Paweł Dymora, Mirosław Mazurek, Jakub Rywka

Dydaktyczne aspekty projektowania aplikacji webowych z wykorzystaniem frameworka Djano w Pythonie

Edukacja - Technika - Informatyka nr 3(13), 302-307

2015

Paweł DYMORA, Mirosław MAZUREK, Jakub RYWKA
Politechnika Rzeszowska, Polska

Dydaktyczne aspekty projektowania aplikacji webowych z wykorzystaniem frameworka Django w Pythonie

Wstęp

Dynamiczny rozwój internetu i technologii z nim powiązanych wymusza na projektantach aplikacji dostosowanie ich produktów do wymagań odbiorców. Obserwuje się zmianę procesu projektowego produktu dążącego do pełnej personalizacji, co z kolei eliminuje zbędny transfer i obciążenie serwerów, ograniczając ruch tylko do faktycznie potrzebnych „treści". Aplikacje internetowe były początkowo aplikacjami statycznymi udostępniającymi jedynie określone dane. Artykuł pokazuje środowisko Pythona, które w prosty i intuicyjny sposób pozwala na dynamiczną zmianę treści w dowolnym momencie. Język Python to uniwersalne i nowoczesne środowisko programowania aplikacji konsolowych, desktopowych oraz internetowych aplikacji bazodanowych z rosnącą popularnością, dzięki czemu może być wykorzystane w dydaktyce przedmiotów programistycznych zarówno na studiach, jak i w szkołach ponadgimnazjalnych, uzupełniając istniejące technologie programistyczne. Celem artykułu jest przygotowanie w pełni użytecznego i wykonalnego projektu aplikacji zapewniającego jej właściwą strukturę i funkcjonalność. Jednym z najważniejszych etapów procesu jest analiza wymagań użytkownika, na podstawie której zespół projektowy może dopiero przystąpić do tworzenia modelu funkcjonalnego aplikacji. Model funkcjonalny zostaje w dalszych etapach procesu wykorzystany do budowy modelu logicznego aplikacji oraz do stworzenia podstaw jej struktury fizycznej. W projekcie wykorzystano język Python, framework Django oraz bazę SQLite.

Język programowania Python

Język Python jest obiektowym, dynamicznym językiem programowania wysokiego poziomu. Dzięki otwartej licencji i ogólnodostępnym bibliotekom w łatwy sposób otrzymuje się dostęp do tego profesjonalnego i coraz bardziej popularnego narzędzia programistycznego. Wykorzystywany jest w wielu znanych portalach, takich jak YouTube czy Google. Aplikacje napisane w tym języku działają pod znaczną większością systemów – Windows, Linux/Unix, Mac, Amiga oraz w smartfonach. Python wywodzi się od języka ABC, który został wynaleziony w CWI (hol. Centrum voor Wiskunde en Informatica). Dalsze prace nad udoskonalaniem produktu miały miejsce m.in. w CNRI (ang. Corporation for National Research Initiatives) – amerykańskiej organizacji

znajdującej się w Reston w stanie Virginia w USA [Norton i in. 2006]. Python zawdzięcza swoją popularność prostej obsłudze, oferując przy tym duże możliwości. Kod jest łatwy w pisaniu oraz przejrzysty, co znacznie ułatwia odnajdywanie błędów w celu ich poprawienia. Duża liczba bibliotek umożliwia dostęp do wielu baz danych, dając możliwość edycji audio i wideo oraz tworzenia stron internetowych. Otwarty kod źródłowy pozwala na dowolne używanie języka i publikowanie aplikacji bez ponoszenia żadnych kosztów. Dzięki swojej elastyczności pierwotny kod w łatwy sposób można edytować i rozszerzać [Krajka 2011].

