|  |  |  |
| --- | --- | --- |
|  | **UNIVERSITATEA „VASILE ALECSANDRI” DIN BACĂU**  **Facultatea de Științe**  **Str. Calea Mărăşeşti, nr. 157, Bacău, 600115**  **Tel. ++40-234-542411, tel./ fax ++40-234-571012** [www.ub.ro](http://www.ub.ro/); e-mail: [stiinte@ub.ro](mailto:stiinte@ub.ro) |  |

**PROGRAMUL DE STUDII INFORMATICĂ**

**LUCRARE DE LICENŢĂ**

**Coordonator ştiinţific:**

Prof./Conf./Lector univ. dr. **Cerasela Crișan Absolvent: Conea Sorin - Ionuț**

**Bacău**

**2020**

|  |  |  |
| --- | --- | --- |
|  | **UNIVERSITATEA „VASILE ALECSANDRI” DIN BACĂU**  **Facultatea de Științe**  **Str. Calea Mărăşeşti, nr. 157, Bacău, 600115**  **Tel. ++40-234-542411, tel./ fax ++40-234-571012** [www.ub.ro](http://www.ub.ro/); e-mail: [stiinte@ub.ro](mailto:stiinte@ub.ro) |  |

**PROGRAMUL DE STUDII INFORMATICĂ**

**SISTEM DE MANAGEMENT PENTRU SPITALE**

**Coordonator ştiinţific:**

Prof./Conf./Lector univ. dr. **Cerasela Crișan Absolvent:**

**Conea Sorin - Ionuț**

**Bacău   
2020**

**Introducere**

Progresele din industria medicală nu se limitează doar la soluțiile medicale de calitate, ci și la schimbarea sistemului de comunicare cu pacienții și a experienței pacientului în ansamblu. Toate industriile au înțeles potențialul înnăscut al automatizării, iar rezultatele sunt destul de optimiste. Ca sector care se străduiește să creeze o lume sănătoasă, digitalizarea modului de operare este o necesitate, la fel și renunțarea la hârtiile scrise, întrucât totul poate fi făcut on-line. Este primul pas care trebuie făcut pentru o lume mai verde pentru generațiile ce vor urma, dar este și cel mai rapid mod de a livra informații utile despre pacienți.

Ca sector care se străduiește să creeze o lume sănătoasă, digitalizarea sau sistemul de operații fără hârtie în industria de sănătate este o necesitate decât o nevoie. Este primul pas care trebuie făcut pentru a dezvălui o lume mai verde pentru generația viitoare.

Un sistem de management este modul în care o organizație gestionează părțile interrelaționate ale activității sale pentru a-și atinge obiectivele. Aceste obiective pot face referire la o serie de subiecte diferite, inclusiv calitatea produsului sau serviciului, eficiența operațională, performanța de mediu, sănătatea și siguranța la locul de muncă și multe altele.

**Aplicațiile web** sunt niste programe care folosesc o arhitectura de tip client-server și tehnologiile World Wide Web.

Omniprezența și accesul foarte rapid la browserele web rezolvă această problemă. Astfel au avoluat serverele ce găzduiesc aplicații, iar pe lângă limbajele de programare folosite în dezvoltarea aplicațiilor web au apărut si framework-uri si tehnologii web dedicate ce-și au ca scop, dezvoltarea mai rapidă de aplicații web. Web-ul a apărut cu scopul distribuirii de pagini web , în cadrul său a apărut formatul HTML urmat de protocolul HTTP.

În anul 1990 avut loc prima dezvoltare ce avea ca scop folosirea unui server web și a dus la aparitia CGI (Common Gateway Interface), aceasta permitea rularea de script-uri pe server ce au ca rezultat diferite răspunsuri dinamice în format HTML.

În 1995 Netscape crează limbajul de programare JavaScript care scriptează comportamentul browserelor web si introduce elemente dinamice, intre timp functionalitatea limbajului JavaScript a fost exstinsa prin introducerea de API-uri, cel mai important fiind XmlHttpRequest, prin care se poate programa procesarea și efectuarea de răspunsuri cu timp mic de execuție, fără a mai reîncărca întreaga pagină.   
Pe partea de server, Common Gateway Interface a urmat apariția Java Servlet ce a fost inclusă in Java Enterprise Edition, fapt ce a transformat Java intr-un limbaj de programare dominant în rândul aplicațiilor web pe partea de server.   
Anii 2010 au adus o extindere a standardului HTML4, cu HTML5, ce impune browserelor implementarea internă a unor functionalități sau funcții care până atunci au fost dezvoltate de terți prin plugin-uri.

