



Universitatea Tehnică a Moldovei

Proiectarea aplicațiilor Web distribuite în baza framework-ului Django

Student:

ȘERBUL Iustin

Conducător:

**Moraru Victor,
conf.univ., dr.**

Chișinău 2022

Adnotare

la teza de masterat cu tema "Proiectarea aplicațiilor web distribuite bazate pe framework-ul Django" a masterandului Iustin Șerbul.

Cuvinte cheie: Framework Django, aplicații distribuite, autentificarea factor dublu, mașină virtuală.

Structura tezei de masterat cuprinde: introducere, trei capitole de bază, concluzii, bibliografie și referințe, 64 de pagini de text și 32 de figuri.

În cadrul tezei de masterat, a fost elaborată și creată o aplicație web bazată pe Django Framework. Această aplicație web a fost concepută pentru orice tip de organizații în care utilizatorii se gândesc la securitatea și confidențialitatea datelor sale personale. În această lucrare, accentul a fost pus în primul rând pe măsurile de securitate legate de autentificarea utilizatorilor, pe privilegiile utilizatorilor și pe posibilitatea de a vizualiza o anumită parte a aplicației în funcție de privilegiile acestora. De asemenea, a fost adăugată autentificarea cu factor dublu, cu scopul ca numai utilizatorii care au o cheie unică generată la fiecare 30 de secunde să se poată autentifica și să vizualizeze informațiile necesare.

De asemenea, pentru realizarea anumitor sarcini, cum ar fi realizarea de grafice și alte tipuri de diagrame, a fost utilizat javascript în paralel cu css.

Annotation

for master thesis on "Designing Distributed Web Applications based on the Django Framework" by master student Iustin Serbul.

Keywords: Django Framework, distributed applications, dual factor authentication, virtual machine.

The structure of the master's thesis includes: introduction, three basic chapters, conclusions, bibliography and references, 64 pages of text and 32 figures.

In the master's thesis, was developed and created a web application based on the Django Framework. This web application was designed for the any kind of organizations where users are thinking about security and privacy of their personal data. In this work, the focus was primarily on security measures related to user authentication, user privileges, and the possibility of viewing a certain part of the application depending on their privileges. Double-factor authentication has also been added, with the aim that only users who have a one-time key generated every 30 seconds can log in and views the necessary information.

Javascript was also used in parallel with css to perform certain tasks, such as making graphs and other types of diagrams

Cuprins

Introducere	2
1 Noțiuni generale despre conceptul de Aplicație web, Framework, și istoria dezvoltării acestora.	4
1.1 Istoria apariției aplicațiilor web	4
1.2 Framework-ul Django ca parte componentă al limbajului Python	9
1.2.1 Istoria creării Framework-ului Django	9
1.2.2 Generalități despre limbajul de programare Python	10
1.3 Măsurile de protecție implementate în Framework-ul Django	13
2.2.1 Cross Scripting-ului (XSS)	13
2.2.2 Falsificarea cererilor încrucișate pe site (CSRF)	15
2.2.3 Protecție SQL Injectarea și protecție Clickjacking	16
1.3 Avantajele Utilizării Framework-ului Django	17
1.3.1 Avantajele utilizării Framework-ului Django față de analogii acestuia	17
1.3.2 Descrierea altor frameworkuri	19
2 Proiectarea unui site web utilizând Django	23
2.1 Utilizarea Django CMS	24
2.2 OpenLDAP ca instrument de autentificare	26
2.3 Utilizarea serverului pe bază de Ubuntu pentru rularea aplicației web.	29
3 Implimentarea și testarea proiectului	31
3.1 Instalarea dependențelor necesare	31
3.1.1 Instalarea limbajului de programare Python.	31
3.1.2 Instalarea IDE Pycharm.	32
3.1.3 Ridicarea mașinii virtuale Ubuntu server	33
3.1.4 Instalarea OpenLDAP și phpLDAPadmin pe serverul ridicat	38
3.2 Configurarea Framework-ului Django	45
3.2.1 Instalarea modulului pentru autentificare prin LDAP	47
3.2.2 Instalarea modulului pentru autentificare dublu-factor.	48
Concluzie	52
Bibliografie	53
Anexa 1. Codul înscris în settings.py	55
Anexa 2. Codul înscris în urls.py	63
Anexa 3. Codul înscris în index.html	64

Introducere

Un cadru web este o colecție de instrumente modulare care elimină o mare parte din dificultățile - și repetițiile - inerente dezvoltării web. De exemplu, majoritatea site-urilor web au nevoie de aceeași funcționalitate de bază: capacitatea de a se conecta la o bază de date, de a seta rutele URL, de a afișa conținut pe o pagină, de a gestiona securitatea în mod corespunzător și așa mai departe. În loc să recreeze toate acestea de la zero, programatorii au creat de-a lungul anilor cadre web în toate limbajele de programare importante: Django și Flask în Python, Rails în Ruby și Express în JavaScript, printre multe, multe altele.

