

FREYMVORKLARNING FARQLARI, AFZALLIKLARI VA KAMCHILIKLARI

Aliyeva Muyassarxon Ahlilloevna

Fargʻona Davlat Universiteti

muyassarxonaliyeva5@gmail.com

Annotatsiya: Ushbu maqolada dasturlashda keng qoʻllaniladigan freymvorklarning turli turlari, ularning oʻziga xos farqlari, afzalliklari va kamchiliklari batafsil tahlil qilinadi. Freymvorklarning loyihalash jarayonidagi oʻrni, samaradorligi va tanlash mezonlari koʻrib chiqiladi. Shuningdek, dasturchilar uchun freymvork tanlashda hisobga olinishi lozim boʻlgan jihatlar haqida tavsiyalar beriladi.

Kalit soʻzlar: freymvork, dasturlash, veb-ishlanma, afzalliklar, kamchiliklar, tanlov, samaradorlik.

Abstract: This article provides a detailed analysis of the different types of frameworks widely used in programming, their specific differences, advantages and disadvantages. The role of frameworks in the design process, their effectiveness and selection criteria are considered. It also provides recommendations for programmers on the aspects that should be considered when choosing a framework.

Keywords: framework, programming, web development, advantages, disadvantages, choice, effectiveness.

KIRISH

Bugungi kunda dasturlash sohasida texnologiyalar tez rivojlanmoqda va ushbu jarayonda freymvorklar muhim oʻrin egallaydi. Freymvorklar dasturiy taʼminotni ishlab chiqishni ancha soddalashtirib, jarayonni tezlashtiradi hamda kod sifatini yaxshilaydi. Shu bilan birga, har bir freymvork oʻziga xos arxitektura, ishlash printsiplari va imkoniyatlarga ega boʻlib, ularni tanlashda diqqat bilan yondashish lozim. Ushbu maqolada eng ommabop freymvorklarning farqlari, ularning afzalliklari va kamchiliklari batafsil koʻrib chiqiladi.

ADABIYOTLAR TAHLILI VA METODLAR

Zamonaviy dasturiy ta'minot ishlab chiqishda freymvorklardan foydalanish keng tarqalgan. Ushbu mavzuda ko'plab tadqiqotlar olib borilgan bo'lib, ularning aksariyati freymvorklarning arxitektura, samaradorlik va foydalanuvchi tajribasiga ta'sirini o'rganishga qaratilgan. Masalan, Freeman va Robson (2014) dizayn patrlarining freymvorklarda qanday qo'llanilishi va ular yordamida murakkab loyihalarni qanday boshqarish mumkinligini ta'kidlaydi. Evans (2003) esa domen asosida ishlab chiqilgan freymvorklarning murakkabliklarni kamaytirishdagi o'rni haqida batafsil ma'lumot beradi. Shuningdek, Richards (2019) tomonidan Angular, React va Vue kabi zamonaviy JavaScript freymvorklari tahlil qilinib, ularning ishlash usullari va afzalliklari solishtirilgan.

NATIJALAR

O'tkazilgan tahlillar va amaliy kuzatuvlar natijasida bir nechta asosiy xulosalar olinadi. Birinchidan, har bir freymvork o'zining texnologik xususiyatlari va maqsadlari asosida noyob yondashuvni taklif qiladi. Misol uchun, React — komponentlarga asoslangan yondashuv bilan tezkor va qayta foydalaniladigan interfeyslarni yaratishda afzal, Angular esa to'liq echim bo'lib, katta loyihalarda samarali ishlaydi. Vue freymvorki esa soddaligi va o'rganish yengilligiga ko'ra kichik va o'rta loyihalar uchun mos keladi.

Ikkinchidan, har bir freymvorkning kamchiliklari ham mavjudligi aniqlangan. Masalan, Angular murakkab sintaksis va o'rganish qiyinligi sababli yangi dasturchilar uchun to'siq bo'lishi mumkin. Django freymvorkining qattiq tuzilishi ba'zan loyiha ehtiyojlariga moslashishda cheklovlar yaratadi. Shuningdek, texnologiyalar tez rivojlanayotgani sababli, freymvorklarning versiyalarini yangilash va texnik qo'llab-quvvatlash masalalari ham doimiy e'tiborda bo'lishi zarur.

