w łatwy sposób edytowana i uaktualniana. Przykładowe działanie strony opartej o CMS widzimy na obrazku nr 1. Twórcy treści, dzięki CMS tworzą nowe treści. Edytorzy mają dodatkowo prawo je upubliczniać i edytować. Moderatorzy mogą też zarządzać komentarzami odwiedzających stronę, blokować ich konta, zmieniać wygląd strony i zarządzać udostępnionymi przez administratora modułami (pluginy, widgety, motywy). Aplikacja CMS może (choć nie musi) być zainstalowana na osobnym serwerze.



1. Przykładowe działanie strony oparte o CMS [55]

Większość tworzonych obecnie stron wyposażonych jest w taki system – strony tworzone na zamówienie są też zwykle wzbogacane o, chociażby prosty, autorski system pozwalający na zarządzanie zawartością strony lub danymi. CMS oddziela treść strony od

tego, w jaki sposób jest prezentowana. Zawartość strony jest zapisywana w bazie danych, a potem, gdy odwiedzamy stronę CMS generuje ją w oparciu o nasze ustawienia i zapisane dane. CMS zwykle pisane są w ASP, JSP, PHP czy Pythonie.

Większość CMS to rozwiązania modułowe - do naszego systemu sami możemy dodawać kolejne elementy zwane wtyczkami (ang. plugin), które wzbogacają funkcjonalność naszej witryny, ułatwiają administrację, czy nawet poprawiają bezpieczeństwo.

CMS mogą być tworzone na zamówienie i udostępniane konkretnemu klientowi. Mają one tę wadę, że dodatki do nich (wtyczki) tworzy zwykle tylko i wyłącznie producent oprogramowania. Jeśli klient będzie chciał zwiększyć swoje możliwości zarządzania, czy funkcjonalność strony musi się skontaktować z producentem. Do tego, CMS o zamkniętym kodzie źródłowym mogą zawierać dużo więcej błędów. Jest to spowodowane tym, że kod udostępniony jest tylko klientom. Szansa na to, że błąd zostanie wykryty i naprawiony jest mniejsza. Z tego powodu wrażliwość takiego rozwiązania na ataki jest zwiększona.

Druga grupa CMS, to te z otwartym kodem źródłowym (ang. open source). Kod udostępniany jest każdemu, a producent zwykle nie pobiera opłat za używanie jego produktu. Gdy taki system jest wystarczająco dobry, zyskuje popularność, a co za tym idzie – zwiększa się liczba dodatków pisanych przez użytkowników. Jako, że użytkownicy znają kod źródłowy mogą sami tworzyć nowe wtyczki i udostępniać je innym. Chociaż większość wtyczek jest darmowa, to częstym zjawiskiem jest tworzenie wtyczki i zezwolenie na korzystanie tylko z części jej funkcjonalności. Dodatkową twórcą udostępnia za opłatą. Oprócz wtyczek społeczności skupione wokół danego CMS tworzą też nowe szaty graficzne i inne dodatki (np. widżety – małe dodatki na stronie, wyświetlające kalendarz, zawierające przyciski zmiany języka, przydatne linki itp., które tworzą się na każdej z podstron i ciągle są dostępne dla użytkownika). Systemy otwarte są często aktualizowane. Ze względu na wgląd do kodu można łatwiej znaleźć błędy w oprogramowaniu, które jednak dość szybko są odkrywane i publikowane. Twórcy CMS naprawiają błąd i proszą użytkownika o aktualizację. W związku z tym regularne sprawdzanie, czy nasza wersja jest najnowsza ma bardzo duże znaczenie, jako prewencja przed atakami na naszą stronę.



Ponadto istnieją CMS „hostowane”. Są to systemy zainstalowane na serwerach producenta. Kod nie jest nam znany – dostajemy tylko możliwość wykorzystywania dostarczonego narzędzia. Zwykle nie ma możliwości dodania swoich wtyczek, a funkcjonalność jest bardzo ograniczona. Dodatkowo, usługodawca często wyświetla swoje reklamy, gdy strona jest wyświetlana. Przykładem takiego systemu jest edito

## 2.2 Różne systemy zarządzania treścią

### 2.2.1 Drupal [4, 19]

Drupal reklamuje się jako system wysokiej wydajności i bezpieczeństwa. System rozszerzamy o dodatkowe funkcjonalności poprzez instalację modułów (ma prawie 33 tysiące dodatków – stan na 13 I 16r [19]). Nie ogranicza nas żadna liczba działających jednocześnie wtyczek. Ten CMS jest dystrybuowany na licencji GNU [22] – jego użytkowanie nie wymaga żadnych opłat i możemy go bez ograniczeń edytować. Wokół Drupalu skupionych jest wiele pasjonatów. Jest to system popularny na całym świecie – dostępnych jest w ponad 100 wersjach językowych (stan na 13 I 16r).



logo Drupala [19]

### 2.2.2 Joomla! [3, 18]

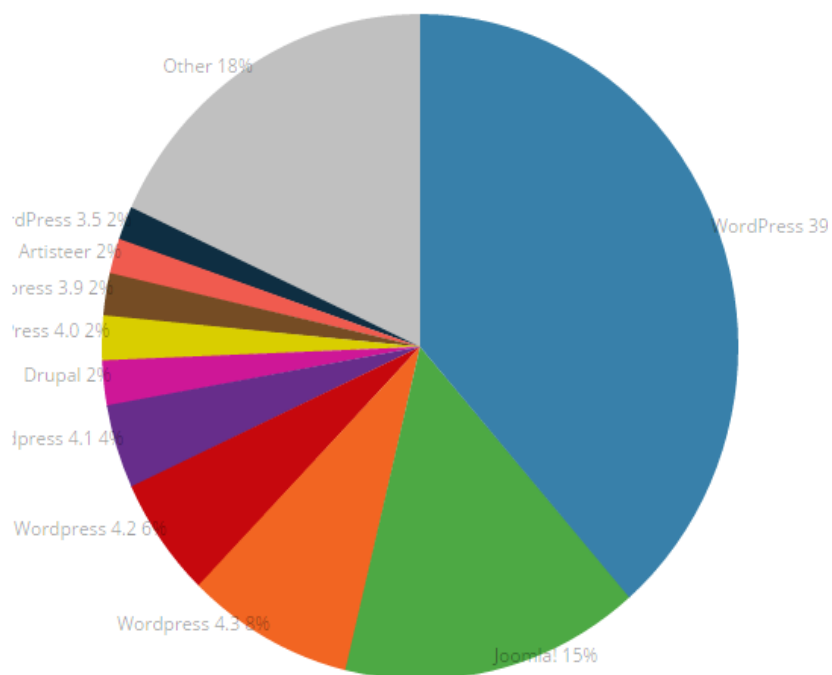
Joomla! to CMS dość popularny w Polsce (zob. rysunek nr 2). Możliwości są bardzo podobne do Wordpressa. Joomla! świetnie nadaje się do bardziej zaawansowanych stron z rozbudowanymi bazami danych.



Joomla!®

logo Joomla! [18]

Joomla! ma mniej wbudowanych funkcji, a dostosowuje się ją do swoich potrzeb używając wtyczek. Nie ma tak przyjaznego i intuicyjnego panelu, jak Wordpress – może być trudniejsza w obsłudze dla osoby bez doświadczenia z systemami CMS. Posiada ponad 7500 wtyczek (stan na 13 I 16r [18])



2. Wykorzystywane w Polsce systemy CMS [6]

### 2.2.3 Wordpress [1, 7, 17]

Jest to najpopularniejszy obecnie CMS. CMS jest dystrybuowany na licencji GNU [22]. Oferuje ponad 42 tysięcy wtyczek, oraz ponad 1500 szablonów graficznych (istnieje także oprogramowanie do tworzenia własnego wyglądu strony). Już bez instalacji wtyczek nasza możliwość konfiguracji jest wystarczająca do prowadzenia swojego bloga. Zarządzanie jest bardzo intuicyjne i przejrzyste.



logo Wordpressa [17]

Popularność systemu wiąże się z dużą społecznością moderską (wiele wspierających się nawzajem twórców wtyczek i motywów graficznych), a co za tym idzie – szerokim wsparciem dla developerów i opisanym na forach internetowych sposobem rozwiązania większości problemów, z którymi borykają się początkujący.

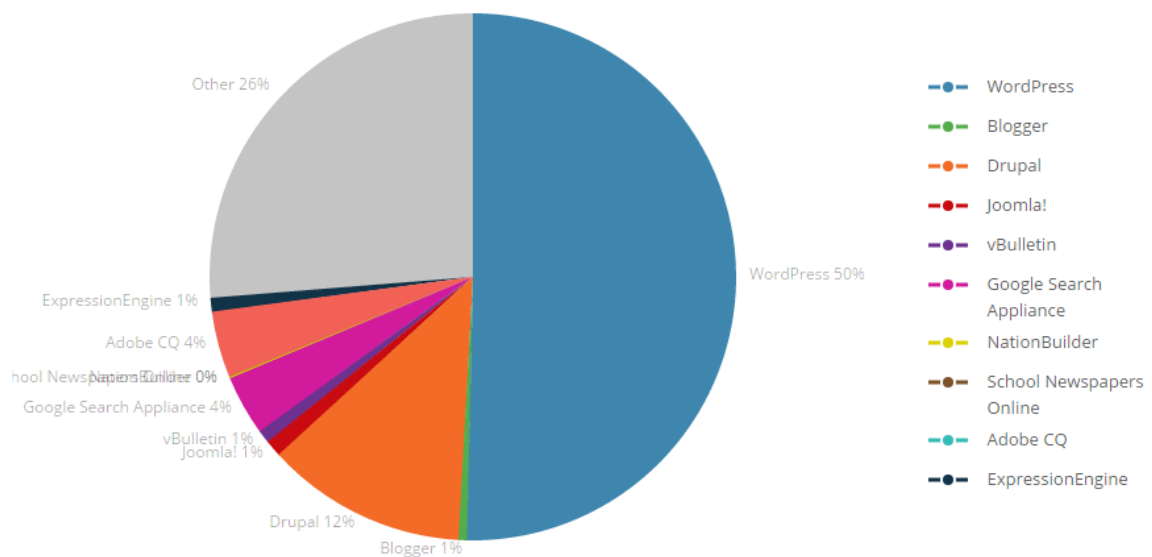
Wordpress jest bardzo popularny do prowadzenia blogów. Do tego strony wordpressa są wyżej notowane w wyszukiwarkach.

### 2.2.4 Porównanie [5, 6]

Każdy z tych systemów ma swoje wady i zalety, jednak w każdym z nich jesteśmy, przy odpowiednim nakładzie pracy i determinacji, stworzyć taką samą stronę. Kwestią decydującą mogą być nasze osobiste sympatie, jednak powinniśmy się też kierować:

- popularnością systemu
- częstotliwością aktualizacji
- bogactwem dodatków
- społecznością

Systemy tracące popularność lub niszowe w niedalekiej przyszłości mogą przestać się rozwijać, co będzie prowadzić do spadku jakości naszej strony w porównaniu do innych, a także zmniejszy bezpieczeństwo.