**HTML**

Dezvoltat în 1989 la CERN de Tim Berners-Lee, HTML (HyperText Markup Language) a constituit un mare avantaj pentru fizicientii ce folosesau computere diferite și schimbau

informații între ei utilizănd conexiunea la internet. HTML constă dintr-o serie de coduri

scurte introduse într-un fișier text de către autorul site-ului - acestea sunt etichetele. Textul

este apoi salvat ca fișier html și vizualizat printr-un browser, cum ar fi Internet Explorer

sau Netscape Navigator. Acest browser citește fișierul și traduce textul într-o formă

vizibilă, urmând să redea pagina așa cum intenționase autorul.Scrierea propriului dvs. cod

HTML presupune utilizarea corectă a etichetelor pentru a vă crea pagina dorită.

Puteți utiliza orice, de la un editor de text rudimentar la un editor grafic puternic pentru a crea pagini HTML. HTML livrează soluții prin care conținutul unei pagini poate fi adnotat cu diferite tipuri de metadate și indicații de redare.

Indicațiile de redare variază de la forma textului până la script-uri compexe. Datele de tip

medatada sau metadatele pot include informații despre titlul paginii, informații despre cum

este strucurata pagina în diferite segmente, paragrafe, etc. dar și informații cruciale care să

permită paginii web să se lege de alte pagini pentru a create hiperlink uri.

Codul HTML se poate genera utilizând tehnologii de codare de tip server-side cum ar fi

PHP, JSP sau ASP. Multe aplicații precum forumuri, sisteme de gestiune a conținutului

(CMS) generează astfel de pagini HTML. Odată cu apariția HTML5 se încearcă definirea

unui singur limbaj de marcare ce se poate scrie folosind atât sintaxa HTML

cât și XHTML. Acesta include modele detaliate de prelucrare pentru a sustine cât mai multe

implementări, pune la dispozitia programatorilor aplicații de tip API pentru aplicații

complexe. Din aceste motive, limbajul HTML5 este un candidat pentru aplicațiile mobile

întrucât o parte dintre acestea pot fi înlocuite cu un site web aferent.  
  
Exemplu de pagina HTML:

<!DOCTYPE html>  
<html>  
 <head>  
 <title> O pagina HTML </title>

</head>  
<body>  
 <h1>Pagina HTML </h1>  
</body>  
</html>

**CSS**

Cascading Stle Sheets (CSS) este un limbaj de design simplu menit să simplifice procesul de prezentare a paginilor web.

Conceptul său a fost creat de Håkon Wium Lie în 1994. În decembrie 1996, limbajul CSS a făcut o specificat de către W3C (World Wide Web Consortium) și astăzi permite dezvoltatorilor web să modifice aspectul paginilor lor web. De exemplu, CSS poate fi folosit pentru a schimba fontul utilizat în anumite elemente HTML, precum și dimensiunea și culoarea acestuia.

Acesta poate fi utilizat pentru a defini stiluri de text, dimensiuni de tabel și alte aspecte ale paginilor Web care anterior au putut fi definite doar în HTML-ul unei pagini. CSS ajută dezvoltatorii Web să creeze un aspect uniform pe mai multe pagini ale unui site Web. În loc să definească stilul fiecărui tabel și al fiecărui bloc de text din HTML-ul unei pagini, stilurile utilizate frecvent trebuie definite o singură dată într-un document CSS. După ce stilul este definit în foaia de stil în cascadă, poate fi folosit de orice pagină care face referire la fișierul CSS.

În plus, CSS simplifică schimbarea stilurilor în mai multe pagini simultan. De exemplu, un dezvoltator Web poate dori să crească dimensiunea implicită a textului de la 10pt la 12pt pentru cincizeci de pagini ale unui site Web. Dacă paginile se referă la aceeași foaie de stil, dimensiunea textului trebuie modificată doar pe foaia de stil și toate paginile vor afișa textul mai mare. Deși CSS este excelent pentru crearea de stiluri de text, este util și pentru formatarea altor aspecte ale aspectului paginilor Web. CSS oferă dezvoltatorilor web un control mai exact asupra aspectului în care vor arăta paginile Web decât HTML. Acesta este motivul pentru care astăzi majoritatea paginilor Web includ fișiere de tip CSS, pentru a stiliza pagina web.

