Uniwersytet Ekonomiczny w Krakowie

Wydział Zarządzania

Kierunek: Informatyka Stosowana

Specjalność: Inżynieria Oprogramowania

**Anna Bajkowska**

Nr albumu: 185786

**Tworzenie stron internetowych dla użytkowników**

Praca **licencjacka**

napisana pod kierunkiem

dr. hab. Dariusza Puta

Kraków 2017

Spis treści

[Wstęp 4](#_Toc492984061)

[1. Strony internetowe i ich odbiorcy 5](#_Toc492984062)

[1.1 Wprowadzenie 5](#_Toc492984063)

[1.2 Określenie celu w jakim tworzy się strony internetowe 5](#_Toc492984064)

[1.2.1 Serwer WWW 6](#_Toc492984065)

[1.2.2 Cel tworzenia stron internetowych 7](#_Toc492984066)

[1.3 Użytkownicy stron internetowych 11](#_Toc492984067)

[1.3.1 Użytkownik realny a unikalny 11](#_Toc492984068)

[1.3.2 Gemius/PBI i Google Analytics 12](#_Toc492984069)

[1.3.3 Profile internautów 15](#_Toc492984070)

[1.4 *User experience* i *web usability* 17](#_Toc492984071)

[1.4.1 *User experience* 17](#_Toc492984072)

[1.4.2 *Web usability* 20](#_Toc492984073)

[2. Technologie wykorzystywane podczas tworzenia stron WWW 23](#_Toc492984074)

[2.1 Wprowadzenie 23](#_Toc492984075)

[2.2 Języki stosowane do tworzenia stron internetowych 23](#_Toc492984076)

[2.2.1 HTML i XHTML 23](#_Toc492984077)

[2.2.2 CSS 27](#_Toc492984078)

[2.2.3 JavaScript 31](#_Toc492984079)

[2.2.4 PHP 35](#_Toc492984080)

[2.3 Systemy zarządzania relacyjnymi bazami danych SQL 40](#_Toc492984081)

[2.3.1 MySQL 40](#_Toc492984082)

[2.3.2 PostgreSQL 43](#_Toc492984083)

[2.4 Gotowe rozwiązanie Joomla! 44](#_Toc492984084)

[2.4.1 Instalacja systemu Joomla! 45](#_Toc492984085)

[2.4.2 Podstawowe funkcje Joomla! 48](#_Toc492984086)

[3. Tworzenie własnej strony internetowej 50](#_Toc492984087)

[3.1 Wprowadzenie 50](#_Toc492984088)

[3.2 Projekt strony internetowej 50](#_Toc492984089)

[3.3 Tworzenie strony internetowej przy zastosowaniu systemu Joomla! 55](#_Toc492984090)

[Zakończenie 65](#_Toc492984091)

[Bibliografia 66](#_Toc492984092)

[Netografia 67](#_Toc492984093)

[Spis tabel 68](#_Toc492984094)

[Spis wykresów 69](#_Toc492984095)

[Spis rysunków 70](#_Toc492984096)

# Wstęp

Tworzenie stron internetowych jest coraz bardziej powszechne i w dzisiejszych czasach każdy może takową zaprojektować i napisać. Różnorodność technologii i porad dostępnych w Internecie jest ogromna, dzięki czemu nawet mało doświadczony twórca może stworzyć dynamiczną i ciekawą stronę internetową. Również mnogość gotowych i darmowych systemów CMS sprawia, że coraz więcej osób postanawia stworzyć swój własny serwis. Praca ta ma na celu pokazanie jak w łatwy sposób można zbudować swój własny serwis, który przyciągnie użytkownika. Postanowiłam napisać niniejszą pracę, ponieważ w Internecie jest wiele stron, które zostały albo „porzucone”, albo „odstraszają” użytkowników. Dlatego chciałam napisać jak można uniknąć takich sytuacji oraz sama w przyszłości stworzyć serwis, który gromadził by użytkowników, którzy interesują się różnymi tematami. Moja praca jest właśnie o tworzeniu stron internetowych od początku. Zaczynając od projektu, strategii user experience, przez napisanie strony od początku lub dostosowanie systemu CMS do naszych potrzeb i na publikacji w Internecie kończąc. Opisuje ona także różne technologie z jakimi twórca może się spotkać podczas procesu pisania strony.

Pierwszy rozdział jest ogólnie o tym czym są strony internetowe i serwery WWW. Opisany został także cel tworzenia własnych serwisów internetowych. Następnie wyjaśniona została różnica między realnym a unikalnym użytkownikiem, techniki liczenia ilości internautów odwiedzających dany portal oraz ich profile. Na koniec opisane zostały pojęcia takie jak: user experience i web usability.

Drugi rozdział opisuje technologie jakie mogą zostać użyte podczas pisania własnego serwisu internetowego. Są to języki stosowane do tworzenia stron internetowych, systemy zarządzania relacyjnymi bazami danych SQL oraz oprogramowanie Joomla! (instalacja i omówienie podstawowych funkcji).

Trzeci rozdział opisuje tworzenie strony od początku czyli projektowanie, określnie tematyki, wymyślenie nazwy serwisu oraz na koniec dostosowanie gotowej strony wygenerowanej przez system Joomla! do naszych preferencji.

# 1. Strony internetowe i ich odbiorcy

## 1.1 Wprowadzenie

W czasach powszechnego dostępu do Internetu coraz częściej przeróżnych informacji szukamy, wpisując dane pojęcie w wyszukiwarce internetowej niż zaglądając do encyklopedii czy też szukając odpowiedniej książki w bibliotece. Nawet zakupy online stają się dużo powszechniejsze. Korzystanie z Internetu jest po prostu prostsze, szybsze i wygodniejsze. Istnieje wiele stron o danej tematyce, gdzie można znaleźć potrzebne informacje bądź przeprowadzić dyskusję na dany temat. Dużo łatwiej jest znaleźć potrzebny produkt i kupić go za pośrednictwem Internetu, niż jechać do sklepu i szukać go na półkach. Od razu także znany jest status dostępności produktu.

Tworzenie stron internetowych jest coraz bardziej powszechne. Istnieje wiele pomocnych narzędzi, które ułatwiają budowanie oraz administrację danym portalem. Dzięki nim, w dzisiejszych czasach nie trzeba posiadać specjalistycznej wiedzy, by prowadzić własny serwis internetowy. Jednak oznacza to również, że nie wszystkie strony internetowe są godne zaufania.

W rozdziale tym opisane zostało, czym jest strona internetowa i w jaki sposób widoczna jest w Internecie, w jakim celu tworzy się strony WWW, scharakteryzowane są także profile internautów. Poruszone zostały także tematy pomiaru badania widowni internetowej, *web usability* oraz *user experience*.

## 1.2 Określenie celu w jakim tworzy się strony internetowe

Każdy widzi jaka jest strona internetowa, ale nie każdy zdaje sobie sprawę z tego, co kryje się za grafikami, animacją oraz tekstem wyświetlanym na ekranie monitora. Dla zwykłego użytkownika Internetu to źródło informacji lub rozrywki. Dla firmy lub sklepu jest to źródło dochodów, a dla programisty są to często setki lub tysiące linii kodu. I chociaż dzięki nowoczesnym technologiom tworzenie stron WWW stało się dużo prostsze to i tak, by w pełni korzystać z potencjału dostępnych narzędzi, trzeba posiadać chociaż minimalną wiedzę na temat funkcjonowania stron internetowych.

Na początek trzeba odpowiedzieć na pytanie: czym są strony internetowe? Najprościej stronę internetową można określić jako dokument HTML (*HyperText Markup Language*). Jest to nic innego jak plik tekstowy, który zawiera instrukcje dotyczące struktury strony internetowej. Dokumenty te, oprócz tekstu, zawierają specjalne znaczniki, które opisują każdy element na stronie. *„HTML definiuje dziesiątki elementów tekstowych, które składają się na dokument, na przykład nagłówki, akapity, tekst zaakcentowany, a także oczywiście odnośniki. Istnieją również elementy HTML dodające informacje o samym dokumencie (na przykład jego tytule) czy wstawiające na stronę media, takie jak obrazki, filmy wideo, animacje Flash czy aplety.”[[1]](#footnote-1)*.

Do tworzenia stron internetowych wykorzystuje się wiele języków. Najpopularniejsze to: HTML, XHTML (*eXtensible HTML*), CSS, Java Script, PHP, ASP.NET, SQL i inne.

### 1.2.1 Serwer WWW

Samo napisanie dokumentu HTML nie wystarczy. By strona była widoczna w Internecie, musi znaleźć się na serwerze WWW. Serwer WWW to usługa, która polega na udostępnieniu przestrzeni dyskowej. Można oczywiście uruchomić własny serwer WWW, w tym celu istnieje specjalnie oprogramowanie (tj. Apache, WampSerwer, XAMPP ect.). Własny serwer WWW przydaje się, gdy chcemy mieć podgląd tworzonej strony internetowej. Jest to bardzo wygodne, ponieważ nie trzeba co chwilę „wrzucać” plików na osobny serwer, by sprawdzić czy wszystko działa.

Jednak do udostępniania strony publicznie najwygodniej jest wynająć miejsce na serwerze, który jest specjalnie przeznaczony do tego zadania. Znalezienie hosta dla strony, czyli miejsca na serwerze, nie jest trudne. Istnieje wiele opcji hostingowych: darmowe lub płatne. Wysokość opłat zależy od tego, ile miejsca potrzebujemy, wielkości transferu na miesiąc i wielu innych czynników. W zamian dostawca zobowiązany jest, by nasza strona działała przez cały czas.

Pliki strony przesyłamy na serwer WWW za pomocą protokołu FTP (*File Transfer Protocol*). Do tego potrzebujemy takich informacji jak: nazwa serwera WWW, nazwa (identyfikator) użytkownika, hasło oraz katalog, w którym mają zostać umieszczone. W celu transferu plików za pomocą protokołu FTP musimy użyć specjalnie stworzonego w tym celu oprogramowania zwanego klientem FTP (np. FileZilla, CuteFTP, SmartFTP etc.).

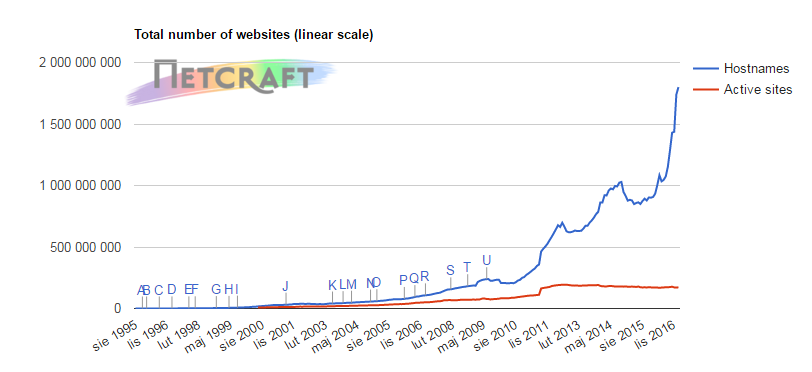
Gdy już zdecydujemy się na którąś z opcji, własny serwer WWW czy hosting, zaprojektowana i gotowa strona internetowa może zacząć pozyskiwać korzystających z niej użytkowników.

### 1.2.2 Cel tworzenia stron internetowych

Pierwsze strony internetowe tworzone były jako źródło informacji. Korzystali z nich głównie pracownicy naukowi. Strony te zawierały informacje głównie z dziedzin fizyki, matematyki czy informatyki. Wraz z biegiem czasu i rozwojem technologii możliwe stało się tworzenie także stron multimedialnych. Obecnie w Internecie można znaleźć strony o dowolnej tematyce.

Przyjrzymy się wzrostowi liczby stron internetowych dostępnych w Sieci. Z początku istnienia Internetu stron było stosunkowo niewiele. Na poniższym wykresie przedstawiony został wzrost liczby stron internetowych na przestrzeni ostatnich dwudziestu lat.

Wykres 1. Liczba stron internetowych na przestrzeni lat



Źródło: https://news.netcraft.com

*„W badaniu ze stycznia 2017 otrzymaliśmy odpowiedzi od 1 800 047 111 stron[…].”[[2]](#footnote-2)*. W badaniu z sierpnia 1995 stron tych było zaledwie 19 732. Oznacza to, że w ciągu niespełna 22 lat ich ilość wzrosła o prawie 2 miliardy. Spowodowane jest to wzrostem popularności Internetu, oraz rozwojem technologii. Poniżej umieszczony został opis poszczególnych punktów na wykresie (tłumaczenie własne):

1. Październik 1995: wsparcie Internet Explorer dla ciasteczek (ang. „cookies”). Listopad 1995: HTML 2.0
2. Grudzień 1995: Netscape przedstawia JavaScript. Styczeń 1996: Sun publikuje Java’ę
3. Cisco publikuje bilans obciążeń Local Director
4. Grudzień 1996: Macromedia Flash 1.0, CSS. Styczeń 1997: HTML 3.2
5. Rejestracja domeny Google.com
6. HTML 4.0
7. Wrzesień 1998: Akami, ICANN. Październik 1998: Rackspace oferuje serwery dedykowane
8. Marzec 1999: RSS. Kwiecień 1999: kończy się monopol rejestracji domen Network Solutions
9. Red Hat IPO. Startuje Blogger
10. VMware startuje z wirtualizacją serwerów
11. Startuje Wordpress
12. Xen 1.0
13. Startuje Facebook
14. Mozilla Firefox 1.0
15. Ajax
16. Startuje Twitter
17. Amazon EC2
18. Firma Apple prezentuje iPhone’a
19. HTML5
20. Google Chrome
21. CloudFlare

Rozwój technologii pozwolił na zwiększenie dostępu do Internetu jak i ułatwił korzystanie z jego zasobów. Jednak, jak można zauważyć na wykresie, duża liczba stron internetowych nie oznacza, że wszystkie są aktywne. W styczniu 2017 roku aktywnych stron było zaledwie 172 353 235. Jest to ledwo 9,6% (9,57%) wszystkich zarejestrowanych.

W jakim więc celu tworzonych jest tyle stron? Odpowiedzi na to pytanie może być wiele i każda z nich poprawna. Strony WWW, w zależności od potrzeb, tworzone są w różnym celu. Firmy prywatne tworzą stronę w celu zareklamowania swojej działalności, by klient mógł zapoznać się z ofertą firmy. Jest to też dobra forma kontaktu z klientem. W dzisiejszych czasach firma bez własnej strony internetowej prawie nie istnieje. Jest mniej konkurencyjna, przez co ma mniej klientów i mniejsze zyski. Państwowe strony internetowe (w Polsce zakończone przyrostkiem .gov.pl) mają na celu ułatwienie obywatelom dostępu do wszelakich informacji dotyczących prawa, które obowiązuje w danym państwie lub też informacji dotyczących danych przedsięwzięć czy regionów. Strony internetowe pełnią też rolę gazety (np. portal [www.gazeta.pl](http://www.gazeta.pl), [www.wp.pl](http://www.wp.pl) etc.), które są elektroniczną wersją papierowych gazet lub też służą do szybkiego przekazywania informacji w formie artykułów. Wszelkie najnowsze informacje od razu trafiają do czytelnika, a nie dopiero po wydrukowaniu i sprzedaży gazety. Dzięki własnej stronie internetowej przeróżne sklepy mogą poszerzyć swoją działalność do skali krajowej lub też światowej ([www.amazon.com](http://www.amazon.com), [www.othertees.com](http://www.othertees.com), [www.dragonus.pl](http://www.dragonus.pl), ect.) a nie tylko prowadzić lokalną, gdzie liczba klientów ograniczona jest do mieszkańców miasta i okolic, w którym sklep posiada siedzibę. Użytkownik jednak ma inny cel, gdy tworzy stronę internetową. Może być to blog, gdzie opisuje, co go interesuje, np. podróże, recenzje filmów, książek lub gier, odczucia względem czegoś, przepisy kulinarne, porady motoryzacyjne i wiele, wiele innych. Bardziej rozbudowane strony internetowe zwykle mają na celu gromadzenie informacji na różnorakie tematy. Może być to portal z przepisami kulinarnymi, dotyczący jakiegoś filmu lub serii, gry, książek, serialu lub też wszystkiego, co jest związane z kulturą.

Serwisy WWW pełnią przeróżne role, od informacyjnych po rozrywkę. Każdy może posiadać swoją własną stronę WWW, jeśli tylko ma pomysł na nią i spędzi trochę czasu na dopracowanie wyglądu oraz prowadzenie swojego portalu tak, by użytkownicy chętnie ją odwiedzali.

## 1.3 Użytkownicy stron internetowych

Strony internetowe tworzy się w różnych celach. Jednak podczas ich tworzenia trzeba zawsze pamiętać, do jakiej grupy odbiorców chcemy dotrzeć i co my sami chcemy przez to osiągnąć. Jeśli naszym celem jest pozyskanie klienta, to strona internetowa, jak i proces jej tworzenia, czy też strategia *User Experience* (strategia wrażeń użytkownika), będzie diametralnie różnić się od strony, która powstaje w celach informacyjnych czy hobbystycznych. By nasza strona internetowa mogła dobrze funkcjonować w Internecie musimy przede wszystkim dobrze się zastanowić, kim jest nasz odbiorca.

### 1.3.1 Użytkownik realny a unikalny

Strony internetowe tworzy się głównie dla użytkowników Internetu, czyli internautów. Na podstawie definicji zawartej w „Słowniku języka polskiego PWN” internauta to: *użytkownik Internetu[[3]](#footnote-3)*. Natomiast według słownika na portalu [www.portalwiedzy.onet.pl](http://www.portalwiedzy.onet.pl) to *osoba komunikująca się poprzez Internet, przemierzająca jego zasoby, szukająca w nim informacji.”[[4]](#footnote-4)*. Każda osoba, która korzysta z Internetu jest internautą, czyli realnym użytkownikiem. To właśnie oni sprawiają, że nasza strona internetowa jest popularna, „żyje”, daje satysfakcję z własnej pracy oraz generuje zyski.

Realny użytkownik (ang. *Real User*) to pojedyncza osoba, natomiast unikalny użytkownik (ang. *Unique User*) to termin, który określa przeglądarkę (liczbę pojedynczych programów „cookie”, które identyfikują użytkownika). Powyższe terminy służą głównie do obliczania „popularności” strony. Różnica pomiędzy nimi wynika z problemu liczenia realnych użytkowników odwiedzających stronę internetową. *„[…] problem pojawia się już na początku kiedy to z jednego komputera np. domowego może korzystać więcej niż jedna osoba, co sprawia że do jednej przeglądarki internetowej będzie zaglądać więcej ludzi. Kolejny problem do pokonania to fakt, że jedna osoba może:*

* *każdorazowo po wyjściu z przeglądarki kasować cookie,*
* *posiadać więcej niż jedną przeglądarkę internetową na komputerze,*
* *posiadać więcej niż jeden komputer.*

*Te problemy sprawiają, że policzenie w prosty sposób realnych użytkowników ze wskazanej strony internetowej jest praktycznie nie możliwe.”[[5]](#footnote-5)*.

Rys. 1 Realni użytkownicy a unikali



Źródło: <http://marketingnaluzie.pl/realny-uzytkownik-vs-unikalny-uzytkownik/>

Powyższy rysunek obrazuje problem z jakim muszą mierzyć się wydawcy podczas szacowania ilości realnych użytkowników, którzy odwiedzają ich stronę. Problem polega głównie na tym, że z jednego komputera (przeglądarki) może korzystać więcej niż jedna osoba co znacznie utrudnia ich policzenie.