### 3. Wykorzystywane na świecie CMS (pierwszy tydzień 2016r) [6]

Jak pokazuje wykres zamieszczony na ilustracji nr 3, Wordpress jest zdecydowanie najpopularniejszym systemem zarządzania treścią (stan na 13 I 16r). W Polsce popularny jest też system Joomla!, który ma u nas 15% udziałów w rynku (zob. rysunek nr 2). Na świecie natomiast drugim najpopularniejszym systemem jest Drupal z 12% wynikiem. Joomla! oraz Blogger na światowym rynku mają po ok. 1%.

Duża popularność przekłada się także na ilość dodatków, których będziemy mogli użyć na naszej stronie. Zmniejsza to koszty dostosowania strony do naszych potrzeb. Wordpress posiada ponad 35 tysięcy dodatków, podczas gdy Drupal oferuje ich ok. 16 tysięcy, a Joomla! 8. Niezbyt popularne systemy mają niewiele dodatków, a nowe trzeba programować samemu, albo zamawiać u developerów. Warto też zaznaczyć, że popularność Drupala i Joomla! maleje, podczas gdy społeczność Wordpressa ciągle się zwiększa.

## **2.3 Samodzielne tworzenie strony internetowej**

Podstawą każdej strony internetowej jest język HTML (ang. HyperText Markup Language) [23]. Język ten opiera się na systemie znaczników określających układ strony internetowej. Został stworzony w CERNie w 1980r przez fizyka Tim Berners-Lee [24]. Najnowszą wersją standardu jest 2.0. Niestety, obecnie do stworzenia estetycznie wyglądającej strony sam HTML nie jest wystarczający. W celu tworzenia prezentacyjnego aspektu strony należy użyć CSS (ang. Cascading Style Sheets) [25]. Połączenie tych dwóch technologii wystarcza do stworzenia statycznej strony.

Kaskadowe arkusze stylów to język do definiowania sposobu prezentacji danych na stronach WWW. Dzięki niemu możemy modyfikować wygląd i położenie elementów HTML.

W celu dodania dynamiki stosuje się dodatkowe technologie. Najczęściej są to języki programowania działające po stronie klienta lub serwera. W rodzinie tych pierwszych najpopularniejszym jest język JavaScript [26], który jest interpretowany przez wszystkie przeglądarki internetowe (każda przeglądarka posiada swój własny silnik JS, nie ma więc wspólnej wersji dla każdej z nich, ale jest to wspólny standard języka JS, więc te silniki dają podobne wyniki).

JS to skryptowy język programowania. Przy tworzeniu stron WWW wykorzystywany jest do interakcji użytkownika ze stroną bez przeładowania jej zawartości.

Po stronie serwera najczęściej wykorzystywany jest język PHP, a także coraz częściej Python i Ruby. Ze względu na dużą popularność, skupimy się na języku PHP [27]. Jest to język przypominający C/C++, udostępniany na licencji PHP [28]. Umożliwia on

stworzenie witryn, które np. kontaktują się z bazą danych, autoryzują użytkowników strony lub przesyłają pliki/informacje. To podejście sprawia, że nasze możliwości są nieograniczone.

Jedną z najczęściej wykorzystywanych baz danych w CMS jest MySQL [29]. Jest to najpopularniejsza na świecie relacyjna baza danych na licencji Open. Jest efektywna i skalowalna. Językiem, który służy do zarządzania relacyjnymi bazami danych, jest SQL [29]. Język ten jest przykładem języka deklaratywnego (w odróżnieniu od paradygmatu imperatywnego) – programista określa co chce otrzymać nie podając algorytmu.

Jeśli zdecydujemy się na tworzenie strony wraz z innymi osobami lub chcemy mieć narzędzie do archiwizacji, dzięki któremu będziemy mieli szybki dostęp do poprzednich wersji naszej pracy, możemy wykorzystać jeden z systemów kontroli wersji. Jednym z najpopularniejszych jest Git [30]. Jest to system rozproszony, ale może też działać w architekturze klient-serwer. Jego twórcą jest Linux Torvalds, twórca jądra Linuxa. Obecnie często jest używany do pracy nad programami Open Source. Ze względu na stabilność i brak opłat wzrasta również jego użycie w zastosowaniach komercyjnych. Istnieje też wiele darmowych serwisów hostujących projekty pod kontrolą tego systemu [31][32].

Inną drogą jest zdecydowanie się na skorzystanie z platform programistycznych, czy frameworków, które udostępniają gotowe, często złożone komponenty. Można je traktować jako zasobniki gotowych elementów, z których możemy złożyć stronę. Mimo ułatwionej pracy, ciągle wymagana jest znajomość HTML, CSS, JS.

Jeśli dysponujemy odpowiednią wiedzą z tego zakresu możemy zdecydować się na samodzielne wykonanie strony internetowej. Jeśli nasze potrzeby są specyficzne, niemożliwe do zrealizowania jako samodzielna wtyczka do CMS lub naszym celem jest udostępnienie usługi poprzez przeglądarkę internetową, to jest to czasami jedyne wyjście. Nie dotyczą nas żadne ograniczenia narzucane przez producentów CMS, a nasza strona jest tak „lekka”, jak tylko możliwe – możemy udostępnić tylko to, co w rzeczywistości jest funkcjonalne i użyteczne, co oznacza dla użytkownika mniejsze wykorzystanie łącza internetowego ergo szybszy dostęp do strony.

## **3. Technologie**

### **3.1 Czemu Wordpress?**

Do realizacji projektu wybrałem Wordpresa z następujących powodów:

- wersja polskojęzyczna
- najpopularniejszy z dostępnych CMS
- ciągle aktualizowany
- posiada funkcję multisite, która pozwala na zakładanie strony każdemu z pracowników. Użytkownik z własną stroną może zamieszczać na niej swoją treść (zadania laboratoryjne, informacje kontaktowe)
- wiele dostępnych wtyczek i motywów graficznych
- pisanie wtyczek jest ułatwione dzięki bogatej dokumentacji i licznej społeczności developerskiej
- otwarty kod zapewnia naprawy zgłaszanych błędów i bezpieczeństwo
- współpraca z systemem Linux i popularnymi serwerami
- łatwy w obsłudze nawet dla laika
- uprzednie doświadczenie z systemem

### **3.2 Przeznaczenie wykorzystanych technologii**

Wykonując projekt (szczególnie tworząc wtyczki), wykorzystaliśmy następujące technologie:

- HTML - wykorzystany w projekcie przy tworzeniu pluginów: składają się z niego wyświetlane formularze i tabele
- CSS - wykorzystano przy edycji jednej z zainstalowanych wtyczek wordpresa, oraz przy tworzeniu własnych pluginów. Skorzystano też częściowo z bibliotek Wordpresa, by dostosować wygląd tabel w panelu administratora do reszty layoutu strony administracyjnej

- JS - wykorzystany został do walidacji danych wpisywanych przez administratora do formularzy. Ukrywa także i pokazuje odpowiednie dane, na podstawie zachowań użytkownika podczas przeglądania instrukcji laboratoryjnych. By ułatwić pisanie pluginów wykorzystano też bibliotekę jQuery. Ułatwia ona odwołania do konkretnych bloków strony, a także jest wyposażona w funkcje estetycznie zmieniające wygląd strony (np. powolne znikanie/pojawianie się danego elementu)
- PHP - używany przez Wordpressa. W projekcie został wykorzystany do komunikacji z bazą danych przy tworzeniu pluginów
- MySQL – służy do obsługi relacyjnej bazy danych, wykorzystywanej przy tworzeniu wtyczek
- Git – w projekcie Git jest używany jako narzędzie do archiwizowania poprzednich wersji strony. Gdyby po jakiejś aktualizacji strona przestała działać, lub pojawiłyby się błędy, to posłuży do przywrócenia poprzedniej wersji strony. Dzięki niemu też przeniesiono stworzoną na wirtualnej maszynie stronę na serwery politechniki

### **3.3 Linux**

Linux to rodzina systemów opartych na jądrze Linux. Kod źródłowy został upubliczniony i obecnie jest modyfikowane przez różne grupy, tworzące jego różnorakie wersje. Linux to systemy znane z niezawodności i bezawaryjności. Wykorzystywany często jako system operacyjny dla serwerów, ze względu na łatwość konfiguracji, niezawodność oraz dostępność aktualizacji. Obecnie liczba systemów linuxowych jest bardzo duża i ciągle rośnie [33].

Wykorzystywany w projekcie ze względu na to, że z tego systemu korzysta serwer Instytutu Fizyki PK, na którym strona ma docelowo działać. Wybrano wersję Ubuntu [34] charakteryzującą się wygodnym interfejsem graficznym. Jest to wersja stabilna i niezawodna. Jeszcze do niedawna był najpopularniejszym Linuxem wykorzystywanym przez osoby prywatne.

## **4. Tworzenie strony**

### **4.1 Wymagania**

Określenie wymagań to podstawowe zadanie stojące przed twórcą lub zleceniodawcą. Dzięki sprecyzowaniu potrzeb możemy zbudować aplikację. Gdy tworzy się na własny użytek, lista wymagań pomaga w organizacji pracy – nie musimy się martwić o tym, że jakiś mniej znaczące pomysły nie zostaną zrealizowane. Gdy zleca się pracę, wymagania mogą stanowić kryterium oceny wykonania zlecenia. Zapewniają też, że developer będzie mógł wywiązać się z powierzonych zadań.

Wymagania funkcjonalne określają, jak ma działać strona internetowa. Wymagania нефункционалне to przedstawienie założeń, które nie są konkretnymi potrzebami, a ogólnymi wymaganiami dotyczącymi pracy – określają jak wymagania funkcjonalne mają być spełnione i obrazują intencje klienta.

#### ***4.1.1 Wymagania funkcjonalne***

W związku z tym, że sam CMS realizuje wiele z podstawowych założeń (takich jak logowanie, czy też edycja zawartości), wymienione zostały tylko wymagania, które zrealizować należy już tworząc stronę na wybranym CMS.