Codul CSS poate fi scris direct in codul HTML:  
Exemplu:  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {  
  color: RED; font-size:20px;  
}  
</style>  
</head>  
<body>  
<h1>Pagina HTML </h1>  
</body>  
</html>  
  
Codul CSS poate fi pus intr-un fisier extern si apelat in codul html:  
folosind tag-ul <link>

<link rel="stylesheet" href="/calea\_fisierului/fisierul.css">

**JavaScript**

JavaScript (JS) este un limbaj ușor, interpretat, orientat pe obiecte, cu funcții de primă clasă și este cel mai cunoscut ca limbaj de script pentru paginile Web, dar este folosit și în multe medii non-browser. Este un limbaj de script multi-paradigm bazat pe prototipuri, care este dinamic și acceptă stiluri de programare orientate pe obiecte, imperativ și funcțional.

JavaScript rulează pe partea clientului web, care poate fi utilizat pentru a proiecta sau programa modul în care paginile web se comportă la apariția unui eveniment.

Spre deosebire de concepția greșită populară, JavaScript nu este „Java interpretat”. Pe scurt, JavaScript este un limbaj de script dinamic care acceptă construcția de obiecte bazate pe prototipuri. Sintaxa de bază este intenționat facută sa semene cu atât Java cât și C ++ pentru a reduce numărul de concepte noi necesare pentru învățarea limbii. Construcțiile de limbaj, cum ar fi instrucțiunile, **for, while, switch** sunt la fel ca in Java sau C++ .

JavaScript funcționează cu HTML și CSS pentru a crea aplicații web sau pagini web. JavaScript este acceptat de majoritatea browserelor web moderne precum Google Chrome, Firefox, Safari, Microsoft Edge, Opera, etc. Majoritatea browserelor mobile pentru Android și iPhone acceptă și JavaScript.

JavaScript controlează elementele dinamice ale paginilor web. Funcționează și în browserele web și, mai recent, și pe serverele web. Interfețele de programare a aplicațiilor (API) sunt, de asemenea, acceptate de JavaScript, oferindu-vă mai multe funcționalități.

JavaScript poate funcționa atât ca limbaj procedural, cât și ca limbaj de programare orientat pe obiecte. Obiectele sunt create programatic în JavaScript, prin atașarea de metode și proprietăți la obiecte altfel goale în timpul rulării, spre deosebire de definițiile sintactice de clasă comune în limbaje compilate precum C ++ și Java. Odată construit un obiect, acesta poate fi folosit ca model pentru crearea de obiecte similare.  
JavaScript este acum utilizat pe scară largă și în afara browserului. Creșterea Node.js în ultimii ani a crescut dezvoltarea backend, odată cu Java, Ruby, Python și PHP și mai multe limbaje tradiționale de tip server-side.

JavaScript poate folosi acum bazele de date și multe alte aplicații și se pot dezvola chiar aplicații încorporate, aplicații mobile, aplicații pentru televizoare și multe altele. Ceea ce a început ca un limbaj simplu de programare în browser este acum unul dintre cele mai cunoscute și folosite limbaje de programare.

Exemplu:  
<!DOCTYPE html>  
<html>  
<script>

alert(‘ Buna ziua ! ’);

</script>   
<body>  
<h1>Pagina HTML </h1>  
</body>  
</html>

PHP

PHP este un limbaj de scriere Web încorporat în HTML. Acest lucru înseamnă că codul PHP poate fi inserat în HTML-ul unei pagini Web. Când este accesată o pagină PHP, codul PHP este citit sau „analizat” de serverul pe care se află pagina. Output-urile funcțiilor PHP de pe pagină sunt returnate de obicei sub forma codului HTML, care poate fi citit de browser. Deoarece codul PHP este transformat în HTML înainte de încărcarea paginii, utilizatorii nu pot vedea codul PHP pe o pagină. Acest lucru face ca paginile PHP să fie suficient de sigure pentru a accesa bazele de date și alte informații sigure.

Codul PHP este rulat în mod normal în scripturi cu text simplu, care vor fi rulate numai pe computere cu un parser ce recunoaste PHP-ul, în lucrarea de față s-a folosit programul XAMPP. PHP preia cea mai mare parte a sintaxei sale din C, Java și Perl. Este o tehnologie open source și rulează pe majoritatea sistemelor de operare și cu majoritatea serverelor Web.

Limbajul PHP a fost scris în limbajul de programare C de către Rasmus Lerdorf în 1994 pentru a împărtăși CV-ul său online și informațiile personale aferente. Din acest motiv, PHP a fost inițial pentru „Personal Home Page”. Lerdorf a combinat PHP cu propriul său parser, făcând publică combinația ca PHP / FI (denumită în general PHP 2.0) la 8 iunie 1995.