### 1.3.2 Gemius/PBI i Google Analytics

Problemy z policzeniem realnych użytkowników sprawiają, że ich liczbę możemy tylko szacować. Do mierzenia liczby realnych użytkowników na stronie może posłużyć oprogramowanie stworzone przez Gemius (Gemius Explorer) lub oprogramowanie Google Analytics. Badania takie są przydatne, gdy chcemy dowiedzieć się, czy nasza strona internetowa jest popularna, czy użytkownicy chętnie ją odwiedzają, ile czasu na niej spędzają oraz dostarcza informacji, co możemy zmienić, by strona funkcjonowała lepiej i była bardziej przyjazna użytkownikowi. Firma Gemius wprowadziła na rynek własną metodę szacowania realnych użytkowników odwiedzających stronę. Do obliczania ich liczby służą wskaźniki, takie jak[[6]](#footnote-6):

* ***OW*** *– Liczba odsłon wygenerowana przez wszystkich użytkowników na danej witrynie*
* ***CD*** *– Liczba cookies, które na pewno istniały w badanym okresie*
* ***OD*** *– Liczba odsłon wygenerowana przez CD*

*Mając te dane obliczana jest liczba cookies występujących na witrynie, gdyby nie występowało zjawisko kasowalności cookies.*

***CW = (OW/ OD)\*CD***

*W podobny sposób obliczana jest liczba cookies zarejestrowana na wszystkich badanych witrynach, gdyby nie występowało zjawisko kasowalności cookies –* ***CP.***

*Zasięg danej witryny obliczany jest ze wzoru:*

***ZW = CW/CP***

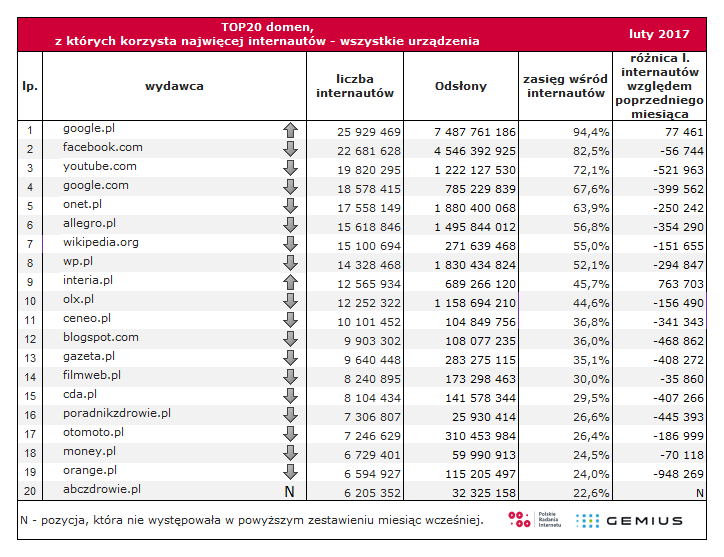
*Znając zasięg danej witryny (****ZW****) i populację internautów w danym okresie (****P****) (obecnie ok. 19 mln) obliczana jest liczba faktycznych użytkowników, którzy odwiedzili daną witrynę (Real Users).*

***UW = ZW\*P***

W Polsce pomiarem widowni internetowej zajmują się „Polskie badanie Internetu” czyli Gemius/PBI. *„Metodologia badania opiera się na pomiarze user-centric (dane gromadzone metodą panelową na grupie internautów, którzy zgodzili się na udział w badaniu) oraz audycie site-centric (dane o oglądalności witryn gromadzone na podstawie kodów zliczających umieszczonych w kodach źródłowych stron www).”[[7]](#footnote-7)*. Badanie to przydatne jest do gromadzenia danych, z których korzysta się przy tworzeniu ofert reklamowych. Gemius/PBI uwzględnia wszystkie typy urządzeń, PC, telefon oraz tablet. Ruch z PC dzieli na ruch z pracy i ruch z domu oraz rozpoznaje użytkownika na różnych urządzeniach.

Dzięki tym badaniom firmy pozyskują dane na temat ruchu na swojej stronie oraz o popularności wśród internautów. Według badania z lutego 2017 roku ranking TOP20 domen prezentował się następująco:

Rys. 2 Ranking TOP20 domen, luty 2017



Źródło: <http://pbi.org.pl/badanie-gemius-pbi/polski-internet-lutym-2017/>

Google Analytics jest narzędziem oferowanym przez Google i jest dostępne dla każdego użytkownika posiadającego stronę internetową i konto Google. Narzędzie to zawiera wiele przydatnych funkcji takich jak:

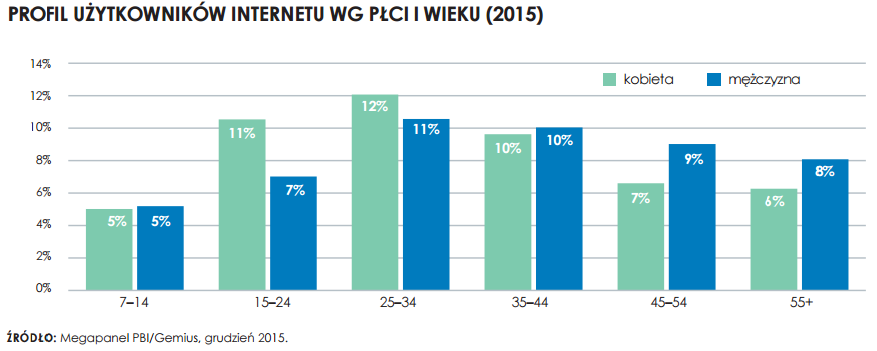
* Narzędzia analityczne
* Analityka treści
* Analityka danych mobilnych
* Pakiet konwersji
* Raporty społecznościowe
* Analityka reklam

Funkcje te pozwalają na tworzenie raportów na temat ruchu na stronie, nawet w czasie rzeczywistym, dzięki czemu możemy sprawdzić, ilu użytkowników przegląda stronę oraz co w danym momencie wyświetlają. Pozwalają dowiedzieć się także, jakiej treści szukają na stronie. „*Usługa Google Analytics nie tylko umożliwia pomiar wielkości sprzedaży i liczby konwersji, ale również zapewnia bieżący wgląd w to, jak użytkownicy korzystają z Twojej witryny, jak do niej dotarli i co możesz zrobić, by chętnie do niej wracali.*”[[8]](#footnote-8).

### 1.3.3 Profile internautów

Każdy internauta jest na swój sposób unikalny i ma różne potrzeby. Dlatego też tworząc stronę internetową musimy na początku zastanowić się, jaką grupę odbiorców chcemy pozyskać. *„Jeśli myślisz, że Twoimi klientami są wszyscy, to zastanów się jeszcze raz. Jeśli tego nie zrobisz, napotkasz olbrzymie problemy na etapie pozyskiwania klientów. Jak sądzisz, co jest łatwiejsze: skłonić wszystkich, by korzystali z Twojej aplikacji, czy skłonić do tego osoby, które naprawdę tej aplikacji potrzebują? […] Pod pojęciem klienta należy tu rozumieć grupę czy też segment ludzi dzielących wspólną potrzebę lub zmagających się z takim samym problemem.”[[9]](#footnote-9)*. Ważny jest cel strony WWW. Jeśli tworzymy stronę internetową dla jakiejś firmy, to chcemy zaprezentować jej produkty lub usługi, w taki sposób by klient chciał je zakupić. Blogi skierowane są do osób o podobnych zainteresowaniach lub dla hobbystów. Tak samo fora dyskusyjne, które zrzeszają mniejszą lub większą ilość internautów, którzy mogą dyskutować na jakiś temat lub udzielać porad. Tematyczne strony internetowe mają natomiast na celu udzielać informacji. Wiek oraz płeć użytkownika także mają duże znaczenie. Według „Raportu strategicznego: Internet 2015/2016” w Polsce z Internetu korzysta 76,4% Polaków.

Wykres 2 Profil użytkowników Internetu wg płci i wieku w 2015 roku



Źródło: Raport strategiczny: Internet 2015/2016

Z powyższego wykresu widać, że z Internetu korzystają nie tylko dorośli i nastolatkowie, ale także dzieci. Każdy z nich szuka w Internecie czegoś innego. Młodsi użytkownicy poszukują w sieci głownie rozrywki (gier, filmów etc.) oraz spędzają dużo czasu na portalach społecznościowych. Trochę starsi internauci zainteresowani są nie tylko rozrywką ale także informacjami na temat nowych technologii czy edukacji. Dorośli natomiast, bardziej niż młodsi, szukają w Internecie informacji na temat pracy, nieruchomości, motoryzacji i turystyki. Płeć także ma znaczenie. Kobiety chętniej przeglądają strony związane z modą niż te o motoryzacji. Wszystko też zależy od zainteresowań użytkowników. Jedni zainteresują się witryną o tematyce motoryzacyjnej, inni znów będą woleli poszukać strony o tematyce fantastycznej.

Profili użytkowników jest wiele i każdy z nich może przeplatać się z innym. Każdy internauta ma inne zainteresowania, potrzeby i cele. Zwraca uwagę na inne rzeczy. Inaczej zachowuje się przeglądając strony internetowe. Dlatego tak bardzo ważne jest, by tworząc własną witrynę skupić się NA tym segmencie użytkowników, dla których jest ona tworzona.

## 1.4 *User experience* i *web usability*

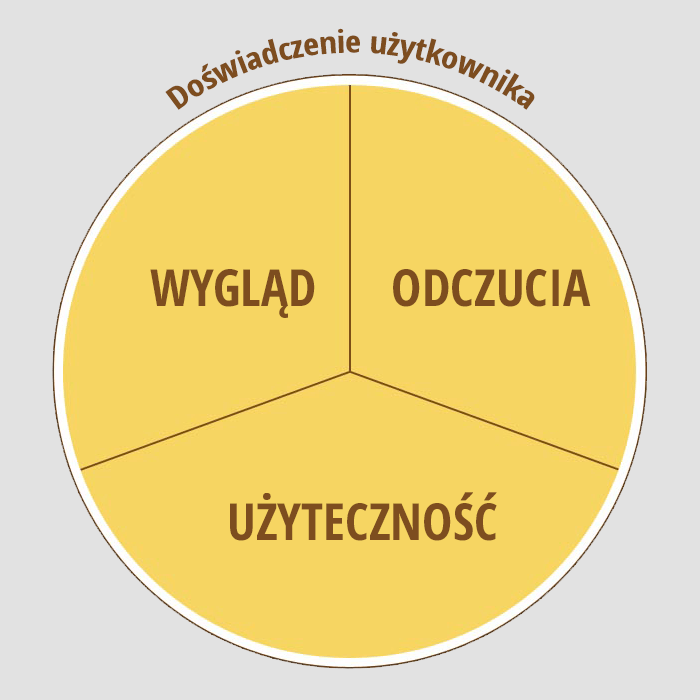
Użyteczność strony internetowej mocno wpływa na jej popularność. Stron w Internecie jest wiele, często o tej samej tematyce, ale nie każdą z nich internauci chętnie odwiedzają. Najczęściej dzieje się tak z powodu słabej funkcjonalności witryny. Jeśli stronę źle się przegląda lub nawigacja pozostawia wiele do życzenia, użytkownik zamyka stronę i szuka innej. Internauta nie chce zastanawiać się nad tym, gdzie ma szukać tego, czego potrzebuje. On chce, by potrzebna informacja pojawiła się tuż przed jego oczami. „*Jednym z wielu udokumentowanych faktów dotyczących korzystania z Internetu jest to, że ludzie poświęcają niewiele czasu na ‘czytanie’ stron WWW. Przeglądają je (rzucają na nie okiem), szukając słów lub wyrażeń przyciągających wzrok. […] Zazwyczaj staramy się skupić na słowach i wyrażeniach, które odpowiadają zadania, jakie mamy do wykonania lub temu, czego akurat poszukujemy. No i oczywiście słowom-kluczom, które na stałe przywiązane są do naszego układu nerwowego, np. ‘darmowy’, ‘wyprzedaż’, ‘seks’ czy nasze własne imię.*”[[10]](#footnote-10).

Tworząc stronę internetową należy zatem pamiętać o *user experience* (wrażeniach użytkownika) jak i *web usability* (użyteczności, funkcjonalności strony). Są to dwa nierozłączne pojęcia podczas tworzenia strony internetowej. Kluczem do sukcesu i popularności witryny nie jest sama treść lecz użytkownicy, którzy, korzystając z naszej strony WWW, są zadowoleni i chętnie do niej wracają.

### 1.4.1 *User experience*

„*Wrażenia użytkownika (UX) to doznania człowieka wynikające z kontaktu z interfejsem produktu cyfrowego w związku z wykonaniem jakiegoś zadania lub osiąganiem jakiegoś celu. Owszem, możemy zaryzykować stwierdzenie, że przykładem interfejsu jest klamka, a potem otworzyć drzwi i ruszyć autostradą prowadzącą do świata produktów w stu procentach fizycznych. W praktyce jednak utarło się, że termin ‘wrażenia użytkownika’ dotyczy przyjemnych lub nieprzyjemnych odczuć związanych z korzystaniem z produktu cyfrowego*”[[11]](#footnote-11). UX w projektowaniu strony internetowej jest tak samo ważny jak przy tworzeniu innych aplikacji. *„Projektowanie User Experience ma charakter interdyscyplinarny, zawiera w sobie elementy psychologii, ergonomii, sztuki, użyteczności, wzornictwa przemysłowego oraz technologii.”[[12]](#footnote-12)*.

Rys. 3 Trzy składowe UX



Źródło: <http://mobiletry.com/blog/psychologia-stron-internetowych>

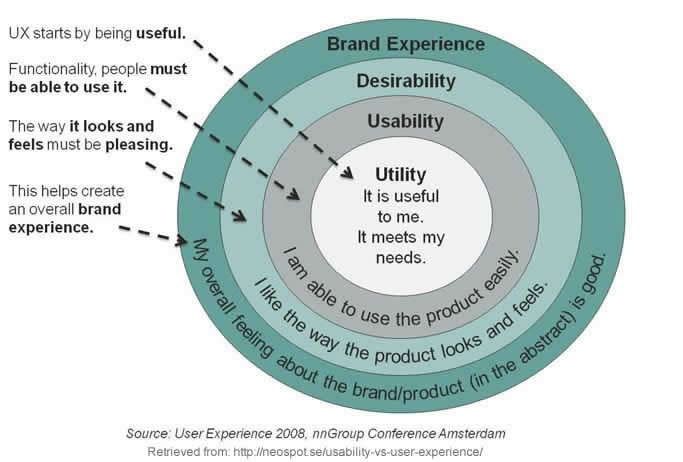
Projektowanie *user experience* to nic innego jak tworzenie czegoś, co jest wartościowe, tworzy emocjonalną więź z użytkownikiem i coś dla niego znaczy. UX jest związane z *web usability*, jest jednak pojęciem dużo szerszym. *User experience* składa się z czynników takich jak:

* Czynniki ludzkie
* Projektowanie
* Ergonomia
* Marketing
* Użyteczność
* HCI (*Human–computer interaction*)

Pojęcie *user experience* można podzielić na kategorie:

* Użyteczność (*utility)*­­­­­­– UX musi być użyteczny
* Funkcjonalność (*usability*)– ludzie muszą być wstanie tego używać
* Atrakcyjność (*desirability*)– wygląd i odczucia muszą być przyjemne
* *Brand experience*– pomaga w stworzeniu zupełnie nowej marki

Rys. 4 Związek pomiędzy user *experience* i *web usability*



Źródło: <http://usabilitygeek.com/the-difference-between-usability-and-user-experience/>

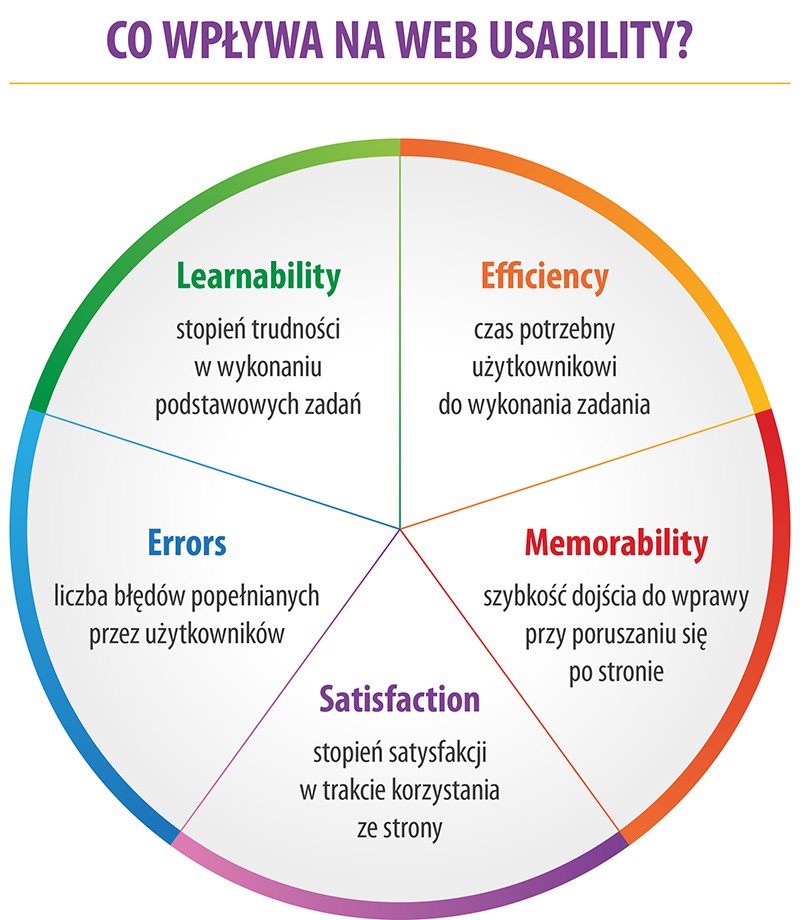
Pojęcie UX jest bardziej zawiłe od *web usability*. Skupia się na odczuciach użytkownika. Odpowiada na pytania: „kim naprawdę są moi użytkownicy?”, „co ich przyciąga?”, „co chcą osiągnąć?”. By znaleźć odpowiedź na te pytania nie ma innego sposobu jak zapytać samych zainteresowanych. *„Badania z udziałem użytkowników pozwalają zweryfikować, czy prace nad propozycją wartości zmierzają w odpowiednim kierunku. […] To proces kontrolny, opierający się na bezpośredniej informacji zwrotnej uzyskiwanej w ramach interakcji z użytkownikiem. W ten sposób zespołowi projektowemu łatwiej jest ocenić, czy wizja produktu jest pięknym marzeniem czy raczej sennym koszmarem.”[[13]](#footnote-13)*. Bez badań z udziałem potencjalnych użytkowników nie dowiemy się, czy to co chcemy im dać, będzie dla nich satysfakcjonujące.

Należy także pamiętać o tym, by internauta odczuwał satysfakcję z użytkowania naszej strony internetowej musi być ona funkcjonalna. I tutaj właśnie wchodzimy na podwórko *web usability*.

### 1.4.2 *Web usability*

*User experience* skupia się na odczuciach użytkownika, natomiast *web usability* jest skierowane w stronę funkcjonalności strony. Sprawia to, że te dwa pojęcia przeplatają się ze sobą. Jednak przy *web usability* mamy do czynienia bardziej z techniczną stroną tworzenia witryny internetowej. Użytkownik nie chce zastanawiać się długo, gdzie ma kliknąć by odnaleźć ważne dla niego informacje. Wygląd strony też ma znaczenie. Nieestetyczna lub zbyt „zaśmiecona” grafikami witryna może odstraszyć internautę. Strona internetowa powinna być prosta w użytkowaniu i przejrzysta.

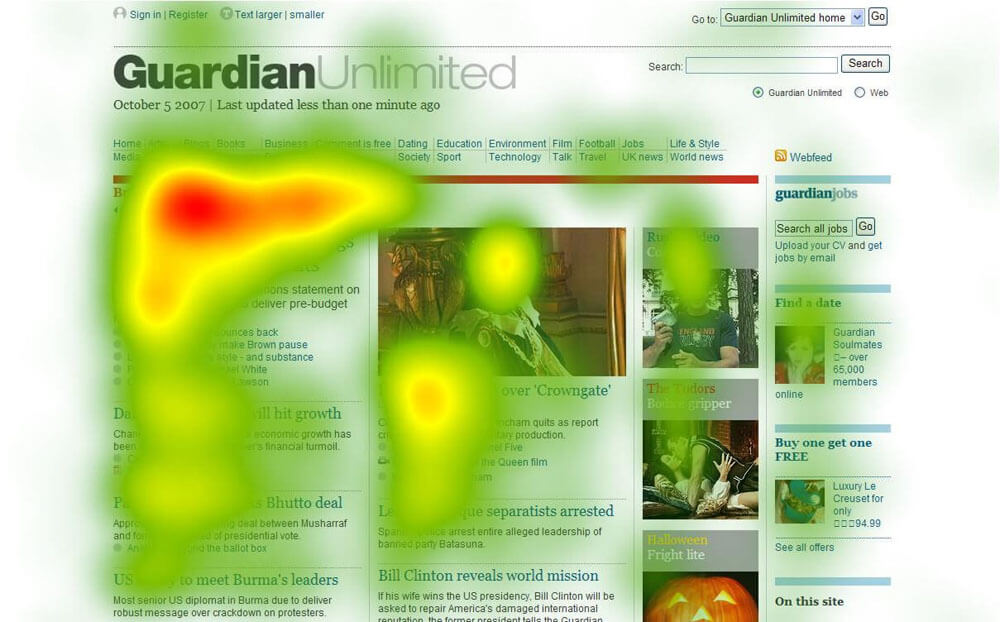
Rys. 5 Składowe *web usability*



Źródło: <https://www.eactive.pl/pozycjonowanie-stron/web-usability-czyli-jak-wazna-jest-uzytecznosc-stron-www/>

Powinniśmy także wykorzystać wiedzę o tym, jak użytkownicy przeglądają stronę w poszukiwaniu interesujących ich treści. Pomoże nam ona budować stronę, tak by ułatwić internautom to zadanie. Powinniśmy umożliwić im szybkie przejrzenie treści, w taki sposób by szybko i bez wysiłku znaleźli to czego szukają. *„Przeprowadzone przez Nilsen Norman Group badania dotyczące śledzenia wzroku pokazują, że użytkownicy w większości czytają strony internetowe według wzoru przypominającego literę "F". Po pierwsze, użytkownicy przeglądają stronę w od lewej do prawej, wzdłuż górnej części strony. Następnie ich wzrok przesuwa się nieco ku dołowi strony i znów przeglądają ją sposób horyzontalny, z tą różnicą, że drugi ruch jest krótszy od poprzedniego. Na koniec użytkownicy skanują lewą stronę witryny od góry do dołu, formując literę "F". Trzeci ruch jest zazwyczaj wolniejszy i bardziej systematyczny.”[[14]](#footnote-14)*.

