***WWSIS@***

***Wrocławska Wyższa Szkoła***

***Informatyki Stosowanej***

Wydział Informatyki

Marcin Maciaszek

Nr albumu: 4291

Projekt i implementacja aplikacji do inwentaryzacji.

Praca magisterska

Kierunek: Informatyka

Specjalność/specjalizacja: Zaawansowane Bazy Danych

Praca wykonana pod kierunkiem:

dr. Inż. Katarzyna Pentoś

*Wrocław 2016*

Spis treści

[*1.* *Wstęp* 3](#_Toc457238700)

[*2.* *Analiza systemu* 3](#_Toc457238701)

[*2.1 Diagram przypadków użycia* 3](#_Toc457238702)

[*2.2 Schemat Bazy Danych* 3](#_Toc457238703)

[*2.3 Opis systemu* 3](#_Toc457238704)

[*2.4 Wymagania funkcjonalne* 3](#_Toc457238705)

[*2.5 Wymagania niefunkcjonalne* 3](#_Toc457238706)

[*3.* *Technologie* 4](#_Toc457238707)

[*3.1 Język HTML* 4](#_Toc457238708)

[*3.2 CSS* 4](#_Toc457238709)

[*3.3 JavaScript oraz jQuery* 6](#_Toc457238710)

[*3.4 Python* 7](#_Toc457238711)

[*3.5 Wzorzec projektowy MVC* 7](#_Toc457238712)

[*3.6 Framework* 8](#_Toc457238713)

[*3.7 Flask Framework* 9](#_Toc457238714)

[*3.8 Baza danych MySQL/PostgresSQL* 9](#_Toc457238715)

[*3.9 Wykorzystane narzędzia* 9](#_Toc457238716)

[*4.* *Porównanie interfejsu systemów dostępnych na rynku.* 9](#_Toc457238717)

[*5.* *Wnioski* 10](#_Toc457238718)

1. Wstęp

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. Analiza systemu

2.1 Diagram przypadków użycia

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.2 Schemat Bazy Danych

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.3 Opis systemu

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.4 Wymagania funkcjonalne

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

2.5 Wymagania niefunkcjonalne

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. Technologie

3.1 Język HTML

Język HTML (Hypertext Markup Language), jest to język ustanowiony przez konsorcjum World Wide Web Consortium (W3C), które tworzy specyfikację nakreślającą postać dokumentów prezentowanych w Internecie. Twórcą powyższego języka jest Tim Berners-Lee. HTML umożliwia publikowanie w Internecie dokumentów zawierających nagłówki, tekst, tabele, listy, zdjęcia oraz formularze. HTML jest językiem prezentacji tzw. “front-end”, bez niego aplikacje webowe były by nieczytelne.

Nie jest to język programowania. Analizując jego nazwę, możemy stwierdzić iż Hyper oznacza, że nie jest to język liniowy. Standardowe języki programowania charakteryzują się liniową strukturą, czyli kolejność wykonywania danego programu zależy od jego struktury. Cała zawartość strony jest cały czas dostępna dla użytkownika. Następnie, Text, oznacza iż wykorzystuje tekst. Jest on umieszczany w znacznikach „< >” i „</>”. Odzwierciedlają ostatnią część nazwy - Mark-up. Znaczniki umożliwiają przeglądarce odróżnienie zwykłego tekstu od kodu HTML. Nadają one atrybuty oraz informują jak powinien być formatowy tekst przez przeglądarkę. Ich budowa wygląda następująco: <znacznik>Zwykły tekst</znacznik> (znacznik otwierający, tekst, znacznik zamykający).

Twórca języka Tim Berners-Lee stworzył go tak, aby ułatwić komunikację naukowców w sieci. Dynamiczny rozwój Internetu w latach 90 przyniósł mu olbrzymią popularność. W dobie globalnego rozwoju język wciąż ewoluował. W roku 1995 została zatwierdzona pierwsza specyfikacja dla HTML 2. W 1996 roku powstała specyfikacja HTML 3.2, która uzyskała rekomendację w 1997 roku. Wraz z kolejnymi specyfikacjami wprowadzano zmiany, które zwiększały przenośność dokumentów HTML. Twórcy doszli do wniosku, iż aby język poprawnie się rozwijał oraz miał wpływ na rozwój Internetu powinien być w pełni kompatybilny ze wszystkimi platformami oraz przeglądarkami.

