**CUPRINS**

**1.** Elemente de dezvoltare a unui site de comerț electronic........................................ 3

1.1. Simplitate..................................................................................................... 3

1.2. Coerență....................................................................................................... 3

1.2.1.Alegerea culorilor în funcţie de publicul ţintă................................... 3

1.2.2. Alegerea culorilor în funcţie de psihologia culorilor........................ 4

1.3.Încredere..................................................................................................... 8

1.3.1. Crearea unor reguli relevante de protecţie a consumatorilor

online........................................................................................................... 8

1.3.2. Acordarea unui grad ridicat de atenţie datelor personale.................. 8

1.3.3. Diversitatea termenelor şi condiţiilor................................................ 8

1.3.4. Creşterea cooperării internaţionale în domeniul comerţului

electronic..................................................................................................... 9

1.3.5. Participarea la discuţiile internaţionale privind comerţul

electronic..................................................................................................... 9

1.4. Transparenţă................................................................................................ 9

1.4.1. Disponibilitatea produsului.............................................................. 10

1.4.2. Stabilirea preţurilor şi a promoțiilor................................................. 10

1.4.3.Taxele de transport şi manipulare...................................................... 10

1.4.4. Introducerea extrem de vizibilă a opțiunilor de plată....................... 11

1.4.5. Descrierea cât mai exactă a produselor............................................ 11

1.5. Caracteristica “mobile friendly” a unui site de comerț electronic.............. 11

1.6. Optimizarea procesului de vânzare............................................................. 13

1.6.1. Pagina de pornire.............................................................................. 13

1.6.2. Pagina de listare produse.................................................................. 14

1.6.3. Pagina de produs.............................................................................. 15

1.6.3.1. Butonul „Adaugă în coș”............................................................... 15

1.6.3.2. Imaginile........................................................................................ 16

1.6.3. Încheierea tranzacţiei........................................................................ 17

1.6.4. Finalizarea......................................................................................... 19

**2.** Tehnologii folosite în crearea site-urilor................................................................. 20

2.1. HTML - (HyperText Markup Language).................................................... 20

2.1.1. Scurtă istorie...................................................................................... 20

2.1.2. Tag-uri HTML................................................................................... 21

2.1.3. Atribute HTML.................................................................................. 24

2.1.4. HTML5.............................................................................................. 26

2.1.5. Editoare HTML.................................................................................. 28

2.2. CSS - (Cascading Style Sheets)................................................................... 30

2.2.1. Inserarea foilor de stil........................................................................ 30

2.2.2. Ordinea de aplicare a foilor de stil..................................................... 32

2.2.3. Selectori............................................................................................. 32

2.3. JavaScript.................................................................................................................... 34

2.3.1. Scurtă istorie...................................................................................... 34

2.3.2. Biblioteci și framework-uri JavaScript.............................................. 39

**3.** Prezentarea site-ului................................................................................................ 46

3.1. Aspecte generale ale proiectării site-ului......................................................... 46

3.1.1. Imagini și pictograme....................................................................... 46

3.1.2. Meniul lateral.................................................................................... 49

3.1.3. Subsolul site-ului.............................................................................. 50

3.2. Paginile de nivel doi de navigare..................................................................... 52

3.2.1. Magazinul ........................................................................................ 52

3.2.2. Pagina de produs.............................................................................. 54

**4.** Concluzii................................................................................................................ 56

**5.** Bibliografie............................................................................................................ 58

**1. Elemente de dezvoltare a unui site de comerţ electronic**

În planificarea arhitecturii unui site de comerț electronic este indicat ca elementul principal în jurul căruia se face dezvoltarea site-ului, să fie chiar utilizatorul. Deoarece acest tip de site propune atât o implicare interactivă a utilizatorului, cât și una materială, în realizarea unui site de comerț electonic, trebuiesc urmate câteva concepte, cu scopul de a atrage atenţia celui ce dorim a se transforma în client.

**1.1. Simplitate**

Interfaţa cu ulilizatorul trebuie să fie una simplă, intuitivă, astfel încât un utlizator nou să nu aibă dileme legate de dispunerea in pagină a secţiunilor site-ului sau a anumitor butoane. Navigarea trebuie să fie uşoară şi fiecare acţiune a utilizatorului să aibe la bază un algoritm care să îi permită acestuia selectarea opţiunii dorite.

