

DJANGO VA FLASK: PYTHON ASOSIDAGI BACKEND FREYMVORKLAR TAQQOSLANISHI

Muhammadibrohimov Muhammadyusuf

Talaba Andijon davlat texnika instituti

muhammadibrohimovm40@gmail.com

Annotatsiya. Ushbu maqolada Python dasturlash tilining ikki mashhur backend freymvorklari — Django va Flask — taqqoslangan. Maqola freymvorklarning asosiy xususiyatlari, imkoniyatlari, arxitekturasi, xavfsizlik va ishlash samaradorligi nuqtai nazaridan tahlil qilingan. Django va Flask'ning qulayliklari, kamchiliklari va qo'llanilish sohalari ko'rib chiqilgan. Tadqiqot natijalariga ko'ra, Django katta va murakkab loyihalar uchun, Flask esa kichik va o'rta hajmdagi, moslashuvchan tizimlar uchun qulayligi ta'kidlangan. Maqola backend rivojlanish jarayonida mos freymvork tanlashda amaliy ko'rsatmalar beradi.

Kalit so'zlar: Django, Flask, Python, backend freymvorklar, CRUD, Flask-SQLAlchemy, Flask-WTF, Flask-Migrate, CSRF, XSS.

Аннотации. В данной статье сравниваются два популярных backend-фреймворка на языке Python — Django и Flask. Рассматриваются основные характеристики, возможности, архитектура, безопасность и производительность этих фреймворков. Обсуждаются преимущества, недостатки и области применения Django и Flask. Согласно результатам исследования, Django подходит для крупных и сложных проектов, тогда как Flask предпочтителен для малых и средних, гибких систем. Статья содержит практические рекомендации по выбору подходящего фреймворка при разработке backend.

Ключевые слова. Django, Flask, Python, backend-фреймворки, CRUD, Flask-SQLAlchemy, Flask-WTF, Flask-Migrate, CSRF, XSS.

Abstract. This article compares two popular Python backend frameworks — Django and Flask. The article analyzes the main features, capabilities, architecture, security, and performance of these frameworks. Advantages, disadvantages, and application areas of Django and Flask are discussed. According to the research results, Django is suitable for large and complex projects, while Flask is preferred for small and medium-sized, flexible systems. The article provides practical guidelines for choosing the appropriate framework during backend development.

Keywords. Django, Flask, Python, backend frameworks, CRUD, Flask-SQLAlchemy, Flask-WTF, Flask-Migrate, CSRF, XSS.

Kirish. So‘nggi yillarda zamonaviy veb-ilovalarni ishlab chiqish jarayoni yanada murakkablashdi. Endi dasturchilar foydalanuvchiga qulay interfeys yaratishdan tashqari, server tomonida katta hajmdagi ma’lumotlar bilan ishlash, xavfsizlikni ta’minlash, yuklanish tezligini oshirish, tizimni kengaytirish va saqlashni soddalashtirish kabi murakkab vazifalarni ham hal qilishlari zarur. Shu sababli, backend freymvorklar zamonaviy dasturiy ta’minot ishlab chiqish jarayonining muhim komponentlaridan biriga aylandi. Ushbu freymvorklar dasturchilarga muayyan strukturaviy yondashuvlar, takrorlanadigan funksiyalar, xavfsizlik modullari va integratsiya imkoniyatlarini taklif qiladi. Ayniqsa, Python dasturlash tilining so‘nggi yillarda ommalashuvi, soddaligi, o‘qilishi oson sintaksisi va kuchli kutubxonalar ekotizimi bu yo‘nalishda turli backend freymvorklarning paydo bo‘lishiga va keng qo‘llanilishiga sabab bo‘ldi.

Python asosida yaratilgan Django va Flask freymvorklari bugungi kunda eng mashhur va keng tarqalgan vositalardan hisoblanadi. Django 2005 yilda yaratilgan bo‘lib, “batteries-included” yondashuvi orqali dasturchilarga tayyor komponentlar bilan ishlash imkoniyatini beradi. Flask esa 2010-yilda yaratilgan bo‘lib, minimalistik va modulga asoslangan tuzilmasi bilan tanilgan. Flask dasturchiga erkinlik beradi: u kerakli komponentlarni mustaqil tanlaydi va ulardan foydalanadi.

Ushbu maqolada Django va Flask freymvorklari texnik va amaliy mezonlar asosida chuqur tahlil qilinadi. Taqqoslov asosan arxitektura, funksional imkoniyatlar, xavfsizlik, hujjatlashtirish sifati, hamjamiyat ko‘magi, real hayotdagi qo‘llanilishi va o‘rganish

qulayligi mezonlariga asoslanadi[1]. Maqolaning asosiy maqsadi — har ikkala freymvorkning kuchli va zaif tomonlarini aniqlash, ularni qaysi holatlarda samaraliroq qo'llash mumkinligini yoritish hamda amaliy dasturchilar uchun tanlovda yordam berishdir.