Przełomowa wersja HTML4 wydana w 1998 roku umożliwiła importowanie dodatkowych komponentów stron internetowych tj.: arkusze stylów, skrypty, ramki i osadzanie obiektów. Obsługa tekstu została udoskonalona. Wprowadzone zostały też narzędzia umożliwiające obsługę stron www przez osoby niepełnosprawne. Wersja 4.01 z 1999 roku, wprowadziła poprawę drobnych błędów.

Aktualnie dostępna jest wersja HTML5 wprowadzona w roku 2010 udoskonala oraz ułatwia tworzenie stron internetowych. Zostało dodanych bardzo dużo nowych elementów, czy znaczników. Jest bardzo często porównywany do Adobe Flash.

World Wide Web Consortium (W3C) jest odpowiedzialne za definiowanie standardów języka HTML. W3C zostało powołane w 1994 roku. Odpowiedzialne jest za rozwój technologii, ścieżki rozwoju, tworzy i zatwierdza specyfikacje, udostępnia oprogramowanie i narzędzia.

3.2 CSS

Kaskadowe Arkusze Stylów(Cascading Style Sheets), czyli CSS jest to prosty mechanizm dodawania stylów(czcionki, kolory, formatowanie) dla stron internetowych. Opracowany został w 1996 roku wraz ze specyfikacja HTML 4.0 przez organizacje W3C.

Arkusz stylów zawiera listę reguł, które informują w jaki sposób mają być interpretowane wybrane elementy języka HTML przez przeglądarkę. Umożliwia to separacje struktury dokumentu HTML od jego prezentacji.

Zapotrzebowanie na CSS zaczęła wzrastać wraz z rozwojem HTMLa i ilości jego elementów oraz możliwościami przeglądarek. Ułatwia on odnalezienie się w szkielecie dokumentu HTML oraz szybką i łatwą edycję jak i błyskawiczną modyfikację dokumentu.

Rozróżniamy trzy możliwości stosowania arkuszy stylów. Możemy definiować style w konkretnym elemencie HTML jest to styl wpisany, jednak gdy osadzimy go w obrębie elementu nagłówka head (znacznik <head> … </head>) taki rodzaj nazywamy osadzonym. Jeżeli wykorzystamy arkusz z pliku zewnętrznego nazwiemy ten arkusz zewnętrznym lub łączonym arkuszem stylów. Wszystkie trzy rodzaje arkuszy można stosować w jednym dokumencie. Łączone arkusze stylów przechowujemy w zewnętrznym dokumencie o rozszerzeniu “.css”.

W związku z tym, iż umożliwiono wykorzystanie typów arkuszy stylów dla jednego dokumentu rozróżniamy ich range(hierarchie):

* Domyślne ustawienia dla przeglądarki
* Łączony arkusz stylu (zewnętrzny dokument o rozszerzeniu “.css”)
* Osadzony arkusz stylu (zamieszczony w nagłówku dokumentu HTML)
* Styl wpisany (dla wybranego elementu HTML)

Styl wpisany otrzymał najwyższy priorytet względem reszty, zaraz po nim znajduje się osadzony w znaczniku head, następnie osadzony oraz ustawienia przeglądarki jako ostatnie.

Sposób osadzania CSS zależy od jego typu. Dla wpisanego, czyli dodanego do konkretnego elementu ma postać:

Dla arkusza osadzonego przedstawia się następująco:

Wewnątrz elementu <style></style> wprowadzamy właściwości oraz ich wartość dla konkretnym elementów.

selektor {właściwość: wartość;}

Selektor jest to znacznik, bądź element HTML, właściwość oznacza jego atrybut dla którego przypisujemy wartość. Należy pamiętać, że właściwość rozdzielamy dwukropkiem od wartości, a je umieszczamy w nawiasach klamrowych. Kiedy dla jednego selektora przypiszemy kilka właściwości rozdzielamy je średnikiem wewnątrz nawiasów klamrowych. W przypadku gdy wartość zbudowana jest z kilka wyrazów, bądź liczb podajemy ją w cudzysłowie:

Stylu wpisanego używamy w momencie, kiedy musimy wprowadzić właściwości dla jednego konkretnego znacznika. Dobrym przykładem jest znacznik <p>, czyli znacznik akapitu. W dokumentach HTML jest używany dość często, dlatego kiedy chcemy wprowadzić zmianę tylko dla jednego akapitu, używamy do tego wyżej wymienionego stylu.