Rys. 6 Mapa ciepła przeglądania strony



Źródło: <http://appletoncreative.com/2015/09/10/users-dont-read-they-scan-making-your-website-understandable-and-scannable/>

Powyższa mapa ciepła pokazuje obszary, na których użytkownik skupia swój wzrok. Czerwone obszary to te, które są najbardziej widoczne, żółte mniej, a zielone najmniej.

Kolejnym krokiem powinno być skupienie się na estetyce strony. Witryna nie powinna być zbytnio zaśmiecona grafikami, a jej treść powinna być czytelna. Akapity nie powinny być zbyt długie, czcionka czytelna i przyjemna dla oka. Kolory tła, czcionki i innych elementów na stronie powinny być spójne i współgrać ze sobą. Bardzo ważna jest hierarchia elementów. Istotne rzeczy powinny być wyróżnione (np. większa czcionka, pogrubienie, podkreślenie). Elementy powiązane ze sobą powinny też wiązać się ze sobą wizualnie.

Powinno się także podzielić stronę na obszary funkcjonalne. Użytkownik powinien od razu wiedzieć, na których częściach strony może się skupić, a które może pominąć. *„Wyniki niektórych badań sugerują, że użytkownicy bardzo szybko podejmują decyzję o tym, które części zawierają przydatne informacje, wskutek czego prawie nigdy nie interesuje ich pozostała zawartość- omijają ją tak, jakby w ogóle nie istniała.”[[15]](#footnote-15)*.

Dobrze zaprojektowana strona internetowa powinna być przyjazna użytkownikowi i zostawiać po sobie dobre wrażenie, dzięki czemu internauci będą do niej chętniej wracać. Musimy zadbać o to by była łatwa w użytkowaniu, przyjazna dla oka, lekka (nie obciążała za bardzo serwera i szybko się ładowała) i by ewentualne reklamy, które pozwolą nam zarabiać na stronie, nie zasłaniały zawartości. Trzeba pamiętać o tym, że witryna internetowa jest wizytówką. Źle wyglądająca i nieestetyczna wizytówka nie zachęci nikogo do kontaktu z nami.

# 2. Technologie wykorzystywane podczas tworzenia stron WWW

## 2.1 Wprowadzenie

Poczynając od roku 1994, kiedy powstała pierwsza specyfikacja HTML, aż do dzisiejszych czasów, technologie, dzięki którym powstają strony internetowe, ciągle się rozwijają. Dzisiaj stworzenie prostego portalu, sklepu czy bloga w Internecie jest dużo prostsze dzięki gotowym rozwiązaniom, które nie wymagają od użytkownika dużej wiedzy. Jednak jeśli chcemy, by nasza strona internetowa wyglądała zachęcająco i reagowała na czynności wykonywane przez użytkownika, to musimy posiąść trochę więcej wiedzy na temat technik jej tworzenia. Zaczynając od poznania podstaw HTML i CSS, poprzez JavaScript i PHP oraz obsługę baz danych. Dzięki temu, dalszej nauce oraz późniejszemu doświadczeniu w używaniu tych języków, w przyszłości będziemy w stanie tworzyć zaawansowane strony internetowe.

W tym rozdziale zostaną opisane podstawowe technologie pomocne przy tworzeniu strony internetowej oraz gotowe rozwiązanie wspomagające tworzenie i obsługę strony internetowej, Joomla! Wytłumaczony zostanie proces instalacji oraz podstawowe funkcje.

## 2.2 Języki stosowane do tworzenia stron internetowych

W dzisiejszych czasach istnieje wiele języków programowania stron internetowych, jak i wiele gotowych rozwiązań, które pozwalają na stworzenie strony WWW nawet tym internautom, którzy nie posiadają dużej wiedzy na temat tworzenia takiej strony od zera. Podstawowymi językami programowania stron internetowych są HTML, XHTML, CSS, PHP, JavaScript. Innymi językami są np. Ruby, Perl czy też Pyhton. Są to jednak mało popularne rozwiązania.

### 2.2.1 HTML i XHTML

Język HTML jest językiem znacznikowym. Opisuje strukturę informacji zawartych wewnątrz strony. Pierwsza specyfikacja języka HTML powstała w 1994 roku. Była ona jednak bardzo uboga. W roku 1996 ukazała się specyfikacja HTML3.2, która została wzbogacona o tabele, które diametralnie zmieniły oblicze stron internetowych. Następna specyfikacja, HTML4.0, pojawiła się w 1998 roku, a rok później ukazała się poprawiona wersja HTML4.01. Na tej wersji prace nad HTML zostały przerwane, a jego następcą został język XHTML. Język ten kładzie większy nacisk na zgodność z formatem XML. Jednak gdy rozpoczęto prace nad XHTML 2.0, który miał być niekompatybilny wstecz, W3C (*World Wide Web Consortium*) porzuciło pracę nad tym standardem i włączyło się w pracę nad HTML5. Standard ten zaczął być opracowywany przez niezależną grupę WHATWG (*Web Hypertext Application Technology Working Group*) w 2004 roku, w 2009 w prace włączyło się W3C by 28 października 2014 roku standard HTML5 został oficjalnie zarekomendowany przez W3C. „*Głównymi celami HTML5 są:*

* Wprowadzenie nowych elementów dla zwiększenia interaktywności i multimedialności stron internetowych.
* Wprowadzenie nowych znaczników semantycznych, aby uczynić sieć bardziej dostępną dla wszystkich.
* Oficjalne załączenie do specyfikacji rozszerzeń, dodanych w przeszłości na własną rękę przez producentów przeglądarek, które i tak stały się już wcześniej standardem de facto.
* Bardziej szczegółowe określenie sposobu obsługi błędów, tak aby dokumenty napisane przez niedouczonych webmasterów, wyświetlały się tak samo w każdej przeglądarce.
* Zachowanie kompatybilności wstecz, tak by użytkownicy starszych przeglądarek również mogli korzystać ze stron, napisanych w nowym języku.”[[16]](#footnote-16)

W rozdziale tym skupimy się na opisaniu struktury dokumentu napisanego w języku HTML5. Przykładowa strona wygląda w następujący sposób:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Tytuł strony </title>

</head>

<body>

Zawartość strony

</body>

</html>

Pierwszy wers to deklaracja typu dokumentu. Znacznik <html> oznaczają definicję dokumentu, <head> zawiera informacje o dokumencie, w tym element <meta>, który definiuje metadane (atrybut charset="UTF-8" odpowiada za kodowanie znaków dokumentu HTML) oraz znaczniki <body>, które zawierają „ciało”, czyli zawartość strony internetowej. Są to podstawowe elementy języka HTML. Trzeba także nadmienić, że znaczniki dzielą się na znaczniki otwierające np. <body> oraz zamykające np. </body>. Stosowanie znaczników zamykających „pomaga” przeglądarce w interpretacji napisanej strony internetowej oraz jej poprawne wyświetlanie. Brak stosowania znaczników zamykających może prowadzić do dwuznaczności i niepoprawnego wyświetlania danej treści lub danego elementu strony.

Dzisiejsze strony internetowe są napisane z użyciem całego szeregu znaczników. W standardzie HTML5 nowością są elementy sekcji strony internetowej. Stosowanie ich pozwala na zwiększenie czytelności kodu. Poniżej przedstawię kilka z nich.

Tabela 1 Nowe znaczniki HTML5 i ich opisy

|  |  |
| --- | --- |
| Znacznik | Opis |
| <header> | Jest to nagłówek strony. Najczęściej sekcja ta zawiera logo strony, wyszukiwarkę i menu strony. |
| <section> | Definiuje sekcję w dokumencie HTML. Zawartość wyróżnia się na tle strony. |
| <nav> | Zawiera nawigację strony internetowej, czyli główne linki służące do poruszania się po naszej stronie. |
| <main> | Wskazuje główną treść na stronie. |
| <footer> | Jest to stopka strony internetowej. Zazwyczaj zawiera informacje o autorze, odnośniki z menu, kontakt lub inne informacje na temat strony. |

Źródło: Opracowanie własne

Powyższe znaczniki są nowymi i najczęściej używanymi elementami podczas tworzenia strony internetowej. Jeśli chodzi o resztę znaczników to język HTML5 zawiera w swojej strukturze także starsze elementy tj.

Tabela 2 Inne wybrane znaczniki HTML i ich opisy

|  |  |
| --- | --- |
| Znacznik | Opis |
| <h1> <h2>  … <h6> | Znaczniki nagłówka. Najważniejszy <h1>, najmniej ważny <h6>. |
| <p> | Treść akapitu. |
| <ol> | Numerowana lista elementów. |
| <ul> | Wypunktowana lista elementów. |
| <li> | Element listy. |
| <table> | Znacznik tabeli. |
| <tr> | Wiersz tabeli. |
| <td> | Komórka tabeli. |
| <th> | Komórka nagłówkowa tabeli. |
| <form> | Znacznik formularza. |
| <input> | Kontrolka formularza. Zawiera różne atrybuty, które odpowiadają za wygląd i typ kontrolki (text, radio, checkbox, submit, itp.). |
| <select> | Znacznik kontrolki pola wyboru. |
| <option> | Opcja w kontrolce <select>. |
| <textarea> | Pole tekstowe. |
| <div> | Znacznik służący do utworzenia sekcji. |

Źródło: Opracowanie własne

Znaczników w języku HTML jest dużo więcej i każdy z nich może posiadać różne atrybuty, które odpowiadają za wygląd tekstu, strony, obrazków itp. Jednak wraz z zastosowanie kaskadowych arkuszy stylów (CSS) odchodzi się od stosowania ich w tym celu. Aby móc użyć arkusza stylów należy w sekcji <head> dodać:

<head>

<link rel=”stylesheet” href=”nazwa\_arkusza.css”>

</head>

gdzie rel=”stylesheet” oznacza relację między dokumentem HTML, a kaskadowym arkuszem stylów, a href=”nazwa\_arkusza.css” określa adres, pod którym znajduje się podany arkusz.

Można oczywiście zastosować znaczniki <style> w sekcji <head> i w nich zdefiniować wygląd poszczególnych elementów strony. Jednak przy dużej ilości definiowanych klas kod strony staje się nieczytelny i trudny do modyfikowania. Kaskadowe arkusze stylów i ich zastosowanie, zostaną dokładniej odpisane w kolejnym podrozdziale.

Oprócz CSS w dokumencie HTML można stosować skrypty napisane w języku JavaScript. Do ich oznaczania służy znacznik <script> umieszczony w sekcji <head>. Skrypt można powiązać z dokumentem w ten sam sposób co kaskadowe arkusze stylów, czyli można opisać go w dokumencie HTML lub w osobnym pliku i powiązać go za pomocą:

<script src=”nazwa\_pliku.js”></script>

, gdzie atrybut src=”nazwa\_pliku.js” określa adres pliku zawierającego skrypt JavaScript. Stosowanie skryptów na stronie internetowej zwiększa jej interaktywność. Dzięki reagowaniu na różne zdarzenia może służyć np. do sprawdzania poprawności wpisywanych danych do formularza. Jednak to jedna z niewielu funkcji stosowania JavaScript na stronach internetowych, zostaną one jednak omówione szerzej w kolejnych podrozdziałach.

Język HTML sam w sobie jest językiem prostym. Każdy może napisać prostą stronę internetową. Jednak jeśli chcemy by nasza strona była bardziej profesjonalna, wtedy musimy sięgnąć także do innych technologii. Na początku mogą one sprawiać problemy mniej doświadczonym użytkownikom, jednak przy odrobinie pracy i praktyce, każdy będzie mógł stworzyć dzięki nim funkcjonalną i przyjemną stronę internetową.

### 2.2.2 CSS

Kaskadowe arkusze stylów (CSS) służą do opisywania wyglądu strony. CSS jest oficjalnym mechanizmem formatowania tekstu i układu elementów. Dzięki użyciu arkuszy stylów, możemy zautomatyzować tworzenie strony internetowej oraz w razie potrzeby zmienić wygląd danych tekstów lub układ elementów, na stronie głównej oraz podstronach. Wystarczy, że dokonamy edycji tylko jednego pliku. Najważniejszymi zaletami stosowania kaskadowych arkuszy stylów są:

* Lepsza kontrola nad wyglądem czcionek i układem elementów na stronie
* Mniej pracy podczas edytowania wyglądu czcionek i układu elementów na stronie
* Szybsze ładowanie strony

Stosowanie plików CSS jest prawie pozbawione wad. W dzisiejszych czasach wszystkie przeglądarki je obsługują, a niespójności można obejść. Działanie kaskadowych arkuszy stylów jest bardzo proste. Wystarczy napisać reguły stylów, które określają, jak mają wyglądać dane elementy. Następnie trzeba dołączyć plik CSS do dokumentu HTML. Reguły deklaruje się w następujący sposób:

p {

color: red;

font-size: 14px;

}

Powyższa instrukcja sprawi, że tekst oznaczony znacznikiem akapitu <p> będzie wyświetlany w kolorze czerwonym o wielkości czcionki 14 pikseli. Reguły CSS są złożone z dwóch części. Część przed nawiasem klamrowym nosi nazwę selektora (należy pamiętać podczas ich definiowania, że CSS uwzględnia wielkość liter, dlatego trzeba być bardzo dokładnym), natomiast część umieszczona w nawiasie to deklaracja. Deklaracja także składa się z dwóch części: właściwości (czyli konkretna specyfika wyglądu) oraz wartość (czyli ustawienie jakie nadajemy danej wartości). Właściwość i jej wartość oddzielone są dwukropkiem, pojedyncza deklaracja natomiast musi być zakończona średnikiem.

Są dwa sposoby stosowania kaskadowych arkuszy stylów. Można używać zewnętrznych plików CSS (sposób ten został opisane w poprzednim rozdziale) lub wewnętrznych arkuszy stylów. W przypadku stosowania drugiego sposobu należy w kodzie strony w sekcji <head> dodać element <style>:

<style type=”text/css”>

p {

color: red;

font-size: 14px;

}

</style>

, gdzie atrybut *type* informuje przeglądarkę, że zawartość elementu zawiera reguły CSS.

Kaskadowe arkusze stylów posiadają wiele różnych typów selektorów. Do najczęściej stosowanych należą:

Tabela 3 Selektory elementów CSS

|  |  |  |
| --- | --- | --- |
| Selektor | Przykład | Znaczenie |
| Uniwersalny | \*{} | Każdy element . |
| Typu | p{} | Odnosi się do elementu p. |
| Klasy | .text{}  p.text{} | Odnosi się do tych elementów, których atrybut *class* ma wartość podaną po kropce. Jeśli przed znakiem kropki znajduje się odwołanie do elementu, to wtedy odnosi się on do wszystkich tych elementów, których atrybut *class* ma wartość podaną za kropką. |
| Identyfikatora | #text{} | Odnosi się do tych elementów, których atrybut *id* ma wartość podaną za znakiem *hash.* |
| Elementu dziecka | li > a{} | Odnosi się do dzieci innego, danego elementu. |
| Elementu potomnego | p a{} | Odnosi się do elementów potomnych innego, danego elementu. |

Źródło: Opracowanie własne

Tabela powyżej przedstawia tylko niewielką część selektorów jakimi mogą dysponować kaskadowe arkusze stylów. Istnieją bardziej zaawansowane selektory, które mogą odwoływać się do elementów na podstawie ich atrybutów i ich wartości.

Tabela 4 Selektory atrybutów CSS

|  |  |  |
| --- | --- | --- |
| Selektor | Przykład | Znaczenie |
| Istnienie | p[class] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class* |
| Równość | p[class=”cat”] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class* o wartości „cat” |
| Odstęp | p[class~=”cat”] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class*, którego jedna z wartości to „cat” |
| Początkowy fragment | p[class^=”cat”] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class*, który posiada wartość rozpoczynającą się ciągiem znaków „cat” |
| Fragment łańcucha | p[class\*=”cat”] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class*, który posiada wartość, która posiada ciąg znaków „cat” |
| Zakończenie | p[class$=”t”] | Odnosi się do wszystkich elementów *p*, które mają zdefiniowany atrybut *class*, którego wartość kończy się na literę „t” |

Źródło: Opracowanie własne

Każda reguła CSS oczywiście nie składa się tylko z samego selektora. By efekt był widoczny na naszej stronie internetowej musimy ją także zadeklarować. W tym pomocna jest znajomość właściwości CSS. Dzięki nim możemy zmieniać formatowanie tekstu, wygląd czcionek, rozmiary danych elementów i ich pozycjonowanie na stronie. Dzięki stosowaniu kaskadowych arkuszy stylów możemy również zmienić standardowy wygląd formularzy na ciekawszy. Właściwości CSS jest ponad setka, tabela poniżej prezentuje tylko niektóre z nich.

Tabela 5 Właściwości CSS

|  |  |  |
| --- | --- | --- |
| Lp. | Właściwość | Znaczenie |
| 1 | background | Określa kolor tła. |
| 2 | border | Określa rodzaj, kolor i grubość obramowania z czterech stron. |
| 3 | border-top | Określa rodzaj, kolor i grubość obramowania górnego. |
| 4 | border-color | Określa kolor obramowania z czterech stron. |
| 5 | border-top-color | Określa kolor obramowania górnego. |
| 6 | border-style | Określa rodzaj obramowania z czterech stron. |
| 7 | border-top-style | Określa rodzaj obramowania górnego. |
| 8 | color | Określa kolor tekstu. |
| 9 | font-family | Określa rodzaj czcionki. |
| 10 | font-weight | Określa grubość czcionki. |
| 11 | margin | Określa margines z czterech stron. |
| 12 | margin-top | Określa margines górny. |
| 13 | padding | Określa atrybuty odstępu z czterech stron. |
| 14 | text-align | Określa poziome wyrównanie tekstu. |
| 15 | z-index | Określa pozycję w stosie elementów . |

Źródło: Opracowanie własne

Jak można zauważyć CSS jest bardzo rozległy i tworzy wiele możliwości by nasza strona przyciągała internautę swoim wyglądem. Ułatwia on także prace nad szatą graficzną witryny co w przyszłości i w miarę rozwoju strony będzie miało dużo większe znaczenie. Jeśli poświęcimy nauce CSS wystarczająco dużo czasu, to nasza strona internetowa będzie mogła cieszyć oko użytkownika dzięki czemu będzie on czuł się dużo swobodniej.

### 2.2.3 JavaScript

JavaScript to skryptowy język programowania, który najczęściej używany jest przy tworzeniu stron internetowych. Dzięki użytym skryptom strona staje się bardziej dynamiczna poprzez reagowanie na zdarzenia. Może także służyć do sprawdzania poprawności danych wprowadzanych przez użytkownika w formularzach. Dzięki czemu można uniknąć różnorakich błędów. JavaScript służy także do tworzenia ciasteczek. Zastosowanie tego języka jest bardzo rozległe i wraz z rozwojem powstała technika Ajax (ang. *Asynchronous JavaScript and XML*) czyli asynchroniczny JavaScript i XML oraz jQuery czyli biblioteka JavaScript, która ułatwia pisanie skryptów w tym języku.

Należy wiedzieć, że skrypt napisany w języku JavaScript jest wykonywany przez przeglądarkę a nie po stronie serwera (tutaj należy krótko wspomnieć o środowisku uruchomieniowym JavaScript- Node.js, które działa po stronie serwera, dzięki czemu można uruchomić skrypt bez użycia przeglądarki). Język ten najczęściej stosowany jest między innymi do:

* Sprawdzania poprawności wprowadzonych danych w formularzach
* Tworzenia plików *cookies*
* Stwarza możliwość reakcji na różne zdarzenia (np. kliknięcie)

Strony internetowe, które posiadają skrypty napisane w języku JavaScript są bardziej zaawansowane od tych, które tych skryptów nie używają. Spowodowane jest to trudnością w nauczeniu się tego języka. By sprawnie posługiwać się językiem JavaScript musimy nauczyć się „myśleć” jak komputer. Jednak gdy już opanujemy podstawy to pisanie tych skryptów nie powinno sprawiać dużych trudności.

JavaScript działa na obiektach. Tablice, liczby, daty, teksty itp. to wszystko są obiekty. Wszystkie dane możemy podzielić na typy proste i złożone. Typ prosty służy do przechowywania liczb, tekstów lub wartości. JavaScript dokonuje konwersji tego typu na typ złożony i przywraca go na nowo do typu prostego. Dlatego nawet gdy mamy do czynienia z typem prostym to i tak wykonujemy operacje na typie złożonym. Drugi typ natomiast służy do przechowywania obiektów.