**1.2. Coerenţă**

Pentru a atrage şi vinde publicului dorit, design-ul site-ului trebuie să fie în concordanţă cu ceea ce acesta oferă spre vânzare. Toate componentele care alcătuiesc design-ul site-ului, de la culori, imagini şi până la fonturile folosite, trebuie să se potrivească. Pentru a atinge performanţa de a îmbina toate aceste variabile într-o manieră perfectă, este necesar să cunoaştem foarte bine produsul pe care dorim să îl vindem, dar şi publicul ţintă.

**1.2.1. Alegerea culorilor în funcţie de publicul ţintă**

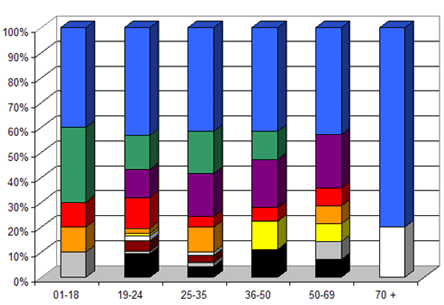
Psihologia culorilor este studiul nuanţelor ca determinant al comportamentului uman.

Culorile pot influenţa percepţii care nu sunt evidente şi au calităţi care pot provoca emoţii la oameni. Astfel, conform studiilor, 85% dintre cumpărători spun ca principalul motiv pentru care cumpără un produs îl reprezintă culoarea, 66% din oameni spun că nu ar cumpăra un produs dacă nu are culoarea lor preferată, iar reclamele colorate sunt remarcate cu 26% mai mult decât cele în negru şi alb[[1]](#footnote-1).

La fel ca în orice strategie de marketing, şi în comerţul electronic, este foarte important să cunoaştem publicul căruia se adresează produsul pe care îl vindem şi astfel vom alege culoarea principala a magazinului pe care dorim să îl dezvoltăm.

Putem aborda mai multe categorii, astfel: pentru produse destinate femeilor vom alege culori ca roz, violet, pentru bărbaţi, negru, albastru, pentru cumpărători impulsivi albastru si portocaliu, iar pentru cei limitaţi de un buget, albastru închis şi verde.

Figura 1. Preferințe ale culorilor în funcție de vârstă



Sursă: <https://www.quicksprout.com/how-colors-affect-conversion-rate/>

După o astfel de analiză, am decis ca albastru să fie culoarea dominantă a site-ului, fiind conform graficului de mai sus culoarea preferată a tuturor grupurilor de vârstă şi una dintre culorile preferate ale bărbaţilor. Cel mai important motiv însă, în alegerea culorii dominante a site-ului meu, îl reprezintă publicul ţintă reprezentat de simpatizanţi ai partidului Uniţi Salvăm România, ai cărui membri au fost cei mai vocali in spaţiul public legat de importanţa participării la vot.

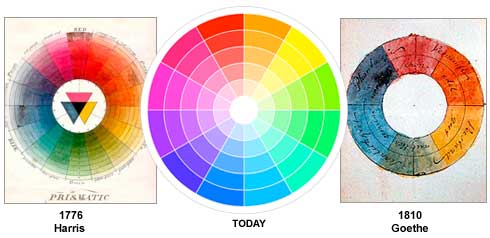
**1.2.2. Alegerea culorilor în funcţie de psihologia culorilor**

Teoria culorilor cuprinde o multitudine de definiţii, concepte şi aplicaţii de proiectare, însă cu toate acestea, există trei categorii de bază ale teoriei culorilor: roata culorilor, armonia culorilor şi contextul modului de utilizare a culorilor.

**1.2.2.1. Roata culorilor**

Sir Isaac Newton a dezvoltat prima diagramă circulară a culorilor în anul 1666[[2]](#footnote-2). De atunci oamenii de ştiinţă şi artiştii au studiat şi proiectat numeroase variante ale acestui concept.

Figura 2. Variante ale roții culorilor



Sursă: <https://www.colormatters.com/color-and-design/basic-color-theory>

Roata culorilor împarte în trei mari categorii culorile, astfel:

**Culori primare:** roşu, galben şi albastru.

În teoria traditională a culorilor, culorile primare sunt cele trei culori de pigment care nu pot fi amestecate sau formate prin combinaţia altor culori. Toate celelalte culori provin din aceste trei nuanţe.

**Culori secundare:** verde, portocaliu şi violet, culori obţinute prin combinarea celor trei culori principale.

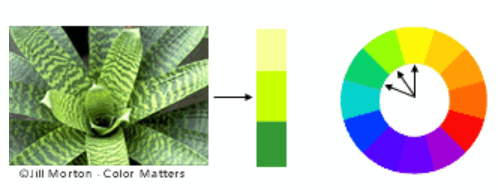
**Culori terţiare:** galben-portocaliu, roşu-portocaliu, roşu-violet, albastru-violet, albastru-verde, galben-verde. Acestea sunt culorile formate prin amestecarea unei culori primare cu una secundare, de aceea nuanţa este un nume cu două cuvinte.