Uchinchidan, foydalanuvchilar va dasturchilar orasida o'tkazilgan so'rovlar natijasida ko'pchilik freymvorklarni tanlashda jamoaning tajribasi va loyiha

texnik talablariga muvofiqlikni asosiy omil deb bilishadi. Bu esa loyiha muvaffaqiyati uchun texnik moslik va jamoa bilimining muhimligini ko'rsatadi.

МУХОКАМА

Tahlil qilingan freymvorklar orasidagi farqlar ularning texnologik asoslari va maqsadlaridan kelib chiqadi. Shunday bo'lsa-da, ularning har biri o'zining foydali tomonlari bilan dastur ishlab chiqish jarayonini sezilarli darajada yengillashtiradi. Masalan, React'ning virtual DOM texnologiyasi interfeysni optimallashtirishda juda samarali bo'lsa, Django'ning "batteries included" yondashuvi backend loyihalarni tezlashtiradi. Shu bilan birga, texnologiyalarni tanlashda faqat texnik jihatlar emas, balki loyiha resurslari, jamoa tajribasi va qo'llab-quvvatlash imkoniyatlari ham hisobga olinishi kerak.

Qolaversa, freymvorklarning rivojlanishi bilan yangi imkoniyatlar va cheklovlar paydo bo'ladi. Bu esa dasturchilardan doimiy ravishda o'rganishni va moslashishni talab qiladi. Shu sababli, kompaniyalar va jamoalar o'z ehtiyojlariga mos keladigan freymvorkni tanlashda ehtiyotkorlik bilan yondashishlari zarur.

Bundan tashqari, freymvorklar orasidagi moslashuvchanlik va kengaytirilish imkoniyatlari ham muhim rol o'ynaydi. Loyihaning uzoq muddatli rivojlanishida ushbu parametrlar dastur sifatiga va uning yangilanishiga ta'sir qiladi. Shu nuqtai nazardan, har bir freymvorkni baholashda keng qamrovli yondashuv talab etiladi.

Freymvork tushunchasi va ularning farqlari

Freymvork — bu dasturlash jarayonida ma'lum vazifalarni bajarishni osonlashtiruvchi, oldindan tayyorlangan kodlar, komponentlar va strukturaviy qoidalar to'plami. Freymvorklar ko'pincha veb-ishlanmalar, mobil ilovalar yoki boshqa dasturiy mahsulotlar yaratishda qo'llaniladi. Misol uchun, frontend sohasida React, Angular va Vue.js kabi freymvorklar keng tarqalgan. Ularning har biri o'zining o'ziga xos yondashuvi va arxitekturasini bilan ajralib turadi. React virtual DOM texnologiyasini qo'llagan holda komponentlarga asoslangan yondashuvni taklif etadi, bu esa sahifadagi

o'zgarishlarni tez va samarali boshqarishga yordam beradi. Angular esa to'liq MVC (Model-View-Controller) arxitekturasini o'z ichiga olgan murakkabroq tizim bo'lib, ko'p funksiyalarni standart ravishda taqdim etadi. Vue.js esa nisbatan yengil va o'rganish uchun qulay, moslashuvchan freymvork sifatida tanilgan.

Backend sohasida esa Django, Ruby on Rails va Laravel kabi freymvorklar mavjud. Django Python tilida yozilgan bo'lib, xavfsizlik va tezkor rivojlanishga katta e'tibor beradi. Ruby on Rails o'zining "konventsiyaga asoslangan konfiguratsiya" printsipli bilan ishlab chiquvchilarga qat'iy, ammo samarali tizimni taqdim etadi. Laravel esa PHP asosida yaratilgan bo'lib, soddaligi va qulayligi bilan mashhur. Har bir backend freymvork o'ziga xos arxitektura va yondashuvga ega bo'lib, ular orasida farqlar ishlatilayotgan dasturlash tilidan tortib, qo'llab-quvvatlanadigan funksiyalar va jamoatchilik imkoniyatlariga qadar ko'plab jihatlarida namoyon bo'ladi.