Obiekt to zbiór zmiennych i funkcji. Służy do tworzenia modelu czegoś rzeczywistego np. auta. W takim obiekcie każda zmienna staje się właściwością a funkcja metodą. Właściwość to informacja o obiekcie, metoda natomiast przedstawia zadanie obiektu. Tworzenie obiektu jest proste i może odbyć się za pomocą dwóch metod:

* Notacji literału
* Notacji z użyciem konstruktora

Pierwsza z nich przedstawia się następująco:

var car = {

mark: ‘Audi’,

model: ‘Q5’,

delivered: 5,

sold: 2,

available: function() {

return this. Delivered – this.sold;

}

};

gdzie *mark, model, delivered* i *sold* to właściwości a *available* to metoda. Natomiast słowo kluczowe *this* odwołuje się do właściwości danego obiektu. Druga metoda wygląda trochę inaczej:

var car = new Object();

car.mark = ‘Audi’;

car.model = ‘Q5’;

car.delivered = 5;

car.sold = 2;

car.available = function() {

return this. Delivered – this.sold;

};

Różnica polega na tym, że na początku tworzymy pusty obiekt *car* (połączenie słowa kluczowego *new* oraz konstruktora Object()). Dodawanie właściwości i metod odbywa się za pomocą notacji z użyciem kropki. W ten sposób uzyskujemy dostęp do obiektu i nadajemy mu dane właściwości i metody. Notacji z użyciem kropki używamy za każdym razem gdy chcemy uzyskać dostęp do właściwości lub metod obiektu. W tym przypadku kropkę nazywamy operatorem elementu składowego. Jeśli chcemy np. wyświetlić nazwę modelu auta lub ile jeszcze takich aut zostało do sprzedania wtedy możemy stworzyć nową zmienną w podany poniżej sposób:

var modelName = car.model;

var availableCars = car.available();

,gdzie nazwa obiektu musi znajdować się po lewej stronie operatora elementu składowego, po prawej natomiast nazwa właściwości lub metody (nosi nazwę elementu składowego).

W języku JavaScript istnieje klkadziesiąt obiektów a większość z nich posiada własne metody i własności.

Tabela 6 Wybrane obiekty JavaScript i ich niektóre metody i własności

|  |  |  |  |
| --- | --- | --- | --- |
| Obiekt | Opis | Metody | Własności |
| anchor | Po kliknięciu w obszar, użytkownik przenoszony jest do innego odsyłacza. | Brak | Brak |
| button | Używany w formularzach, definiuje przycisk. | click() | name  value |
| Date | Pozwala na pracę z datą i czasem. | getDate()  getDay()  getTime()  getYear()  setDate()  setTime()  setYear() | Brak |
| document | Zawiera informacje o stronie oraz informuje przeglądarkę o sposobach wyświetlania strony internetowej. | clear()  close()  open()  write()  writeln() | alinkColor  anchors  bgColor  linkColor  links  title  vlinkColor |
| hidden | Wykorzystywany do ukrycia pola *input*. | Brak | name  value |
| Math | Przechowuje wartości matematyczne takie jak: funkcje matematyczne czy wartość liczby Pi. | cos(liczba)  log(liczba)  pow(liczba)  random() | E  LN10  PI  SQRT2 |
| navigator | Zwraca informacje o przeglądarce użytkownika. | Brak | appCodeName  appName  appVersion  userAgent |
| window | Reprezentuje okno przeglądarki, najważniejszy obiekt w hierarchii obiektów. | alert()  close()  open() | document  frame  location  name  self  window |

Źródło: Opracowanie własne

JavaScript przy tworzeniu stron internetowych wykorzystywany jest głównie do walidacji formularzy oraz reagowania na zdarzenia. Zdarzenia dzielą się na:

* UI (ang. user interface)
* Klawiatury
* Myszy
* Aktywności
* Formularzy

Zdarzenia aktywują kod JavaScript w trzech krokach. Autor strony internetowej musi wybrać węzeł skryptu, do którego ma się odwołać. Następnie musi wskazać zdarzenie, które spowoduje reakcję po wykonaniu akcji z danym elementem. Na koniec należy zdefiniować kod, który zostanie uruchomiony po wystąpieniu danego zdarzenia. Aby dołączyć zdarzenie do elementu należy użyć instrukcji przedstawionej poniżej:

element.onZdarzenie = nazwaDanejFunkcji;

gdzie *element* to węzeł elementu, *onZdarzenie* to dane zdarzenie poprzedzone przedrostkiem *on*, a *nazwaDanejFunkcji* to nazwa funkcji, która ma zostać wykonana po wystąpieniu zdarzenia. Poniżej przedstawione zostały wybrane zdarzenia, z którymi użytkownik może się spotkać podczas przeglądania różnych stron internetowych.

Tabela 7 Wybrane zdarzenia JavaScript

|  |  |
| --- | --- |
| Zdarzenie | Opis |
| load | Zakończenie wczytywania strony. |
| scroll | Przewinięcie strony przez użytkownika. |
| keypress | Wstawienie znaku. |
| click | Naciśnięcie i zwolnienie przycisku myszy przez użytkownika nad tym samym elementem. |
| mouseover | Przesunięcie kursora myszy nad danym elementem przez użytkownika. |
| submit | Wysłanie formularza przez użytkownika. |

Źródło: Opracowanie własne

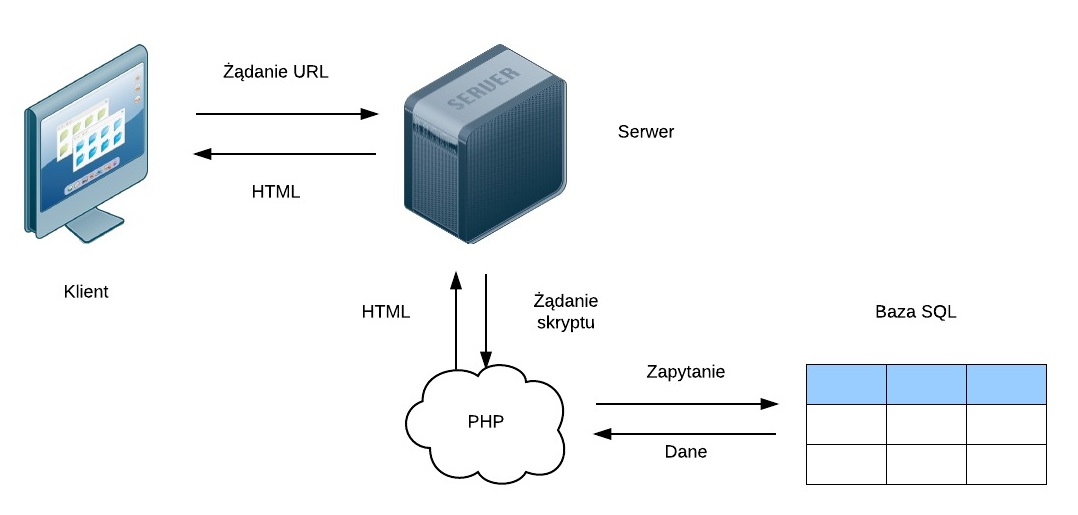
Język JavaScript jest bardzo rozległy i daje mnóstwo możliwości podczas tworzenia stron internetowych. Jednak jest to język programowania i jego nauka może zająć trochę czasu. I chociaż w Internecie można znaleźć ogrom przykładów zastosowania oraz rozwiązań to dla osób bez minimalnej wiedzy na temat programowania będzie to całkowicie niezrozumiałe. Dlatego zanim zaczniemy myśleć o dołączeniu skryptów napisanych w języku JavaScript do naszej strony warto zacząć od kursu, który nas przeprowadzi przez podstawy. Następnym krokiem, jeśli nie czujemy się wystarczająco na siłach, można znaleźć bardziej zaawansowany kurs lub skorzystać z pomocy np. serwisu <http://codewars.pl>, oferuje on nie tylko naukę języka JavaScript ale także kilkunastu innych, w tym:

* PHP
* SQL
* C#
* Java
* inne

2.2.4 PHP

PHP to skryptowy język programowania, który najczęściej stosuje się do dynamicznego generowania stron internetowych. Jest to potężne narzędzie w rękach twórców stron internetowych, które wymaga doświadczenia by w pełni wykorzystać jego możliwości. Język PHP jest dostępny jako *open source* oraz może współpracować z prawie każdym systemem operacyjnymi oraz z wieloma bazami danych. Dlatego też PHP jest najczęściej stosowany gdy strona internetowa korzysta z baz danych.

Rys. 7 Działanie stron, które wykorzystują PHP i bazę danych SQL



Źródło: Opracowanie własne

Działanie takich stron wygląda następująco:

1. Internauta wchodzi na stronę internetową, wysyła żądanie URL.
2. Serwer przetwarza kod PHP.
3. Opcjonalnie do bazy danych (jeśli strona korzysta z takowej) zostaje wysłane zapytanie i następuje odebranie danych.
4. Serwer odczytuje instrukcje zapisane w języku PHP, przetwarza je i wysyła do klienta w formie kodu HTML.

Jak można zauważyć, PHP działa po stronie serwera, więc by móc mieć wgląd na stronę, która została napisana przy użyciu języka PHP należy mieć zainstalowany serwer, który obsługuje ten język (np. Apache). Bez tego kod PHP nie będzie działał i strona nie wyświetli się poprawianie. Sam kod natomiast musi być umieszczony w odpowiednich znacznikach:

<?php

?>

a plik, w którym zapiszemy kod strony musi mieć rozszerzenie .php. Należy także pamiętać, że każdy błąd w kodzie także będzie skutkował nieprawidłowościami w wyświetlaniu się strony.

Aby zacząć pisanie stron internetowych przy użyciu języka PHP należy zapoznać się z kilkoma przydatnymi funkcjami. Jedną z nich jest funkcja *echo()* lub *print()*. Przesyłają one dane do przeglądarki dzięki czemu możliwe jest wyświetlenie tekstów, tabel, zmiennych i innych. Prostym przykładem ich zastosowania jest:

echo 'Ala ma kota';

echo "Ala ma kota";

Można korzystać z cudzysłowów czy apostrofów zamiennie. Różnica polega na tym, że w przypadku gdy chcemy wyświetlić wartość wcześniej zdeklarowanej zmiennej to musimy ją umieścić w cudzysłowach. W innym przypadku wyświetli się dokładnie to co umieściliśmy w apostrofach. Innymi słowy: jeśli mamy zmienną $x = 5; (zamienne w języku PHP deklaruje się poprzedzając ich nazwę znakiem dolara) w przypadku gdy wydamy polecenie:

echo "Ala ma $x kotów";

, to na ekranie wyświetli nam się napis: Ala ma 5 kotów. W przypadku użycia apostrofów na ekranie zobaczymy: Ala ma $x kotów. Oprócz tego język PHP posiada kilka znaków, które umieszczone w cudzysłowach mają specjalne znaczenie.

Tabela 8 Znaki specjalne PHP

|  |  |
| --- | --- |
| Znak | Znaczenie |
| \" | Wyświetlenie cudzysłowu. |
| \' | Wyświetlenie apostrofu. |
| \\ | Wyświetlenie backspace’a. |
| \n | Przejście do nowego wiersza. |
| \r | Powrót karetki (powrót do początku linii). |
| \t | Użycie tabulatora. |
| \$ | Wyświetlenie znaku dolara. |

Źródło: Opracowanie własne

Jeśli chcemy przechować jakąś wartość, wtedy używamy wcześniej wspomnianych zmiennych. Ich nazwę, która może być kombinacją dowolnych liter, cyfr i znaków podkreślenia, musi poprzedzać znak dolara. Jedyne zastrzeżenia są takie, że bezpośrednio po znaku dolara nie może znajdować się cyfra oraz, że w języku PHP ważna jest wielkość liter. Innymi słowy $x i $X nie oznaczają tej samej zmiennej. Zmienne mogą przechowywać wartości logiczne, liczby, teksty, tablice i obiekty.

Wartość zmiennych, jak sama nazwa wskazuje, może ulec zmianie podczas wykonywania skryptu. Czasem jednak potrzebujemy wartości stałych. Stałą definiuje się w następujący sposób:

define('NAZWA', 'wartosc');

gdzie *define()* to funkcja języka PHP przeznaczona specjalnie do definiowania stałych.

Podczas tworzenia strony internetowej z użyciem języka PHP przydatne może okazać się łączenie łańcuchów, czyli np. łączenie w jeden ciąg wartości kilku zmiennych. Do łączenia łańcuchów stosuje się kropkę, która w tym przypadku nosi nazwę operatora konkatenacji. Zastosowanie można pokazać na prostym przykładzie:

$imie = 'Stefan';

$nazwisko = 'Kowalski';

$imie\_i\_nazwisko = $imie.' '.$nazwisko;

W tym przypadku połączyliśmy imię i nazwisko w jedną zmienną, dodając między zmienną $imie i $nazwisko, spację.

Najczęstszym wykorzystywanym procesem podczas tworzenia strony internetowej jest obsługa formularzy. Czy to przy rejestracji użytkowników, przesyłania opinii czy wypełniania danych do wysyłki, formularze stanowią ważną część serwisu. Przesyłanie danych, którymi użytkownik wypełnił formularz może odbyć się za pomocą dwóch metod: *get* oraz *post*. Metoda *get* służy głównie do przesyłania niewielkich ilości danych, natomiast metodę *post* stosuje się gdy chcemy wywołać jakąś akcję, np. zaktualizować dane w bazie danych. W celu stworzenia formularza musimy zastosować następujący znacznik HTML:

<form action ="akcja.php" method="post"></form>

gdzie atrybut *action* określa dokąd dane z formularza zostaną wysłane, a atrybut *method* określa metodę wysyłania danych. Działanie takich formularzy jest proste:

1. Internauta wypełnia formularz.
2. Po zatwierdzeniu formularza przez użytkownika, dane są wysyłane do podanego w atrybucie *action* skryptu.
3. Zostaje wywołany skrypt PHP.

Podany skrypt PHP może zawierać różne instrukcje dotyczące wykorzystania wprowadzonych danych. Może wyświetlić podziękowania za wypełnienie formularza, wysłać dane do bazy danych lub na adres e-mail. Wszystko zależy od przeznaczenia formularza.

Kolejną przydatną funkcją PHP jest połączenie z bazą danych. By móc pobrać dane z bazy należy najpierw zdefiniować połączenie z nią. W tym celu będziemy potrzebować: nazwy użytkownika, hasło do bazy, adres hosta oraz nazwę bazy danych, z którą chcemy się połączyć. Musimy także połączyć się z serwerem. Służy do tego odpowiednia funkcja. W zależności od tego z jakiego systemu zarządzania bazami danych korzystamy może ona przybrać postać:

$dbconnMySQL = mysqli\_connect(adres\_hosta, nazwa\_uzytkownika, haslo, nazwa\_bazy);

$dbconnPostgreSQL = pg\_connect("host=adres\_hosta dbname=nazwa\_bazy user=nazwa\_uzytkownika password=haslo");

Pierwsza zmienna przechowuje informacje dotyczące połączenia z systemem MySQL, druga natomiast z PostreSQL. Jednak do tworzenia stron internetowych najczęściej stosuje się system zarządzania bazami danych MySQL. Warto także zdeklarować nie tylko zmienną, dzięki której połączymy się z serwerem ale także kilka stałych, które przechowają informacje potrzebne do połączenia się z bazą. Jeśli nasza strona korzysta z bazy danych na kilku podstronach najlepiej jest stworzyć osobny plik PHP, który będzie zawierał te dane, dzięki czemu nie będziemy zmuszeni do deklaracji tych samych stałych w kodzie różnych podstron ale wystarczy, że dołączymy ten plik do odpowiedniego pliku.

Języka PHP można używać nie tylko do tworzenia stron internetowych, ale także do pisania aplikacji. Ma bardzo szerokie zastosowanie i wyżej opisane podstawy to tylko niewielki czubek góry lodowej jego możliwości. Dlatego warto przejść przez kurs, który pomoże nam w opanowaniu podstaw programowania w PHP, zanim zaczniemy pracę nad ambitniejszymi projektami.

## 2.3 Systemy zarządzania relacyjnymi bazami danych SQL

System zarządzania relacyjną bazą danych to program, który służy do korzystania z relacyjnej bazy danych. Pozwala użytkownikowi na wykonywanie operacji na danych takich jak:

* Wprowadzanie
* Odczytywanie
* Aktualizację
* Przetwarzanie
* Wyszukiwanie

System zarządzania bazą danych pozwala także użytkownikowi na stworzenie nowej bazy danych, umożliwia dostęp do danych za pomocą języka SQL oraz zapewnia ochronę danych. Do najpopularniejszych systemów zarządzania relacyjnymi bazami danych należą: MySQL, PostgreSQL oraz SQLite.

Bazą danych natomiast nazywamy zbiór danych, który zapisany jest zgodnie z dokładnie określonymi regułami. Zbiór danych powinien zawierać istotne dane, jego struktura powinna być zgodna z określonym modelem danych, powinna przedstawiać związki pomiędzy danymi, które muszą być użyteczne i znajdować zastosowanie praktyczne.

Bazy danych często zawierają bardzo dużą ilość danych i to właśnie systemy zarządzania relacyjnymi bazami danych umożliwiają dostęp do nich, pracę i wykorzystanie ich w danym celu. Dzięki tym systemom możemy efektywnie zarządzać danymi.

### 2.3.1 MySQL

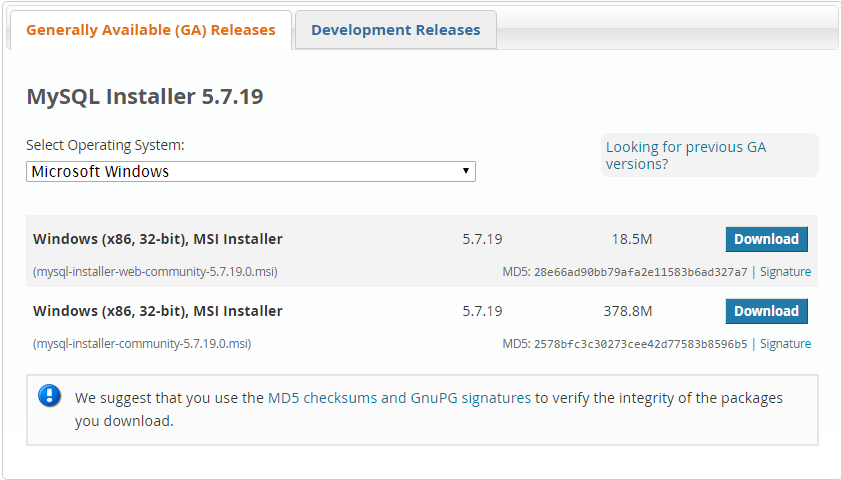
MySQL jest systemem zarządzania relacyjnymi bazami danych na licencji GPL lub komercyjnej. Od roku 2010 jest rozwijany przez firmę Oracle. Jest to najpopularniejszy system do zarządzania bazami danych, dzięki czemu pomoc do niego jest bardzo łatwo dostępna. MySQL obsługuje następujące typy danych:

* Tinyint
* Smallint
* Mediumint
* Integer
* Bigint
* Float
* Double
* Decimal
* Date
* Datetime
* Timestamp
* Time
* Year
* Char
* Varchar
* Tinytext
* Text
* Mediumtext
* Longtext
* Enum
* Set

MySQL jest łatwy w obsłudze, funkcjonalny, szybki i wydajny a przy tym bezpieczny. Swoją szybkość i wydajność MySQL zawdzięcza ograniczeniu niektórych funkcji co przy niektórych aplikacjach może być kłopotliwe, jednak do tworzenia stron internetowych nadaje się idealnie.

Instalacja serwera MySQL dla systemu Windows jest bardzo prosta. Wystarczy pobrać najnowszą wersję ze strony: <https://dev.mysql.com/downloads/installer/> i wybrać wersję instalatora (pełny lub *web*).