**Armonia culorilor**

Armonia este un echilibru dinamic şi prin definiţie exclude extremele.

Următoarele ilustraţii şi descrieri prezintă câteva formule de bază.

Figura 3. Schema culorilor bazată pe culori analogice



Sursă: <https://www.colormatters.com/color-and-design/basic-color-theory>

Culorile analogice sunt oricare trei culori care sunt poziţionate una lângă pe roata culorilor şi sunt folosite astfel, să creeze o armonie vizuală, contrastul între ele fiind redus.

Figura 4. Schema culorilor bazată pe culori complementare

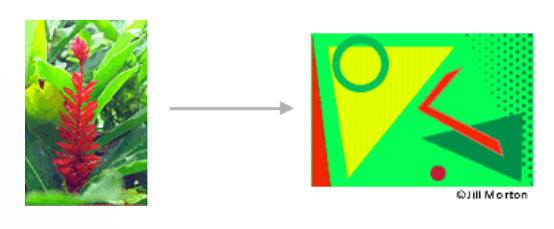


Sursă: <https://www.colormatters.com/color-and-design/basic-color-theory>

Culorile complementare sunt oricare două culori poziţionate diametral opus pe roata culorilor, cum ar fi spre exemplu, roşu şi verde, roşu-violet şi galben-verde.

În ilustraţia de mai sus, există mai multe variaţii de galben-verde în frunze şi mai multe variaţii de roşu-purpuriu în orhidee. Aceste culori opuse creează un maxim de contrast şi stabilitate.

Figura 5. Schema culorilor bazată pe exemple din natură



Sursă: <https://www.colormatters.com/color-and-design/basic-color-theory>

Natura oferă un punct de plecare perfect pentru teoria armoniei culorilor.

În ilustraţia de mai sus, galbenul, verdele şi roşu, creeză un design armonios, indiferent dacă această combinaţie se încadrează într-o formulă tehnică pentru armonia culorilor.

**1.2.2.2. Psihologia culorilor**

Psihologia culorilor este studiul modului în care ne influenţează culorile, fiind în acelaşi timp un subiect controversat din cauza problemelor generate de diferenţele culturale, preferinţa culorilor în funcţie de sex şi multe altele.

Cu toate toate acestea, există anumite interpretări, semnificaţii şi percepţii comune legate de simbolistica culorilor.

* Roşu – Atenţie, emoţie, furie, căldură, confort, viaţă, sânge, oprire.
* Galben – Aventură, fericire, competenţă, entuziasm, bogăţie, rafinament.
* Verde – Echilibru, bun gust, invidie, sănătate, bani, armonie.
* Albastru – Onestitate, corporativ, de înaltă calitate, masculinitate, competenţă, loialitate, încredere, fiabilitate.
* Roz – Dragoste, compasiune, rafinament, sinceriatate, romantism, blândeţe.
* Violet – Competenţă, autoritate, regalitate, respect, mister.
* Maro – Prietenie, rezistenţă, tristeţe, organic, natural.
* Negru – Rafinament, scump, inteligent, moarte, doliu.
* Alb – Simplitate, ordine, inocenţă, puritate, curăţenie, neutralitate.
* Gri şi argintiu – Atemporalitate, practic, neutralitate, rafinament[[3]](#footnote-3).

**1.3. Încredere**

Site-ul web trebuie să inspire încredere potenţialilor clienţi. Un design care pare profesionist va şterge instantaneu majoritatea îndoielilor pe care le-ar putea avea un utilizator atunci când ajunge la procesul de finalizare.

Conform unui sondaj efectuat pe un eşantion de 24.000 de utilizatori de internet din 24 de ţări, 22% din ei au declarat că nu cumpără niciodată online, iar dintre aceştia, 49% au dat ca motiv, lipsa de încredere.

Conform unui referat al Forumului Economic Mondial, s-au evidenţiat cinci moduri prin care putem creşte încrederea utilizatorilor în comerţul electronic[[4]](#footnote-4).

**1.3.1. Crearea unor reguli relevante de protecţie a consumatorilor online**

Normele de protecţie a consumatorilor online pot fi private, publice sau sisteme mixte.