Django a moștenit abordarea "cu baterii incluse" din Python și include suport din start pentru sarcini comune în dezvoltarea web: autentificarea utilizatorului, șabloane, rute și vizualizări, interfață de administrare, securitate robustă, suport pentru mai multe backend-uri de baze de date și multe, multe altele.

Această abordare permite dezvoltatorilor web să se concentreze pe ceea ce face ca o aplicație web să fie unică, mai degrabă decât să reinventeze de fiecare dată roata pentru funcționalitatea standard și sigură a aplicațiilor web.

În schimb, mai multe cadre populare - mai ales Flask în Python și Express în JavaScript - adoptă o abordare de tip "microframework". Acestea oferă doar strictul necesar pentru o pagină web simplă și lasă la latitudinea dezvoltatorului să instaleze și să configureze pachete terțe pentru a reproduce funcționalitatea de bază a site-ului web. Această abordare oferă mai multă flexibilitate dezvoltatorului, dar generează și mai multe posibilități de a face greșeli.

Începând cu 2019, Django se află în dezvoltare activă de peste 14 ani, ceea ce îl face un veteran grizzly în anii de software. Milioane de programatori au folosit deja Django pentru a-și construi site-uri web, ceea ce este, fără îndoială, un lucru bun. Dezvoltarea web este dificilă. Nu are sens să repetăm același cod - și aceleași greșeli - atunci când o comunitate mare de dezvoltatori geniali a rezolvat deja aceste probleme pentru noi.

În același timp, Django rămâne în dezvoltare activă și are un program de lansare anual. Comunitatea Django adaugă în mod constant noi caracteristici și îmbunătățiri de securitate. Și, cel mai bine, este scris în limbajul de programare Python, minunat de ușor de citit, dar încă puternic. Pe scurt, dacă construiești un site web de la zero, Django este o alegere fantastică.

Deoarece designul World Wide Web nu era în mod inerent dinamic, primul hipertext era format din HTML codificat manual și publicat pe servere web. Orice modificare a paginilor publicate trebuia să fie efectuată de către autorul paginilor respective. În 1993, a fost introdus standardul Common Gateway Interface (CGI) pentru interfațarea aplicațiilor externe cu serverele web, pentru a oferi o pagină web dinamică care să reflecte intrările utilizatorului. Cu toate acestea, implementările inițiale ale interfeței CGI au avut, de obicei, efecte negative asupra încărcării serverului, deoarece fiecare solicitare pornea un proces separat. Implementările mai recente utilizează procese persistente, printre alte tehnici, pentru a reduce amprenta în resursele serverului și pentru a oferi un spor general de performanță. În 1995, au apărut pentru prima dată medii de dezvoltare server/limbaj complet integrate și au fost introduse noi limbaje specifice web, cum ar fi ColdFusion, PHP și Active Server Pages. Deși marea majoritate a limbajelor pentru crearea de pagini web dinamice dispun de biblioteci care să ajute la îndeplinirea sarcinilor comune, aplicațiile web necesită adesea biblioteci specifice pentru anumite sarcini, cum ar fi crearea de HTML (de exemplu, JavaServer Faces).

Cadrelor web trebuie să funcționeze în conformitate cu regulile arhitecturale ale browserelor și protocoalelor, cum ar fi HTTP, care este fără stare. Paginile web sunt servite de un server și pot fi apoi modificate de către browser cu ajutorul JavaScript. Ambele abordări au avantajele și dezavantajele lor. Modificările paginilor de pe server necesită, de obicei, reîmprospătarea paginii, dar permit utilizarea oricărui limbaj și folosirea unei puteri de calcul mai mari. Modificările de pe partea clientului permit actualizarea paginii în bucăți mici, ceea ce seamănă cu o aplicație desktop, dar sunt limitate la JavaScript și sunt rulate în browserul utilizatorului, care poate avea o putere de calcul limitată. De obicei, se folosește o combinație între cele două. Aplicațiile care folosesc foarte mult JavaScript se numesc aplicații cu o singură pagină și, de obicei, folosesc un cadru web JavaScript client-side pentru a organiza codul.