Rys. 8 Instalatory darmowej wersji MySQL

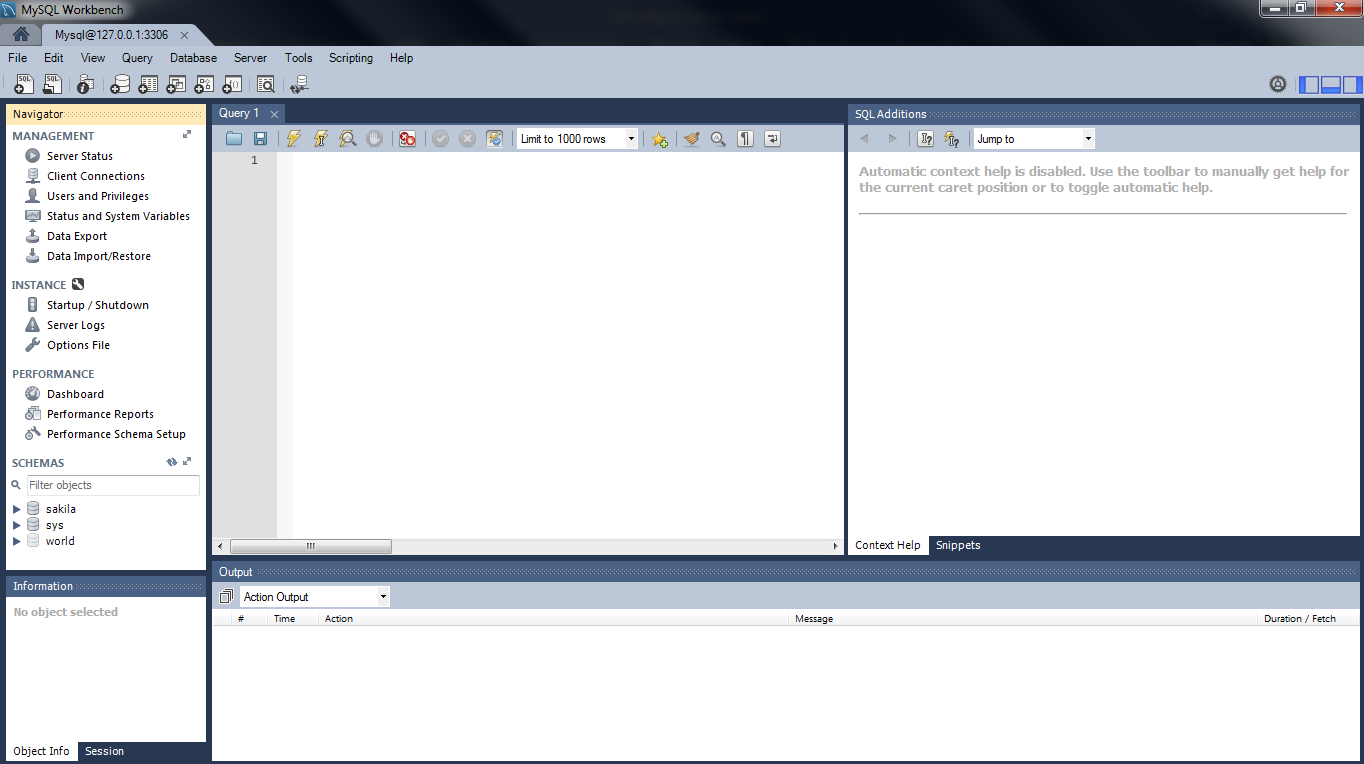


Źródło: Zrzut okna przeglądarki

Głównymi elementami jakie należy zainstalować są: MySQL Serwer oraz MySQL Workbench. Podczas instalacji należy postępować zgodnie z instrukcjami instalatora. Następuje wtedy wstępna konfiguracja serwera i podajemy wtedy informacje takie jak nazwa administratora bazy danych (domyślnie *root*) oraz hasło.

Następnie, gdy instalacja zakończy się, należy stworzyć nową bazę danych. W tym celu otwieramy MySQL Workbench i logujemy się na serwer za pomocą nazwy użytkownika i hasła podanych podczas instalacji. Naszym oczom powinien ukazać się taki widok:

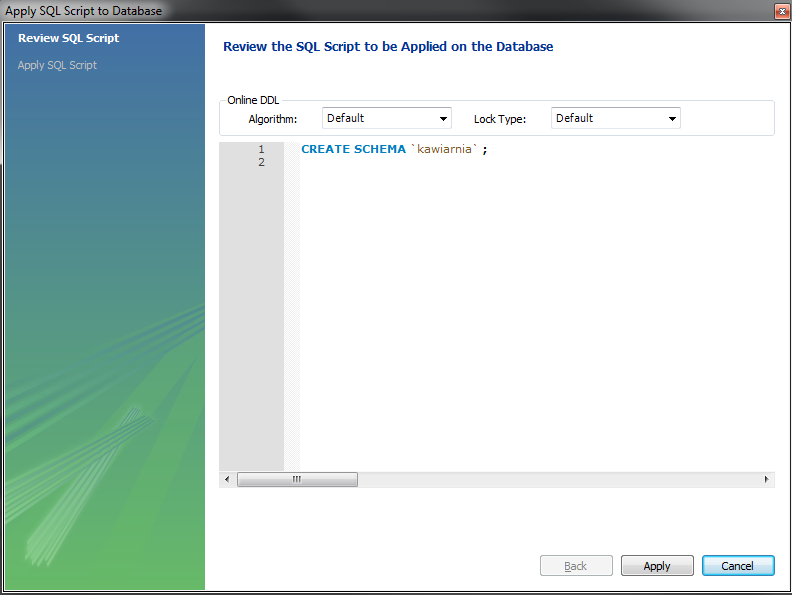
Rys. 9 Główny ekran MySQL Workbench



Źródło: Zrzut ekranu

By utworzyć nową bazę, klikamy przycisk *Create a new schema in the connected server*, w polu *Name* wpisać nazwę bazy, wybrać odpowiednie dla nas kodowanie znaków i na końcu kliknąć przycisk *Apply*. Wyświetli nam się okno:

Rys. 10 Tworzenie nowej bazy danych



Źródło: Zrzut ekranu

Klikamy *Apply*, który sprawi, że zostanie uruchomiony skrypt tworzący nową bazę danych. Później opcjonalnie możemy dodać uprawnienia nowemu użytkownikowi by mógł zarządzać nowo utworzoną bazą danych.

W ten oto prosty sposób mamy podstawę do dalszej pracy przy tworzeniu strony internetowej, która korzysta z baz danych. Należy także nadmienić, że MySQL Workbench nie jest jedynym narzędziem do administracji naszą bazą danych. Możemy w tym celu używać także phpMyAdmin, który jest darmowym narzędziem, napisanym w języku PHP, które działa za pośrednictwem przeglądarki.

### 2.3.2 PostgreSQL

PostgreSQL to kolejny system zarządzania relacyjnymi bazami danych z licencją GPL. Jest najbardziej zaawansowanym systemem na licencji *open source* oraz najbardziej zgodnym ze standardami. Jest mniej popularny od MySQL ale dużo bardziej funkcjonalny i przeznaczony do bardziej skomplikowanych operacji bazodanowych. PostgreSQL obsługuje poniższe typy danych:

* Bigint
* Bigserial
* Bit
* Bit varying
* Boolean
* Box
* Bytea
* Character varying
* Character
* CIDR
* Circle
* Date
* Double precision
* Inet
* Integer
* Interval
* Line
* LSEG
* Macaddr
* Money
* Numeric
* Path
* Point
* Poligon
* Real
* Smallint
* Serial
* Text
* Time
* Timestamp
* Tsquery
* Tsvector
* Txid\_snapshot
* Uuid
* Xml

PostgreSQL posiada doświadczoną społeczność, która zawsze służy pomocą i radą. Ogrom funkcjonalności systemu powoduje, że jest mniej wydajny od MySQL i trudniej jest znaleźć hosting, który obsługuje serwer PostgreSQL’a. Jednak wybór tego systemu jest najlepszym z możliwych gdy liczy się dla nas integralność danych czy jeśli pracujemy nad bardzo złożonym projektem.

Instalacja serwera PostgreSQL jest bardzo prosta i nie wymaga od nas dużych umiejętności. Pierwszym krokiem jest pobranie instalatora odpowiedniego dla naszego systemu operacyjnego ze strony <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads#windows>. Podczas instalacji wybieramy folder, w którym zostaną wypakowane pliki serwera oraz folder, w którym będą przechowywane dane serwera. Można je ustawić samemu lub zostawić domyślne. Kolejnym krokiem jest ustawienie hasła administratora serwera. Potem wybieramy numer portu, na którym serwer będzie nasłuchiwał żądań (domyślnie to 5432). Na koniec instalacji możemy jeszcze zaznaczyć opcje *Launch Stack Builder at exit*. Jest to narzędzie do pobierania i instalowania dodatkowych sterowników i aplikacji do PostgreSQL’a. Jeśli nie potrzebujemy nic więcej to odznaczamy tą opcję i klikamy przycisk *Finish*.

Domyślnym narzędziem do administracji bazami danych jest pgAdmin. Jest prosty w użyciu i tworzenie baz danych oraz tabel przy jego pomocy nie powinno być problematyczne.

Gdy już zainstalujemy serwer PostgreSQL i zapoznamy się z dostępnymi narzędziami do administracji bazami danych, możemy zacząć pracę nad aplikacjami bazodanowymi.

## 2.4 Gotowe rozwiązanie Joomla!

Jeśli nie chcemy zaczynać tworzenia strony internetowej od podstaw lub jeśli nie czujemy się na siłach by taką stronę stworzyć możemy posłużyć się szeregiem dostępnych w sieci gotowych rozwiązań. W tym celu można użyć oprogramowania CMS (ang. *Content Management System*) czyli systemu zarządzania treścią. Pozwala ono na utworzenie serwisu internetowego i jego administrację poprzez przeglądarkę internetową. Jest to najprostszy sposób na stworzenie własnej strony internetowej. Jednak obsługa takiego rozwiązania także wymaga nieco wiedzy na temat funkcjonowania takiego systemu.

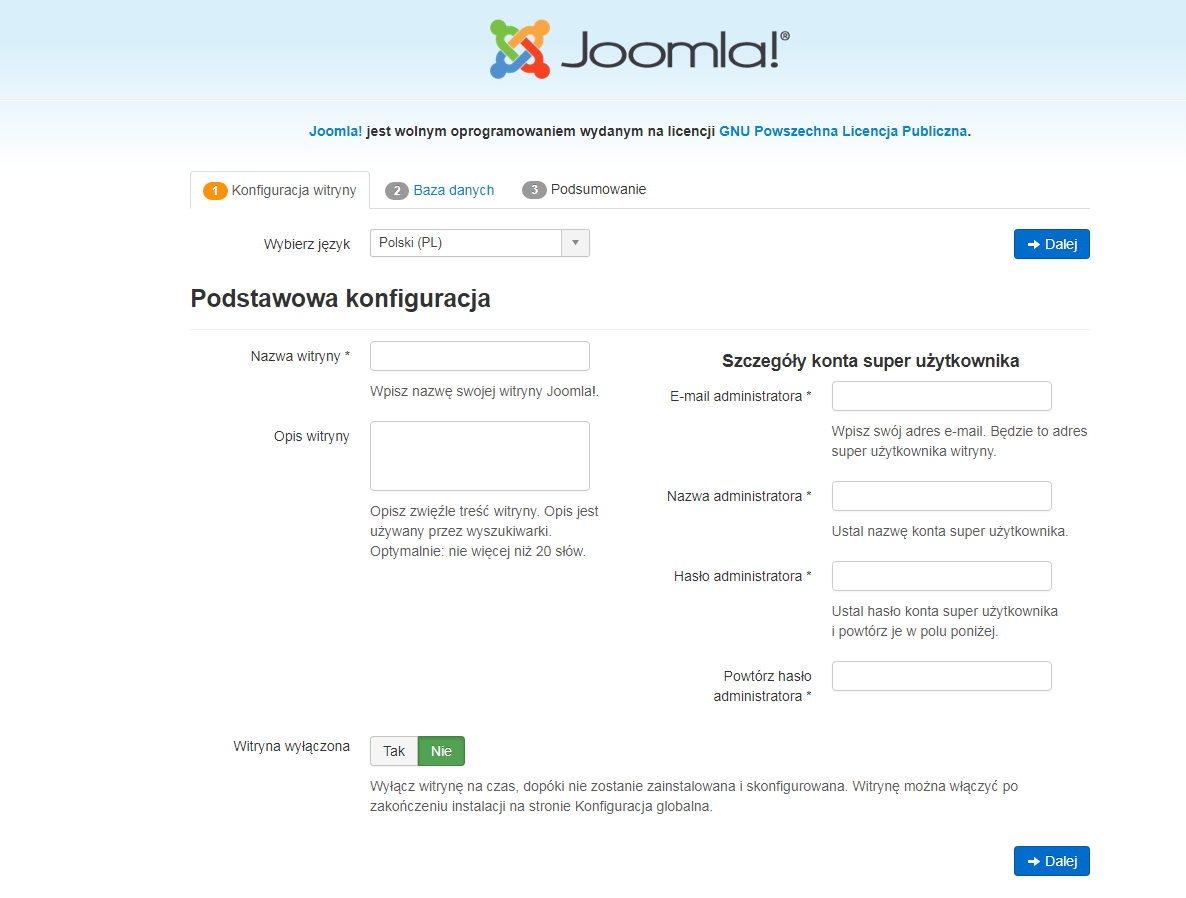
Najpopularniejszymi z nich są WordPress, Joomla! oraz Drupal. Istnieją także systemy do zarządzania sklepem internetowym takie jak: PrestaShop, Magneto, OpenCart i wiele innych. Większość z nich jest na licencji *open source* dzięki czemu możemy z nich korzystać za darmo. W tym rozdziale omówiony zostanie system Joomla!, w szczególności instalacja i podstawowe funkcje w panelu administracyjnym.

### 2.4.1 Instalacja systemu Joomla!

Zanim zaczniemy instalację systemu Joomla! potrzebujemy serwera WWW. Możemy wykupić hosting i od razu instalować nasz CMS na wybranym serwerze lub najpierw stworzyć stronę na lokalnym serwerze zainstalowanym na naszym komputerze. W tym celu mogą posłużyć narzędzia takie jak WampServer lub XAMPP. Są to pakiety, które składają się z serwera Apache, bazy danych MySQL i interpretera PHP. Oba są darmowe i łatwe w obsłudze. Dla naszych potrzeb wykorzystywany będzie WampServer, więc instalacja przebiegać będzie na serwerze lokalnym (*localhost*).

Pierwszym krokiem jest popranie pakietu Joomla! ze strony internetowej dostawcy: <https://www.joomla.org/>. Następnie skompresowany folder wypakowujemy i nazywamy jak nam wygodnie do katalogu serwera. Następnie w przeglądarkę wpisujemy adres: localhost/nazwa\_folderu. Powinna nam się wyświetlić strona startowa, gdzie wykonywana jest wstępna konfiguracja systemu. Wybieramy tam język, wpisujemy nazwę witryny, adres e-mail administratora, jego nazwę i hasło. Opcjonalnie możemy także opisać naszą witrynę. Po wypełnieniu wymaganych pól klikamy przycisk „Dalej”.

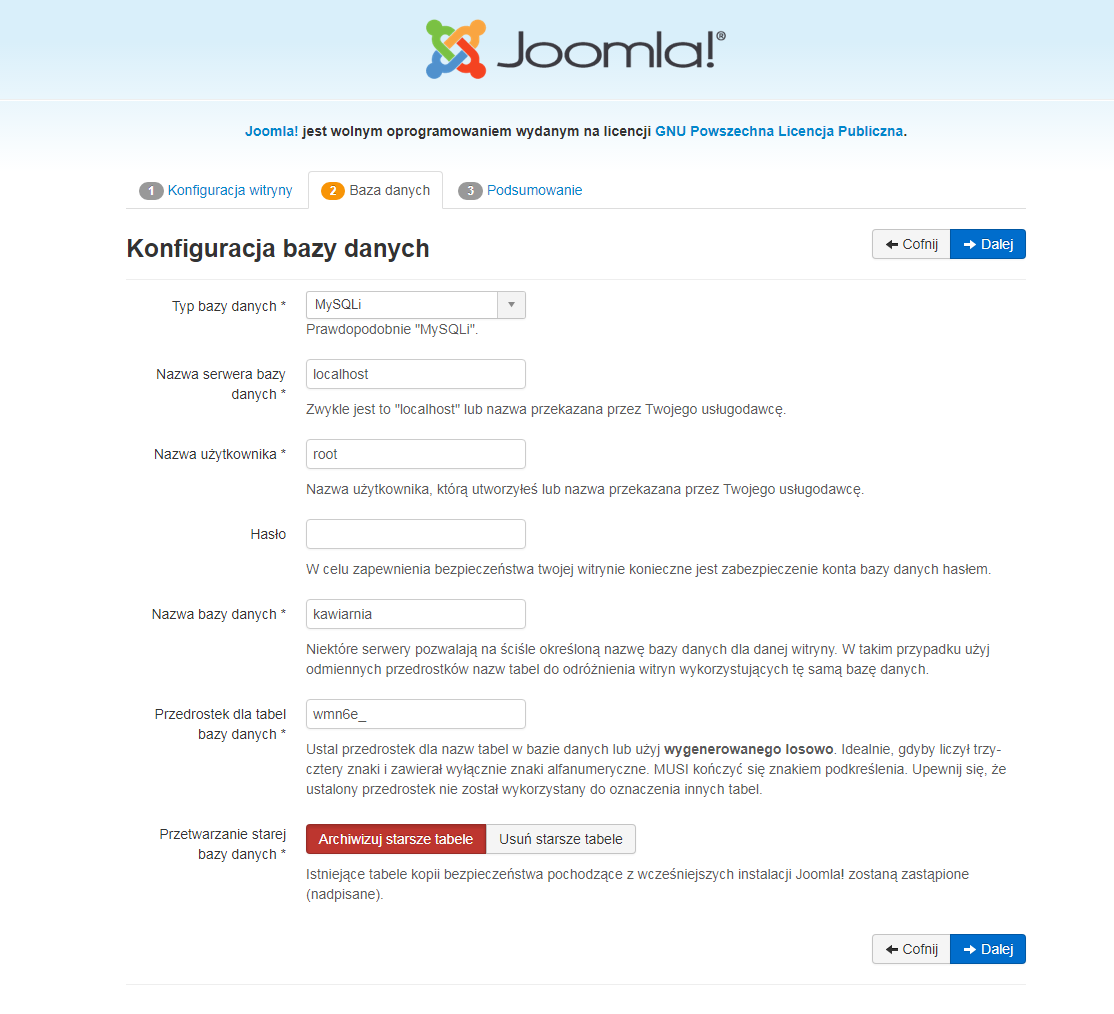
Rys. 11 Strona startowa instalacji systemu Joomla!



Źródło: Zrzut ekranu przeglądarki

Kolejnym krokiem jest konfiguracja bazy danych. Najlepiej jest stworzyć nową bazę danych dla naszego systemu co było już opisane wcześniej. Tutaj wybieramy typ bazy danych (najczęściej jest to MySQL), nazwę serwera bazy danych (najczęściej *localhost*), nazwa użytkownika, hasło (w przypadku gdy używamy WampServer’a lub XAMPP’a konto bazy danych nie jest zabezpieczone hasłem), nazwę bazy danych, z której chcemy korzystać oraz przedrostek dla tabel (w ten sposób łatwo możemy odróżnić tabele należące do innych aplikacji). Gdy wypełnimy wszystkie pola klikamy przycisk „Dalej” i przechodzimy do ostatniego etapu instalacji.

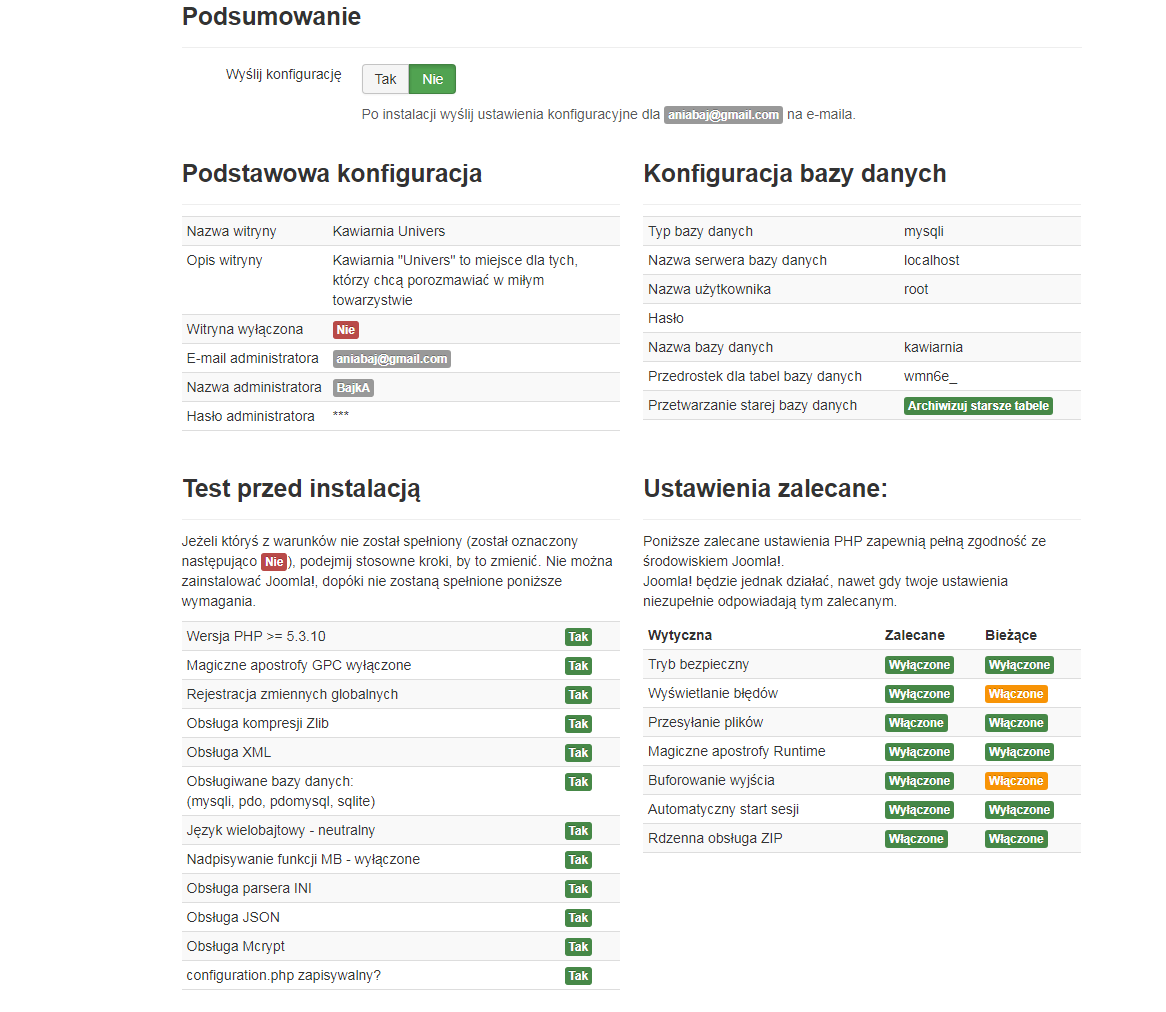
Rys. 12 Konfiguracja bazy danych



Źródło: Zrzut ekranu przeglądarki

Przygotowania do instalacji kończą się podsumowaniem konfiguracji podstawowej, bazy danych oraz testu serwera przed instalacją. Jeśli wszystkie opcje są oznaczone zieloną ikoną to możemy zakończyć przygotowania i zainstalować system Joomla! W tym celu klikamy przycisk „Instaluj”. Ostatnim etapem jest usunięcie katalogu *Installation*. Po tym możemy zalogować się do panelu administracyjnego i zacząć pracę nad naszą stroną internetową.