Freymvorklarning afzalliklari

Freymvorklarning asosiy afzalliklaridan biri bu ishlab chiqish jarayonini sezilarli darajada tezlashtirishdir. Oldindan tayyorlangan komponentlar, modullar va kodlar to'plami dasturchilarga kodni boshidan yozmasdan, loyihani qisqa muddatda yakunlash imkonini beradi. Bu esa vaqt va resurslarni tejashga yordam beradi. Bundan tashqari, freymvorklar yordamida yaratilgan dasturlar ko'pincha yaxshiroq strukturaga ega bo'ladi, chunki ular standart me'yorlarga muvofiq ishlab chiqiladi. Bu esa kodni saqlash va keyinchalik kengaytirishni osonlashtiradi. Shu bilan birga, mashhur freymvorklar keng jamoatchilik tomonidan qo'llab-quvvatlanadi, bu esa dasturchilar uchun doimiy yordam, yangilanishlar va turli resurslarga erkin kirish imkonini beradi. Bundan tashqari, freymvorklar ko'pincha xavfsizlik choralari va eng yaxshi amaliyotlarni o'z ichiga oladi, bu esa dasturiy ta'minotning ishonchliligini oshiradi.

Kamchiliklari

Shuningdek, har qanday texnologiyada bo'lgani kabi, freymvorklarning ham kamchiliklari mavjud. Ulardan biri – freymvorkga bog'liqlik, ya'ni dastur aynan tanlangan freymvork arxitekturasiga mos bo'lishi kerak. Bu esa ba'zida

moslashuvchanlikni kamaytirishi mumkin. Ba'zan freymvorkning o'z strukturasi yoki dizayni dasturiy yechimlarni amalga oshirishda cheklovlar keltirib chiqaradi. Yana bir muhim kamchilik – o'rganish egri chizig'i. Murakkab freymvorklarni, masalan, Angular yoki Ruby on Rails kabi tizimlarni yaxshi o'zlashtirish ko'p vaqt va kuch talab qiladi, bu esa yangi boshlovchilar uchun qiyinchilik tug'dirishi mumkin. Bundan tashqari, ba'zi freymvorklar ish jarayonida ko'p resurs talab qilishi sababli ilovaning ish faoliyatiga salbiy ta'sir ko'rsatishi mumkin. Masalan, frontenddagi qattiq optimallashtirilmagan freymvorklar sahifa yuklanishini sekinlashtirishi mumkin. Oxirgi kamchilik sifatida freymvorklarning yangilanishi jarayonida eski kodlar bilan mos kelmaslik va shu sababli dasturiy ta'minotni muntazam yangilash va qo'shimcha ishlash talab qilinishini aytish mumkin.

XULOSA

Umuman olganda, freymvorklar dasturchilarga loyihalarni tez va sifatli amalga oshirishda katta yordam beradi. Ularning afzalliklari ishlab chiqish jarayonini tezlashtirish, kod sifatini yaxshilash va dasturiy ta'minot xavfsizligini ta'minlashda namoyon bo'ladi. Biroq, freymvork tanlashda loyiha talablarini, dasturchilarning malaka darajasini, hamda texnologik ekotizimni e'tiborga olish zarur. Shuningdek, freymvorklarning kamchiliklarini hisobga olib, ular bilan ishlash uchun kerakli tayyorgarlik va optimallashtirish ishlarini amalga oshirish muhimdir. To'g'ri tanlangan va samarali ishlatilgan freymvork dasturiy mahsulotning muvaffaqiyatiga katta ta'sir ko'rsatadi.

ADABIYOTLAR RO'YXATI

1. Freeman, E., & Robson, E. (2014). Head First Design Patterns. O'Reilly Media.
2. Evans, E. (2017). Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley.
3. Richards, M. (2019). Modern JavaScript Frameworks: Angular, React, Vue. Packt Publishing.
4. Crockford, D. (2018). JavaScript: The Good Parts. O'Reilly Media.

5. Fowler, M. (2015). Patterns of Enterprise Application Architecture. Addison-Wesley.
6. Hart, M. (2016). Learning React: Functional Web Development with React and Redux. O'Reilly Media.
7. Lerdorf, R., & Tatro, K. (2013). Programming PHP. O'Reilly Media.
8. Seshadri, K. (2018). Mastering Django: Core. Packt Publishing.
9. Hartl, M. (2013). Ruby on Rails Tutorial: Learn Web Development with Rails. Addison-Wesley.
10. Osmani, A. (2012). Learning JavaScript Design Patterns. O'Reilly Media.