**Tehnologii de baza:**

**->HTML :** -descrie formatul primar in care documentele sunt distribuite si vizualizate pe web, fiind interpretat de browsere; - reprezinta un limbaj de marcare; -standard official HTML: W3C;

**->CSS:** defineste stulurile si modurile de afisare ale elem. HTML; -permite stilizarea in loc a elementelor;

**->XML**: -conceput sa incapsuleze date, nu sa le afiseze; -repr. Un limbaj complementar cu HTML;

**Tehnologii client-slide:**

**->JavaScript:** -conceput sa adauge interactivitate pag HTML; -repr un limbaj interpretat, se executa fara a fi compliat; Permite: inserarea de text dynamic, reactia la diferite evenimente; scrierea/citirea continutului elem. HTML, validarea datelor, crearea si gestionarea cookie-urilor;

**->Flash:** -permite crearea si dezvoltarea de elemente multimedia; -ActionScript;

**Tehnologii server-side:**

**->PHP:** -cod executat pe serverul web; -paginile sunt returnate clientului ca cod HTML; - suporta multi-platforma; -software open-source;

**->ASP.NET :** -solutia Microsoft pt crearea de pagini, aplicatii si servicii web

**-> Java EE:** -standard de facto folosit pt dezvoltarea de aplicatii server-side scrise in Java;

**-> Java Services**: -se executa pe server;- proceseaza cererile HTTP dinamice;

-**>JSP**: -similar cu PHP si Java; -ermite dezvoltarea de pag. Dinamice ce contin HTML, XML sau Java

**MySQL-executia unei interogari:**

1)Browserul clientului inițiază o conexiune http către serverul Web

2)InterpretorulPHP pre-proceseazăcodulscrisîn PHP

3)InterogareaPHPdin codulPHPinterpretat este trimisăcătreserverulMySQL

4)Serverul MySQL returneazărezultatulinterogăriiSQL

5)CodulHTML rezultatîn urmaprocesăriiîntreguluiprogram PHPeste returnat serverului Web

6)DocumentulHTML este returnat browser-ului.

**GET vs. POST**

**->GET:** -idempotenta; -solicita o resursa speficata; -toate datele sunt vizibile in URL; -parametrii sunt salvati de catre browser; -metoda mai putin sigura decat POST;

**->POST:** -nu este idemponenta; -trimite date spre procesare catre resursa specifica; -datele nu sunt vizibile in URL; -parametrii nu sunt salvati de catre browser; -meth mai sigura decat GET;

**Width vs max-width / height vs max-height**

-> proprietățilewidthși heightsetează, în cazul unor valori explicite,lățimea și înălțimea unui element la o dimensiune fixă

-> dacă fereastra browser-ului este mai mică / minimizată sub acea dimensiune, browser-ul adaugă opțiunea de scrollingîn pagină, însă nu redimensioneazăelementele

-> max-heightși max-widthpermit acest lucru, browser-ul urmând a redimensiona containerele automat

**Tag-ul <form>… </form>**

->permite interactiunea utilizatorului ->server web;

->etape: 1) utilizatorul completeaza formularul si il expediaza serverului; 2)serverul analizeaza si parseaza formularul complet(daca e necesar poate stoca datele intr-o BD); 3) serverul raspunde utilizatorului(optional);

Atribute: 1)action -> precizeaza ce se intampla cu datele formularului, odata ajunse pe server; 2) method -> folosit de browser pt expedierea datelor formularului;

**HTML 5- noi elemente grafice**

**<canvas>**

->permite introducerea de elemente grafice in codul unei pag web; -reprezinta containerul pt acest tip de obj; realizarea efectiva a graficului se face cu JavaScript;- depenent de rezolutia ecranului; -fara support pt gestionarea evenimentelor; - capabilitati reduse pt gestionarea textului; -potrivit pt jocuri cu grafica sofisticata;

**<svg>**

->Scrable vector graphics; -limbaj ce permite descrierea de elem grafice 2D; -independent de rezolutia ecranului; -confera support pt gestionarea evenimentelor; -incet in cazul redarilor complexe; -nerecomandat in cadrul jocurilor;