Rys. 13 Podsumowanie instalacji

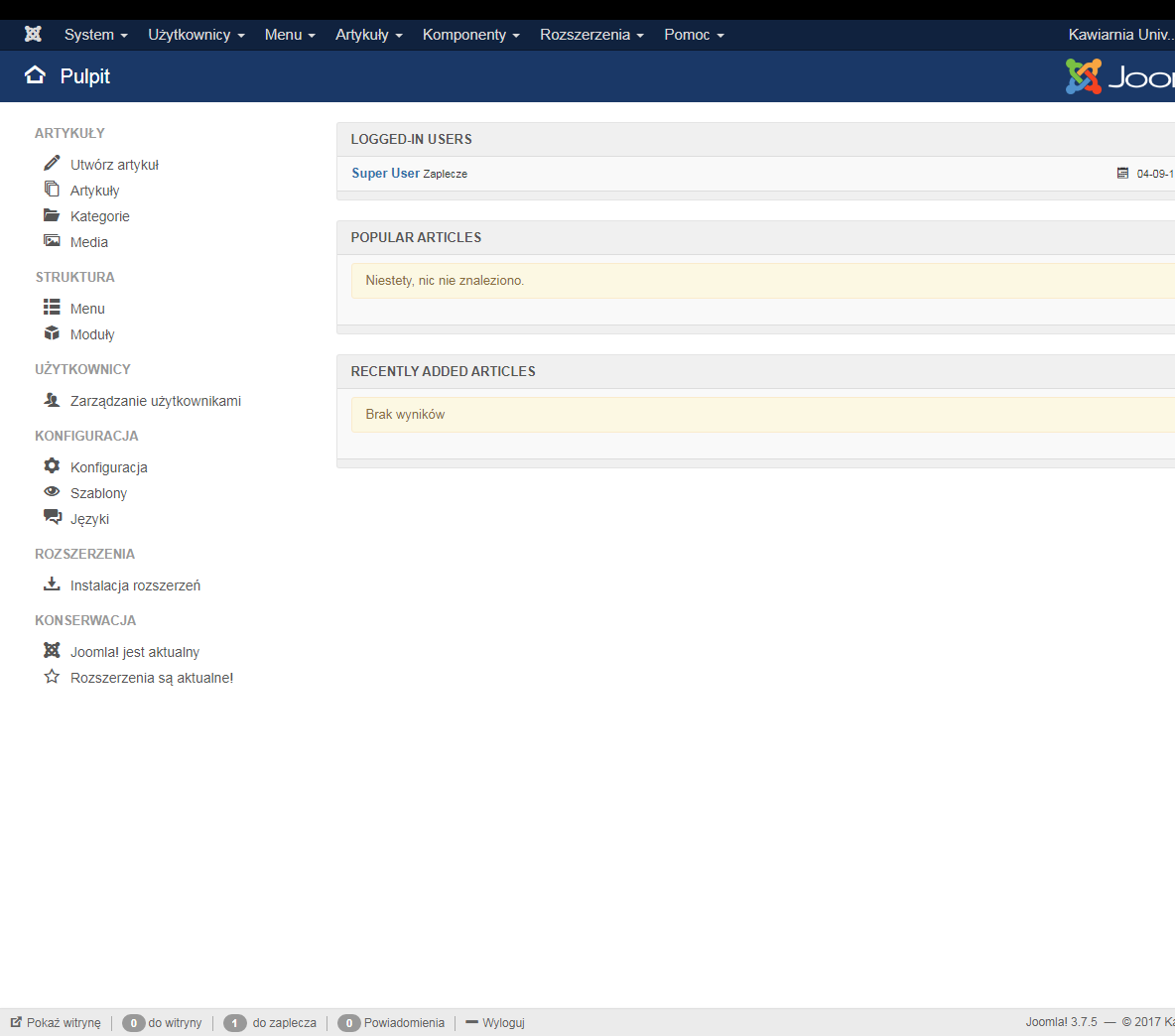


Źródło: Zrzut ekranu przeglądarki

### 2.4.2 Podstawowe funkcje Joomla!

Po zalogowaniu się do panelu administracyjnego pojawia nam się taki widok:

Rys. 14 Panel administracyjny Joomla!



Źródło: Zrzut ekranu przeglądarki

Całość nie wygląda zbyt imponująco, jednak z biegiem czasu ten stan rzeczy ulegnie zmianie.

W górnym *menu* widzimy opcje, które dotyczą konfiguracji naszej strony internetowej. Możemy dzięki nim zarządzać ustawieniami strony, poszczególnych modułów, dodawać i usuwać użytkowników czy korzystać z pomocy. Po lewej stronie mamy do dyspozycji skróty do najważniejszych funkcji. Są nimi: *Artykuły, Struktura, Użytkownicy, Konfiguracja, Rozszerzenia* oraz *Konserwacja*.

W zakładce *Artykuły* Możemy tworzyć i zarządzać już powstałymi treściami zamieszczonymi na naszej stronie. Możemy je przypisywać do uprzednio zdefiniowanej, odpowiedniej kategorii, ustawić kto ma dostęp do danego artykułu oraz dodać słowa kluczowe. Możemy także ustawić datę publikacji oraz wygaśnięcia artykułu. Wpisać autora tekstu oraz odniesienie zewnętrzne.

W sekcji *Struktura* możemy zarządzać nawigacją na stronie (czyli modyfikować *menu*) oraz dodawać i edytować już powstałe moduły strony.

Zakładka *Użytkownicy* pozwala na zarządzanie zarejestrowanymi użytkownikami. Możemy przypisać ich do danej grupy, sami taką grupę stworzyć, przypisać uprawnienia dostępu, dodawać notatki.

W *Konfiguracji* możemy zmienić ustawienia globalne serwisu i poszczególnych komponentów, szablon strony (domyślnie zainstalowane są dwa) oraz zmienić język w panelu administracyjnym i na stronie.

System Joomla! w swojej podstawowej konfiguracji oferuje swoim użytkownikom wiele opcji. Do tego możemy jeszcze instalować przeróżne rozszerzenia. Niektóre z nich są płatne a niektóre możemy zainstalować całkowicie za darmo.

Ostatnia pozycja w *menu* informuje nas o tym czy nasz system oraz dodatki są aktualne.

Wymienione wyżej funkcje są przedstawione tylko w niewielkim stopniu. Bardziej szczegółowy opis niektórych z nich znajdzie się w kolejnym rozdziale, gdzie ukazany zostanie proces stworzenia prostej strony internetowej dla użytkowników.

# 3. Tworzenie własnej strony internetowej

## 3.1 Wprowadzenie

Napisanie własnej strony internetowej nie jest bardzo szybkim procesem. Bez względu na to czy piszemy stronę od podstaw, czy użyjemy gotowego rozwiązania, trzeba na dopracowanie różnych szczegółów poświęcić trochę czasu. I nie powinniśmy się z tym śpieszyć, ponieważ cały poświęcony nasz czas zaowocuje wyglądem, który przyciąga uwagę oraz treścią, która sprawi, że użytkownik będzie chciał wracać po więcej. I przede wszystkim będziemy mieć satysfakcję z tego, że nasza strona „żyje” i przyciąga użytkowników a dzięki ewentualnym reklamom będziemy mogli także zarabiać.

W tym rozdziale opisany zostanie proces tworzenia strony internetowej. Zaczniemy od wstępnego projektu czyli określimy tematykę serwisu, jego nazwę i wygląd. Zastanowimy się czy napiszemy stronę internetową sami czy użyjemy w tym celu gotowego rozwiązania. Zastanowimy się także do jakiej grupy użytkowników chcemy dotrzeć.

Na koniec stworzymy przykładową stronę internetową przy użyciu oprogramowania Joomla!, gdzie zostaną opisane poszczególne etapy pracy oraz opcje jakie oferuje sam system.

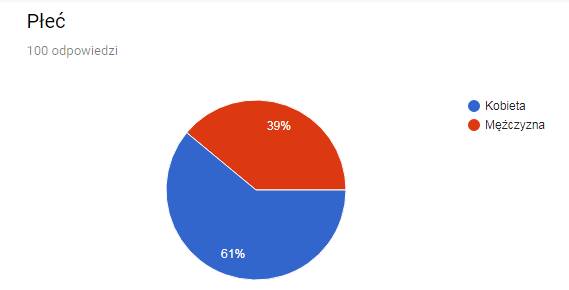
## 3.2 Projekt strony internetowej

Zanim zaczniemy pisać stronę internetową, na początek powinniśmy ją zaprojektować. Określić tematykę, grupę użytkowników, do których chcemy dotrzeć, wymyślić tytuł oraz wstępnie zaprojektować wygląd strony. Są to niezbędne kroki, których nie możemy ominąć. Nie tylko ze względu na czas jaki musimy poświęcić na stworzenie takiej strony ale często także ze względu na pieniądze jakie musimy włożyć w nasz projekt.

Zacznijmy od tematyki. W dzisiejszych czasach bardzo popularne stało się określenie *popkultura* czyli kultura popularna. Oczywiście istniała ona od dawna a samo pojęcie także nie jest nowe. Jest to typ kultury, który jest lubiany przez rzeszę ludzi i powszechnie dostępny. Dzięki Internetowi dostęp do zasobów popkulturowych jest dużo prosty a kontakty z innymi fanami nie stanowią problemu. Idąc tym tropem, strona internetowa o tej tematyce, może przyciągnąć uwagę różnych użytkowników.

Następnie należy sobie zadać pytanie czy chcemy skupić się na jednym temacie (jednej grupie internautów), czy pójdziemy krok dalej i stworzymy serwis, który przyciągnie więcej użytkowników. Doświadczenie pokazuje, że większość osób zainteresowanych jakimś wytworem kultury popularnej nie skupia się tylko na nim samym. Zazwyczaj interesuje się także innymi tworami. Na forach dyskusyjnych często poruszane są także inne wątki, które są zupełnie nie związane z tematyką danego forum. Obrazuje to poniższa ankieta utworzona w celu zbadania zainteresowania internautów wytworami kultury popularnej. W ankiecie wzięło udział sto osób. Większość z nich stanowiły kobiety (61%).

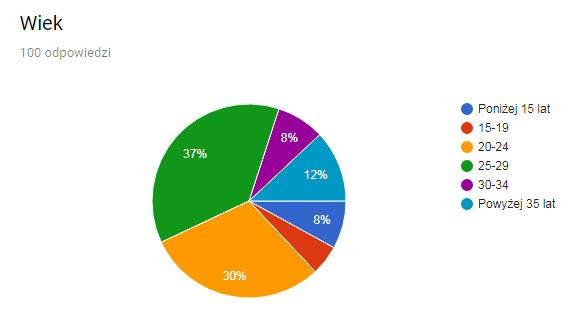
Wykres 3 Procentowy udział każdej z płci



Źródło: Narzędzia Google

Byli to ludzie w różnym wieku, jednak zdecydowana większość z nich najczęściej mieściła się w przedziałach: 20-24 oraz 25-29 (odpowiednio 30% i 37%).

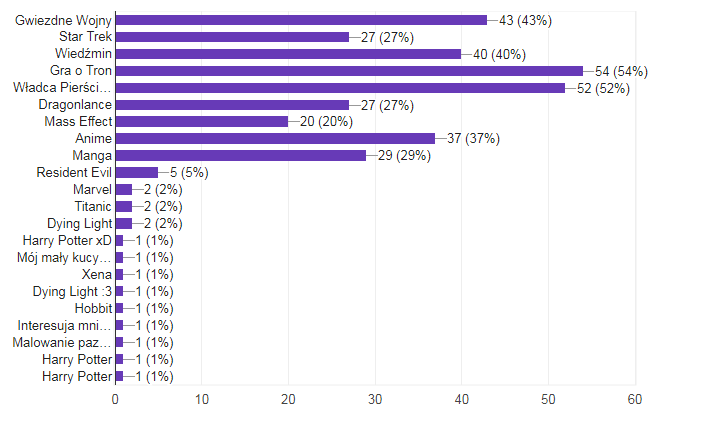
Wykres 4 Procentowy udział wiekowy



Źródło: Narzędzia Google

Jak widać naszą grupą odbiorców będą kobiety i mężczyźni w różnym wieku. Natomiast jeśli chodzi o preferencje tematyczne to odpowiedzi przedstawiały się w różny sposób. Jednak większość zaznaczyła więcej niż jedną opcję odpowiedzi. Może to oznaczać, że miejsce gdzie użytkownik może dyskutować na najróżniejsze tematy i przy okazji dowiadywać się o nowościach, jest pożądane przez internautę.

Wykres 5 Preferencje kulturowe internautów



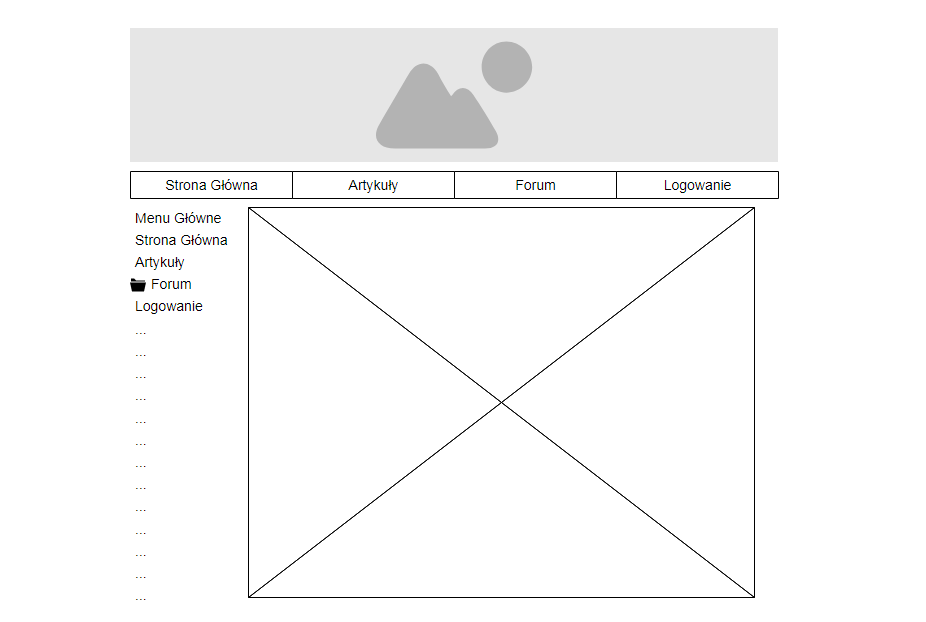
Źródło: Narzędzia Google

Kolejnym krokiem po określeniu tematyki i grupy odbiorców jest ustalenie jakie funkcje będą dostępne w naszym serwisie. Czy chcemy by było to forum, blog czy serwis z informacjami. Chcąc stworzyć stronę internetową, która przyciągnie użytkowników nie tylko samą treścią ale także funkcją społecznościową warto zastanowić się nad połączeniem serwisu informacyjnego z forum. Dzięki temu użytkownicy i goście będą mieli dostęp do informacji na dany temat oraz będą mogli dyskontować na różne tematy na forum. Można także w przyszłości zastanowić się nad czatem dla zarejestrowanych użytkowników.

Teraz powinniśmy pomyśleć nad tytułem naszej strony internetowej. Powinna ona nawiązywać do tematyki serwisu i być na tyle oryginalna by przykuć uwagę. Skoro chcemy stworzyć między innymi forum dyskusyjne to można zastanowić się co nam się z tym kojarzy. Rozmowy na różne tematy mogą przypominać spotkania np. w kawiarni. Natomiast tematyka ma się obracać wokół wielu wytworów kultury popularnej. Krążąc między tymi różnymi światami możemy sobie to skojarzyć ze wszechświatem. Jednak „Kawiarnia Wszechświat” nie brzmi wystarczająco dobrze w moim własnym odczuciu. W języku angielskim słowo *wszechświat* brzmi *universe*. Dodajmy do tego coś od siebie jeszcze by nie być dosłownym. Słowo *universe* wymawia się: /ˈjuːnɪˌvɜːs/ czyli na końcu nie wymawia się litery *e*. Spolszczając, więc nieco to słowo wyjdzie nam taki tytuł: „Kawiarnia Univers”. Oczywiście prace nad wymyśleniem własnej nazwy mogą być dużo trudniejsze w zależności co chcemy osiągnąć. Jednak jeśli mamy do czynienia z młodymi odbiorcami to „Kawiarnia Univers” jako tytuł powinna przyciągnąć uwagę.

Ostatnim krokiem przed rozpoczęciem prac nad stroną jest zaprojektowanie jej wyglądu. Do tego powinniśmy użyć odpowiednich narzędzi do projektowania lub, jeśli wolimy, użyć kartki papieru i ołówka by naszkicować wygląda naszej strony. Przykład projektu strony znajduje się poniżej.

Rys. 15 Projekt układu elementów strony



Źródło: Opracowanie własne

Projekt ten pokazuje układ poszczególnych elementów strony takich jak:

* Logo
* Menu poziome i pionowe
* Miejsce na kontent strony

i jak można zauważyć jest on dość prosty, dzięki czemu nawet poczatkujący twórca stron internetowych nie powinien mieć problemów z jego zrealizowaniem.

Kiedy już wiemy jak będzie mniej więcej wyglądać nasza strona internetowa, to możemy przejść do wyboru między użyciem gotowej technologii a pisaniem strony od podstaw. Na początek trzeba się zastanowić jakimi umiejętnościami dysponujemy oraz budżetem. Jeśli mamy duże doświadczenie w tworzeniu stron internetowych i kolejna nie prawi nam problemu to bez przeszkód możemy ją napisać od podstaw. Możemy także komuś to zlecić ale to z kolei wiąże się z wydatkami jakie musimy ponieść by móc opłacić zleceniobiorcę. Gdy nie posiadamy dużego doświadczenia oraz mamy bardzo ograniczony budżet, to najlepszym rozwiązaniem będzie skorzystanie z gotowego rozwiązania typu *open source*. Zaletami jak i wadami takiego wyjścia są:

Tabela 9 Wady i zalety stosowania gotowych rozwiązań

|  |  |
| --- | --- |
| Zalety | Wady |
| Niskie koszty (praktycznie zerowe) | Problemy z dostosowaniem do swoich preferencji |
| Krótki czas wdrożenia | Trudność w wyróżnieniu się |
| Prostota w obsłudze |  |
| Prosta edycja poszczególnych modułów |  |
| Wsparcie techniczne |  |
| Prosta aktualizacja |  |

Źródło: Opracowanie własne

Jak można zauważyć użycie gotowego rozwiązania ma więcej zalet niż wad. Jeśli jednak najbardziej zależy nam na wyróżnieniu się oraz możliwości dostosowania strony do swoich preferencji, to najlepszym rozwiązaniem będzie napisanie swojej strony od podstaw.

Gdy już zdecydujemy się na któreś wyjście, czy to będzie pisanie od podstaw, czy użycie któregoś z gotowych rozwiązań, to możemy rozpocząć pracę nad stworzeniem strony internetowej.