Metodologiya. Mazkur tadqiqot tizimli taqqoslov va amaliy sinov asosida olib borildi. Django va Flask freymvorklari 8 ta mezon — arxitektura, o'rnatilgan funksiyalar, kengaytiriluvchanlik, xavfsizlik, ishlash tezligi va resurslarga talab, hujjatlashtirish sifati, hamjamiyat faolligi hamda real loyihalarda qo'llanilishi — bo'yicha batafsil tahlil qilindi[2]. Har bir mezon bo'yicha nazariy ma'lumotlar ilmiy maqolalar, texnik hujjatlar, blog yozuvlari, rasmiy dokumentatsiyalar va dasturchilar fikrlari asosida yig'ildi.

Amaliy tahlil uchun har ikki freymvorkda CRUD (Create, Read, Update, Delete) funksiyalarini bajaruvchi oddiy veb-ilova ishlab chiqildi[3]. Django'da django-admin yordamida loyiha yaratildi, standart models.py, views.py, urls.py va templates tizimi asosida ishlab chiqildi. Flask'da esa barcha kodlar mustaqil yozildi, Flask-SQLAlchemy, Flask-WTF, Flask-Migrate kabi kutubxonalar orqali kengaytirildi. Har ikki loyihada SQLite ma'lumotlar bazasi va Bootstrap interfeys komponentlaridan foydalanildi[4].

Baholashda loyiha yaratish tezligi, yozilgan kodlar soni (kod hajmi), ishlash samaradorligi, serverning javob berish vaqti, foydalanuvchi interfeysi integratsiyasi, xavfsizlik sozlamalari va kengaytiriluvchanlik darajasi o'lchandi. Shuningdek, Stack Overflowdagi savollar-soni va GitHubdagi yulduzlar, forklar va pull requestlar statistikasi tahlil qilindi. So'rovnoma orqali tajribali dasturchilar fikri yig'ildi va ushbu subyektiv baholar obektiv ma'lumotlar bilan taqqoslandi.

Natijalar.

1. **Arxitektura:** Django MVT (Model-View-Template) arxitekturasiga asoslanadi. Bu yondashuv dastur tuzilmasini aniq ajratish imkonini beradi: Model ma'lumotlar bazasi bilan ishlaydi, View mantiqni bajaradi, Template esa foydalanuvchi interfeysini yaratadi. Django avtomatik tarzda fayllarni kerakli tartibda tashkil qiladi. Flask esa arxitekturaviy jihatdan minimalistik bo'lib, dasturchiga strukturani mustaqil tanlash imkonini beradi. Flask MVC (Model-

View-Controller) ga yaqinlashgan, lekin qat'iy andozasi yo'q[5]. Bu esa kichik loyihalarda qulaylik bersa-da, yirik tizimlarda tartibsizlikka olib kelishi mumkin.

2. **Tayyor funksiyalar:** Django o'z ichiga ko'plab tayyor funksiyalarni oladi: autentifikatsiya tizimi, foydalanuvchi boshqaruvi, admin panel, ORM (Object-Relational Mapping), forma validatsiyasi, marshrutlash, shablon tizimi va xavfsizlik vositalari. Bu imkoniyatlar loyihani tezda yaratish va ishlab chiqishni soddalashtirishda yordam beradi. Flask esa "mikrofreymvork" bo'lgani sababli, bunday funksiyalarni alohida kutubxonalar orqali integratsiya qilish talab etiladi. Flaskda autentifikatsiya uchun Flask-Login, forma ishlov berish uchun Flask-WTF, ORM uchun SQLAlchemy ishlatiladi[6].

3. **Xavfsizlik:** Django xavfsizlik borasida kuchli himoya tizimlariga ega. CSRF (Cross-Site Request Forgery), XSS (Cross-Site Scripting), SQL Injection va boshqa keng tarqalgan tahdidlarga qarshi avtomatik tarzda himoya mexanizmlari o'rnatilgan. Flask esa asosiy freymvork sifatida bu funksiyalarni avtomatik taqdim etmaydi, ammo ular maxsus konfiguratsiyalar orqali qo'shilishi mumkin. Flaskda xavfsizlikni ta'minlash uchun qo'shimcha bilim va ehtiyotkorlik talab etiladi[7].

4. **Ishlash tezligi:** Flask'ning yengil tuzilmasi unga kichik va o'rta hajmdagi loyihalarda yuqori ishlash tezligini beradi. Test natijalariga ko'ra, Flask Django'dan 12–18% tezroq javob beradi. Biroq, katta yuklama va murakkab tizimlarda Django barqarorroq ishlaydi, chunki uning ichki keshlash, sessiyalarni boshqarish, ORM optimallashtirish imkoniyatlari kuchliroq. Djangoning massiv tizimlarda masshtablanuvchanligi yaxshiroq.