**CSS-Cascading Style Sheets**

-limbaj de design; -defineste modul de afisare al elementelor HTML; -permite aplicarea de formatari unei pagini intregi sau chiar unui site intreg, in bloc; -elementele de stil CSS se pot aplica: unui element individual din pagina, asupra unui grup de elemente, sau la nivelul intregului document;

**Avantaje ale limbajului CSS:**

1)Organizare mult mai eficientă a codului

•Permite definirea de proprietăți comune pentru elemente din pagini HTML diferite

•Același fișier cu proprietăți CSS poate fi inclus / referențiat în oricâte fișiere HTML

2)Viteză de încărcare a paginilor sporită

•Elimină necesitatea scrierii atributelor HTML pentru fiecare element HTML în parte

•O regulă CSS definită la nivelul unui document se aplică pentru toate aparițiileelementului vizat

3)Gestionare mai facilăa codului

•Efectuarea unei modificări de stil asupra tuturor elementelor de același tip presupune doar modificarea fișierului CSS

4)Elemente de stil superioare limbajului HTML

•CSS suportă o varietate de proprietăți de stil mult mai maredecât cea de la nivelul atributelor HTML

•Astfel, permite o stilizare mai avansată a paginilor

5)Portabilitate sporită

•Foile de stil CSS permit optimizarea conținutului pentru vizualizarea mai multe dispozitive

•Același document HTML poate fi afișat diferit, în funcție de dispozitiv

6)Compatibilitatea sporită ⇨respectarea standardelor Web globale

•Din ce în ce mai multe elemente HTML sunt deprecated(depășite)

•Utilizarea limbajului CSS asigură compatibilitatea

Tipuri de selectori CSS:

1. Selectori pt elemente(tag-uri) HTML: permit setarea de proprietati pt elementele html specifice; sunt reprezentate prin numele efectiv al tag-ului elem vizate;
2. Selectorul universal: permite setarea de proprietati pt toate elem HTML din domeniul de vizibilitate
3. Selectorul descendent: permite setarea de proprietati pt elem HTML imbricate; se poate merge pe oricate niveluri;
4. Selectorul pt clase: permite setarea apartenentei acestora la diverse clase definite de utilizator;
5. Selectorul pt ID: permite setarea de identificatori pt acestea, de catre utilizator, poate fi utilizat in combinatie cu cel pt tag-uri HTML
6. Selectorul bazat pe atributele elem HTML: permite o filtrare suplimentara fata de selectorul pt elem HTML, prin posibilitatea de a impune si valori pt atributele acestora;

**Avantajele utilizarii Java Script:**

Java Script este un limbaj conceput pt a adauga interactivitate paginilor HTML; este case-sensitive;

1. **Reducerea interactiunii cu serverul web:** J.S. permite prevalidarea datelor introduse de utilizatori, inaintea trimiterii la server; reduce semnificativ traficul si incarcarea serverului;
2. **Interactivitatea sporita**: crearea de interfete ce permit reactia la diferite evenimente; reactia la evenimente de tip hover, focus, vizitare de link-uri, etc;
3. **Raspuns mai rapid la input-ul utilizatorului:** JavaScript poate semnala, inca de la nivelul browser-ului, lipsa sau incorectitudinea anumitor actiuni din partea utilizatorului; elimina necesitatea asteptarii unui raspuns din partea serverului, pt fiecare actiune in parte
4. **Realizarea de interfete mai avansate**: permite integrarea de diverse componente in pagina; controlul programatic asupra acestora;

**Arhitecturi web- elemente de baza:**

1. **Cerinte de functionare:** -adaptare la necesitatile clientilor, categorii de vizitatori, scalabilitate data de potentialul nr de vizitatori, raportare la concurenta, in pas cu evolutia tehnologica actuala, raportare la contextual social cotidian;
2. **Factori calitativi:** -disponibilitate(utilizabilitatea site-ului); - performanta(viteza de raspuns adaptata necesitatilor clientilor); -aspect(design modern); -securitate(politici de securitate)
3. **Aspecte tehnologice:** -tehnologiile utilizate(software); -limbajele de programare folosite; -platforma hardware utilizata; -sisteme de operare; -servicii puse la dispozitie;- backward-compatibility;
4. **Experienta anterioara:** -management de proiecte; -integrarea/utilizarea de arhitecturi si platform existente; -design patterns; folosirea de biblioteci; -utilizarea de framework-uri; -Feedback;