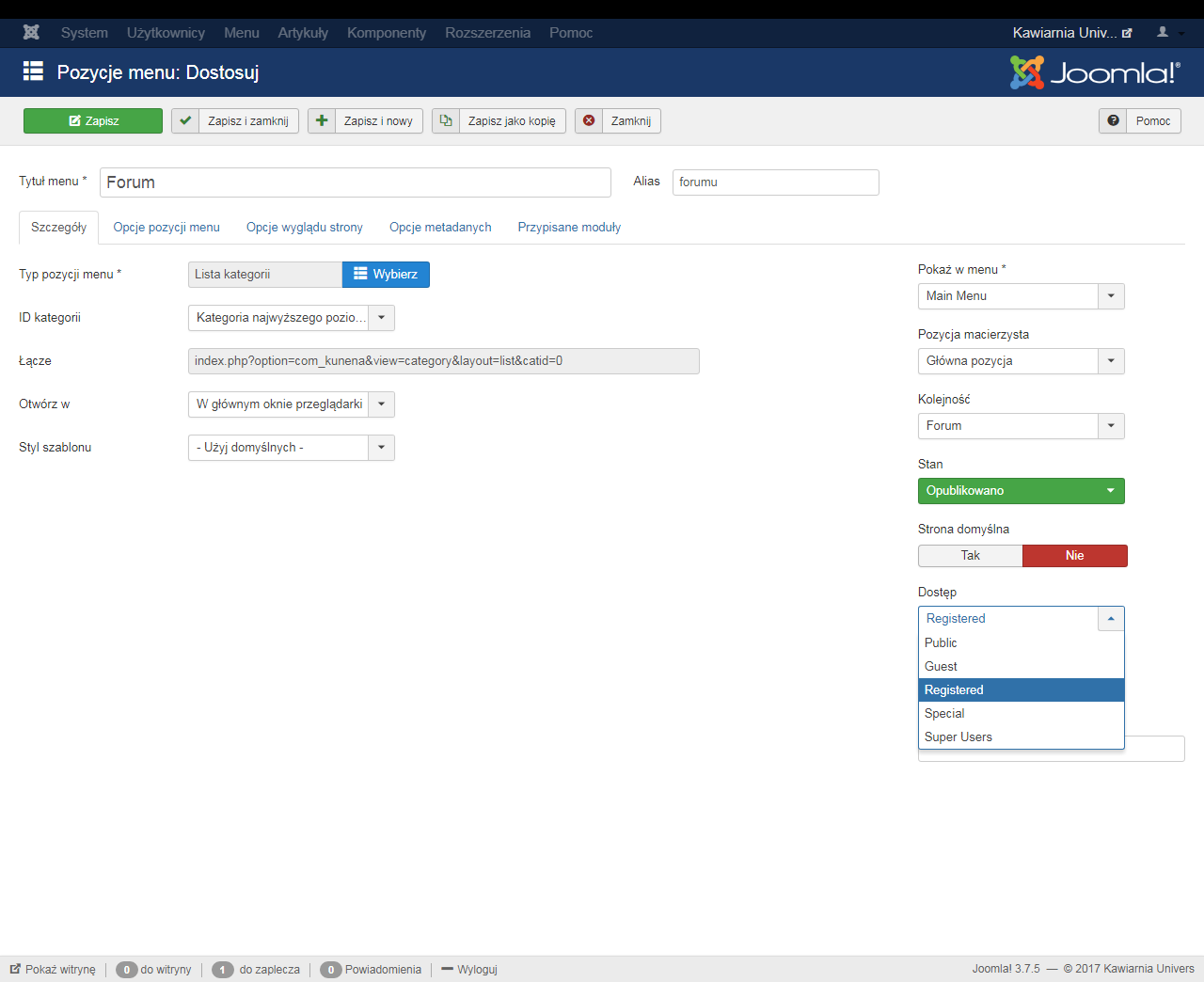
## 3.3 Tworzenie strony internetowej przy zastosowaniu systemu Joomla!

W poprzednim rozdziale zainstalowaliśmy oprogramowanie Joomla! na naszym serwerze. Od teraz możemy zacząć konfigurować naszą stronę.

Na samym początku zacznijmy od instalacji języka polskiego dla systemu Joomla!, czego możemy dokonać w bardzo prosty sposób. Po pierwsze wchodzimy w zakładkę *Install extensions-> Install Languages*. Następnie szukamy języka polskiego, klikamy instaluj i prawie gotowe. Na koniec musimy zmienić język z angielskiego na polski. W tym celu klikamy zakładkę *Languages* i zmieniamy domyślny język poprzez kliknięcie ikony gwiazdki. Teraz z łatwością możemy zacząć konfigurację naszej strony.

Na początek możemy zacząć od edycji menu głównego. W tym celu klikamy zakładkę *Menu*, następnie *Main Menu* i w tym miejscu możemy dodawać i usuwać podstrony naszego serwisu oraz wybrać, która ma być stroną główną. W opcjach każdej pozycji w *menu* możemy ustawić jaka grupa osób będzie miała dostęp do danej podstrony. Przykładowo chcemy by do podstrony *Forum* miały dostęp tylko osoby zarejestrowane. W tym celu w nowo utworzonej pozycji, w zakładce *Szczegóły* zmieniamy dostęp z *Public* na *Registered.*

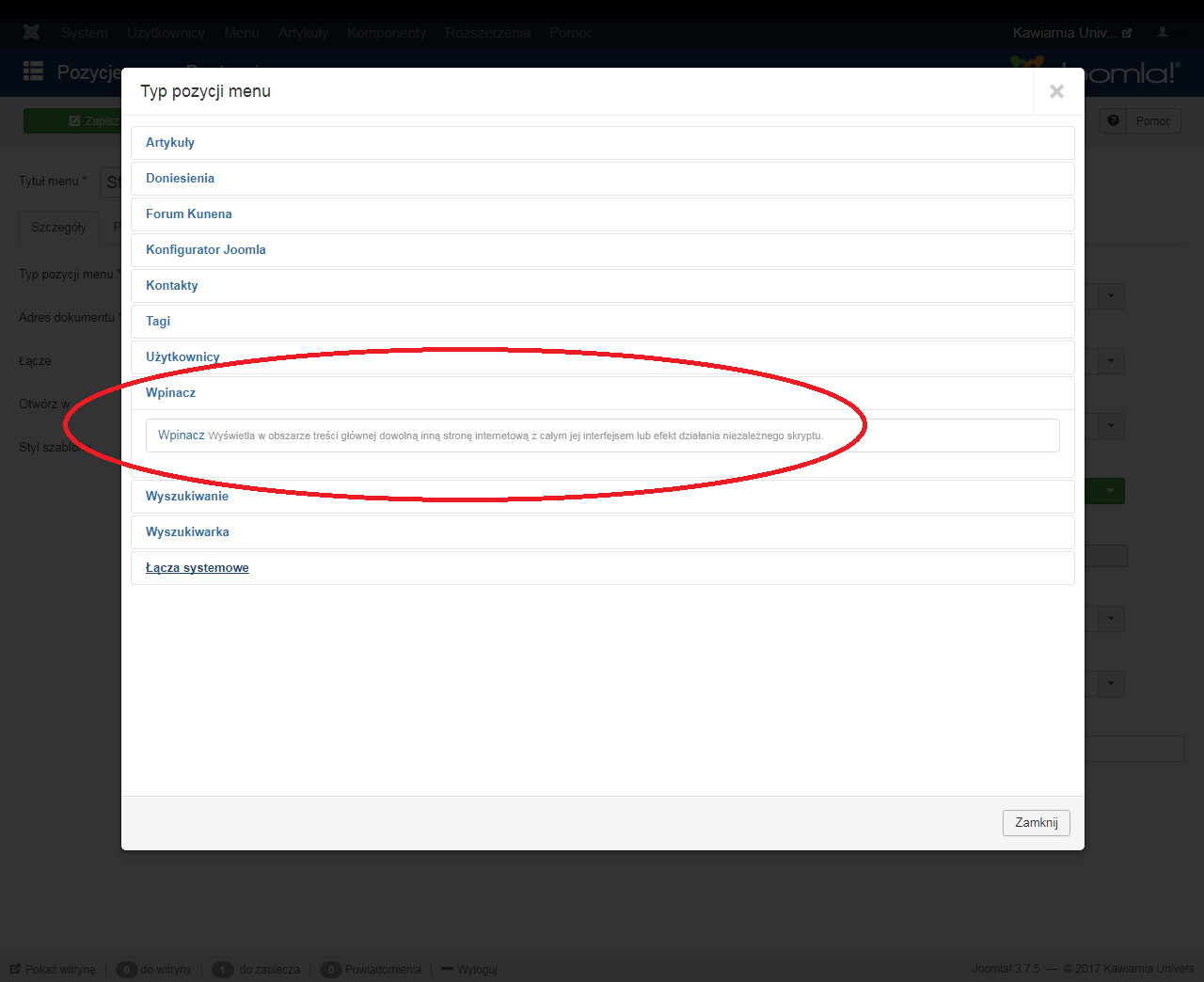
Rys. 16 Zmiana opcji dostępu do podstrony



Źródło: Zrzut ekranu przeglądarki

Dzięki temu podstrona ta będzie widoczna tylko i wyłącznie po zalogowaniu się do serwisu. W zakładce tej możemy także ustalić co zostanie wyświetlone na danej podstronie. Jeśli chcemy by na stronie głównej wyświetlał się ekran powitalny, informacje o stronie lub jakikolwiek inny wybrany przez nas kontent wtedy wystarczy wybrać odpowiedni *Typ pozycji menu.* W tym przypadku jest *Wpinacz.*

Rys. 17 Typy pozycji menu



Źródło: Zrzut ekranu przeglądarki

Następnie wpisujemy adres dokumentu, który chcemy wyświetlić. Przykładowo jest to plik .html, który został specjalnie stworzony by na naszej stronie głównej wyświetlało się odpowiednie powitanie. Poniżej prezentuję cały kod utworzonej w ten sposób prostej strony.

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<style>

body{

background: #d8d8d8;

background: -webkit-linear-gradient(-45deg, #d8d8d8, #cccccc, #d8d8d8);

background: -o-linear-gradient(-45deg, #d8d8d8, #cccccc, #d8d8d8);

background: -moz-linear-gradient(-45deg, #d8d8d8, #cccccc, #d8d8d8);

background: linear-gradient(-45deg, #d8d8d8, #cccccc, #d8d8d8);

}

p{color: #005e8d;}

img{

max-height: auto;

max-width: 690px;

}

</style>

</head>

<body>

<h2>Witaj przybyszu!</h2>

<p>Oto Kawiarnia <i>Univers</i> miejsce, gdzie każdy znajdzie coś dla siebie.<br>

Rozgość się i usiądź wygodnie przy którymś ze stoików. <br></p>

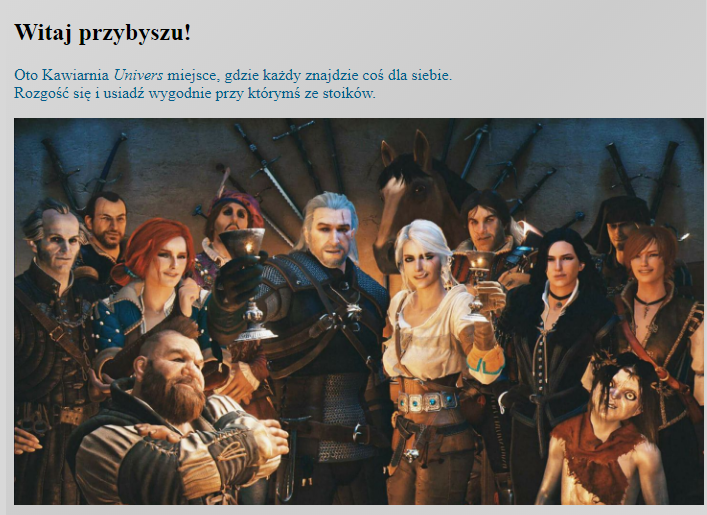
<img src="/joomla/grafiki/witaj.jpg"/>

</body>

</html>

Strona wyświetla prosty tekst powitania i jeden obrazek. Zastosowany kod CSS służy do zdefiniowania wyglądu tła strony, koloru tekstu oraz wielkości obrazka. Co w efekcie wygląda w następujący sposób:

Rys. 18 Strona główna

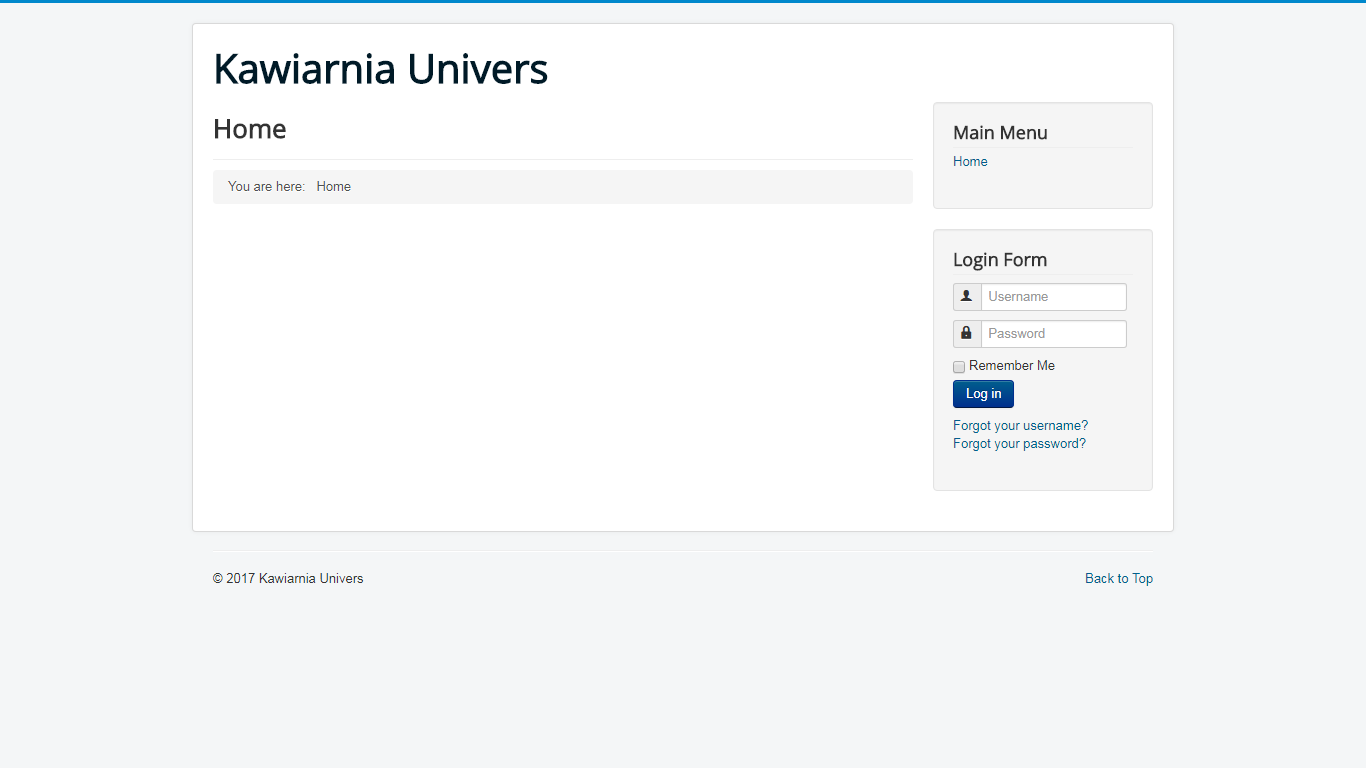


Źródło: Zrzut ekranu przeglądarki

Pozycja *Forum* potrzebuje zainstalowanego rozszerzenia *Kunena Forum*, które w prosty sposób pomoże nam w zarządzaniu nim. Najprościej jest pobrać pakiet *Kunena* na dysk, przejść do zakładki *Instalacja rozszerzeń->Instaluj z pakietu*, przeciągamy skompresowany katalog w odpowiednie miejsce i instalacja sama się rozpoczyna. Po kilku chwilach otrzymujemy komunikat, że pakiet został zainstalowany. Teraz możemy podpiąć nasze forum do strony tworząc nową pozycję w *menu* i wybierając odpowiedni *Typ pozycji menu.*

Samą konfiguracją naszego forum zajmiemy się nieco później. W tej chwili naszym kolejnym krokiem będzie dostosowanie wyglądu naszej strony do tego co wcześniej zaprojektowaliśmy. Ustawienia domyślne pustej strony w systemie Joomla! sprawiają, że strona wygląda tak:

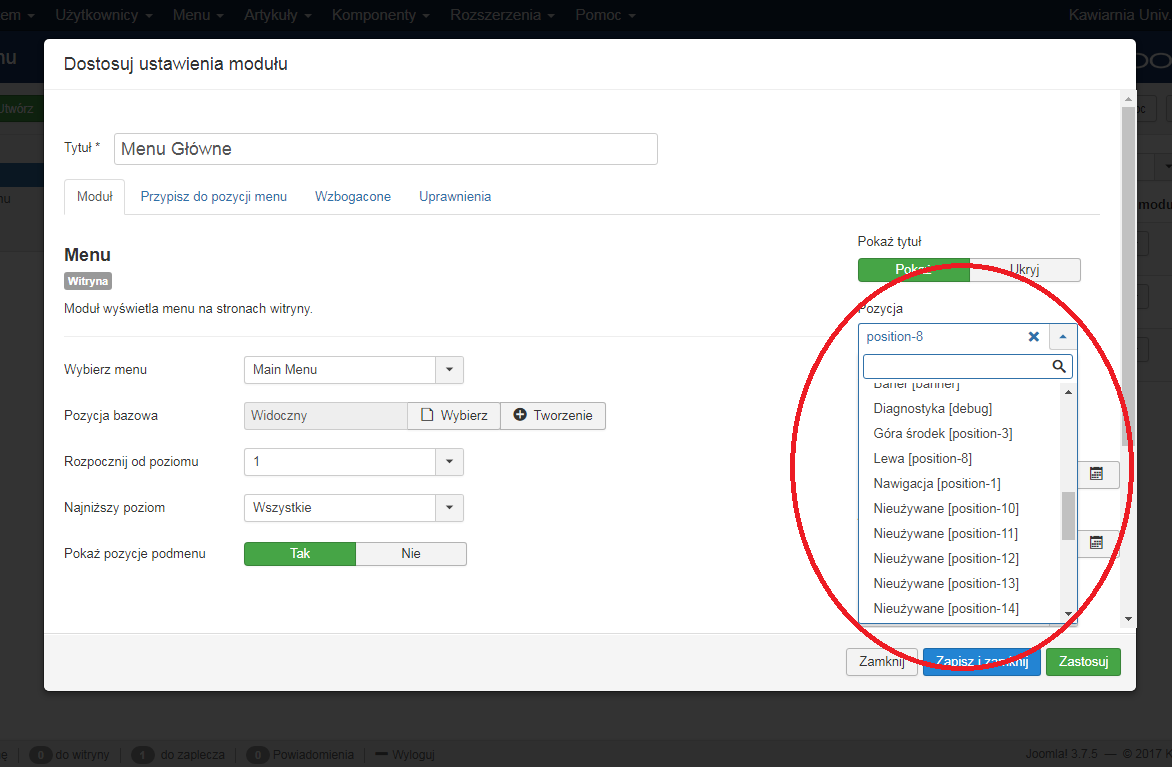
Rys. 19 Domyślny wygląd strony głównej



Źródło: Zrzut ekranu przeglądarki

W ustawieniach *Menu*  możemy zmieniać położenie niektórych elementów na stronie oraz wyświetla, bądź ukrywać je.

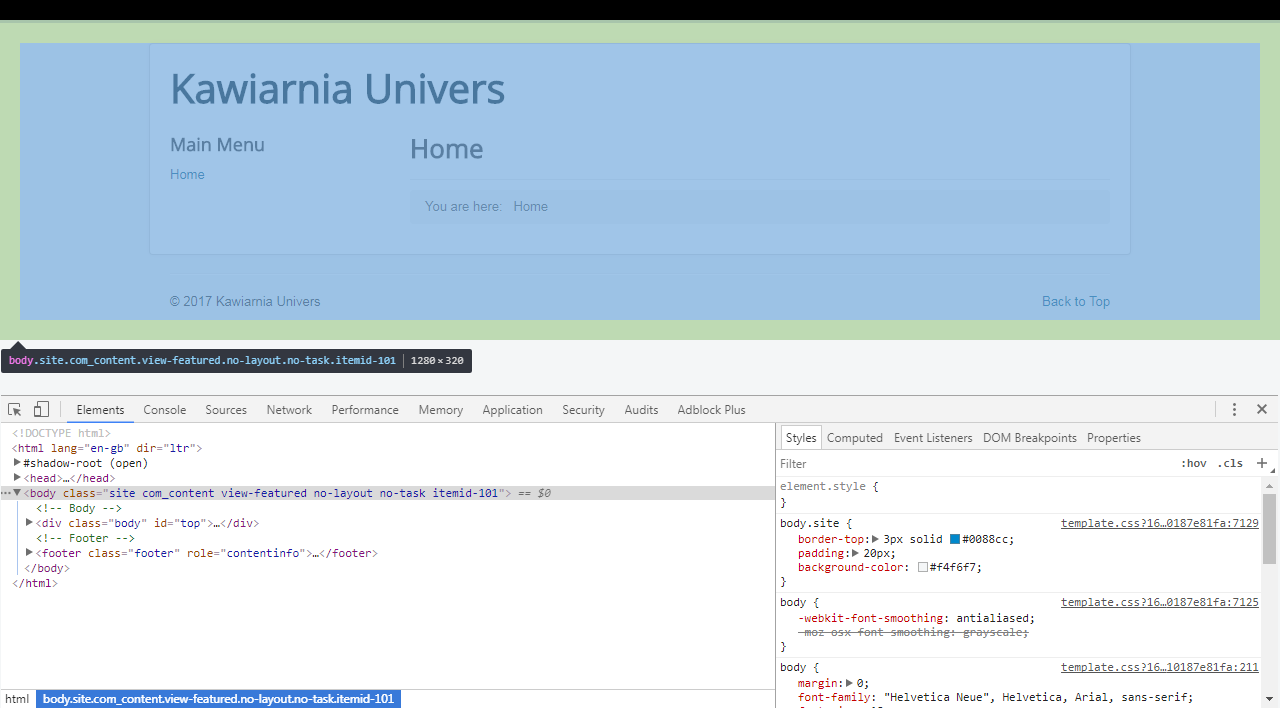
Rys. 20 Ustawienie pozycji menu



Źródło: Zrzut ekranu przeglądarki

„Przerzucamy” więc nasze *menu* na lewą stronę. Kolejnym krokiem będzie zmiana kolorystyki oraz loga strony. Chcemy by po kliknięciu na nie możliwy był powrót na stronę główną. W tej chwili tą funkcję pełni napis „Kawiarnia Univers”. Te zmiany wymagają już od nas trochę większej wiedzy na temat pisania stron internetowych, ponieważ musimy ich dokonać w kodzie szablonu. By odnaleźć interesujące nas klasy oraz linie kodu najprościej jest skorzystać z narzędzi, które udostępnione są w przeglądarkach Chrome i Mozilla Firefox. *Narzędzie dla programistów* można otworzyć za pomocą skrótu klawiszowego F12. Następnie klikamy prawym klawiszem myszy na interesującym nas elemencie i wybieramy opcję *Zbadaj*. W ten sposób z łatwością odnajdujemy, który element kodu należy zmienić. W przypadku zmiany koloru tła będzie to plik *template.css* a logo musimy podmienić w pliku *index.php*, który znajduje się w katalogu odpowiedniego szablonu.