Doi programatori, Zeev Suraski și Andi Gutmans, au reconstruit nucleul PHP, lansând rezultatul actualizat ca PHP. / FI 2 în 1997. În acest moment, acronimul a fost schimbat în PHP: HyperText Preprocessor. (Acesta este un exemplu de acronim recursiv: unde acronimul însuși este în propria definiție.)

În 1998, PHP 3 a fost lansat, care a fost prima versiune la scară largă. PHP 4 a fost lansat în mai 2000, cu un nucleu nou, cunoscut sub numele de Zend Engine 1.0. PHP 4 a prezentat viteză și fiabilitate îmbunătățite față de PHP 3. În ceea ce privește caracteristicile, PHP 4 a adăugat referințe, tipul boolean, suport COM pe Windows, buffering de ieșire, multe funcții noi de matrice, programare orientată pe obiecte, includerea bibliotecii PCRE, și altele. Versiunile de întreținere ale PHP 4 sunt încă disponibile, în principal pentru actualizări de securitate.

<!DOCTYPE html>  
<html>

<title> Exemplu PHP </title>  
<body>  
  
<?php  
echo "Acesta este un exemplu de utilizare al codului pHp în HTML”;  
?>  
</body>  
</html>

**SQL**

Limbajul SQL a fost inițial dezvoltat la laboratorul de cercetare IBM din San José, în cadrul unui proiect care dezvoltă un prototip pentru un sistem relațional de gestionare a bazelor de date numit System R la începutul anilor 70.

Primele sisteme de gestionare a bazelor de date bazate pe SQL au devenit disponibile din punct de vedere comercial până la sfârșitul anilor '70. În același timp, au fost publicate sisteme relaționale de gestionare a bazelor de date bazate pe alte limbaje de programare.

Cea mai cunoscută alternativă pentru SQL a fost limbajul QUEL în sistemul Ingres.

SQL a fost ales pentru standardizare. Atât standardul ANSI american, cât și ISO internațional au luat parte la standardizare.

În 1986, a fost aprobat primul standard SQL (SQL-ul de bază aproximativ 40 de pagini + limbajul modulului și încorporarea în limbaje de programare, în total aproximativ 100 de pagini).

În 1992, noua versiune SQL-92 (numită și SQL2) a fost aprobată. Acesta conținea îmbunătățiri mari ale limbii. Mărimea standardului a fost de aproximativ. 600 de pagini + standardul de dicționar de date; cu totul c. 1000 de pagini. Au fost specificate trei niveluri pentru SQL, nivelul de bază, care conține în principal nucleul vechiului SQL / 89

nivelul intermediar, care conține noi tipuri de date, operații și structuri

SQL complet cu mai multe tipuri și structuri de date

În loc de un standard masiv, s-a luat decizia de a continua dezvoltarea în anumite părți.

În 1995, a fost aprobată interfața la nivel de apel SQL / CLI, adică specificația de interfață pentru utilizare prin programe. Standardul specifică interfața ODBC.

În 1996, SQL / PSM (module stocate persistente) a fost aprobat ca limbaj de specificare a procedurii bazei de date.

În 1999, noua versiune SQL-99 (numită și SQL3) a fost aprobată. Standardul este împărțit în cinci părți.

SQL este un limbaj de interogare structurat. SQL este utilizat pentru a comunica cu o bază de date. Conform ANSI (American National Standards Institute), este limbajul standard pentru sistemele relaționale de gestionare a bazelor de date. Instrucțiunile SQL sunt utilizate pentru a efectua sarcini precum actualizarea datelor dintr-o bază de date sau pentru a prelua date dintr-o bază de date. Unele sisteme comune de gestionare a bazelor de date relaționale care folosesc SQL sunt: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Deși majoritatea sistemelor de baze de date folosesc SQL, majoritatea au și propriile lor extensii proprii care sunt utilizate de obicei numai în propriile sisteme . Cu toate acestea, comenzile standard SQL, cum ar fi "Select", "Insert", "Update", "Delete", "Create", și "Drop" pot fi utilizate pentru a realiza aproape toate operațiile posibile cu o bază de date.