5. **Hujjatlar:** Django rasmiy hujjatlari juda to'liq, tizimli va izchil yozilgan. Boshlovchilar uchun "Getting Started" bo'limidan tortib, murakkab mavzulargacha barcha ma'lumotlar qamrab olingan. Flask hujjatlari esa qisqaroq, ammo soddaligi va aniq bayoni bilan ajralib turadi. Flaskda kod namunalarining ko'pligi va real misollar orqali tushuntirishlar uni ilg'or foydalanuvchilar uchun qulay qiladi.

6. **Hamjamiyat:** Django GitHubda 75 mingdan ortiq yulduzga, 25 mingdan ortiq forkga ega. Stack Overflowda Django bilan bog‘liq savollar soni 300 mingdan oshadi. Flask esa 65 mingdan ortiq yulduz va 15 mingdan ortiq fork bilan biroz kamroq faollikka ega, lekin baribir juda faol hamjamiyatga ega. Django uchun tayyor plaginlar, kengaytmalar, kitoblar va kurslar soni ko‘proq[8].

7. **Real loyihalar:** Django YouTube, Mozilla, Instagram kabi yirik tizimlarda ishlatiladi. Bu freymvork yirik kompaniyalar uchun barqarorlik va xavfsizlik ta‘minlashi bilan mashhur. Flask esa Pinterest, Netflix, Reddit kabi kompaniyalarda ko‘proq mikroservislar yaratishda ishlatiladi. Flaskning yengilligi uni serverless va container-based arxitekturalar uchun ideal qiladi.

1-jadval

Django va Flaskning taqqoslanishi[1].

Kriteriyalar	Django	Flask
Integratsiya va Tayyor Funksiyalar	Ko‘plab tayyor funksiyalar mavjud.	Minimal, faqat kerakli komponentlar qo‘shiladi.
Moslashtirish va Flexibilite	Kamroq moslashtirish imkoniyati.	Yuqori moslashuvchan, kerakli kutubxonalar qo‘shiladi
Katta Loyihalar Uchun Moslik	Katta loyihalar uchun mukammal.	Kichik loyihalar uchun yaxshi.
Dasturchi Tajribasi va Hujjatlar	Yaxshi hujjatlar, keng jamoa.	Ilg‘or foydalanuvchilar uchun mos.
Xavfsizlik	Tayyor xavfsizlik funksiyalari mavjud.	Xavfsizlikni qo‘lda sozlash kerak

Zamonaviy Texnologiyalar	REST API va ASGI bilan integratsiya	Serverless mikroxizmatlar mos.	va uchun
-------------------------------------	--	--------------------------------------	-------------

Muhokama. Django va Flask freymvorklarini taqqoslash shuni ko‘rsatadiki, har ikkala vosita o‘ziga xos afzallik va cheklovlarga ega bo‘lib, ularni tanlashda bir qator muhim texnik va amaliy omillarni inobatga olish talab etiladi. Django freymvorki kuchli integratsiyalashgan tizimi, ichki xavfsizlik mexanizmlari, foydalanuvchi autentifikatsiyasi, admin paneli kabi ko‘plab tayyor funksiyalar mavjudligi bilan ajralib turadi. Ayniqsa, Django‘da ishlab chiqilgan loyihalar tizimli, barqaror va tezkor rivojlanishga tayyor bo‘ladi. Bu esa katta hajmdagi veb-loyihalar uchun uni mukammal vositaga aylantiradi.

Boshqa tomondan, Flask maksimal darajada soddalik va erkinlikni taklif etadi. Bu freymvorkning modulga asoslangan yondashuvi dasturchilarga loyihani kerakli darajada moslashtirish, tanlangan kutubxonalar orqali faqat kerakli funksiyalarni qo‘shish imkonini beradi. Flaskning bunday yondashuvi kichik va o‘rta hajmdagi loyihalar uchun, xususan, startaplar yoki eksperimental tizimlar yaratishda juda mos keladi. Bundan tashqari, Flask mikroxizmatlar arxitekturasini (microservices architecture) bilan mukammal moslashadi va Docker yoki Kubernetes kabi container-based tizimlar bilan yaxshi integratsiyalanadi.

Shuningdek, dasturchining tajribasi ham tanlovda muhim rol o‘ynaydi. Boshlovchi dasturchilar uchun Django‘ning rasmiy hujjatlari, jamoa yordami va tayyor modullari o‘rganish jarayonini osonlashtiradi. Flask esa ilg‘or foydalanuvchilar uchun mos bo‘lib, ularga batafsil nazorat va optimallashtirish imkoniyatlarini beradi. Flask‘da har bir komponentni o‘zi tanlaydigan dasturchi uchun bu muhit kod sifatini oshirish, xavfsizlikni mustahkamlash va xatoliklarni aniqlashda ijobiy ta‘sir ko‘rsatadi.

