



UNIwersYTET ŁÓDZKI
WYDZIAŁ MATEMATYKI I INFORMATYKI
KIERUNEK ANALIZA DANYCH

Praca dyplomowa inżynierska

Projektowanie aplikacji internetowych z wykorzystaniem frameworka
Django

Autor: inż. Damian Wąsik
Promotor pracy: dr Piotr Fulmański

Łódź, 2021

Spis treści

1	Wstęp	1
1.1	Wprowadzenie	1
1.2	Cel pracy	1
1.3	Struktura pracy	2
2	Aplikacje internetowe	3
2.1	Rodzaje aplikacji	3
2.1.1	Aplikacje internetowe	3
2.1.2	Aplikacje desktopowe	3
2.1.3	Aplikacje mobilne	4
2.1.4	Aplikacje wiersza poleceń	4
2.2	W jaki sposób działa aplikacja internetowa?	4
2.2.1	Podział na frontend i backend	4
2.2.2	Architektura klient-serwer	5
2.2.3	Frameworki	5
2.2.4	API	5
2.3	Zbieranie wymagań	6
2.4	Projektowanie interfejsu graficznego	6
2.5	Tworzenie szkieletu aplikacji	6
2.6	Rodzaje frameworków	6
2.7	Framework Django	6
3	Pozyskiwanie danych do aplikacji	7
3.1	Web scraping	7
4	Systemy kontroli wersji	8
5	Współpraca na platformie typu Github	9
6	Projekt aplikacji śledzącej ceny telefonów	10
6.1	Opis problemu	10
7	Podsumowanie	11
	Bibliografia	13

Rozdział 1

Wstęp

1.1 Wprowadzenie

Szybki rozwój technologii sprawił, że w obecnych czasach są nieograniczone możliwości wdrażania pomysłów za pomocą ogólnodostępnych narzędzi. Przy poświęceniu określonej ilości czasu oraz pieniędzy ludzie są w stanie rozwiązać przeróżne problemy za pomocą aplikacji. Aplikacji używamy codziennie. Przy odpowiednim ich doborze oraz umiejętnym wykorzystaniu, mogą w bardzo wygodny sposób zaoszczędzić nam bardzo dużo czasu (a czas jest najcenniejszym zasobem). Uważam, że projektowanie aplikacji jest fascynującym doświadczeniem i postaram się dostarczyć jak najwięcej wartościowych informacji na ten temat w mojej pracy.

1.2 Cel pracy

W mojej pracy zawarte są informacje na temat projektowania aplikacji internetowej. Przed rozpoczęciem pisania założyłem, że opiszę krok po kroku jak wygląda proces tworzenia rozwiązania danego problemu, opiszę narzędzia jakie są potrzebne oraz całe środowisko pracy. Po to, żeby osoba nie specjalizująca się w tym temacie po przeczytaniu tego tekstu, była w stanie zaciekać się tematem i postawić sama pierwsze kroki bazując na wiedzy w nim zawartej. Wszystko oparte na przykładzie mojego projektu aplikacji internetowej, aby można było sobie od razu zwizualizować przedstawioną wiedzę.

1.3 Struktura pracy

Rozdział 2

Aplikacje internetowe

2.1 Rodzaje aplikacji

Najbardziej popularne kategorie aplikacji [2] to:

1. aplikacje internetowe
2. aplikacje desktopowe
3. aplikacje mobilne
4. aplikacje wiersza poleceń

2.1.1 Aplikacje internetowe

Aplikacje internetowe (webowe) to oprogramowanie dostępne z poziomu przeglądarki internetowej (takiej jak Google Chrome, Safari, czy Firefox). Przeglądarki są wbudowane w każdy popularny system operacyjny. Ich ogromną zaletą z perspektywy użytkownika jest wysoka dostępność. Żeby skorzystać aplikacji webowej, wystarczy znać jej adres internetowy. Przykładem aplikacji internetowej jest YouTube [1]

2.1.2 Aplikacje desktopowe

Aplikacje desktopowe instaluje się bezpośrednio na komputerze. Są one dystrybuowane na każdy system operacyjny oddzielnie (np. program napisany na system Win-

dows nie zadziała na Linux). Mogą być szybsze w działaniu, ponieważ zazwyczaj nie są uzależnione od szybkości łącza internetowego.

2.1.3 Aplikacje mobilne

Aplikacje mobilne są używane codziennie przez użytkowników smartfonów. Dzięki nim we własnej kieszeni mamy dostęp do wszystkich potrzebnych nam szybko informacji.

