Un architettura web si classifica in una architettura a n livelli, ove alcune sono fissi o standard, es internet o application server, altri invece presentano scelte.

A livello più basso vi sono le risorse dati (come base dati o servizio di terze parti es. meteo), sopra vi è l’application server (ovvero il server che contiene l’applicazione, ovvero, per esempio le funzionalità che la piattaforme offre piuttosto che le funzionalità per parlare con il database), sopra vi è il web server (è un server che sa come parlare su internet), sopra le infrastrutture internet (i cavi, wifi, la rete in generale, usata per ricevere e trasmettere i dati, alla cima della piramide vi sono browser e altri client (es terminale, tablet, telefono). Noi collasseremo tutto assieme, ma in genere le cose sono ben separate. Si sfrutta il protocollo http che è un tipo di protocollo richiesta-risposta, è un protocollo per cui il browser fa una richiesta a cui arriva una risposta (non può arrivare la risposta senza la richiesta).

Browser: capiscono poche risorse e pochi linguaggi, tutti linkati da un html che viene ricevuto dal browser, queste informazioni sono ricevute dal server elaborate (se non disponibile) e viene rimandata al client.

Il browser legge HTML (detta il contenuto e la struttura della pagina), CSS (decide colori, fonti, layout della pagina, etc.) come linguaggi di markup, che javascript come linguaggio di programmazione (che da la parte dinamica, la logica, l’interattività e l’animazione del sito).

Html ha un linguaggio semantico associato: usa una serie di tag che corrispondono a quello che si sta creando, es header=intestazione, quindi contiene le informazioni di intestazioni, ma non viene definito come e dove visualizzarlo, ad esempio se in basso o in alto, se grande o piccolo

Se manca un file css, viene visualizzato lo stile di default definito dal browser (ognuno ha uno stile simile ma diverso). Clicclando su una pagina web vengono generati nuovi eventi es click, scroll, che javascript può interpretare per poi sfruttarli per dare dinamismo.

Dom = document object model, permette di recuperare la struttura dati del documento della pagina html, javascript lavora su questo file per dare la dinamicità. Potrebbero esserci informazioni quali di cosa parla la pagina web, che linguaggio usa ecc (nel root element), header h1= primo header che c’è ed è il più importante. <a> = ancora = serve per andare da una pagina a un’altra o altre parti della stessa pagina.

CSS: indica lo stile di una pagina e come appare, permette di adattarsi a dimensione schermo diverse (es telefono, pc), risoluzioni differenti, schemi diversi es daltonici, media differenti. Differenza risoluzione diversa vs dispositivo diverso: quando si guarda da smartphone lo schermo gira da orizzontale a verticale e viceversa, quindi il dispositivo deve adattare il layout in base all’applicazione. I media diversi servono per esempio per avere una modalità facilitata per la stampa del sito

Come si scrive un foglio css:: H1 {color:blue; font-size:12px;}

In questo l’h1 deve avere colore blue e dimensione del font di 12 pixel

Differenza tra query e url è il fatto che la query è preceduta da ?

La risposta alla richiesta di http può essere un file vuoto, un errore, una pagina statica presa dal server o una pagina dinamica creata al momento dal server, http non frega del contenuto della pagina

Server logico : la funzione detta prima

Serve fisico: la posizione fisica del server

Il server web da le risorse in rispsota alla richiesta, il web server manda una richiesta che viene aperta e chiusa una volta conclusa, ha il compito anche di servire più richieste multiple allo stesso momento.

ARCHITETTURA TRADIZIONALE (nostro approccio)

Si usa l’approccio tradizionale: il server manda la risposta dopo la richiesta, rendendo talvolta la pagina web dinamica (server side rendering), creando una applicazione web multi pagina, per ogni richiesta si crea una nuova pagina

ARCHITETTURA MODERNA

1. Single page application (SPA): Il server manda la stessa pagina per ogni url unico, il lavoro è scaricato dal server al client della singola persona (es Instagram ha sempre la stessa pagina, ma con diverse informazioni). Come vantaggio si ha che è potenzialmente più veloce, riceve solo il testo, non l’intera pagina di nuovo e come svantaggi si ha che c’è più carico sul client e meno sul server, la cronologia non funziona (non posso usare la freccetta indietro), si hanno problemi di sicurezza
2. Isomorphic application: mix tra approccio tradizionale e approccio spa (più usato)
3. Progressive web app (PWA): applicazioni web che emulano le applicazioni native mobile, es generare notifiche