Exemple:

SELECT Numele , Prenumele FROM medici;

INSERT INTO pacienti (CNP, Parola, Email, Nume, Prenume, Varsta, Inaltime, Greutate, Grupa\_sange,Numar\_telefon)

VALUES ('$CNP', '$Parola','$Email', '$Nume', '$Prenume', '$Varsta', '$Inaltime', '$Greutate', '$Grupa\_sange','$Numar\_telefon');

**BootStrap**

Bootstrap este un framework de front-end ce a fost dezvoltat pentru a sprijini crearea de site-uri web și aplicații web dinamice. Este unul dintre cele folosite framework-uri de front-end, deoarece ajută la o procesare ușoară și rapidă pentru a dezvolta un site web. Acceptă toate browserele majore și încărcarea rapidă a paginilor web responsive.  
 Bootstrap este format din șabloane de design bazate pe HTML și CSS pentru diverse componente de interfață și are ca scop ușurarea dezvoltării web. Prin actualizarea CSS-ului, se pot obține rapid design-uri unice si responsive. Dezvoltatorii se concentrează mai mult pe componentele de interacțiune, deoarece framework-ul în sine va avea grijă de afișarea standard a datelor, care pot fi modificate ulterior foarte ușor.

Bootstrap este compatibil cu aproape toate browserele de ultimă versiune, cum ar fi Internet Explorer, Google Chrome, Opera, Firefox și Safari. Suporta designul web receptiv și ajustează dinamic aspectul paginilor web, luând în considerare caracteristicile dispozitivului utilizat. Viteza de dezvoltare este unul dintre avantajele sale majore. Dacă se dorește dezvoltarea unei aplicații web responsive care să respecte standardele moderne, BootStrap vine în ajutorul dezvoltatorilor. Acesta salvează efortul de codare oferind mai puțină funcționalitate CSS și blocuri de cod pre-construite, scutind foarte mult timp întrucât clasele CSS nu vor mai fi create de la zero.

Conform previziunilor CISCO, traficul global de date mobile a crescut de aproximativ 11 ori între 2013 și 2018. Aceste statistici indică necesitatea unui site web responsive în diverse tipuri de dispozitive mobile.

Bootstrap este responsive si are un set de grile cu 12 coloane care ajută la ajustarea dinamică a site-ului web la o rezoluție de ecran adecvată. Funcția „responsive utility classes” a Bootstrap-ului permite ascunderea sau afișarea unei anumite secțiuni de conținut pentru o anumită dimensiune a ecranului.

Framework-ul BootStrap facilitează personalizarea site-urilor web și îi ajută pe dezvoltatori să proiecteze site-uri web adaptate preferințelor lor. Dezvoltatorul are posibilitatea de a selecta doar părtile de cod care îi sunt utile. Cu această caracteristică, se poate scăpa de codul neutilizat pentru o optimiza viteza de încărcare a paginii web.

Bootstrap ajută la remedierea promptă a problemelor cu o comunitateimensă. Bootstrap lansează, de asemenea, actualizări continue pentru a remedia orice probleme noi. În prezent, este dezvoltat, găzduit și întreținut de GitHub cu peste 9000 de angajamente și 500 de contribuabili.

Exemplu:

<div class="card border-left-primary shadow h-100 py-2">

<div class="card-body"><div class="row no-gutters align-items-center">

<div class="col mr-2">

<div class="text-xs font-weight-bold text-primary text-uppercase mb-1">Medici de garda</div>

<div class="h5 mb-0 font-weight-bold text-gray-800">15</div>

</div>

<div class="col-auto">

<i class="fas fa-notes-medical fa-2x text-gray-300"></i></div></div>

**Punerea problemei**

Sistemul de sănătate este una dintre activitățile socio-economice esențiale; prin urmare, necesită un management rațional și eficient. Pentru asta, este necesar să existe un instrument care să permită un control adecvat al informațiilor generate în instituțiile de sănătate.  
 Spitalele, au ca și oameni de bază medici, asistenți, etc, aceștia generează un volum mare de informații esențiale, dar, în cele mai multe cazuri nu este disponibil în timpul și maniera necesară, fapt ce produce pierderi mari de timp și bani.

În ultimii ani, sistemele informatice din domeniul sănătății au contribuit la îmbunătățirea calității vieții oamenilor din toate sectoarele societății noastre, astfel încât este inevitabil să amânăm dezvoltarea de noi soluții software care să vină . În prezent, managementul clinic și administrativ al spitalelor și centrelor de sănătate este posibil printr-o singură platformă, cu sprijinul tehnologiei de vârf, dezvoltat pentru a optimiza procesele care permit funcționarea organizațiilor dedicate tratării pacienților din orice ramură a medicamentului.