Framework Django

Webowy framework wysokiego poziomu Django napisany w języku programowania Python cechuje się przejrzystą architekturą, szybkim rozwojem aplikacji, dużą swobodą w pisaniu oraz prostotą wykonania. Opiera się na wzorcu projektowym MVT (*model – view – template*), dzięki czemu jego struktura jest prosta i przejrzysta, dająca możliwość prostego, a zarazem szybkiego rozwoju. Django powstało z myślą o szybko rozwijającym się portalu dziennikarskim, w którym zamieszczane są nowości ze świata. Doskonale nadaje się także do aplikacji webowych, gdzie informacje zmieniają się często [djangobook.com]. Struktura katalogowa projektu o nazwie `nowyprojekt` obejmuje następujące elementy:

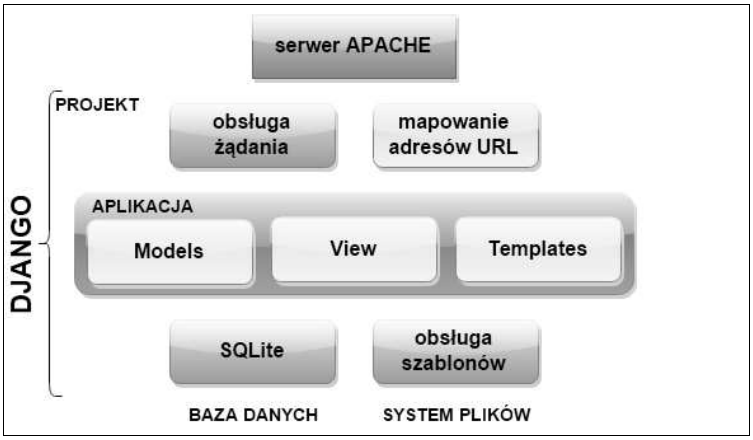
- `__init__.py` – zawiera informacje o tym, że katalog nadrzędny powinien być traktowany jako pakiet Pythona,
- `manage.py` – pozwala na interakcję z projektem i działa z wiersza poleceń,
- `settings.py` – pozwala na konfigurację projektu,
- `urls.py` – znajduje się tu deklaracja adresów URL dla tego projektu, pewnego rodzaju mapa strony zbudowanej na bazie Django.

Zaletą środowiska Django jest dołączony do niego serwer deweloperski. Jest to „lekki” serwer WWW napisany w całości w Pythonie. Dzięki niemu można sprawdzić postępy pracy bez potrzeby konfiguracji serwera produkcyjnego. Domyślnie serwer deweloperski działa na porcie 8080 (http://127.0.0.1:8080/).

Projekt aplikacji webowej

Proces tworzenia aplikacji zademonstrowano na przykładzie witryny internetowej kina. Głównym zadaniem takiego systemu informatycznego jest natychmiastowe informowanie klientów o zmianach w repertuarze, podglądzie sali czy nowościach. Użytkownik zdalnie ma możliwość sprawdzenia, jaki repertuar jest w danej chwili wyświetlany, oraz zarezerwowania wybranego dla siebie miejsce. Dzięki systemowi rejestracji każdy klient traktowany jest indywidualnie. Po zalogowaniu i przeglądzie witryny może wyrazić swoją

opinię, przyczyniając się do ciągłego rozwoju aplikacji. Projektowany system i jego strukturę przedstawiono na rys. 1.