Rys. 21 Narzędzie dla programistów



Źródło: Zrzut ekranu przeglądarki

W przypadku tła chcemy zmienić je na odpowiedni obrazek, który będzie nawiązywał do „klimatu” strony. W tym celu otwieramy odpowiedni plik, szukamy odpowiednej reguły ( w tym przypadku body.site) i zamieniamy deklarację:

background-color: #kolor;

na

background-image: url("adres\_obrazka");

Następnie zmieniamy kolory tła elementu, którego atrybut *class* przyjmuje wartość *container* i w pliku *template.css* zmieniamy deklarację reguły .body .container:

background-color: #fff;

na

background-color: #d8d8d8;

Jak widać zmiany w wyglądnie nie są skomplikowane i wystarczą do tego jedynie podstawy znajomości CSS’a oraz HTML’a.

Kolejnym krokiem jest zmiana domyślnego loga na obrazek. Nasze własne logo możemy stworzyć przy pomocy darmowego narzędzia, które znajduje się pod adresem: <https://www.onlinelogomaker.com/>. Dzięki niemu możemy szybko stworzyć własny obrazek. W celu podmiany musimy otworzyć plik *inex.php*, który znajduje się w katalogu używanego szablonu. Szukamy zmiennej, która przechowuje informacje o tym jak ma wyglądać logo i zmieniamy jej wartość:

$logo = '<span class="site-title" title="' . $sitename . '">' . $sitename . '</span>';

na

$logo = '<img src="adres\_obrazka"/>';

Po wprowadzonych zmianach nasza strona powinna wyglądać miej więcej w ten sposób:

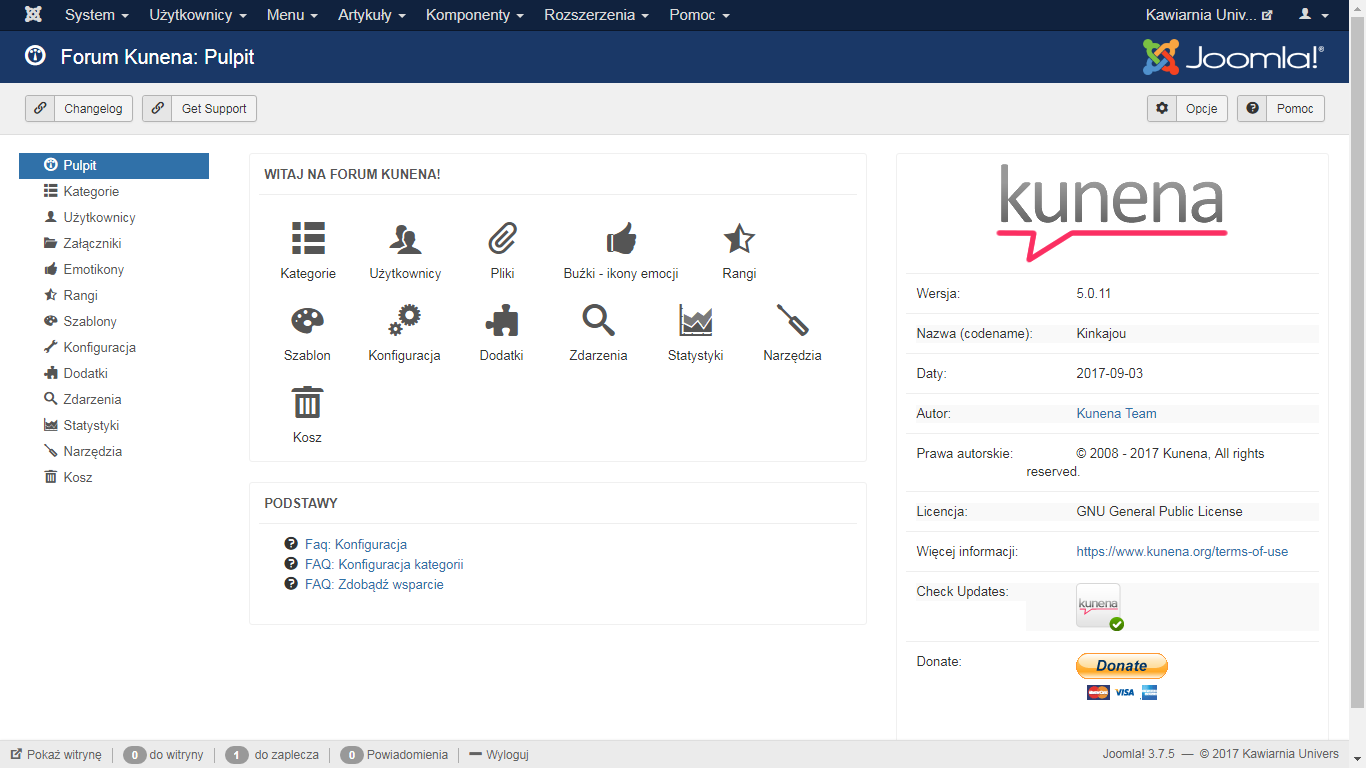
Rys. 22 Strona główna po edycji



Źródło: Zrzut ekranu przeglądarki

Gdy już nasza strona wygląda tak jak chcemy, to kolejną rzeczą na jakiej powinniśmy się skupić to konfiguracja forum. Wchodzimy w zakładkę *Kunena Forum*. Widzimy różne opcje dotyczące konfiguracji naszego forum.

Rys. 23 Opcje *Kunene Forum*



Źródło: Zrzut ekranu przeglądarki

W zakładce *Kategorie* możemy szybko i łatwo dodawać i usuwać nowe kategorie do naszego forum. Regulować dostęp do danej kategorii oraz wyznaczyć moderatorów. Kolejna, *Użytkownicy*, pozwala na przeglądanie użytkowników forum, gdzie można nadać im uprawnienia moderatora lub zablokować go. *Pliki* pozwalają na przeglądanie plików, które użytkownicy mogą załączyć do swojego postu a w zakładce *Buźki- ikony emocji* możemy przeglądać listę emotikon, których można używać. Opcje forum pozwalają nam również zmieniać rangi forumowiczów, szablony, włączać i wyłączać dodatki, przeglądać statystyki oraz konfigurować ustawienia forum.

Dodawanie nowych kategorii jest bardzo proste. Wystarczy wejść o odpowiednią zakładkę i dodać nową kategorię. Na początku wybieramy czy ma być to kategoria najwyższego poziomu, czyli zbiór wszystkich forów, czy może jednak kategoria podrzędna, która pozwala na utworzenie forum i dodawanie tematów i postów. Na początek warto najpierw stworzyć kilka kategorii najwyższego poziomu by potem spokojnie móc do nich dodawać kolejne fora.

Obsługa *Kuneny* sama w sobie jest bardzo prosta i intuicyjna. Jest to bardzo dobre rozwiązanie, które współgra z oprogramowaniem Joomla! i nie wymaga ogromnej wiedzy. W przyszłości, gdy forum zyska na popularności i bez problemu, jednym kliknięciem, możemy ustanowić zaufanych użytkowników moderatorami by pomagali w utrzymaniu porządku.

Jednym z ostatnich kroków jakie będą nam potrzebne w konfiguracji strony to dodawanie artykułów. Warto jednak najpierw stworzyć odpowiednie kategorie tematyczne. Ułatwi to nawigację oraz utrzyma porządek na stronie. Będzie też dużo łatwiej zarządzać napisanymi artykułami.

Tworzenie kategorii jak i podkategorii artykułów jest bardzo proste. Wystarczy wpisać nazwę, ustawić opcje dostępu i ewentualnie krótko opisać. Tak samo proste jest utworzenie nowego artykułu. Klikamy na opcję *Utwórz artykuł* i możemy zaczynać jego edycję.

Na początku warto od razu wpisać tytuł oraz określić kategorię do jakiej się odnosi. W tym miejscu także możemy utworzyć nową kategorię jeśli odpowiednia jeszcze nie istnieje. Określamy dostęp do artykułu, wpisujemy tagi (czyli słowa kluczowe) i możemy zacząć pisać. Pisać możemy sami ale warto (jeśli sami nie posiadamy takich umiejętności) zatrudnić kogoś, kto wyróżnia się łatwością pisania czyli ma tzw, „lekkie pióro”. Nic nie przyciąga użytkowników i nie zachęca do czytania tak jak dobrze napisany tekst. Toporny i niedający się do czytania tekst na pewno zrazi potencjalnego użytkownika i sprawi, że nie będzie miał ochoty pozostać na stronie. Następnie możemy dodać obrazek, który pojawi się obok tekstu w wprowadzeniu oraz inną do pełnego tekstu. W kolejnym kroku możemy zmienić opcje artykułu. Głównie są to opcję wyświetlania różnych elementów takich jak: tytuł artykułu, nazwa kategorii, nazwa autora, ilość odsłon, ocena i inne. W opcjach publikacji możemy wybrać datę publikacji czyli przygotować artykuł na jakiś temat i opublikować go np. dwa dni później oraz datę, kiedy to artykuł zostanie wycofany z publikacji. W tym miejscu możemy także określić, kto jest autorem artykułu oraz dodać odnośnik zewnętrzny (jeśli treść nie należy do nas).

Jednym z ostatnich kroków jest opublikowanie strony w Internecie. Większość usług hostingowych oferuje instalację różnego oprogramowania w tym systemu Joomla! dzięki czemu nie będziemy mieć problemów z przeniesieniem naszej strony z serwera lokalnego na hosting. Najlepiej, po instalacji systemu na serwerze hostingu, podmienić wszystkie pliki (w tym celu używamy np. programu *FileZlla*) oprócz *configuration.php*, gdzie należy jedynie dokonać zmiany w linijce:

public $dbprefix = 'naszprefix\_;

zmieniamy prefix tabel na ten, którego używaliśmy podczas edycji strony na serwerze lokalnym. Następnie eksportujemy bazę danych z serwera i importujemy na hosting. Po takiej podmianie zapewne będziemy musieli dokonać jeszcze kilku kosmetycznych zmian w niektórych ustawieniach i nasza strona powinna być już gotowa do pokazaniu jej światu.

Gdy nasza strona jest już dostępna w Internecie należy ciągle dbać o to by się rozwijała oraz słuchać rad użytkowników. Dzięki temu strona będzie „żyła” a my będziemy mogli być dumni ze swojej pracy.

# Zakończenie

W powyższej pracy zawarłam istotne informacje jakie powinien posiadać początkujący twórca stron internetowych. Omówiłam między innymi dlaczego tworzymy różne serwisy internetowe, badania dotyczące wzrostu ilości stron WWW dostępnych w Internecie na przestrzeni lat oraz metody liczenia użytkowników odwiedzających naszą stronę internetową. Opisałam także jak ważna jest strategia user experience oraz web usability. Następnie opisałam najpopularniejsze języki służące do tworzenia stron internetowych i systemy zarządzania relacyjnymi bazami danych wraz z opisem ich instalacji na komputerze oraz instalację oprogramowania Joomla! wraz z omówieniem jego podstawowych funkcji. Całość była wprowadzeniem do stworzenia własnej strony internetowej w oparciu o system Joomla! z wykorzystaniem wiedzy na temat HTML’a, CSS’a i PHP.

Wnioskiem z powyższej pracy jest fakt, iż stworzenie własnej strony internetowej nie musi się wiązać z posiadaniem dużej wiedzy na ten temat oraz z dużymi kosztami. Jednak trzeba poświęcić na to trochę czasu by móc uporać się z różnymi, w większości przypadków, błahymi problemami. Ze wszystkimi jednak można sobie poradzić w prosty sposób próbując samemu je rozwiązać lub znaleźć pomoc w Internecie.

# Bibliografia

J. Chudley, J. Allen „Projektowanie witryn internetowych User eXperience”, Helion, Gliwice, 2013

J. Duckett „HTML i CSS. Zaprojektuj i zbuduj witrynę WWW. Podręcznik Front End Developera”, Helion, Gliwice, 2014

J. Levy „Strategia UX: jak tworzyć innowacyjne produkty cyfrowe, które spotkają się z uznaniem na rynku”, Helion, Gliwice, 2017

J.N. Robbins „Projektowanie stron internetowych. Przewodnik dla początkujących webmasterów po HTML5, CSS3 i grafice.”, Helion, Gliwice, 2008

L. Ullman „PHP6 i MySQL5. Dynamiczne strony WWW. Szybki start.”, Helion, Gliwice, 2009

S. Krug „Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych”, Wydanie II, Helion, Gliwice, 2010

# Netografia

<http://appletoncreative.com/2015/09/10/users-dont-read-they-scan-making-your-website-understandable-and-scannable>

<http://marketingnaluzie.pl>

<http://semahead.pl/blog/unikalny-uzytkownik-vs-realny-uzytkownik-html.html>

<http://www.comarch.pl/erp/nowoczesne-zarzadzanie/numery-archiwalne/user-experience-projektowanie-pozytywnego-doswiadczenia>

<http://www.kurshtml.edu.pl/html/wstep,html5.html>

<https://news.netcraft.com>

<https://www.gemius.pl>

<https://www.google.pl/intl/pl/analytics/features/index.html>

[www.poetalwiedzy.onet.pl](file:///C:\Users\Ania\Desktop\www.poetalwiedzy.onet.pl)

[www.sjp.pwn.pl](file:///C:\Users\Ania\Desktop\www.sjp.pwn.pl)

# Spis tabel

[Tabela 1 Nowe znaczniki HTML5 i ich opisy 25](#_Toc492984097)

[Tabela 2 Inne wybrane znaczniki HTML i ich opisy 25](#_Toc492984098)

[Tabela 3 Selektory elementów CSS 28](#_Toc492984099)

[Tabela 4 Selektory atrybutów CSS 29](#_Toc492984100)

[Tabela 5 Właściwości CSS 30](#_Toc492984101)

[Tabela 6 Wybrane obiekty JavaScript i ich niektóre metody i własności 33](#_Toc492984102)

[Tabela 7 Wybrane zdarzenia JavaScript 35](#_Toc492984103)

[Tabela 8 Znaki specjalne PHP 37](#_Toc492984104)

[Tabela 9 Wady i zalety stosowania gotowych rozwiązań 54](#_Toc492984105)

# Spis wykresów

[Wykres 1. Liczba stron internetowych na przestrzeni lat 8](#_Toc492984322)

[Wykres 2 Profil użytkowników Internetu wg płci i wieku w 2015 roku 15](#_Toc492984323)

[Wykres 3 Procentowy udział każdej z płci 51](#_Toc492984324)

[Wykres 4 Procentowy udział wiekowy 51](#_Toc492984325)

[Wykres 5 Preferencje kulturowe internautów 52](#_Toc492984326)

# Spis rysunków

[Rys. 1 Realni użytkownicy a unikali 12](#_Toc492984327)

[Rys. 2 Ranking TOP20 domen, luty 2017 14](#_Toc492984328)

[Rys. 3 Trzy składowe UX 18](#_Toc492984329)

[Rys. 4 Związek pomiędzy user *experience* i *web usability* 19](#_Toc492984330)

[Rys. 5 Składowe *web usability* 20](#_Toc492984331)

[Rys. 6 Mapa ciepła przeglądania strony 21](#_Toc492984332)

[Rys. 7 Działanie stron, które wykorzystują PHP i bazę danych SQL 36](#_Toc492984333)

[Rys. 8 Instalatory darmowej wersji MySQL 41](#_Toc492984334)

[Rys. 9 Główny ekran MySQL Workbench 42](#_Toc492984335)

[Rys. 10 Tworzenie nowej bazy danych 42](#_Toc492984336)

[Rys. 11 Strona startowa instalacji systemu Joomla! 45](#_Toc492984337)

[Rys. 12 Konfiguracja bazy danych 46](#_Toc492984338)

[Rys. 13 Podsumowanie instalacji 47](#_Toc492984339)

[Rys. 14 Panel administracyjny Joomla! 48](#_Toc492984340)

[Rys. 15 Projekt układu elementów strony 53](#_Toc492984341)

[Rys. 16 Zmiana opcji dostępu do podstrony 55](#_Toc492984342)

[Rys. 17 Typy pozycji menu 56](#_Toc492984343)

[Rys. 18 Strona główna 58](#_Toc492984344)

[Rys. 19 Domyślny wygląd strony głównej 59](#_Toc492984345)

[Rys. 20 Ustawienie pozycji menu 59](#_Toc492984346)

[Rys. 21 Narzędzie dla programistów 60](#_Toc492984347)

[Rys. 22 Strona główna po edycji 62](#_Toc492984348)

[Rys. 23 Opcje *Kunene Forum* 62](#_Toc492984349)

1. J.N. Robbins „Projektowanie stron internetowych. Przewodnik dla początkujących webmasterów po HTML5, CSS3 i grafice.”, Helion, Gliwice, 2008, str. 36 [↑](#footnote-ref-1)
2. „Internet Research, Anti-Phishing and PCI Security Services | Netcraft”, https://news.netcraft.com, tłum. własne [↑](#footnote-ref-2)
3. „Słownik języka polskiego PWN”, [www.sjp.pwn.pl](http://www.sjp.pwn.pl) [↑](#footnote-ref-3)
4. „Portal Wiedzy Onet”, [www.poetalwiedzy.onet.pl](http://www.poetalwiedzy.onet.pl) [↑](#footnote-ref-4)
5. „Marketing na luzie”, <http://marketingnaluzie.pl> [↑](#footnote-ref-5)
6. Semahead, <http://semahead.pl/blog/unikalny-uzytkownik-vs-realny-uzytkownik-html.html> [↑](#footnote-ref-6)
7. Gemius Polska, <https://www.gemius.pl> [↑](#footnote-ref-7)
8. Google Analytics, <https://www.google.pl/intl/pl/analytics/features/index.html> [↑](#footnote-ref-8)
9. J. Levy „Strategia UX: jak tworzyć innowacyjne produkty cyfrowe, które spotkają się z uznaniem na rynku”, Helion, Gliwice, 2017, str. 44 [↑](#footnote-ref-9)
10. S. Krug „Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych”, Wydanie II, Helion, Gliwice, 2010, str. 34-35 [↑](#footnote-ref-10)
11. J. Levy „Strategia UX: jak tworzyć innowacyjne produkty cyfrowe, które spotkają się z uznaniem na rynku”, Helion, Gliwice, 2017, str. 34 [↑](#footnote-ref-11)
12. „User Experience – projektowanie pozytywnego doświadczenia”, <http://www.comarch.pl/erp/nowoczesne-zarzadzanie/numery-archiwalne/user-experience-projektowanie-pozytywnego-doswiadczenia/> [↑](#footnote-ref-12)
13. J. Levy „Strategia UX: jak tworzyć innowacyjne produkty cyfrowe, które spotkają się z uznaniem na rynku”, Helion, Gliwice, 2017, str. 32-33 [↑](#footnote-ref-13)
14. „Users Don’t Read, They Scan: Making Your Website Understandable and Scannable”, <http://appletoncreative.com/2015/09/10/users-dont-read-they-scan-making-your-website-understandable-and-scannable/>, tłum. własne [↑](#footnote-ref-14)
15. S. Krug „Nie każ mi myśleć! O życiowym podejściu do funkcjonalności stron internetowych”, Wydanie II, Helion, Gliwice, 2010, str. 48-49 [↑](#footnote-ref-15)
16. <http://www.kurshtml.edu.pl/html/wstep,html5.html> [↑](#footnote-ref-16)