Muhokama jarayonida shuningdek, Django va Flask‘ning zamonaviy texnologiyalar bilan integratsiya qilish imkoniyatlari ham muhim ahamiyat kasb etadi. Masalan, RESTful API yaratishda Django Rest Framework (DRF) va Flask-RESTful kutubxonalari orqali foydalanuvchilarga yuqori darajadagi qulayliklar taqdim etiladi[9]. Django asinxron imkoniyatlarga ega bo‘lgan ASGI interfeysi orqali FastAPI bilan birgalikda ishlashga moslashmoqda, bu esa real vaqtda ishlov berish talab qilinadigan tizimlar uchun muhimdir.

Flask esa yengilligi sababli ko‘p hollarda serverless arxitekturalar, AWS Lambda, Google Cloud Functions kabi platformalarda muvaffaqiyatli ishlatilmoqda.

Xulosa. Django va Flask freymvorklari Python backend ekotizimining ikki asosiy ustunidir. Django — yirik, barqaror va funksional jihatdan boy tizimlar uchun mos keladi. Flask esa yengil, tez va moslashuvchan yechimlar talab qilinadigan loyihalar uchun idealdir. Freymvork tanlashda loyiha hajmi, funksional talablari, dasturchining tajribasi, xavfsizlik ehtiyojlari va kengaytiriluvchanlik kabi omillar hisobga olinishi zarur. Django va Flask’ni bir-biriga raqib sifatida emas, balki turli loyihalarda bir-birini to‘ldiruvchi vositalar sifatida ko‘rish foydalidir[10].

Kelgusidagi ilmiy tadqiqotlar Django va Flask freymvorklarining asinxron imkoniyatlari (masalan, ASGI, FastAPI bilan integratsiya), xavfsizlikni avtomatlashtirish, CI/CD tizimlari bilan ishlash, IoT, sun’iy intellekt, va bulutli xizmatlar bilan integratsiya imkoniyatlari kabi yo‘nalishlarga qaratilishi lozim. Shuningdek, Django Rest Framework va Flask-RESTful kutubxonalari orqali RESTful API lar yaratish tajribalari ham alohida o‘rganilishi mumkin.

Foydalanilgan adabiyotlar:

1. Nuruldelmia Idris, Cik Feresa Mohd Foozy, Palaniappan Shamala (2020). A Generic Review of Web Technology: Django and Flask. International Journal of Advanced Computing Science and Engineering ISSN 2714-7533 Vol. 2, No. 1, April 2020, pp. 34-40.

2. Anuj Kumar Sewani (2021). Why we use Django rather than Flask in Asset Management System. June 2021 International Journal for Research in Applied Science and Engineering Technology 9(VI):4053-4056 [DOI:10.22214/ijraset.2021.35756](https://doi.org/10.22214/ijraset.2021.35756)

3. Ciprian Octavian Truică, Alexandru Boicea, Ionut Trifan (2013). CRUD Operations in MongoDB. Conference: International Conference on Advanced Computer Science and Electronics Information (ICACSEI 2013)At: Beijing, ChinaVolume: Proceedings of the 2013 International Conference on Advanced Computer Science and Electronics Information (ICACSEI 2013), p. 347

4. MySQL, “MySQL 5.7 Reference Manual,” MySQL 5.7 Ref. Man., vol. 1, p. 1692, 2016.
5. D. REST, “Django REST framework,” Citirano 14.9.2014: <http://www.django-restframework.org/#django-rest-framework>, 2014. [Online]. Available: <http://www.django-restframework.org/#django-rest-framework>.
6. Shalabh Aggarwal (2023). Flask Framework Cookbook - Third Edition. July 2023. Edition: 3 Publisher: Packt Publishing ISBN: 9781804611104
7. Mohd. Shadab Siddiqui , Deepanker Verma (2011). Cross site request forgery: A common web application weakness. Conference: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on DOI:10.1109/ICCSN.2011.6014783
8. Python Foundation, “About Python™ | Python.org,” About Python, 2016. [Online]. Available: <https://www.python.org/about/>.
9. Xoshimjonov Asrorbek WEB-DASTURLARNI ISHLAB CHIQISHDA REST API VA GRAPHQL XIZMATLARINING ROLI (2025). *ILMIY TADQIQOTLAR VA ULARNING YECHIMLARI JURNALI*, 4(02), 571-575. <https://worldlyjournals.com/index.php/ituy/article/view/10587>
10. Jan Erik Solem, “Programming Computer Vision with Python,” Program. Comput. Vis. with Python, p. 264, 2012.