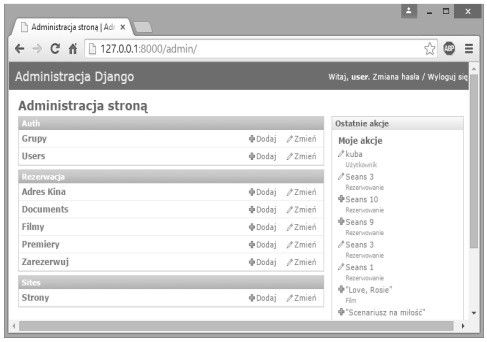
Rys. 1. Struktura aplikacji w Django

Aplikacja składa się z dwóch elementów. Pierwszym z nich jest panel administracyjny służący do zarządzania całą aplikacją. Pełny dostęp do panelu powinien posiadać tylko administrator. Drugi element aplikacji to witryna WWW, do której dostęp mają wszyscy użytkownicy i klienci kina. Python w wersji 2.5 używa bibliotek SQLite [Diamond i in. 2010]. Dzięki modułowi Django nie ma potrzeby odwoływania się do SQLite3 bezpośrednio. Python obsługuje także bazy danych typu Dbm (*databse manager*), takie jak dbm, GNU's dbm, Berkeley DB, i serwery baz danych DB2, MySQL, PostgreSQL, MS SQL oraz Oracle. Bazy danych typu Dbm mogą przechowywać dane w pliku przypominającym słownik, a dostęp do danych jest zazwyczaj po kluczu. Niestety, takie rozwiązanie ma spore ograniczenia. Wartości danych to zwykle tekst, zaś implementacja odbywa się za pomocą tablic haszujących lub drzewa, a brak odrębnego serwera i zapis danych do pliku znacznie ogranicza jego możliwości. Proces tworzenia tabel z danymi wykorzystującymi SQLite3 wygląda następująco:

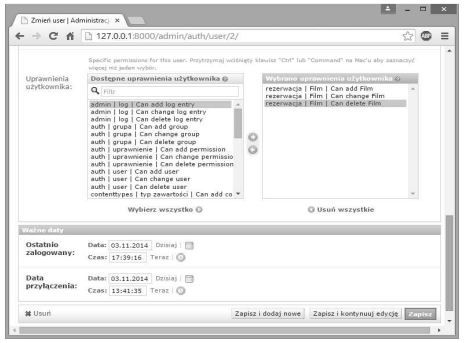
```
>>> import sqlite3
>>> conn = sqlite3.connect('artysci')
>>> c=conn.cursor()
>>> c=c.execute("""create table Artists (
      ArtistID INTEGER PRIMARY KEY,
      ArtistName TEXT);""");
>>> c=c.execute("""create table CDs (
      CDID INTEGER PRIMARY KEY,
```

```
      ArtistID INTEGER NOT NULL,
      Title TEXT NOT NULL,
      Date TEXT);""");
>>> c=c.execute("""insert into Artists (ArtistID,ArtistName) val-
ues (NULL, 'Jan Kowalski')""")
>>>  c=c.execute("""insert  into  CDs  (CDID,ArtistID,Title,Date)
values (NULL,1,'Stary','1995')""")
```

Do zarządzania aplikacją stworzoną we frameworku Django służy wbudowany panel administracyjny. Dzięki niemu można edytować pola klas w modelu, sprawować kontrolę nad użytkownikami i zarządzać całą witryną (rys. 2). Posiadając pełny dostęp, można edytować wszystkie elementy bloków Auth, Rezerwacja oraz Sites. Obiekty znajdujące się w kategorii Auth zawierają informacje na temat użytkowników oraz grup, do których oni należą. Można dodawać i usuwać użytkowników, edytować podane przez nich informacje, zmieniać hasła dostępu do utworzonych kont, nadawać uprawnienia, wyświetlać datę ostatniego logowania, ustawienia dokładnych uprawnień każdemu z użytkowników, co jest niebywałą zaletą np. podczas ustalania przywilejów każdego z pracowników oddzielnie (rys. 3). Stworzoną aplikację charakteryzuje przejrzystość i prostota obsługi. Na rys. 4 przedstawiono przykładowy widok główny witryny. Aplikacja zawiera szereg funkcjonalności, np. przeglądanie informacji dotyczących kina, repertuaru, premier, uruchomienie zwiastuna filmu (użycie multimedialnych typów danych), rezerwację biletu itp [Hoekman 2010].



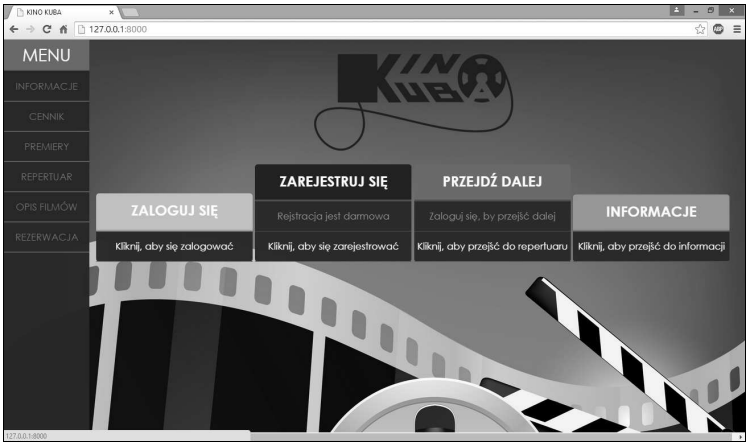
Rys. 2. Przykładowy panel administracyjny