2.1.4 Aplikacje wiersza poleceń

Aplikacje wiersza poleceń to programy bez interfejsu graficznego. Zazwyczaj są wykorzystywane przez programistów, administratorów. Program uruchamia się komendą uruchomieniową (zazwyczaj też jest opcja podania argumentów uruchomieniowych w celu modyfikacji zachowania aplikacji). Po uruchomieniu w powłoce ¹ widoczne są komunikaty do użytkownika świadczące o przebiegu działania programu.

2.2 W jaki sposób działa aplikacja internetowa?

2.2.1 Podział na frontend i backend

Programowanie aplikacji internetowych dzieli się na dwie główne dziedziny: - frontend - backend

Zazwyczaj za każdą dziedzinę jest odpowiedzialny oddzielny zespół programistów. Ułatwia to w ogólnym zarządzaniu w projekcie.

Frontend odpowiada za wszystko co użytkownik widzi jak się czuje korzystając z aplikacji. Za to jak strona wygląda, jakie są na niej animacje, przejścia. Jak elementy są ułożone na stronie, jaka jest szata kolorystyczna. Programiści frontend piszą kod źródłowy serwisu w języku HTML, stylizują wygląd strony w CSS oraz odpowiadają za doświadczenie użytkownika na stronie za pomocą języka javascript.

¹(ang. shell) – program komputerowy pełniący rolę pośrednika pomiędzy systemem operacyjnym lub aplikacjami a użytkownikiem, przyjmując jego polecenia i „wyprowadzając” wyniki działania programów

Backend jest związany z całą logiką aplikacji. Wartości widoczne na stronie, muszą być wcześniej odpowiednio przetworzone i wyciągnięte na przykład z bazy danych. Zespoły backendowe są odpowiedzialne za takie zadania: - komunikacja z bazami danych - bezpieczeństwo danych - tworzenie interfejsów API

2.2.2 Architektura klient-serwer

Aplikacje internetowe są dostępne za pomocą protokołu HTTP. Jest on zaprojektowany na bazie architektury klient-serwer. Klientem jest użytkownik, który chce wejść na stronę. Wysyła przygotowane zapytanie do serwera (serwisu, z którego chce skorzystać). Najprostszym zapytaniem jest po prostu adres internetowy strony.

2.2.3 Frameworki

Oprócz znajomości języków programowania wykorzystuje się różne frameworki. Framework to przygotowany zestaw narzędzi zaprojektowany po to, żeby ułatwiać rozwiązanie danego problemu. Popularne frameworki są utrzymywane przez zespoły programistów, zazwyczaj posiadają też obszerną dokumentację wraz z przykładami użycia. Są też dobrze przetestowane. Dzięki temu, wszystkie sprawy związane z utrzymywaniem tego kodu są po stronie innego zespołu. Oszczędza się również dużo czasu. Zamiast myśleć nad przygotowywaniem potrzebnych narzędzi, tworzy się bezpośrednio logikę aplikacji.

2.2.4 API

API jest to interfejs umożliwiający manipulowanie aplikacją. Jest to ustandaryzowany sposób komunikacji pomiędzy programistami. Sposobem na wykorzystanie API może być komunikacja między zespołem frontend i backend. Zespoły backend przetwarzają dane i wystawiają tzw. końcówki (endpoints)

2.3 Zbieranie wymagań

2.4 Projektowanie interfejsu graficznego

2.5 Tworzenie szkieletu aplikacji

2.6 Rodzaje frameworków

2.7 Framework Django

Rozdział 3

Pozyskiwanie danych do aplikacji

3.1 Web scraping

Web scraping to technika pozwalająca na pozyskiwanie ogólnodostępnych w sieci danych. Dzięki narzędziom do web scrapingu jesteśmy w stanie analizować kod źródłowy strony i wyciągać nieustrukturyzowane dane.

Rozdział 4

Systemy kontroli wersji

W systemach informatycznych występuje bardzo szybkie tempo zmian. Aby wspomóc programistów w zarządzaniu wersjami oprogramowania, wprowadzono systemy kontroli wersji. Najpopularniejszy obecnie jest GIT. Istnieją również platformy internetowe bazujące na tym systemie, które dostarczają interfejs graficzny i umożliwiają łatwiejszą komunikację programistom pracującym nad projektem. Najpopularniejsze platformy to Github, Gitlab czy BitBucket. Są to rozproszone systemy, dzięki temu każdy z programistów ma możliwość pracy w wydzielonym środowisku, nie wchodząc w kolizję innym programistom z zespołu.

Rozdział 5

Współpraca na platformie typu Github

Rozdział 6

Projekt aplikacji śledzącej ceny telefonów

6.1 Opis problemu

Rozdział 7

Podsumowanie

Bibliografia

[1] Youtube.

[2] digitallyher.pl. Podstawowe typy aplikacji.

Spis rysunków