

**WEB SAYT DASTURLASH VA UNING ZAMONAVIY
TEXNOLOGIYALARI**

Nodira Jo'rayeva

Buxoro Davlat Texnika Universiteti

ANNOTATSIYA: *Ushbu maqolada web dasturlashning asosiy tamoyillari, zamonaviy texnologiyalari va rivojlanish tendensiyalari haqida ma'lumot beriladi. Web dasturlashning old va orqa qismi (frontend va backend) texnologiyalari, ularning o'ziga xos jihatlari va qo'llanilish sohalari tahlil qilinadi. Shuningdek, web saytlarni ishlab chiqishda xavfsizlik, foydalanuvchi tajribasi (UX/UI) va optimizatsiya masalalari ham yoritiladi.*

KALIT SO'ZLAR: *Web dasturlash, HTML, CSS, JavaScript, Backend, Frontend, Full-Stack, UX/UI, Framework, Web xavfsizligi.*

KIRISH

Zamonaviy dunyoda internet texnologiyalari hayotimizning ajralmas qismiga aylandi. Bugungi kunda web saytlar nafaqat axborot yetkazish vositasi, balki biznesni rivojlantirish, xizmatlar ko'rsatish va kommunikatsiya uchun ham muhim ahamiyatga ega.

Web dasturlash – bu internetda ishlaydigan sayt va web-illovalarni yaratish jarayoni bo'lib, u frontend (foydalanuvchi interfeysi) va backend (server qismi) texnologiyalaridan iborat. Frontend qismi foydalanuvchiga ko'rinadigan interfeys va dizaynni o'z ichiga oladi, backend esa saytning ma'lumotlar bazasi, server bilan ishlash va autentifikatsiya jarayonlarini boshqaradi.

Ushbu maqolada web dasturlashning asosiy texnologiyalari, zamonaviy dasturlash tillari va frameworklari, shuningdek, web sayt yaratishda e'tiborga olinishi lozim bo'lgan jihatlar haqida so'z yuritiladi.

ASOSIY QISM**1. Web dasturlashning asosiy tamoyillari**

Web sayt yaratish jarayoni quyidagi bosqichlarni o'z ichiga oladi:



Dasturlash tili tanlash – frontend uchun HTML, CSS, JavaScript, backend uchun PHP, Python, Node.js kabi tillar.

Dizayn va foydalanuvchi interfeysi (UI/UX) – saytning vizual ko‘rinishi va qulayligi.

Ma’lumotlar bazasi bilan ishlash – MySQL, PostgreSQL, MongoDB kabi tizimlar yordamida ma’lumotlarni saqlash va boshqarish.

Server konfiguratsiyasi va xavfsizlik – sayt tezligi va xavfsizligini ta’minlash.

Test va optimizatsiya – saytning ishlashini sinovdan o‘tkazish va tezlikni oshirish.

Web dasturlashda yuqoridagi tamoyillarga amal qilish saytni samarali va foydalanuvchilar uchun qulay bo‘lishini ta’minlaydi.

2. Frontend (Old qism) texnologiyalari

Frontend – bu foydalanuvchilar ko‘radigan va ular bilan muloqot qiladigan sayt qismidir. U quyidagi texnologiyalarga asoslanadi:

HTML (HyperText Markup Language) – saytning asosiy tuzilishini yaratish uchun ishlatiladi.

CSS (Cascading Style Sheets) – dizayn va uslubni belgilash uchun qo‘llaniladi.

JavaScript – interaktiv elementlar qo‘shish va foydalanuvchi bilan muloqotni ta’minlash uchun ishlatiladi.

Zamonaviy frontend dasturlashda quyidagi frameworklar keng qo‘llaniladi:

React.js – Facebook tomonidan ishlab chiqilgan va foydalanuvchi interfeyslarini yaratish uchun ishlatiladi.

Vue.js – yengil va tezkor framework bo‘lib, dinamik web ilovalar yaratishda qo‘llaniladi.

Angular – Google tomonidan ishlab chiqilgan va murakkab web ilovalar yaratish uchun mo‘ljallangan.