Rys. 3. Przykład nadawania uprawnień użytkownikom

Przy użyciu skryptu JavaScript istnieje możliwość sortowania seansów według daty wyświetlania lub tytułu. Przy każdym z tytułów widnieje informacja o typie filmu (lektor, napisy, dubbing). Jeśli film jest dostępny, można np. dokonać rezerwacji miejsca. Dla każdego z filmów stworzono oddzielne

podglądy sali. Kolor zielony oznacza miejsca wolne, a czerwony – miejsca zajęte. Wszystkie miejsca można edytować za pomocą panelu administracyjnemu, zaznaczając lub odznaczając odpowiednie pole.



Rys. 4. Widok główny aplikacji internetowej

Podsumowanie

Głównym celem artykułu było przedstawienie pełnego procesu tworzenia aplikacji internetowej wykorzystującej framework webowy Django napisany w języku programowania Python. Podczas tworzenia projektu użyty został język znaczników HTML, skryptowy język programowania JavaScript oraz kaskadowe arkusze stylów CSS. Dzięki dołączeniu bazy danych SQLite3 powstała sprawnie działająca witryna internetowa obsługująca kino. Przy użyciu przeglądarki internetowej użytkownik jest w stanie swobodnie poruszać się po kolejnych widokach stron bez konieczności instalowania dodatkowego oprogramowania. Niebywałą zaletą aplikacji jest fakt, że bazuje ona na darmowej licencji, dzięki czemu nie wymaga żadnych dodatkowych nakładów finansowych. Odbiorca otrzymuje produkt gotowy do wdrożenia. Kod aplikacji jest czytelny, a widoki dla poszczególnych stron zamieszczone są w oddzielnych dokumentach HTML. W bardzo prosty sposób można dodawać kolejne funkcjonalności do aplikacji. Rosnąca popularność nie tylko samego języka Python, ale także frameworka Django sprawia, że aplikacja napisana przy użyciu tego oprogramowania jest rozwiązaniem bardzo przyszłościowym i dobrą alternatywą dla komercyjnych systemów.

Literatura

http:/www.djangobook.com.

Hoekman R. (2010): *Magia interfejsu. Praktyczne metody projektowania aplikacji internetowych*, Gliwice.

Horstmann C.S., Cornell G. (2009): *Java Core. Techniki zaawansowane*, Gliwice.
Krajka A. (2011): *Python – podstawy języka i aplikacje internetowe*, Lublin.
Kreibich J. (2010): *Using SQLite*.
Norton P., Samuel A., Altel D., Foster-Johnson E., Richardson L., Diamond J., Parker A., Roberts M. (2006): *Python. Od podstaw*, Gliwice.
Wilson G. (2006): *Przetwarzanie danych dla programistów*, Gliwice.

Streszczenie

W artykule przedstawiono proces tworzenia interaktywnych aplikacji webowych wykorzystujących język programowania wysokiego poziomu Python, framework sieciowy Django, język znaczników HTML, kaskadowe arkusze stylów CSS oraz skryptowy język programowania JavaScript. Celem artykułu jest zapoznanie studenta z procesem opracowania całego projektu technicznego obejmującego zarówno środowisko programistyczne, wzorce projektowe, biblioteki wraz z przygotowaniem analizy funkcjonalnej aplikacji.

Słowa kluczowe: nauczanie, Python, bazy danych, SQLite, Django.

Educational Aspects of Designing Web Applications Using Django Framework in Python

Abstract

The paper presents the process of creating interactive web applications using high-level programming language Python, Django network framework, HTML markup language, Cascading Style Sheets CSS and scripting language JavaScript. The aim of this article is to familiarize students with the process of the development of the entire project including technical development environment, design patterns, libraries including the preparation of the applications functional analysis.

Keywords: teaching, Python, databases, SQLite, Django.