Multe cadre urmează modelul arhitectural MVC pentru a separa modelul de date cu reguli de afaceri de interfața cu utilizatorul. Acest lucru este în general considerat o practică bună, deoarece modularizează codul, promovează reutilizarea codului și permite aplicarea mai multor interfețe. În cazul aplicațiilor web, acest lucru permite prezentarea unor vizualizări diferite, cum ar fi pagini web pentru oameni și interfețe de servicii web pentru aplicații la distanță.

Bibliografie

1. A Wedge of Django – Daniel and Adrey Foldroy. p.20
2. Django (web framework) [citat 20.08.2021]. Disponibil: [https://en.wikipedia.org/wiki/Django_\(web_framework\)](https://en.wikipedia.org/wiki/Django_(web_framework)).
3. Документация Django [citat 20.08.2021]. Disponibil: <https://djangodoc.ru/3.2/>.
4. Web application, From Wikipedia, the free encyclopedia [citat 20.08.2021]. Disponibil: https://en.wikipedia.org/wiki/Web_application.
5. Django 3.0. Практика создания веб-сайтов на Python –Владимир Дронов, БХВ-Петербург, 2021.704 стр 17-18.
6. Django Knowledge Base [citat 20.08.2021]. Disponibil: <https://django.fun/>.
7. Пошаговая установка PyCharm Base [citat 20.08.2021]. Disponibil: <https://pythonru.com/baza-znaniy/poshagovaja-ustanovka-pycharm>.
8. Django for beginners - William S. Vincent. p. 1-5.
9. Cadrul web - Web framework [citat 20.08.2021]. Disponibil: https://ro.abcdef.wiki/wiki/Web_framework.
10. Install Ubuntu 20.04 LTS server on VMware [citat 20.08.2021]. Disponibil: <https://blog.eldernode.com/install-ubuntu-20-04-lts-server-on-vmware/>.
11. Django cms documentation Release 3.9.0 [citat 20.08.2021]. Disponibil: <https://readthedocs.org/projects/django-cms/downloads/pdf/latest/>.
12. What Is OpenLDAP?, Written by Kate Lake on October 12, 2021 [citat 20.10.2021]. Disponibil: <https://jumpcloud.com/blog/what-is-openldap>.
13. Install and Configure OpenLDAP Server on Ubuntu [citat 20.10.2021]. Disponibil: <https://computingforgeeks.com/install-and-configure-openldap-server-ubuntu/>.
14. Tutorial How To Install and Configure OpenLDAP and phpLDAPAdmin on an Ubuntu 14.04 Server [citat 20.10.2021]. Disponibil: <https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-openldap-and-phpldapadmin-on-an-ubuntu-14-04-server>. (Accesat 20.08.2021).
15. Bootstrap (front-end framework) [citat 20.10.2021]. Disponibil: [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)).

16. Django Authentication using LDAP Users [citat 20.10.2021]. Disponibil: <https://linuxamination.blogspot.com/2021/07/django-authentication-using-ldap-users.html>.
17. One-time password [citat 20.10.2021]. Disponibil: https://en.wikipedia.org/wiki/One-time_password.
18. Aplicatie web - Web application [citat 20.10.2021]. Disponibil: https://ro.abcdef.wiki/wiki/Web_application.
19. LDAP (Lightweight Directory Access Protocol) [citat 20.10.2021]. Disponibil: <https://searchmobilecomputing.techtarget.com/definition/LDAP>.
20. Ubuntu Server: A cheat sheet [citat 20.10.2021]. Disponibil: <https://www.techrepublic.com/article/ubuntu-server-the-smart-persons-guide/>.
21. LDAP (Lightweight Directory Access Protocol) (Română) [citat 20.10.2021]. Disponibil: <https://iphone-journal.com/ro/ldap-lightweight-directory-access-protocol-9/>
22. Bootstrap (cadru frontal) [citat 20.10.2021]. Disponibil: <https://ro.naturalparentingcenter.com/750551-bootstrap-front-end-framework-ZXIIJO>.
23. Cross-site scripting [citat 20.10.2021]. Disponibil: <https://portswigger.net/web-security>.
24. Django vs. Flask [citat 20.10.2021]. Disponibil: <https://trio.dev/blog/django-vs-flask>.
25. Difference Between Flask and Django [citat 20.10.2021]. Disponibil: <https://www.guru99.com/flask-vs-django.html>.
26. Django-CMS overview - Jorge Quiterio. p. 180-181