<p style="font-size: 12pt; font-weight: bold; font-family: Verdana, sans-serif">Hello</p>

Dla osadzonych arkuszy jego zawartość umieszczamy pomiędzy znacznikami <style></style> w sekcji <head> dokumentu. Dla znacznika <style> należy dodać atrybut type="text/css". Atrybut ten pozwala przeglądarce interpretować zawartość znacznika jako jako arkusz stylów.

Łączony arkusz jest wręcz idealnym rozwiązaniem dla witryn zawierających wiele strom, czy też podstron. Umożliwia on nam na wykorzystanie go dla wielu dokumentów HTML.

W nagłówku dokumentu HTML wykorzystującego styl łączony musimy pamiętaj, aby go zdefiniować

Plik musi mieć rozszerzenie “.css”. Dzięki temu przeglądarka będzie wstanie czytać jego zawartość i odpowiednio interpretować. Atrybut href musi zawierać ścieżkę do dokumentu CSS.

3.3 JavaScript oraz jQuery

JavaScript jest to wysokopoziomowy, dynamiczny, nietypowany oraz interpretowany. Został on ustandaryzowany przez specyfikację ECMAScript. Razem z HTML oraz CSS jest traktowany jako podstawowe technologie WWW, jest podstawą nowoczesnego “design’u” stron internetowych wspieranych przez wszystkie przeglądarki internetowe.

JavaScript jest bardzo często mylony z językiem Java - są to dwa całkowicie różne od siebie języki programowania.

JavaScript osadzony na stronach internetowych wykorzystuje sie między innymi do:

* Automatycznej zmiany formatowania zawartości strony WWW.
* Umożliwia uruchamianie tzw. “okienek pop-up”.
* Pozwala na zmianę tekstu, bądź grafiki po wykonaniu określonej akcji myszką.

Aktualnie wszystkie strony WWW, które są interaktywne wykorzystują JavaScript do w/w akcji.

Skrypty JavaScript możemy osadzić na trzy sposoby:

* W nagłówku strony.
* Wewnątrz sekcji <body>.
* Jako zewnętrzny plik.

Wykorzystujemy do tego znacznik HTML <script>, który osadzamy w dokumencie HTML na w/w trzy sposoby.

jQuery jest lekką biblioteka napisaną dla języka JavaScript, umożliwia na prostsze i szybsze korzystanie z JavaScript. Bardzo popularna biblioteka, która nie obniża wydajności, a w zamian ułatwia tworzenie animacji, na dynamiczne zmiany strony oraz udostępnia korzystanie z AJAX’a. Skrypty napisane za pomocą jQuery nie wymagają żadnych zmian w dokumencie HTML.Głównym zyskiem z wykorzystania jQuery jest czysty, mało skomplikowany kod.

* Cechy biblioteki jQuery:
* Manipulacja HTML/DOM
* Manipulacja CSS
* Efekty oraz animacje
* AJAX
* Użyteczność

3.4 Python

3.5 Wzorzec projektowy MVC

Wzorzec projektowy jest to uniwersalne, sprawdzone rozwiązanie dla powtarzających się czynności, przypadków lub problemów. Wykorzystuje się je w rozwiązaniach opartych o programowanie obiektowe.

Model-View-Controller(MVC), czyli Model-Widok-Kontroler architektoniczny wzorzec projektowy umożliwiający organizację struktury aplikacji wyposażonych w interfejs graficzny dla użytkownika. Głównym założeniem tego wzorca jest oddzielenie głównych części aplikacji od siebie:

* Model odpowiada za reprezentację logiki aplikacji
* Widok opisuje widoki, czyli reprezentuje logikę aplikacji(tworzy dla nich interfejs graficzny)
* Kontroler przetwarza dane uzyskane od użytkownika, przetwarza przekazane dane, aktualizuje widoki, oraz aktualizuje model

MVC jest kluczem do stworzenia dobrego projektu dla aplikacji wyposażonej w GUI(graficzny interfejs użytkownika). Istnieje możliwość ponownego wykorzystania wielu obiektów, a ich interfejsy lepiej zdefiniowane. Aplikacje zaprojektowane z wykorzystaniem MVC są dużo łatwiej rozszerzalne w porównaniu do innych aplikacji.

Jak na schemacie powyżej, widok przekazuje akcje użytkownika do kontrolera. Ten reaguje, następnie jeżeli jest potrzeba to komunikuje się z modelem, który zwraca mu logikę, czy dane lub od razu zwraca do widoku rezultat akcji użytkownika.

