*DJANGO STATE OF THE ART*

1. *MOTIVATIE PYTHON 3 + DJANGO*

Am ales framework-ul Django, implementat in Python, deoarece am considerat ca acest framework este cel mai fiabil in cazul aplicatiei noastre.

Alegerea limbajului de programare Python in vederea implementarii partii de backend se bazeaza pe cateva criteria importante:

1. Este usor de folosit, deoarece Python este un limbaj netipizat(nu se defines tipurile de variabile), aplicatia concentrandu-se astfel, din punct de vedere al dezvoltatorilor, mai mullt pe functionalitatile esentiale ale aplicatiei, in detrimentul anumitor detalii de implementare.
2. Din punct de vedere al detaliilor tehnice, si din punct de vedere statistic, Python este un limbaj care se invata mai usor fata de alte limbaje de programare de aceeasi categorie. Prin urmare, este un limbaj usor de inteles, ceea ce il face automat usor de citit. In majoritatea proceselor de dezvoltare a aplicatiilor software, o problema care se ridica reprezinta intelegerea modulelor scrise de catre coechipieri de catre alti coechipieri. Prin constructia limbajului, Python cauta sa reduca aceasta problema.
3. Costurile timp de implementare sunt mai reduse, deoarece Python dispune de un sistem de import al librariilor foarte puternic. In ciuda acestui fapt, timpul de executie este, in unele cazuri, mai mare, deoarece Python este un limbaj pur interpretat, iar procesul de interpretare dureaza mai mult decat procesul de compilare.
4. Python este un limbaj multifunctional, care poate fi folosita pe o gama larga de aplicatii. In ceea ce priveste popularitatea limbajului, aceasta este in crestere(momentan este pe locul 3, conform <https://www.tiobe.com/tiobe-index/> ) fiind folosit de la aplicatii web, pana la aplicatii statistice, machine learning, prelucrarea limbajului natural, si implementarea celor mai complexe aplicatii. Python este folosit de companii precum Google, Facebook, BitDefender, si multe altele. In anul 2018, Python a obtinut din partea aceluiasi site premiul pentru cel mai popular limbaj de programare al anului.

Django este un framework pentru aplicatii web(si nu numai) implementat in limbajul Python. Django este un framework care implementeaza arhitectura MVT(Model View Template)-partea de controller este implementata automat de catre Django prin respectarea unui contract template. Este un instrument de a crea robustete si simplitate modului de implementare al partii de backend a oricarei aplicatii Web. Este folosit de companii precum NASA, YouTube, Google, Instagram, BitDefender etc. tocmai datorita eficientei sale.

Avantaje(+):

Django este un framework care suporta foarte multe pluginuri, inclusive integrare cu Google Cloud

Comunitatea de software developeri, si comunitatea de support Django este una numeroasa

Django ofera un support foarte puternic pentru ORM( Object-Relational-Mapping), ceea ce usureaza foarte mult comunicatia serverului de aplicatii cu serverul de baze de date, indiferent care ar fi sistemul de gestiune, astfel accelerand timpul de constructie al programului. Fara acest support, crearea tabelelor si definirea interogarilor si procedurilor eera necesara sa fie implementata manual.

Django ofera support pentru implementarea design patternului DRY(Don’t Repeat Yourself)

Colectia de librarii*(‘batteries’)* pe care o dispune Django este foarte puternica, facilitand o multime de aspecte care tin de domeniul Middleware. Astfel, Django, prin utilitarele acesteia. Cateva dintre lucrurile pe care le putem controla prin intermediul Django: autentificare prin *auth package,* administrarea interfetei prin *admin package*, gestionarea sesiunilor prin *Session Package,* gestionarea mesajelor sesiunilor prin *Messages Package,* etc

Scalabilitate pe orizontala si pe vertical

Este fiabil pentru orice SGBD, in special Oracle, PostgreSQL. Am ales la nivel de solutie SGBD PostgreSQL deoarece am considerat ca se potriveste cel mai bine cu tipul si cantitatea de date pe care le integram in aplicatie. In plus, PostgreSQL dispune de o interfata atractiva pentru a gestiona datele la nivel de baze de date. Astfel, ofera suport puternic pentru teste pe baza de date sau pe date in sine.

Are caracteristici si utilitare de securitate foarte puternice

(-) Dezavantajele sunt minore, spre exemplu Django nu este potrivit pentru aplicatii reduse ca si magnitudine, raportul timp de executie/complexitatea aplicatiei nefiind optim in acest caz. De asemenea, este necesara cunoasterea intregului sistem ce urmeaza a fi dezvoltat, pentru a putea fi folosit in mod optim acest framework.

1. *COMPARATII INTRE DJANGO SI ALTE FRAMEWORKURI PENTRU BACKEND*

LARAVEL (PHP)

(+): Are o comunitate de developeri pe GitHub foarte mare

Suporta cicluri de dezvoltare software foarte rapide, prin constructia limbajului

Documentatia frameworkului este foarte bine constituita

Ca si Django, Laravel dispune de un sistem de librarii cu arbore de gestiune al dependentelor, prin care poate sa efectueze servicii middleware si comunicatiile cu bazele de date

(-): Este o platform relativ noua, nerecomandata novicilor

Comunitatea de support este relativ mica fata de alte platform

Procesul de reverse routing este ingreunat la executia codului

Comparativ cu Django, Laravel are un timp de executie in ceea ce priveste timppul de acces la baza de date, la parsarea JSON, etc.

Comparativ cu Django, conform cercetarilor, desi are implementate standarde de securitate inalte, Laravel nu le dispune pe ale celui din urma.

SPRING MVC(JAVA)

(+): Comunica cu serverele Tomcat extreme de usor(prin Maven de exempplu)

Asigura o multitudine de API-uri

Configurarea unui program de tip Spring se poate face in trei moduri: ori prin arbore de dependente XML, ori prin adnotari de tipul @notation, sau manual prin java config

(-) Adaugarea dependentelor se face relativ dificil ori prin importarea de JAR sau WAR, ori prin modificarea arborelui de dependente

Nu este bazat pe Java Enterprise Edition Standard

Invatarea acestui framework ar necesita mai mult timp

NODE.JS (JAVASCRIPT)

(+): Performanta este ridicata

Comunitatea de pe GitHub este mai mare, comparativa cu cea a utilizatorilor de Django

Arhitectura este una care se bazeaza pe dirijarea evenimentelor, exista un fir de executie pentru toate procesele

Dispune de o gama larga de librarii, iar node package manager(npm) reprezinta una din cele mai mari resurse de packages din totalitatea framework-urilor

(-): Asincron

Din cauza ca toate procesele sunt distribuite pe un fir de executie, consumul de CPU este mai mare decat in cazul Django

Sunt necesare o multime de callback-uri imbricate, ceea ce ar putea reduce viteza proceselor.

ASP.NET MVC (C#)

(+): Codul este relativ stabil

Integrare puternica cu IDE Microsoft Visual Studio Code

Integrare puternica cu MVC Model

Integrare puternica cu servicii cloud, in special Micrososft Azure

(-): Toata platforma este dependent de servicii Microsoft

1. *CONCLUZII*

Analizand argumentele de mai sus, si luand in seama princlipalele avantaje si dezavantaje ale principalelor frameworkuri pentru aplicatii web – partea de backend, putem trage concluzia ca DJANGO este o alternativa fiabila in realizarea backend-ului aplicatiei site-ului Facultatii de Informatica din Iasi.