<b>ID</b>	<b>NAZWA</b>	<b>OPIS</b>
W01	Dodawanie użytkowników	Należy rozróżnić administratora serwera od administratora strony/treści. Jedynie administrator strony może dodawać nowych użytkowników i nadzoruje treści na stronie
W02	Profil użytkownika	Zmiana danych (tytuł, telefon, adres email, konsultacje, pokój). Wprowadzaniem danych zajmuje się administrator
W03	Strona użytkownika	Dodany przez administratora użytkownik może samodzielnie tworzyć własną stronę w obrębie CMS. Należy udostępnić publicznie link do strony prywatnej użytkownika. Użytkownik może dodawać swoje własne pliki i umieszczać je na swojej stronie
W04	Strony laboratoriów	Zakładki zawierające informacje o zasadach BHB, regulaminie laboratoriów, a także wykaz ćwiczeń. Każde ćwiczenie powinno móc być wyposażone w stosowny opis, mogący składać się z tytułu i numeru ćwiczenia, instrukcji wykonania zadania, sali, w której odbywają się zajęcia, a także załączonej tabeli stałych. Wymaganie może być zrealizowane jako część CMS, albo strona zewnętrzna w stosunku do CMS
W05	Lista pracowników	Lista pracowników instytutu z podziałem na aktualnych i emerytowanych pracowników
W06	Lista publikacji	Lista publikacji pracowników instytutu
W07	Aktualności	Dział "wydarzenia" na stronie głównej. W dziale tym pojawiałyby się informacje o planowanych konferencjach, warsztatach, wydarzeniach. Dobrze, aby te informacje można było importować/eksportować do oficjalnych profili na portalach społecznościowych

W08	Lista doktoratów	Lista doktoratów. Dodawanie nowych pozycji do listy z poziomu administratora. Lista ma zawierać imię i nazwisko osoby wykonującej pracę, tytuł pracy, datę publikacji, uczelnię, promotora
W09	Lista habilitacji	Lista podobna do przedstawionej w W08. Różni się tym, że nie jest wymagane podanie promotora
W10	Katedry	Lista przedstawiający członków dwóch katedr: Fizyki Materiałów i Struktury Materii. Realizacja podobna do W05, jednak powinna być też możliwość oznaczenia osoby jako kierownika katedry
W11	Galeria	Możliwość udostępniania przez administratora zdjęć wraz z opisem. Galeria jest dostępna do wglądu publicznie
W12	Kontakt	Udostępnienie danych teleadresowych do Instytutu oraz mapę z zaznaczonym wydziałem. Musi być to opcja bardzo łatwo dostępna z poziomu strony głównej. Warto również dodać informacje na temat oficjalnych danych dotyczących rozliczania delegacji
W13	Podstrona informacyjna	Strona informacyjna z nazwiskami i kontaktem do osób odpowiedzialnych za instytut (dyrektor, zastępcy, kierownicy). Strona powinna zawierać także opis instytutu, jego historię, link do sylabusu, koła naukowe, link do Wikipedii na temat Pałacu w Łobzowie tj. siedziby Wydziału Fizyki, Matematyki i Informatyki
W14	Inne podstrony	Strona z wykazem demonstracji do wykładów, opis użytecznych programów komputerowych
W15	Strona główna	Nagłówek ze zmienianym przez admina banerem Linki do stron: Politechniki Krakowskiej, Wydziału FMiI oraz poczty SquirrelMail Przycisk zmiany języka

		Tekstowa (kopiowalna) nazwa instytutu w języku polskim i angielskim
W16	Lokalizacja strony	Możliwość tłumaczenia zawartości strony na inne języki. Wymagany językiem jest angielski. Możliwość obsługi innych języków
W17	FAQ	Podstrona zawierająca najczęściej zadawane pytania
W18	Integracja z portalem Facebook	Instytut Fizyki nie ma własnej strony facebook - warto taką założyć i sprząc ze stroną internetową, tak, aby publikacja informacji na jednej skutkowała aktualizacją na drugiej
W19	Zadbanie o bezpieczeństwo i zabezpieczenie przed spamem	Sprawdzenie możliwości zabezpieczenia adresów email podawanych na stronie przed spamem. Zabezpieczenie komentarzy przed spamem. Zabezpieczenie strony przed włamaniami
W20	Zabezpieczenie przed awarią	Archiwizacja starszych wersji strony, możliwość powrotu do starszej wersji w przypadku awarii po aktualizacji lub innych problemów ze stroną

Tabela 1. Wymagania funkcjonalne

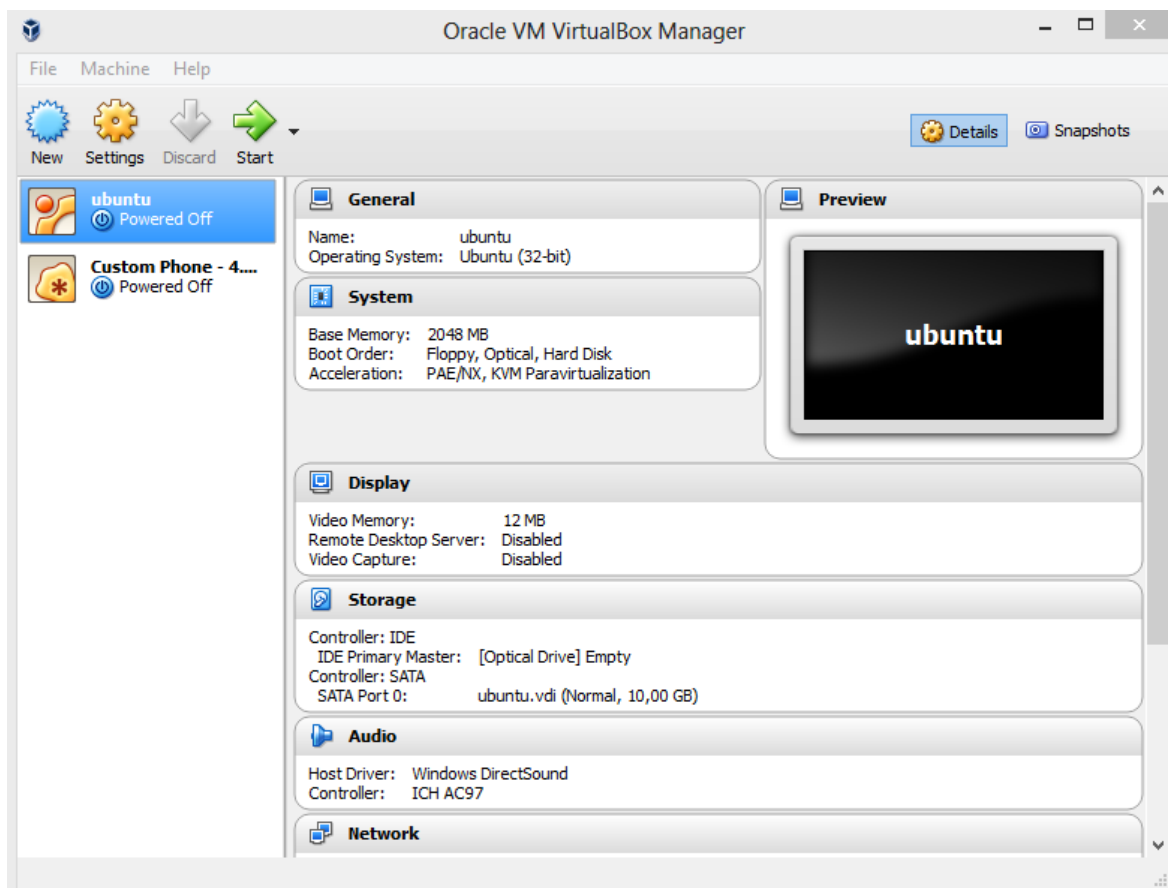
#### **4.1.2 Wymagania niefunkcjonalne**

- Strona łatwa w obsłudze dla osoby z niskim doświadczeniem w korzystaniu z technologii informacyjnych (w tym CMS). Osoba zajmująca się stroną nie powinna mieć problemów z obsługą, ewentualne trudności powinny być rozwiązane poprzez skorzystanie z dołączonej do strony instrukcji, zawierającej opis podstawowych działań
- Przejrzystość i prostota strony – użytkownik (osoba przeglądająca stronę) nie powinien mieć najmniejszych problemów ze znalezieniem interesujących go informacji

- Bezproblemowe działanie na sprzęcie udostępnionym przez Politechnikę Krakowską. Działanie strony na popularnych przeglądarkach: Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Safari
- Technologie używane do realizacji mają być technologiami rozwijającymi się. Nie można dopuścić do sytuacji, w której dana technologia przestanie być aktualizowana
- Technologie i oprogramowanie mają być darmowe, na licencji „Open” (GNU, BSD, itp.)

## **4.2 Instalacja Linuxa**

System postawiono przy użyciu programu VirtualBox od Oracle [35]. Po ściągnięciu najnowszego obrazu systemu Ubuntu w wersji 32-bitowej, montujemy go w programie. Wirtualna maszyna daje możliwość stworzenia wirtualnego komputera z zadanymi parametrami (wielkość pamięci RAM, pojemność wirtualnego dysku twardego), umożliwiającego instalację dowolnego systemu operacyjnego. Przykładowy wygląd VB przedstawiony jest na rysunku nr 4.



4. Program VirtualBox z zamontowanym systemem Ubuntu

## 4.3 Instalacja komponentów i wordpressa

By móc korzystać z CMS należy uprzednio skonfigurować odpowiednio system.

### 4.3.1 Przygotowanie Linuxa [11]

Pierwszym krokiem, jakim musimy wykonać przed konfiguracją systemu jest zalogowanie się jako użytkownik z prawami administratora (*root*) lub wykorzystanie polecenia *sudo* przed podaniem komendy w konsoli. Jako, że wykorzystywany przez nas system (Ubuntu) ma zablokowane konto *root*, to przed każdym poleceniem dopiszemy polecenie *sudo*. Polecenia wpisywane w konsoli oznaczone są czcionką Courier New.

Do poprawnego działania Wordpress wymaga LAMP/LEMP (Linux, Apache MariaDB/MySQL, Perl/PHP/Python/Primate) [36], czyli zestawu oprogramowania open source, będącą podstawą serwera dla dynamicznych stron WWW. Składa się on z systemu Linux (z którego już korzystamy), serwera Apache, serwera bazy danych i języka

skryptowego. Jest to jeden z najpopularniejszych zestawów oprogramowania dla serwerów hostujących strony WWW.

By zainstalować powyższy zestaw oprogramowania upewniamy się, że nasza wersja systemu jest aktualna i ewentualnie ją aktualizujemy. Służą do tego polecenia

```
sudo apt-get update  
sudo apt-get dist-upgrade  
sudo apt-get autoremove
```

Możliwe, że po wykonaniu się poleceń będzie wymagane ponowne uruchomienie systemu.

```

SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

Set root password? [Y/n] Y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] Y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] Y
... Success!

By default, MySQL comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] Y
- Dropping test database...
ERROR 1008 (HY000) at line 1: Can't drop database 'test'; database doesn't exist
... Failed! Not critical, keep moving...
- Removing privileges on test database...
... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] Y
... Success!

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!

Cleaning up...

```

5. Wykonanie programu `mysql_secure_installation` z wybraniem odpowiednich opcji

Następnym krokiem jest instalacja serwera Apache2 [37] poleceniem

```
sudo apt-get install apache2
```

a następnie bazy MySQL poleceniem

```
sudo apt-get install mysql-server mysql-client
```

Dodatkową czynnością, którą możemy wykonać jest wykorzystanie programu *mysql\_secure\_installation* pozwalającego nam na ustawienie hasła dla użytkownika *root*, usunięcie kont anonimowych i bazy testowej. Program ten usuwa też przywileje użytkowników z przedrostkiem *test\_*. Kolejny zainstalowanym oprogramowaniem jest PHP5:

```
sudo apt-get install php5 libapache2-mod-php5
```

oraz jego moduły rozszerzające:

```
sudo apt-get install php5-mysql php5-curl php5-gd php5-intl php-pear php5-imagick php5-imap php5-mcrypt php5-memcache php5-ming php5-ps php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xsl
```

Moduły te są wymagane do poprawnego działania Wordpressa.

### ***4.3.2 Instalacja wordpressa [11]***

Gdy system ma już zainstalowane odpowiednie moduły możemy przejść do pobrania i wypakowania plików Wordpressa ze strony [38]. Pobieramy je do folderu */tmp/* z oficjalnej strony Wordpressa:

```
cd /tmp/ && wget http://wordpress.org/latest.tar.gz
```

i wypakowujemy:

```
tar -xvzf latest.tar.gz
```

Folder */tmp/* to folder do przechowywania danych tymczasowych i nie należy tam przechowywać danych przez dłuższy czas. Często jest czyszczony przy wyłączenia systemu ze względów bezpieczeństwa. Używamy go tylko dlatego, że pobrane pliki po wypakowaniu nie będą nam już potrzebne.