În timp ce majoritatea ţărilor au un cadru legal pentru protecţia consumatorilor, doar 52% le-au actualizat pentru a acoperi activităţile online, potrivit Conferinţei Naţiunilor Unite pentru Comerţ şi Dezvoltare (UNCTAD) CyberLaw Tracker. Pentru alte ţări, legile sunt în stadiul de proiect.

Astfel că, firmele îşi publică propriile politici de rambursare, returnare şi anulare a comenzilor şi oferă clienţilor mecanisme de evaluare şi feedback.

**1.3.2. Acordarea unui grad ridicat de atenţie datelor personale**

Multe părţi interesate consideră regulile care protejează informaţiile personale ca făcând parte din setul de instrumente de protecţie a consumatorilor online.

Potrivit UNCTAD, 58% dintre ţări au reguli în viguare, 21% nu au, restul datelor fiind incerte.

Cercetările sugerează că aceste măsuri joacă un rol important în stimularea încrederii online.

Într-un sondaj realizat de KPMG pe un eşantion de 18.000 de consumatori online din 50 de ţări, 41% dintre respondenţi au spus că deţinerea controlului asupra modului în care au fost utilizate datele lor personale era mai probabil să îi facă să aibe încredere într-o companie, în special în America de Nord, Europa şi Africa de Sud.

**1.3.3. Diversitatea termenelor şi condiţiilor**

Normele de protecţie a consumatorilor online pot acoperi toate aspectele unei relaţii B2C (business to consumer). Cele mai provocatoare politici sunt cele care abordează asimetria informaţională, practicile comerciale neloiale, e-spam-ul, clauzele contractuale, securitatea plăţilor, responsabilitatea şi returnările.

Ţările pot avea abordări diferite, iar preferinţele privind autoreglarea şi standardizarea industriei pot varia.

**1.3.4. Creşterea cooperării internaţionale în domeniul comerţului electronic**

Comeţul electronic a devenit din ce în ce mai important în ultimele două decenii.

Unele estimări sugerează că este o piaţă în creştere de 25,3 trilioane de dolari. Cu toate acestea, pentru comercianţii online, accesul global şi conectarea consumatorilor din diferite ţări rămâne o provocare. Activitatea transfrontalieră de comerţ electronic a reprezentat doar 7% din totalul vânzărilor B2C online în 2015.

Percepţia consumatorilor online că achiziţile din străinătate ar putea fi mai puţin sigure şi că nu există soluţii ale problemelor ce pot apărea, are un impact negativ în raportul tranzacţionării globale a produselor online.

Este necesară consolidarea cooperării politicii internaţionale pentru a creşte încrederea online la nivelul întregului sistem, pe lângă reducerea divergenţelor internaţionale, dacă cererea trebuie să corespundă ofertei.

**1.3.5. Participarea la discuţiile internaţionale privind comerţul electronic**

Ţările au depus eforturi pentru a-şi alinia abordările privind regulile de protecţie a consumatorilor online la Organizaţia pentru Cooperare şi Dezvoltare Economică (OCDE) şi UNCTAD.

Un număr tot mai mare de acorduri comerciale preferenţiale (PTA) vizează consolidarea standardelor şi asigurarea unor abordări transparente. Unii încurajează cooperarea între agenţiile de protecţie a consumatorilor online, în timp ce alţii mandatează instituirea legilor privind protecţia informaţiilor generale.

În ianuarie 2019, 76 de ţări responsabile pentru 90% din comerţul global s-au angajat să înceapă negocieri cu privire la aspectele comerciale ale comerţului electronic.

**1.4. Transparenţă**

Pe măsură ce tărâmul eCommerce continuă să crească, iar opţiunile clienţilor se înmulțesc, aceştia devin din ce în ce mai mofturoşi atunci când vine vorba de pe ce site ar trebui să cumpere. Unul dintre factorii majori care influențează alegerea site-ului este transparența. Clienții vor să simtă că magazinele și site-urile de comerț electronic sunt transparente și le oferă toate informațiile. Transparența nu este bună doar pentru clienți, este benefică și pentru companii. O mai mare transparență duce la mai puține coșuri de cumpărături abandonate și clienți mai mulțumiți. Mai jos sunt câteva domenii pe care ar trebui să se concentreze comercianţii atunci când se doreşte îmbunătățirea transparenței activității de comerț electronic.