3.6 Framework

W programowaniu komputerowym, framework jest szkieletem do tworzenia aplikacji, lub pewnych części aplikacji. Umożliwia też dostarczenie ogólnych funkcjonalności, które użytkownik może rozbudować według potrzeb danego projektu. Może być mylony z biblioteką programistyczną, jednak odróżnić możemy je kilkoma cechami:

Odwrócenie sterowania - przepływ sterowania(flow of control) w aplikacji narzucany jest poprzez framework zamiast programisty.

Rozszerzalność - framework powinien umożliwiać nadpisywanie, dodawanie lub pisanie nowych rozszerzeń do jego funkcjonalności przez programistę.

Niemodyfikowalny kod - framework powinien być stworzony w taki sposób, aby programista nie modyfikował go.

* Typowe elementy frameworka:
* Mechanizm uruchamiania i przetwarzania akcji.
* Mechanizm tworzenia logiki biznesowej aplikacji.
* Zarządzanie konfiguracją.
* Zarządzanie komunikacją z bazą danych.
* Obsługa formularzy.
* System szablonów.
* Obsługa błędów.
* Mechanizmy bezpieczeństwa, uwierzytelniania i kontroli dostępu.
* Generatory kodu.

Framework jest zestawem gotowych komponentów do budowy aplikacji internetowych zaprojektowane przez doświadczonych programistów. Ich dokumentacja jest kompletna oraz zawiera zazwyczaj “tutorial” z budowania przykładowej aplikacji i dzięki temu są łatwe w użyciu. Oszczędza to wiele czasu ponieważ jedyne co należy zrobić, jest nauka posługiwania się. Tak więc wystarczy skupić się na budowaniu danego projektu. Nowy uczestnik projektu, który zna dany framework, z łatwością zrozumie już napisany w nim kod, oraz szybko będzie mógł uczestniczyć w nim. W przeciwieństwie do frameworków autorskie rozwiązania pisane przez mniej doświadczonych programistów często zawierają dużo błędów i nie są tak dobrze zoptymalizowanie jak framework. Powoduje to duże problemy przy inwestygacji błędów, oraz naraża projekt na poważne problemy.

3.7 Flask Framework

Flask jest to micro web framework napisany w języku Python oraz bazowany na Werkzeug toolkit i Jinja2 template engine. Bezpłatnie dostępny na podstawie licencji BSD. Ostatnią stabilną wersją Flask jest 0.10.1, która została wydana w lutym 2016 roku. Przykładowymi aplikacjami, które wykorzystują Flask jest Printerest, Linkedin jak i witryna poświęcona internetowej społeczności Flask. Flask został nazwany jako micro framework ponieważ nie zmusza, czy nakazuje deweloperom na korzystanie ze specjalnych narzędzi, bądź bibliotek. Nie posiada takich warstw abstrakcji jak baza danych, walidacja formularzy, czy inne komponenty. Jednak Flask wspiera rozszerzenia, które mogą rozszerzać jego funkcjonalność, jednak muszą one byc zaimplementowane w nim samym. Istnieją rozszerzenia umożliwiające wykorzystanie technologii ORM, walidacje formularzy, autentykacje, oraz inne przydatne dodatki wykorzystywane w aplikacjach webowych. Wszystkie rozszerzenia są aktualizowane dużo częściej niż sam rdzeń frameworka.

Flask został stworzony przez Armina Ronachera, który należy do grupy Pocoo. Jest to grupa entuzjastów Pythona utworzona w 2004 roku. Oparty jest na Werkzeug WSGI toolkit jak i Jinja2 template engine, oba zostały opracowane przez grupę Pocoo. Pomimo, że Flask dość rzadko wydaje kolejne wersje, ze względu na swoją prostote został bardzo polubiany przez programistów języka Python.

Cechy frameworka:

* Zawiera serwer deweloperski oraz debuuger
* Zintegrowane środowisko testów jednostkowych
* Możliwość tworzenia RESTful API
* Wykorzystuje szablony Jinja2
* Wspiera secure cookies (postronie klienta)
* W 100% kompatybilne z WSGI 1.0
* Unicode-based
* Dokumentacja
* Kompatybilność z Google App Engine
* Dostepność wielu rozszeżeń zwiększających użyteczność aplikacji.

Przykładowy kod aplikacji "Hello World!":

3.8 Baza danych MySQL/PostgresSQL

3.9 Wykorzystane narzędzia

1. Porównanie interfejsu systemów dostępnych na rynku.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

1. Wnioski

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.