Domyślnym folderem Apache2 w systemach Ubuntu jest */var/www/html*. Kopiujemy do tego folderu wypakowaną zawartość będąc w folderze */tmp/*:



```
sudo mv wordpress/* /var/www/html/
```

Niezwykle istotną czynnością, która przez niedoświadczone osoby nie jest wykonana jest skasowanie testowej strony *index.html*, tworzącej się domyślnie w folderze roota. Jeśli tego nie zrobimy, to wpisując adres lokalny w wyszukiwarce internetowej wyświetli się nam ona, a nie strona Wordpressa.

Po wykonaniu powyższych czynności musimy także stworzyć bazę danych, aby Wordpress mógł przechowywać swoje dane. Najpierw łączymy się z bazą używając podanego wcześniej hasła:

```
mysql -u root -p
```

tworzymy bazę:

```
CREATE DATABASE wpdb;
```

kolejnie tworzymy nowego użytkownika i definiujemy dla niego nowe hasło:

```
CREATE USER wpuser@localhost IDENTIFIED BY  
'new_password_here';
```

Dajemy naszemu nowemu użytkownikowi pełny dostęp do stworzonej bazy:

```
GRANT ALL ON wpdb.* to wpuser@localhost;
```

a na sam koniec odświeżamy przywileje bazy i kończymy połączenie z bazą poleceniem

```
exit
```

Do połączenia z bazą Wordpress będzie potrzebował odpowiednich danych: nazwy użytkownika oraz hasła, a także nazwy bazy danych. Będą one odczytywane z pliku konfiguracyjnego. Tworzymy plik *wp-config.php* w katalogu */var/www/html/* i edytujemy go dowolnym edytorem tekstowym:

```
sudo gedit /var/www/html/wp-config.php
```

W pliku muszą znaleźć się następujące definicje:

- `define('DB_NAME', 'wpdb');`
- `define('DB_USER', 'wpuser');`

- `define('DB_PASSWORD', 'hasło_do_bazy_danych');`

Na koniec zmieniamy uprawnienia do folderu html:

```
sudo chown -R www-data:www-data /var/www/html/ oraz sudo
chmod -R 755 /var/www/html/
```

i restartujemy serwer Apache2:

```
sudo service apache2 restart
```

Wpisując adres `/localhost/` w oknie przeglądarki dostaniemy się do naszej strony postawionej na CMS Wordpress, gdzie skończymy konfigurację strony zgodnie z wyświetlanymi instrukcjami (tworzenie konta admina, wybór języka, itp.). Logując się przy pomocy wypełnionych danych na konto z prawami administratora mamy dostęp do naszej strony i możemy ją dowolnie modyfikować.

### **4.3.3 Wordpress Multisite [12]**

By spełnić wymagania z tabeli nr 1, musimy zezwolić rejestrowanym użytkownikom na tworzenie własnych stron. Służy do tego tryb multisite. Użytkownicy z takimi stronami dostają swoją własną stronę w wordpresie. Pomędzy stronami będą dzielone te same wtyczki i motywy graficzne, jednak każdy z użytkowników otrzyma swój własny folder i osobne tabele w bazie danych.

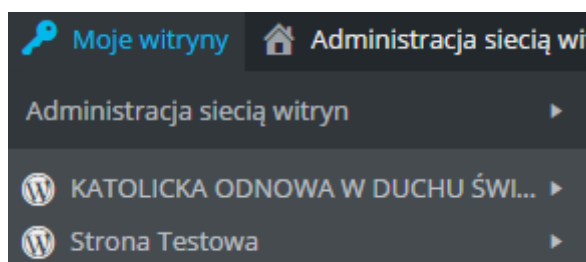
Przed instalacją tego modułu, jeśli strona nie jest nowo stworzona, zaleca się utworzenie kopii zapasowej bazy i danych, czyli plików, które w przypadku awarii zostaną użyte do przywrócenia poprzedniego stanu.

Pierwszym krokiem jest dodanie definicji

```
define('WP_ALLOW_MULTISITE', true);
```

w naszym pliku konfiguracyjnym (*wp-config.php*). To działanie udostępni nam opcję *Ustawienia sieci* (dostępne w stronie administracyjnej, po zalogowaniu na konto administratora) pod opcją menu *Ustawienia* – ustawiamy tam tytuł sieci stron, oraz wpisujemy adres email administratora. Opcja w tym samym podmenu *Uruchamianie sieci witryn* zawiera instrukcje uruchomienia sieci. Będziemy musieli edytować nasze pliki: *wp-config.php* oraz *.htaccess* (jeśli plik nie istnieje, należy go stworzyć). Zgodnie

z instrukcjami w odpowiednie miejsce pierwszego pliku wklejamy definicje uruchamiające działanie Wordpress Multisite, a do drugiego dodajemy podane reguły modyfikujące adresu URL.



6. Nowe opcje menu po aktywowaniu Wordpress Multisite

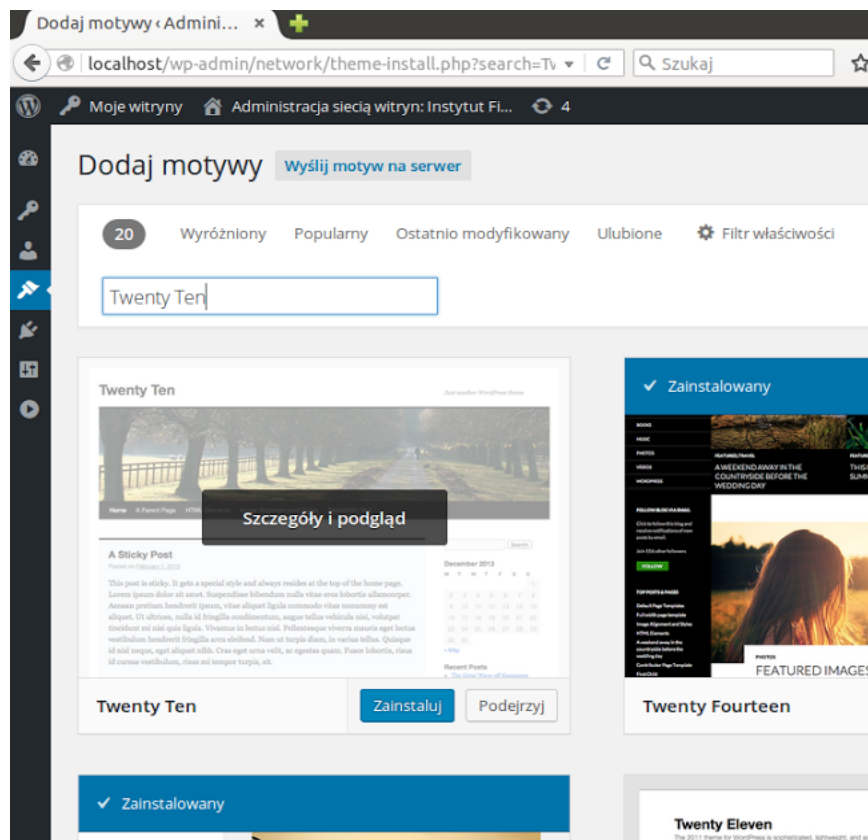
Po odświeżeniu strony na pasku narzędzi (u góry strony – zob. rysunek nr 6), pojawi się opcja pozwalająca na przejście do administracji siecią witryn (gdzie można dodawać nowe wtyczki, motywy graficzne, widżety itp.) i do konkretnych stron.

#### 4.4 Wybór i instalacja motywu

Przed instalacją i konfiguracją wtyczek możemy zadbać o to, żeby nasza strona wyglądała tak, jak chcemy. Wordpress udostępnia wiele motywów – lwią część z nich jest tworzona przez społeczność [39], a część to autorskie rozwiązania Wordpressa (je także trzeba zainstalować – domyślnie zainstalowany jest jeden motyw). Do projektu wybrano właśnie motyw z tej drugiej części – zapewni to, że motyw będzie wykonany w pełni profesjonalnie.

Jako, że działa już Wordpress Multisite, nowe motywy dodajemy z panelu *Administracji sieci witryn*. Wyszukujemy ten, który nas interesuje i instalujemy go (zob. rysunek nr 7). Każdy z motywów możemy podglądać na przykładowej stronie, dzięki czemu nie musimy instalować każdego motywu, aby wiedzieć, czy spełnia on nasze oczekiwania.

Wybrany motyw *Twenty Ten* jest przejrzysty i oszczędny. Wydaje się elegancki, w ciągu dalszego dopasowania strony zlikwidujemy też niepotrzebne elementy.

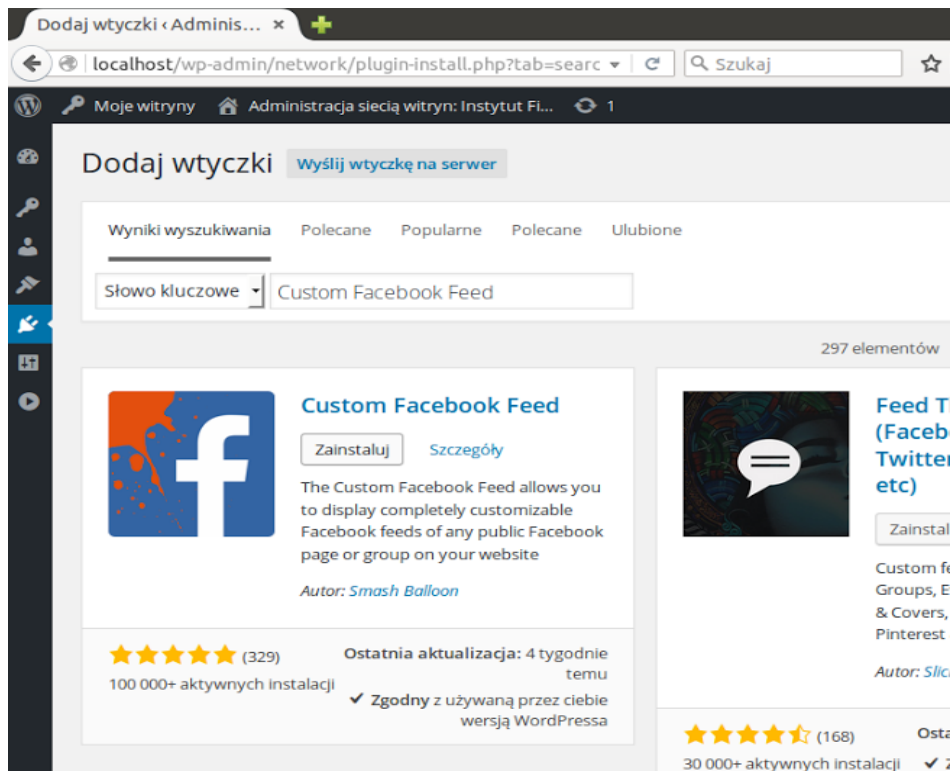


7. Wyszukiwanie i instalacja motywu

#### 4.5 Wyszukiwanie, instalacja oraz konfiguracja wtyczek. Wykorzystanie ich na stronie

Wyszukiwanie wtyczek (ang. plugin), podobnie jak motywów jest dostępne z poziomu zarządzania siecią. Jako administrator instalujemy wtyczki (zob. rysunek 8), a następnie możemy je aktywować w kolejnym podmenu.