**API vs SDK vs IDE:**

**->API:** -reprezinta o interfata; -Permite programelor software să interacționeze între ele;- Specifică rutinele și structurile de date ce trebuie definite pt comunicatia dintre doua aplicatii;(ex: JavaAPI, Google Maps API);

**->SDK**: -Reprezintă un set de instrumente; -Permite dezvoltarea de aplicații software pentru o anumită platformă;- Include utilitare, librăriiși documentațiinecesare unui programator pentru scrierea unei aplicații; ( JavaSDK );

**->IDE:** -Reprezintă un mediu de dezvoltare integrat;- Aplicații software ce pun la dispoziție facilități pentru dezvoltarea de alte aplicațiipentru programatori; - Poate include atât API-uri, cât și SDK-uri (eclipse; Android Studio; Visual Studio);

**Kernel-mode vs User-mode:**

-Un server Web poate fi încorporat în kernel-ul sistemului de operare sau poate rula în spațiul utilizator(la fel ca orice aplicație uzuală);

-User-mode: necesită permisiuni din partea sistemului pentru a utiliza mai multă memorie sau mai multe resurse procesor; Exemple: Kernel-mode Web Servers: Microsoft IIS; User-mode Web Servers: Apache, Nginx etc.

**PHP- Design Patterns:**

1. Factory Design Pattern:-Util în creșterea flexibilității claselor utilizate; -Uzual, modificarea unei metode dintr-o clasă poate avea efectul de cascadă la nivelul aplicației;-Permite crearea de obiecte particularizate, folosind clasa Factory; - Practic, definește interfețe specifice peste obiectele uzuale;
2. Singleton Design Pattern:- Un obiect de tip singletoneste un obiect ce poate fi instanțiat o singură dată de aplicație, la un moment de timp;-Alternativă viabilă la utilizarea variabilelor globale;-Exemplu: conexiunea la baza de date, ce se dorește a fi partajată la nivelul unei resurse (pagini) din aplicație
3. MVC (Model-View-Controller) Pattern; - Model–partea de BackEnd; - View –partea de GUI (Graphical User Interface); - Control–adaptor(componentă ce mediază interacțiunea dintre utilizator și aplicație, precum si intre Model si View;

**GET vs POST (php)**

Ambele metode creează un Array de perechi de forma “nume” => “valoare”;- “nume” ⇨numele controalelor formularului HTML; -“valoare” ⇨valorileintrodusede utilizator;- Accesibile prin intermediul variabilelor super globale $\_GET și $\_POST, ce pot fi accesate din orice funcție, clasă sau fișier;

**$\_GET** ⇨ un Array de variabile trimise prin intermediul parametrilor din cadrul URL-ului

•Informațiile folosind metoda GET sunt vizibile în clar, la nivelul URL-ului

•Limitarela aproximativ 2000 de caractere

•GET se recomandă exclusivpentru trimiterea de date fără caracter sensibil

•NU ar trebui niciodată folosit pentru trimiterea de parole;

**$\_POST** ⇨ un Array de variabile trimise prin intermediul metodeiHTTP POST

•Informațiile folosind metoda POST nu sunt vizibile în clar⇨ suntîncapsulateîn cadrulcererii HTTP

•Nicio limitare a dimensiunii datelor de transmis

•POST se recomandă pentru trimiterea de date cu caracter sensibil

•Suport multi-part pentru upload de fișiere;

**Elemente vs. Atribute (XML):**

* Nu există o restricție la nivelul tehnologiei XML privind utilizarea elementelor sau atributelor
* Totuși, ar trebui luate în calcul următoarele aspecte pentru realizarea structurii unui document XML:

 atributele nu pot conține valori multiple

 atributele nu pot conține structuri

arborescente

 atributele nu pot fi extinse ușor

**Componentele platformei de servicii Web:**

1)SOAP(Simple Object Access Protocol)⇨ protocol bazat pe tehnologia XML, ce permite schimbul de informații între aplicații, folosind HTTP

2)UDDI(Universal Description, Discovery and Integration)⇨ repository pentru înregistrarea și căutarea de servicii Web

3)WSDL(Web Services Description Language)⇨ limbaj bazat pe tehnologia XML pentru descrierea serviciilor Web;