Frontend qismning samarali ishlashi saytning vizual jozibadorligi va foydalanuvchi tajribasiga (UX) bog‘liq.



3. Backend (Orqa qism) texnologiyalari

Backend – bu saytning server qismi bo‘lib, foydalanuvchilarning so‘rovlarini qayta ishlaydi, ma’lumotlarni saqlaydi va boshqaradi. Asosiy backend dasturlash tillari:

PHP – eng mashhur server tili bo‘lib, WordPress kabi tizimlarda ishlatiladi.

Python (Django, Flask) – katta hajmdagi ma’lumotlarni qayta ishlash va sun’iy intellekt bilan bog‘liq web ilovalar yaratish uchun mos keladi.

Node.js – JavaScript asosida ishlaydigan server platformasi bo‘lib, real vaqt rejimidagi ilovalar yaratishda ishlatiladi.

Ruby on Rails – tezkor web ilovalar yaratish uchun ishlatiladigan framework.

Backend qismi, shuningdek, ma’lumotlar bazalari bilan ishlash uchun quyidagi texnologiyalarni qo‘llaydi:

MySQL – eng ommabop relatsion ma’lumotlar bazasi.

PostgreSQL – kuchli va kengaytirish mumkin bo‘lgan ma’lumotlar bazasi.

MongoDB – NoSQL ma’lumotlar bazasi bo‘lib, dinamik ma’lumotlarni saqlash uchun ishlatiladi. Backend qismi sayt xavfsizligi va foydalanuvchi ma’lumotlarini himoya qilish uchun mas’uldir.

4. Web sayt xavfsizligi va optimizatsiyasi

Web sayt yaratishda xavfsizlik va tezlik muhim omillardan hisoblanadi. Quyidagi xavfsizlik choralarini ko‘rish tavsiya etiladi:

HTTPS protokoli – saytning ma’lumotlarini shifrlash orqali himoya qilish.

SQL Injection hujumlariga qarshi himoya – foydalanuvchi kiritgan ma’lumotlarni tekshirish orqali oldini olish.

XSS (Cross-Site Scripting) hujumlariga qarshi choralar – zararli skriptlarni oldini olish.

Tezlikni optimallashtirish uchun quyidagilar qo‘llaniladi:

Sayt kodini minimallashtirish – ortiqcha kodlardan qochish.

Cache tizimlaridan foydalanish – ma’lumotlarni vaqtinchalik saqlash orqali tezlikni oshirish. CDN (Content Delivery Network) – kontentni geografik jihatdan foydalanuvchiga yaqin serverlardan yuklash.



XULOSA

Web dasturlash hozirgi zamonning eng talabgir sohalaridan biri bo'lib, biznes va texnologiya rivojlanishida muhim rol o'ynaydi. Frontend va backend texnologiyalarining o'zaro uyg'unlashuvi natijasida sifatli va funksional web saytlar yaratiladi.

Web dasturlashda dizayn, xavfsizlik va tezlik kabi omillarga e'tibor qaratish muhimdir. Kelajakda sun'iy intellekt, blokcheyn va 5G texnologiyalarining rivojlanishi web dasturlash sohasiga yangi imkoniyatlar olib kelishi kutilmoqda.

Ushbu maqolada web dasturlashning asosiy tamoyillari, zamonaviy texnologiyalari va muhim jihatlari haqida batafsil ma'lumot berildi. Bu yo'nalishda o'z bilimlarini oshirish istagida bo'lganlar uchun doimiy o'rganish va amaliy tajriba orttirish tavsiya etiladi.

FOYDALANILGAN ADABIYOTLAR

1. Duckett, J. (2011). HTML & CSS: Design and Build Websites. Wiley.
2. Freeman, E., & Robson, E. (2018). Head First JavaScript Programming. O'Reilly Media.
3. Flanagan, D. (2020). JavaScript: The Definitive Guide. O'Reilly Media.
4. McDonald, M. (2017). Web Security for Developers. No Starch Press.
5. Resig, J. (2019). Pro JavaScript Techniques. Apress.