Każda wtyczka ma swój opis, oceny społeczności i opcjonalnie zrzuty ekranu pokazujące jej działania, dzięki któremu widzimy, jak działa lub wygląda plugin.



8. Wyszukiwanie i instalacja wtyczki

#### 4.5.1 Custom Facebook Feed [40]

Wtyczka pozwala na integrację strony/wpisu w CMS z daną grupą/stroną na portalu Facebook. Po instalacji i aktywacji wtyczki możemy dodać Facebook ID w odpowiednie pole, a następnie używając shortcode `[custom-facebook-feed]` na danej stronie, udostępnić content z facebooka.

Dodatkowo wtyczka oferuje ustawienia, dzięki którym zdefiniujemy ile wpisów z facebooka ma być wyświetlanych i jak często mają być aktualizowane.

Oprócz tego możemy dopasowywać wygląd do naszych potrzeb: wielkość okna z wpisami, odstępy, wygląd tekstu, ikony itd., a w wersji PRO także jeden z trzech layoutów.

#### 4.5.2 Customize Meta Widget [41]

Prosty plugin umożliwiający edycję domyślnie używanego widgetu Meta, który domyślnie pokazuje odnośniki do stron logowania i strony z szybkim podglądem nowych wpisów ze strony. Udostępnia kod tego widgetu i dostarcza funkcji zastępujący go poprzez

nasz kod. Możemy dzięki niemu usunąć linki, które nas nie interesują. Okno edycji przedstawione jest na rysunku nr 9.

```
#WIDGET BEGINS HERE. PLEASE EDIT AS MUCH AS YOU WANT
?>
    <ul>
    <?php wp_register();?>
    <li><?php wp_loginout(); ?></li>
    <!--<li><a href="<?php bloginfo('rss2_url'); ?
>" title="<?php echo attribute_escape(__('Syndicate this site using
RSS 2.0')); ?>"><?php _e('Entries <abbr title="Really Simple
Syndication">RSS</abbr>'); ?></a></li>
    <li><a href="<?php
bloginfo('comments_rss2_url'); ?>" title="<?php echo
attribute_escape(__('The latest comments to all posts in RSS')); ?>">
<?php _e('Comments <abbr title="Really Simple
Syndication">RSS</abbr>'); ?></a></li>-->
    <?php wp_meta();?>
    </ul>
<?php
#WIDGET ENDS HERE.
```

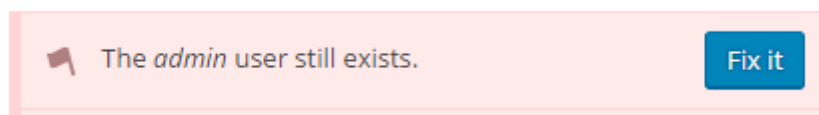
9. Edycja widgetu Meta poprzez zakomentowanie niepotrzebnych linków

### 4.5.3 iThemes Security [42]

iThemes Security to potężna wtyczka znacznie zwiększająca bezpieczeństwo naszej strony. Ma ona za zadanie diagnostykę najczęściej występujących błędów, obniżających bezpieczeństwo strony, jak zagrożenia spowodowane błędami w instalowanych wtyczkach, słabymi hasłami czy przestarzałym oprogramowaniem. Segreguje ona problemy, które wykryje na naszej witrynie przypisując im 3 poziomy priorytetów (High, Medium, Low) oraz oferuje pomoc w ich rozwiązaniu.

Po zakupieniu pełnej wersji, wtyczka daje nam nowe możliwości, jak śledzenie działalności użytkowników, importowanie/eksportowanie ustawień, ustawianie reguł dotyczących haseł, itp.

Przykładowe powiadomienie zaprezentowane jest na rysunku nr 10.



10. Przykładowy problem, sygnalizowany przez wtyczkę iThemes Security

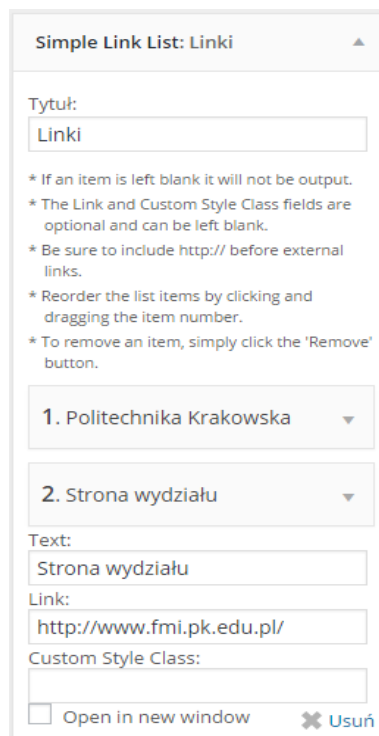
#### 4.5.4 Simple Link List Widget [43]

Wtyczka dodaje widget pozwalający nam na stworzenie własnej listy linków i wyświetlanie jej na stronie. Nie musimy dzięki temu wpisywać kodu html w domyślnym widżecie tekstowym Wordpressa, co jest pomocne dla osób niezaznajomionych z tym językiem. Nasze linki mają opcję otwierania się w nowych oknach. Listę utworzoną przez widget pokazuje rysunek nr 11, okno edycji dodanego widgetu pokazuje rysunek nr 12.

#### Linki

- [Politechnika Krakowska](#)
- [Strona wydziału](#)
- [Squirrel Mail](#)

11. Widget Simple List widoczny na stronie



12. Widget Simple Link List - tryb edycji

#### 4.5.5 Simple Share Buttons Adder [44]

Wtyczka dodająca przyciski udostępniania danego wpisu na różnych portalach społecznościowych. W naszym przypadku wybraliśmy Facebooka [50], Google+ [51] oraz LinkedIn [52], jednak dzięki prostemu menu prostym przeciągnięciem ikony możemy dodać kolejną, do jednego z 12 innych portali. Wersja komercyjna wtyczki umożliwia nam też połączenie z dwoma kolejnymi portalami społecznościowymi. Uruchomioną wtyczkę i jej działanie pod przykładowym postem prezentuje rysunek nr 13.

Wtyczka pozwala nam na edycję tego, gdzie pojawiają się przyciski, pozycji przycisków, towarzyszącego im napisu, itp.

## Inny wpis

Opublikowano [21 grudnia 2015](#), autor: [admin](#)

tekst PL



Opublikowano [Bez kategorii](#) | [Możliwość komentowania została wyłączona](#) | [Edytuj](#)

13. Trzy przyciski udostępniania, załączone do wpisu dzięki wtyczce Simple Share Buttons Adder

### 4.5.6 Simple Staff List [45]

Rozbudowana wtyczka pozwalająca nam na stworzenie listy zdefiniowanych osób na stronie. Prosty formularz pozwala nam na dodanie osoby. Możemy dodać jej imię i nazwisko, zdjęcie, adres email, telefon oraz nasz własny opis. Osoby możemy dzielić na różne grupy (zob. rysunek 14) i zmieniać ich pozycje na liście.



Pod opcję menu *Templates* możemy także zmieniać sposób wyświetlania listy,<sup>14</sup> możliwość dodania osoby do zdefiniowanych grup modyfikując zawarty tam kod HTML i CSS. Znajdują się tam także skróty, których możemy użyć do wstawiania zmiennych w kod.

Pod opcją *Usage* znajdziemy sposób użycia wtyczki – jak dodać listę na stronę. Używamy do tego shortcode, np. `[simple-staff-list group="Robots" order="ASC"]` wyświetli we wpisie osoby z grupy *Robots* w kolejności rosnącej.

Modyfikujemy tę wtyczkę, zmieniając wzór, zgodnie z którym wyświetla się lista, a także dodając do kodu CSS instrukcję odpowiedzialne za podświetlenie i opisanie kierownika katedry listujemy właśnie Katedry, a także pracowników instytutu na odpowiednich podstronach witryny. Ekran modyfikacji kodu CSS i wzoru wyświetlania



ukazany jest na rysunku nr 15.

The screenshot displays the 'Staff Loop Template' editor in WordPress. At the top, there is a 'View Default Template' button. Below it, a text area contains the following PHP code:

```
[staff_loop]
<div class="member">
  <div class="staff-member-info-wrap left-cont">
    [staff-position] [staff-name]<br>
    [staff-email-link]
    [staff-bio]
  </div>
  <div class="right-cont">
    
  </div>
</div>
[/staff_loop]
```

Below the code area is a 'Save ALL Changes' button. Underneath, the 'Staff Page CSS' section is visible, featuring a checkbox for 'Write to external CSS file? (Leave unchecked for WP Multisite.)' and a 'View Default CSS' button. A text area below contains the following CSS code:

```
.headmaster:nth-of-type(1)
{
padding-top: 10px;
background-color: #EEEEEE;
}

.headmaster:nth-of-type(1):before
{
font-weight: bold;
padding-left: 50px;
content: "Kierownik";
}

.left-cont {
width: 472px;
float: right;
}
```

At the bottom of the CSS editor is another 'Save ALL Changes' button.

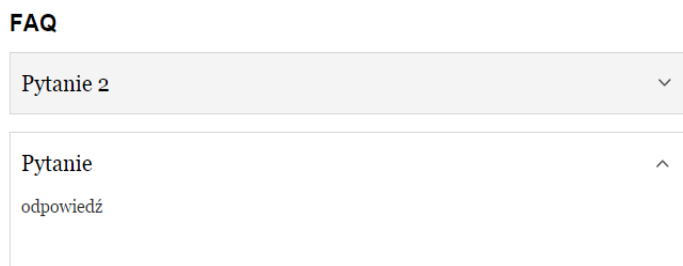
15. Ekran umożliwiający edycję kodu CSS i wzoru wyświetlania listy we wtyczce Staff Members

#### 4.5.7 WP FAQ [46]

Wtyczka pozwalająca w prosty sposób dodać najczęściej zadawane pytania. Każdy

użytkownik może na swojej stronie stworzyć własną listę pytań. FAQ (ang. Frequently Asked Questions – często zadawane pytania) możemy dzielić na kategorie i udostępniać na stronie. Zamieszczona na stronie wtyczka ukazana jest na rysunku nr 16.

W rozszerzonej wersji mamy do wyboru więcej opcji wyglądu naszych pytań i dodane kolejne opcje edycji.

