Veb səhifələrin işləmə prinsiplərinə baxsaq ilk əvvəl bizə məlum olur ki, veb səhifənin tam şəkildə bizə görünür hala gəlməsi işlətdiyimiz brauzerlərin gördüyü işdir. Brauzerlər veb səhifələri oxumaq üçün xüsusi proqramlardır, internetə sorğu göndərərək axtardığımız veb səhifəni bizə tapıb çıxarır. Bəs veb səhifə özü fiziki olaraq harda yerləşir? Veb səhifələr onları saxlamaq üçün xüsusi yerlər olan serverlərdə yerləşir və server bütün mməlumatları bütöv halda özündə cəmləyir və yalnız HTTp və HTTPS sorğularına cavab verə bilir. Serverlər özləri də bir kompyuter olduğu üçün onlarında oxuyub başa düşəcəyi proqramlama dilləri var.

Hər şeydən əvvəl bilmək lazımdır ki, istifadə etdiyimiz brauzerlər yalnız HTML, CSS, JavaScript dillərini oxuya bilir. Əgər belədirsə, serverlərdə istifadə olunan fərqli dilləri necə görə bilirik? Bu proses üçün interpetatorlar işə giri və onlar server üçün yazılan proqramlama dilindəki məlumat və ya proqramı brauzerin oxuya biləcəyi dilə çevirir və brauzer bu məlumatları toplayaraq saytı tam halda bizə sunur.

Serverlər bizə nə üçün lazımdır? Yazdığımız kodu tam məhsul şəklində görmək üçün bizə mütləq server lazımdır. Bu server bizim kompyuterimiz də ola bilər, internetdən online olaraq aldığımız hər hansısa bir yerdə yerləşən bir kompyuter də. Əgər belədirsə bəs niyə veb səhifə qurarkən öz kompyuterimizi server olaraq istifadə etmirik? Çünki, serverlərdə qurulan sistem ola biləcək hər bir kənar müdaxilə və bədbəxt hadisələrə (yanğın, həddən artıq isinmə, elektrik kəsilməsi və s.) qarşı sığortalanıb. Buna görə də dünyanın istənilən yerindəki istifadəçi asanlıqla və istədyi vaxt qurduğumuz veb səhifəyə access edə bilir.

Deyilənlərdən belə nəticə çıxarmaq olar ki, veb səhifə ərsəyə gətirərkən iki tip proqramlama var, 1. Server side və 2. Client side

Server side (back end)- dilində yazılan proqramlamadakı olan məlumatlar istifadəçiyə görünmür. Server side dillərinə Ruby (on rails), Javascript (NodeJS), PHP, Python, C # dilləri aid olsa da bunlarla limitlənmir. Və bu dillərdə yazılan hər bir kod yalnız HTTP və HTTPS sorğularına cavab verir. Server side sayta lazım olan bütün məlumatları özündə saxlayır (şəkillər, videolar, səs yazmaları, HTML, CSS, JS kodları, istifadəçi login və parolları və s.). İstifadəçilər tərəfindən görünmür. Yalnız URL uzantıları ilə girilən adressləri sorğularına cavab verir, istifadəçinin özündən atdığı hər hansı mətnə yox. İstifadəçinin brauzerində görünən veb səhifənin son versiyasını işləyib sorğuya cavab verir. Server-side proqramlama dillərinin çatışmayan cəhəti onların hər bir sorğunu yenidən işləyib istifadəçiyə göndərmə məəcburiyyəti olmasıdır. Əgər saytda məlumatlar həddən artıq çoxdursa və ya istifadə etdiyimiz video vəya şəkil və ya hər hansısa datanın həcmi böyükdürsə bu zaman saytın açılması və requestə cavab verməsində gecikmələr ola bilər.

Client side (front end) – proqramlama- bu hissə isə tamamilə istifadəçi brauzerində görünən proqramlama hissəsidir. Burada istifadə olunan dillər JavaScript, HTML və CSS -dir. Bunun səbəbi isə brauzerlərin yalnız bu dillərdə yazılanları oxuna və işlənə bilməsidir. Istifadəçiyə cavab verə bilmə xüsusiyyəti client side proqramlamadan ayıran əsas xüsusiyyətdir. Server side-dan fərqli olaraq client side proqramlama heç bir məlumatı toplama və ona cavab vermə xüsusiyyətinə malik deyil, lakin ola bilər ki, JavaScript daxil edilən məlumatdan istifadə edib yenidən ona cavab verə bilər lakin, bu da sırf proqramistin əvvəlcədən qərarlaşdırdığı yol üzrə gedir. Serverdən heç bir faylı birbaşa oxuya bilmir və yalnız HTTP və HTTPS sorğularına cavba verir. Client side proqramlamanın bir digər üstün cəhəti yazdığımız kodları brauzeri sitifadə edərək ən son halına baxa bilərik, lakin server -side-da belə bir lüksümüz yoxdur bunun üçün ya onıline serverimiz ya da kompyuterdə qurulu local serverimiz olmalıdır. Client-side istifadəçiyə görünən saytın son halıdır və əsas yük onların üzərinə düşür. Belə ki əgər düzgün tasarlanıbsa o zaman istifadəçinin bütün sorğuların cavab verə bilər məsələn istənilən veb səhifə həm PC-lərdə həm də smartfonlarda açıla bilər. bu zaman saytın hər cihaza uyğun ölçü alması onun responsiveliyindən irəli gəlir və bu client-side programmingin işidir.