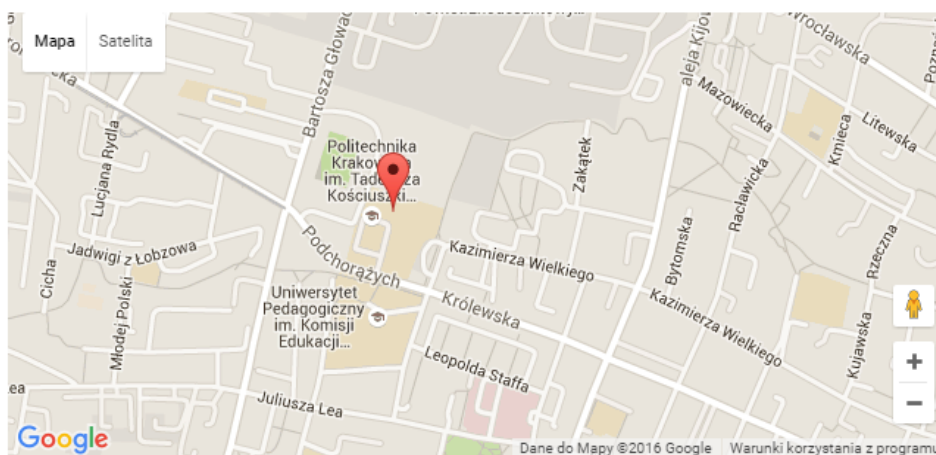


16. Strona FAQ z załączoną rozwijaną listą pytań

#### 4.5.8 WP Google Maps [47]

Wtyczka pozwalająca na dodanie znacznika do mapy i udostępnienie go na stronie dzięki *shortcode* (krótkiemu odnośnikowi). Daje nam możliwość edycji rozmiarów mapy, nazwy, rodzaju (drogowa, satelitarna, hybryda, terenowa), położenia mapy, zmiany wyglądu, itp. Mapa zamieszczona na stronie wygląda tak, jak na rysunku nr 17.

W wersji PRO dodatkowo możemy edytować mapę i znaczniki w dowolny sposób.



17. Mapa z zaznaczonym adresem wydziału, wyświetlana na stronie dzięki wtyczce WP Google Maps

### 4.5.9 WPGlobus [48]

Plugin pozwala nam na zlokalizowanie strony w wybranych przez nas językach. Każdy wpis i każda strona w naszej witrynie zostaje wzbogacona o opcję tłumaczenia. Możemy przejść dzięki tej opcji do kolejnej zakładki, w której przetłumaczymy nasz wpis na wybrany język. Nasza strona może być też oznaczona jako nieprzetłumaczalna.

W menu ustawień wybieramy interesujące nas języki i aktywujemy je (lista aktywnych języków – zob. rysunek 18). Możemy wybrać jeden

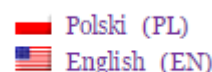


18. Wybór aktywnych języków

z kilkunastu już dostępnych, lub

dodać nowy. Każdy język jest identyfikowany przez ikonę mapy i nazwę lub dwuliterowy skrót. Podstawowy język strony umieszczamy na samej górze listy.

Możemy zmienić tryb wyboru języka, dzięki czemu widget odpowiedzialny za ten wybór zmieni swój wygląd.



Dostępne opcje to kombinacje ww. identyfikatorów.

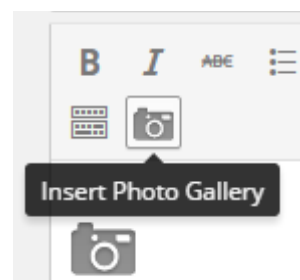
19. Widget zmiany języka

Możemy też edytować ten wygląd przez edycję pliku CSS. Widget zmiany języka wygląda przykładowo tak, jak na rysunku nr 19.

Nasza wtyczka może być wyłączona dla danej kategorii zawartości (podstawowa zawartość witryny to wpisy i strony, ale może być rozszerzona przez inne pluginy).

### 4.5.10 Photo Gallery [49]

Zaawansowany plugin do tworzenie responsywnych galerii zdjęć. Zawiera też dodatkowy plugin do dodawania galerii i albumów zdjęć do postów lub stron. Zawiera szerokie opcje edycji wyglądu zdjęć i sposobu ich wyświetlania, a także opisywania. Umożliwia także dodawanie własnego znaku wodnego, lub reklamy. Po dodaniu zdjęć do galerii możemy dodać je za pomocą specjalnego przycisku (zob. rysunek 20) do wpisu lub strony. Na ilustracjach 21 i 22 widzimy zamieszczoną we wpisie wtyczkę.

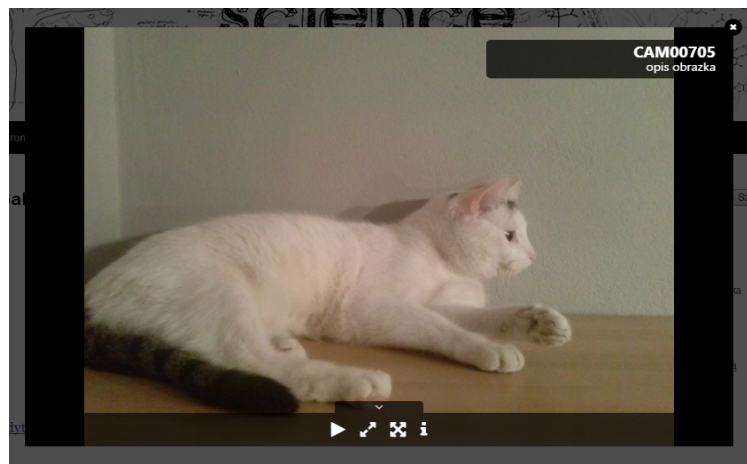


20. Przycisk dodania galerii do wpisu, dostępny przy jego tworzeniu

## Galeria



21. Wyświetlenie galerii na stronie



22. Wybrane zdjęcie powiększa się. Wyświetlona nazwa zdjęcia i dodany opis

Wtyczka pomaga w doinstalowaniu dodatków do importu i integrujących galerię z facebookiem.

## 4.6 Tworzenie własnych wtyczek [13-16]

Gdy nie znajdziemy wtyczki spełniającej nasze wymagania dzięki wyszukiwarce Wordpressa, możemy stworzyć ją sami. Pomoże nam bogate community [53] i rozwinięty manual Wordpressa [54].

### 4.6.1 Tworzenie i instalacja własnych wtyczek

Naszą wtyczkę możemy złożyć z wielu plików .js, .css., .php, .html. Wszystkie te pliki muszą znajdować się w jednym folderze, o nazwie takiej, jak nasza wtyczka. Gotowy produkt umieszczamy w folderze `wp-content/plugins/`, gdzie, jak możemy zaobserwować,

Dodawane są wszystkie zainstalowane przez nas wtyczki (każda zainstalowana wtyczka ma tam swój folder) – jest to równoznaczne z instalacją wtyczki.

Sercem naszej wtyczki będzie plik .php, w którym, na początku umieszczamy komentarz mający identyfikować naszą wtyczkę. Dzięki temu będzie można ją rozróżnić od innych, a także udostępnić ją innym użytkownikom Wordpressa. Przykładowy komentarz z wykonanej na potrzeby pracy wtyczki:

```
/**
 * Plugin Name: Thesis List
 * Plugin URI: http://torus.uck.pk.edu.pl/~mrazny/?wp-
plugin_name=thesis-list
 * Description: Tworzenie i zarządzanie listą doktorantów i
habilitacji
 * Version: 1.0
 * Author: Mikolaj Razny
 * Author URI: http://torus.uck.pk.edu.pl/~mrazny/?wp-
plugin_name=thesis-list
 */
```

Jak widzimy komentarz zawiera nazwę naszej wtyczki, stronę, pod którą możemy znaleźć ją i jej opis, krótki opis wtyczki, wersję, nazwę autora i jego stronę internetową. W tym przypadku podaliśmy tą samą stronę. Plugin URL i nazwa identyfikują wtyczkę i odróżniają ją od innych.

Ten główny plik musi też zawierać (jak każdy plik .php) znacznik `<?php` na początku, oraz `?>` na końcu.

#### **4.6.2 Praca z bazą danych [14]**

Łatwym i prostym sposobem jest połączenie się z bazą danych używając klasy \$wpdb. Pozwala nam ona przekazywać zmienne i instrukcje w mySql, dodatkowo upraszczając proces dzięki wbudowanym funkcjom, dzięki którym np. listę argumentów w insercie podajemy jako listę elementów PHP. Pomaga też w obsłudze danych otrzymanych od bazy.

Zaczynamy od zwołania instancji klasy \$wpdb, co daje nam dostęp do jej funkcji. Funkcje wykorzystane w projekcie to m.in.:

- `get_results(zapytanie_sql)` – zwraca wynik zapytania z bazy
- `insert(tabela, dane, format)` – dodaje nowe dane do bazy
- `delete(tabela, dane, które_usunąć)` – usuwa dane z bazy
- `update(tabela, dane, które_zaktualizować)` – modyfikuje (aktualizuje) dane w bazie
- `query(sql)` – dowolne instrukcje

### 4.6.3 Dodawanie danych na stronę [shortcode]

*Shortcode* (krótki odnośnik) umożliwiają nam na prosty dostęp do danych zdefiniowanych przez wtyczki. Dzięki nim pod prostym kawałkiem tekstu, może ukrywać się właściwie nieograniczona ilość danych. Zdefiniowany tekst wpisujemy w trybie edycji strony i po wyświetleniu ukazuje się to, na co wskazuje nasz odnośnik.

*Shortcode* musimy zdefiniować w naszym pluginie.

```
function bartag_func( $atts ) {
    $a = shortcode_atts( array(
        'foo' => 'something',
        'bar' => 'something else'
    ), $atts );
    return "foo = {$a['foo']}";
}

add_shortcode( 'bartag', 'bartag_func' );
```

Gdy nasz *shortcode* wygląda tak: `[bartag foo="foo-value"]`, to podana funkcja rozumie, że podaliśmy „foo-value”, jako argument *foo* i wyświetli na stronie tekst „foo = foo\_falue”. Funkcją *add\_shortcode* dodaliśmy do wordpressa obsługę tego skrót. Podajemy jego nazwę oraz nazwę funkcji, którą wywołuje. Bardzo istotnym jest fakt, że funkcja nie wyświetla jej ciała, ale musi zwrócić tekst. Jeśli więc chcemy ukryć pod skrótem dużą ilość danych, to musimy w funkcji stworzyć odpowiednio dużą zmienną, a następnie doklejać do niej kolejne fragmenty.

#### 4.6.4 Wtyczka do zarządzania listą habilitacji i doktoratów (*thesis list*)

Pierwszą z tworzonych przez nas wtyczek jest wtyczka, dzięki której stworzymy listę habilitacji i doktoratów, a następnie za pomocą shortcode wyświetlimy ją na stronie. Wyświetlana będzie jako tabela z kolumnami zawierającymi: imię z nazwiskiem, tytuł pracy, datę publikacji, uczelnie i promotora.

Na początku tworzymy funkcję `tl_set_up_db()` - w jej ciele przekazujemy do bazy danych instrukcje tworzenia tabel, w których przechowamy nasze dane. Następnie rejestrujemy tą funkcję używając „haka”.

```
register_activation_hook ( __FILE__, 'tl_set_up_db' );
```

Dzięki tej funkcji przy aktywacji wtyczki wywoła się funkcja `tl_set_up_db()`.

Następnie definiujemy wszystkie funkcje, których będziemy używać, tj.:

- `tl_insert` – dodaje dane z formularza
- `tl_delete` – usuwa wpis
- `tl_admin` – zawiera zawartość strony administracyjnej

Kolejnym krokiem jest stworzenie strony dla administratora. Będzie on miał możliwość (dzięki formularzowi) dodawania nowych wpisów do tabeli habilitacji i drugiej tabeli – doktoratów. Zarejestrowanie zawartości strony odbywa się poprzez użycie funkcji `add_action` wywołującej funkcję `add_options_page`.

```
add_action('admin_menu', 'tl_admin_actions');  
function tl_admin_actions() {  
    add_options_page('Thesis List', 'Thesis List', 'manage_options', 'thesis-list', 'tl_admin');  
}
```

Ta druga funkcja przypisuje zawartość funkcji `tl_admin` do strony administracyjnej strony, a także dodaje przycisk do menu *Ustawienia*. Dzięki temu przyciskowi możemy administrować naszą wtyczką.

W ciele funkcji `tl_admin` możemy załączyć pliki `.js` i `.css` wykorzystywane na stronie administracyjnej. Kod html oblekamy w tagi `<div>` klasy „wrap” - jest to klasa wordpressa, dzięki temu zawartość będzie odpowiednio umiejscowiona. W środku umiejscawiamy nasz kod, który w razie potrzeby przeplatamy kodem PHP.

Jak zauważyliśmy, nasze funkcje mają przedrostkiem *tl\_*, co jest skrótem od nazwy wtyczki – w tym przypadku *Thesis List*. To ważne, żeby poprzedzać nazwy wszystkich funkcji unikalnym przedrostkiem, gdyż dzięki temu nie wystąpi konflikt z innymi funkcjami innych pluginów.

Na samym końcu dodajemy funkcję *add\_shortcode*. W tym przykładzie dzięki kodowi `[tl_thesis_doctorate="true/false"]`, wyświetlimy odpowiednio listę doktoratów (true), albo habilitacji (false).

Gotowy formularz widzimy na rysunku nr 23.

Doktoraty					
Imię i nazwisko	Tytuł pracy	Data	Uczelnia	Promotor	Akcja
Test Test	Przygotowanie strony internetowej dla jednostki uniwersyteckiej, przy użyciu Systemu Zarządzania Treścią(CMS), na przykładzie strony Instytutu Fizyki Politechniki Krakowskiej	2015-12-23	Politechnika Krakowska	Test Testowy	<input type="checkbox"/>
Tester	Lorem ipsum dolor sit amet enim. Etiam ullamcorper. Suspendisse a pellentesque dui, non felis. Maecenas malesuada elit lectus felis, malesuada ultricies. Curabitur et ligula. Ut molestie a, ultricies porta urna. Vestibulum commodo volutpat a, convallis ac, laoreet enim.	2015-01-15	UW	TESTER	<input type="checkbox"/>
a	Lorem ipsum dolor sit amet enim. Etiam ullamcorper. Suspendisse a pellentesque dui, non felis. Maecenas malesuada elit lectus felis, malesuada ultricies. Curabitur et ligula. Ut molestie a, ultricies porta urna. Vestibulum commodo volutpat a, convallis ac, laoreet enim.	2015-12-11	asdf	asdf	<input type="checkbox"/>
b	Lorem ipsum dolor sit amet enim. Etiam ullamcorper. Suspendisse a pellentesque dui, non felis. Maecenas malesuada elit lectus felis, malesuada ultricies. Curabitur et ligula. Ut molestie a, ultricies porta urna. Vestibulum commodo volutpat a, convallis ac, laoreet enim.	2015-12-02	adsdfs	dsa	<input type="checkbox"/>
ss	test	2015-12-31	ttt	ttt	<input type="checkbox"/>

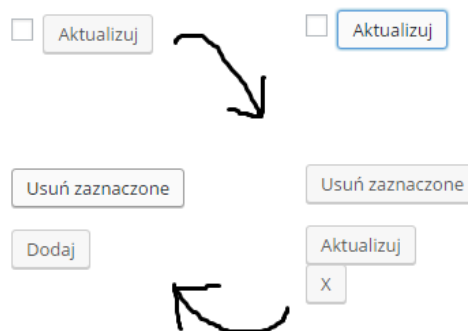
Imię i nazwisko	Tytuł pracy	Data	Uczelnia	Promotor	Akcja
-----------------	-------------	------	----------	----------	-------

23. Formularz i lista doktoratów z wtyczki Thesis List

#### 4.6.5 Wtyczka do zarządzania listą zajęć laboratoryjnych (class instruction)

Wtyczka ta ma podobne działanie do poprzedniej, jednak wzbogacamy funkcjonalność o aktualizowanie/poprawianie danych, a także stosujemy trochę inny mechanizm shortcode.

Tryb aktualizacji danych włącza się, dzięki kliknięciu przycisku „aktualizuj”, który znajduje się przy każdym wpisie w tabeli. Za pomocą Java Script wypełniamy danymi z wiersza formularz, który możemy ciągle edytować. Dodaje on też ID wpisu do ukrytego pola. Możemy zatwierdzić zmiany, albo wrócić do normalnego trybu dodawania poprzez naciśnięcie odkrytego przycisku „X”, który



24. Mechanizm zmiany funkcji przycisków



kasuje zawartość pola ID. Gdy pole ID jest wypełnione, to wysłanie formularza aktualizuje wiersz. W innym przypadku dane z formularza są dodawane jako nowy wpis. Mechanizm zmiany trybu na aktualizację/dodawanie wpisów przedstawia rysunek nr 24.

Inny mechanizm shortcode polega na tym, że argumentu nie podajemy jako wartość jakiejś zmiennej, ale podajemy między znacznikami. Dodając na stronie wpis `[ci_lab]i/ii/elektr[/ci_lab]` wyświetlamy tabelę, odpowiednio laboratoriów I, II, albo elektronicznych.

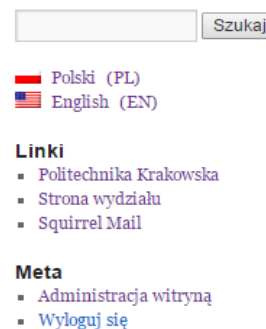
## 4.8 Dodawanie nowych stron

Dodawanie nowych stron odbywa się poprzez wejście w menu *Strony->Dodaj nową*. Wprowadzić można tytuł strony (dzięki WPGlobus także w innym języku) oraz zawartość w postaci formatowalnego tekstu (tryb wizualny), albo w html (tryb tekstowy).

Modyfikując atrybuty strony możemy określić, czy jest to strona samodzielna, czy np. jest podstroną jakiejś innej (wybranie rodzica). Możemy też zmienić szablon i określić pozycję w menu. Tę pozycję można później zmienić w podglądzie wszystkich stron.

## 4.9 Edycja widgetów

Przy podglądzie strony w trybie administratora, możemy zmieniać elementy jej wyglądu, a także dodatki. Przemieszczając widgety w różne miejsca, zmieniać obraz/kolor tła itp. Jednym z założeń projektu jest zachowanie czytelności strony, w związku z tym usuwamy z paska widgetów niepotrzebne dodatki, taki jak kalendarz. Dodajemy także linki do strony Politechniki Krakowskiej, Wydziału FMiI. Dodajemy także widget zmiany języka WPGlobus. Cały widok na widgety prezentuje rysunek nr 25.



25. Obszar widgetów widoczny na stronie

## 4.10 Gotowa strona

Rysunek nr 26 przedstawia stronę testową. Należy ją uzupełnić odpowiednią zawartością korzystając z Panelu Administracyjnego. Jest to zadanie dla osoby, która

będzie odpowiedzialna za stronę.

Kolejne rysunki przedstawiają podstronę zintegrowaną z facebookiem (zob. rysunek 27), działanie menu nawigacyjnego (zob. rysunek 28) i podstronę laboratoriów (zob. rysunek 29).

## Strona Testowa

mały napisik po prawej :v



---

[Strona główna](#) [Facebook](#) [Laboratorium](#) [Pracownicy](#) [Kontakt](#) [Inne](#)

### Inny wpis

Opublikowano [21 grudnia 2015](#), autor: [admin](#)

tekst PL

Share via [f](#) [G+](#) [in](#)

Opublikowano [Bez kategorii](#) | [Możliwość komentowania została wyłączona](#) | [Edytuj](#)

### Wpis PL

Opublikowano [29 listopada 2015](#), autor: [admin](#)

Lorem ipsum dolor sit amet erat. Nulla venenatis blandit justo, at orci consequat tortor. Nam urna. Nulla vehicula wisi, ullamcorper mauris. Nunc non nunc. Etiam sit amet purus sem volutpat at, egestas vitae, tempus malesuada. Donec eget elit sed molestie vitae, facilisis felis, nec risus. Vivamus hendrerit tellus sodales in, dapibus eu, leo. In porttitor risus. Suspendisse nec erat sed ante. In vitae massa at risus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque dapibus risus. Suspendisse ultricies tortor pede in felis ac ipsum. Fusce commodo volutpat eu, neque. Donec a lorem ligula, nonummy velit non nulla. Suspendisse bibendum tempus. Pellentesque orci sit amet sagittis vel, velit. Duis blandit ultrices consetetuer. Aenean libero. Proin gravida pulvinar ligula, nonummy sagittis. Lorem ipsum dolor vel turpis. Praesent ante. Sed aliquet ac, laoreet vulputate eget, nonummy auctor mi. Pellentesque tortor. Cras egestas est ultricies eu, ornare lorem. Mauris interdum faucibus, diam eu felis. Quisque eu lectus. Pellentesque at ante. Suspendisse vitae lorem vitae mauris. Mauris pretium cursus. In vitae lectus semper egestas. Praesent quis est. Nunc elementum. Fusce quam nec est. Aliquam.

Share via [f](#) [G+](#) [in](#)

Opublikowano [Bez kategorii](#) | [Skomentuj](#) | [Edytuj](#)

---

[Strona Testowa](#)

 *Dumnie wspierane przez*  
WordPress.

26. Widok na stronę główną wykonanej strony

## Facebook



**Politechnika Krakowska im. Tadeusza Kościuszki**

1 day ago

Zima wydaje się na razie tylko wspomnieniem :( Wprawdzie nie jesteśmy wielkimi fanami tej pory roku, ale doceniamy urok takich zdjęć :).

P.S. Zwróciliście uwagę na piękną i satysfakcjonującą symetrię ścieżek? ;)

/Fot. Instagram - @viktoryia\_semianiuk/

[View on Facebook](#) · [Share](#)



**Politechnika Krakowska im. Tadeusza Kościuszki**

1 day ago

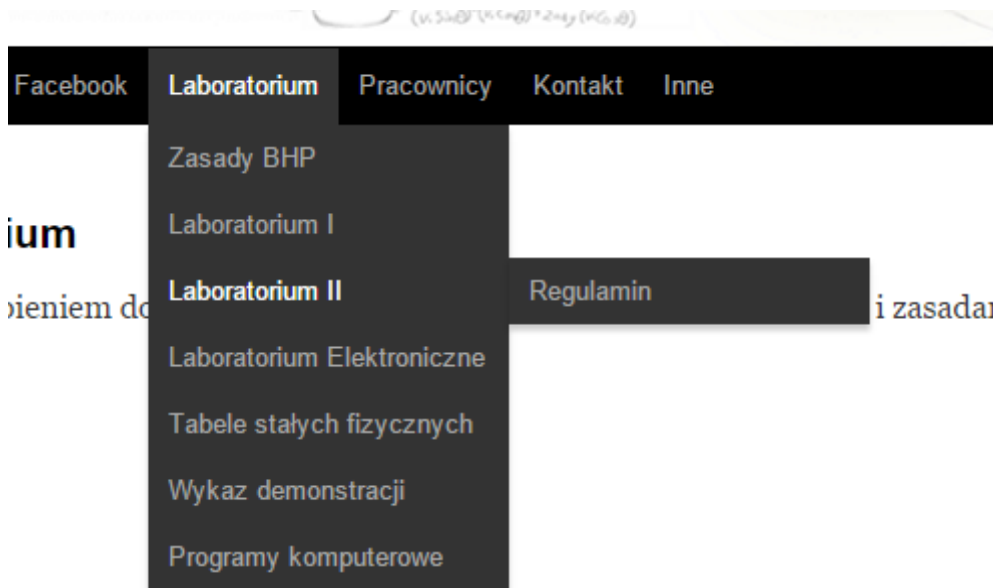
Politechnika Krakowska im. Tadeusza Kościuszki zaktualizował(a) swoje zdjęcie w tle.

[View on Facebook](#) · [Share](#)



**Politechnika Krakowska im. Tadeusza Kościuszki**

27. Fragment z wtyczką odpowiedzialną za integrację z facebookiem



28. Działające menu nawigacyjne

## Laboratorium II

Nr 43

Wyznaczanie prędkości światła



S114

[tabela](#)

Rys.1. Przyrząd do wyznaczania prędkości światła

1. Równania Maxwella.
2. Wyprowadzenie równań fali elektromagnetycznej w próżni i w ośrodku jednorodnym.
3. Zasada pomiaru. Pojęcie drogi optycznej, współczynnika załamania

29. Strona zawierająca listę ćwiczeń laboratoryjnych - wyświetlony opis jednego z ćwiczeń

### 4.11 Przeniesienie strony na serwery Politechniki Krakowskiej - wdrożenie

Wdrożenie odbywa się poprzez zainstalowanie wordpressa na serwerze Instytutu Fizyki zgodnie z instrukcjami przedstawionymi w punkcie 4.3. Po udanym dostosowaniu oprogramowania klonujemy repozytorium git (polecenie *git clone*) z maszyny wirtualnej, a następnie rozpakowujemy je w tej samej lokalizacji na serwerze IF.

Do poprawnego przeniesienia strony wymagane będzie wyznaczenie opiekuna serwera IF, który udzieli dostępu do konta z prawami tworzenia nowych plików, a także do danych dostępowych do zainstalowanej uprzednio bazy MySQL (na serwerze IF zainstalowane jest już odpowiednie oprogramowanie).

## **5. Podsumowanie**

Cel pracy został osiągnięty. Został stworzony od zera projekt w pełni funkcjonalnej strony internetowej Instytutu Fizyki Politechniki Krakowskiej. Strona ta spełnia nowoczesne standardy stawiane stronom internetowym, oraz jest zgodna ze specyfikacją. Choć niektóre z wymagań były trudne do zrealizowania, to zostały spełnione w zadowalający sposób. Praca była satysfakcjonująca i okazała się doskonałym zadaniem, pozwalającym zaznajomić się z tematem CMS oraz tworzeniem własnych wtyczek do Wordpressa.

Wykonany projekt może być dalej rozwijany i wzbogacany o kolejne cechy, takie jak np. wirtualna mapa wydziału. Dzięki zastosowaniu popularnego systemu CMS, strona może być nieograniczenie rozbudowywana o nową funkcjonalność w bardzo łatwy sposób.

## Bibliografia

- [1] *Co to jest CMS i czy jest Ci potrzebny?*, <http://sukcesstrony.pl/co-to-jest-cms> (dostęp 6 I 16r)
- [2] *System zarządzania treścią – Wikipedia, wolna encyklopedia*, [https://pl.wikipedia.org/wiki/System\\_zarządzania\\_treścią](https://pl.wikipedia.org/wiki/System_zarządzania_treścią) (dostęp 6 I 16r)
- [3] *Czym jest Joomla!*, <http://www.joomla.pl/o-joomla/joomla-jako-program/czym-jest-joomla> (dostęp 6 I 16r)
- [4] *About | Drupal.org*, <https://www.drupal.org/about> (dostęp 6 I 16r)
- [5] *Który CMS jest najlepszy: Wordpress, Joomla!, Drupal czy inny?*, <http://sukcesstrony.pl/ktory-cms-jest-najlepszy-wordpress-joomla-drupal-czy-inny>
- [6] *CMS technologies Web Usage Statistics*, <http://trends.builtwith.com/cms>
- [7] *Joomla Vs Wordpress – YouTube*, [https://www.youtube.com/watch?v=SjwKWIGXV\\_Q](https://www.youtube.com/watch?v=SjwKWIGXV_Q), Ryan Moore
- [8] *„SVN, Git i Redmine”*, instrukcje laboratoryjne Politechniki Krakowskiej przedmiotu Administrowanie Systemami Komputerowymi, Kraków 2014
- [9] *MySQL :: MySQL Editions*, <https://www.mysql.com/products/> (dostęp 6 I 16r)
- [10] *Kurs PHP » Podstawy*, <http://phpkurs.pl/podstawy/> (dostęp 6 I 16r)
- [11] *How to Install Wordpress on Ubuntu 15.04 Server | Liberian Geek*, <http://www.liberiangeek.net/2015/06/how-to-install-wordpress-on-ubuntu-15-04-server/> (dostęp 13 XI 15r)
- [12] *Create A Network « WordPress Codex*, [http://codex.wordpress.org/Create\\_A\\_Network](http://codex.wordpress.org/Create_A_Network) (dostęp 13 XI 15r)
- [13] *Tips and Tricks HQ*, <https://www.youtube.com/channel/UCsZdFFESa0E54dl6uaUcK4g> (dostęp 14 XII 15r)
- [14] *WordPress Essentials: Interacting With The WordPress Database*,

- <https://www.smashingmagazine.com/2011/09/interacting-with-the-wordpress-database/> (dostęp 9 XII 15r)
- [15] *Pippin Williamson* – *Youtube*, <https://www.youtube.com/channel/UCSlxGloycwNUEApJfUAIKFg>, Pippin Williamson (dostęp 15 XII 15r)
  - [16] *Develop With TP* – *YouTube*, <https://www.youtube.com/channel/UCnVv6y1FJ2XbbmXvWO9VBDg> (dostęp 23 XII 15r)
  - [17] *WordPress* › *Polska*, <https://pl.wordpress.org/> (dostęp 13 I 16r)
  - [18] *Joomla.pl – informacje*, <http://www.joomla.pl/> (dostęp 13 I 16r)
  - [19] *Drupal - Open Source CMS | Drupal.org*, <https://www.drupal.org/> (dostęp 13 I 16r)
  - [20] *How CMS Works*, <http://it.ok.ubc.ca/service-catalogue/publishing-hosting/cms/help/how-it-works.html> (dostęp 13 I 16r)
  - [21] *ICMS system Edito - Content Management System - System zarządzania treścią w technologii PHP i ASP.NET*, <http://www.edito.pl/> (dostęp 13 I 16r)
  - [22] *System Operacyjny GNU i Ruch Wolnego Oprogramowania*, <https://www.gnu.org/home.pl.html> (dostęp 13 I 16r)
  - [23] *W3C XHTML2 Working Group Home Page* <https://www.w3.org/MarkUp/> (dostęp 13 I 16r)
  - [24] *Tim Berners-Lee - Wikipedia, the free encyclopedia*, [https://en.wikipedia.org/wiki/Tim\\_Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee) (dostęp 13 I 16r)
  - [25] *Cascading Style Sheet*, <https://www.w3.org/Style/CSS/> (dostęp 13 I 16r)
  - [26] *ECMAScript*, <http://www.ecmascript.org/> (dostęp 13 I 16r)
  - [27] *PHP: Hypertext Preprocessor*, <https://www.php.net/> (dostęp 13 I 16r)
  - [28] *PHP: License Information*, <https://secure.php.net/license/> (dostęp 13 I 16r)

- [29] *MySQL*, <https://www.mysql.com/> (dostęp 13 I 16r)
- [30] *Git*, <https://git-scm.com/> (dostęp 13 I 16r)
- [31] *GitHub · Where software is built*, <https://github.com/> (dostęp 13 I 16r)
- [32] *Strona główna - Dropbox*, <https://www.dropbox.com/> (dostęp 13 I 16r)
- [33] *Distrowatch.com*, <http://distrowatch.com/table.php?distribution> (dostęp 13 I 16r)
- [34] *The leading OS for PC, tablet, phone and cloud | Ubuntu*, <http://www.ubuntu.com/> (dostęp 13 I 16r)
- [35] *Oracle VM VirtualBox*, <https://www.virtualbox.org/> (dostęp 13 I 16r)
- [36] *LAMP – Wikipedia, wolna encyklopedia*, <https://pl.wikipedia.org/wiki/LAMP> (dostęp 13 I 16r)
- [37] *Welcome to The Apache Software Foundation!*, <http://www.apache.org/> (dostęp 13 I 16r)
- [38] <http://wordpress.org/latest.tar.gz> (dostęp 13 I 16r)
- [39] *WordPress › Wyróżnione « Darmowe motywy dla WordPressa*, <https://pl.wordpress.org/themes/> (dostęp 13 I 16r)
- [40] *WordPress › Custom Facebook Feed « WordPress Plugins*, <https://pl.wordpress.org/plugins/custom-facebook-feed/> (dostęp 13 I 16r)
- [41] *WordPress › Customize Meta Widget « WordPress Plugins*, <https://pl.wordpress.org/plugins/customize-meta-widget/> (dostęp 13 I 16r)
- [42] *WordPress › Better WP Security « WordPress Plugins*, <https://pl.wordpress.org/plugins/better-wp-security/> (dostęp 13 I 16r)
- [43] *WordPress › Simple Link List Widget « WordPress Plugins*, <https://pl.wordpress.org/plugins/simple-link-list-widget/> (dostęp 13 I 16r)
- [44] *WordPress › Simple Share Button Adder « WordPress Plugins*, <https://pl.wordpress.org/plugins/simple-share-buttons-adder/> (dostęp 13 I 16r)



- [45] *WordPress › Simple Staff List « WordPress Plugins*, <https://pl.wordpress.org/plugins/simple-staff-list/> (dostęp 13 I 16r)
- [46] *WordPress › WP FAQ « WordPress Plugins*, <https://pl.wordpress.org/plugins/wp-faqs/> (dostęp 13 I 16r)
- [47] *WordPress › WP Google Maps « WordPress Plugins*, <https://pl.wordpress.org/plugins/wp-google-maps/> (dostęp 13 I 16r)
- [48] *WordPress › WPGlobus « WordPress Plugins*, <https://pl.wordpress.org/plugins/wpglobus/> (dostęp 13 I 16r)
- [49] *WordPress › Photo Gallery « WordPress Plugins*, <https://pl.wordpress.org/plugins/photo-gallery/> (dostęp 13 I 16r)
- [50] *Facebook*, <https://www.facebook.com/> (dostęp 13 I 16r)
- [51] *Google+*, <https://plus.google.com/> (dostęp 13 I 16r)
- [52] *Witamy w LinkedIn | LinkedIn*, <https://www.linkedin.com/polski> (dostęp 13 I 16r)
- [53] *WordPress › Community*, <https://wordpress.org/news/category/community/> (dostęp 13 I 16r)
- [54] *Main Page « WordPress Codex*, <https://codex.wordpress.org/> (dostęp 13 I 16r)
- [55] *4 Lesser Known CMS That Should Not Be Overlooked | Open Designs*, <http://www.opendesigns.org/4-lesser-known-cms-worth-checking-out/> (dostęp 14 I 16r)