**1.4.1. Disponibilitatea produsului**

Primul pas pentru a îmbunătăți transparența site-ului de comerţ electronic este afișarea disponibilității produsului. Niciun client nu vrea să ajungă la sfârșitul verificării online doar pentru a afla că produsul pe care îl cumpără nu este în stoc și că ar putea dura săptămâni și uneori chiar luni să vină. Acesta este unul dintre cele mai rapide modalități de a face un client nemulţumit, de a anula comanda sau de a abandona coșul de cumpărături. Cea mai bună modalitate de a gestiona articolele din stoc, este afişarea unei note pe pagina produsului care spune că articolul nu este pe stoc și că va putea fi comandat ulterior, împreună cu o estimare a duratei de timp a comenzii. În acest fel, clientul va ști exact când va primi produsul în cazul în care alege să cumpere obiectul.

**1.4.2. Stabilirea preţurilor şi a promoţiilor**

Alte două domenii foarte importante asupra cărora se crește transparența sunt prețurile și promoțiile. Odată cu stabilirea prețurilor, trebuie să afişăm preţul întreg pe care îl vor plăti clienţii pentru un articol. La finalul plății, ei nu vor să vadă taxe diverse, atașate articolului pe care îl cumpără despre care nu știau. De asemenea, dacă prețul este redus pe baza numărului de articole pe care clientul le cumpără, acest lucru trebuie să le fie clar. Clientul nu ar trebui să se aștepte să plătească cinci dolari pentru un produs, dacă prețul de cinci dolari se aplică numai atunci când cumpără zece sau mai multe articole. Același lucru este valabil și pentru promoții. Promoțiile trebuie să fie clar conturate, astfel încât clienții să nu se simtă ca și cum ar fi fost păcăliţi să cumpere ceva ce credeau că este la promoţie, dar de fapt promoţia să nu se aplice.

**1.4.3. Taxele de transport şi manipulare**

Un alt domeniu care trebuie să aibă transparență îl reprezintă costurile de transport și de manipulare. Prețurile percepute pentru transport și manipulare trebuie să fie găsite cu ușurință pe site și ar trebui să existe un link sau o filă pe fiecare pagină de produs cu costul acestora, astfel încât clienții să poată vedea rapid cât costă transportul. Dacă așteptați până la sfârșitul plății pentru a arăta clienților prețul pe care îl vor plăti pentru expediere, mulți își vor abandona coșurile (studiile arată până la 72% vor abandona cumpărăturile dacă nu se oferă transport gratuit)[[5]](#footnote-5). Este mult mai ușor și mai constructiv ca posibilii cumpărători să fie anunţaţi de la început cât va costa transportul, astfel încât, aceştia să ştie dacă se încadrează în bugetul pe care ei îl au disponibil.

**1.4.4.** **Introducerea extrem de vizibilă a opțiunilor de plată**

Cumpărăturile online au devenit din ce în ce mai populare în ultimii ani, dar mulți clienți sunt în continuare precauți cu privire la utilizarea metodelor de plată necunoscute atunci când fac cumpărături online. Indiferent de modul în care compania alege să proceze plățile (Paypal, debit direct, etc.), trebuie furnizate informații detaliate și clare. Clienții nu vor petrece timpul căutând informațiile pe site., vor abandona pur și simplu comanda și vor accesa site-ul web al unui concurent.

**1.4.5. Descrierea cât mai exactă a produselor**

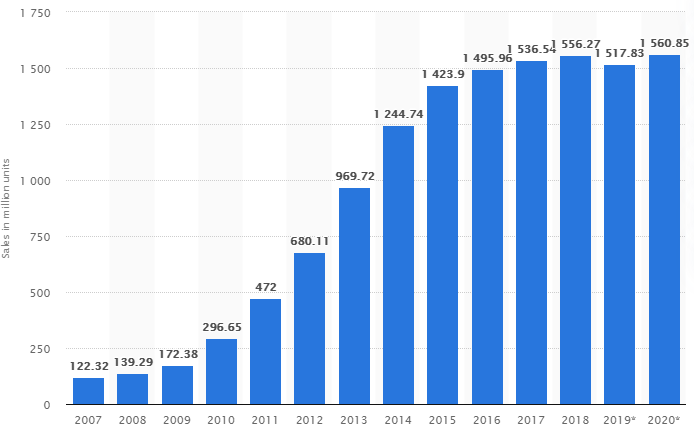
Toate descrierile produselor trebuie să fie cât mai detaliate pentru a evita confuziile. Descrierile înșelătoare conduc la reclamații ale clienților, recenzii negative și ocupă, de asemenea, atât din timpul clienţilor, cât şi din timpul şi resursele comerciantului.

**1.5. Caracteristica “mobile friendly” a unui site de comerț electronic**

Site-urile de comerț electronic trebuie sa fie foarte bine optimizate pentru telefonul mobil, deoarece statisticile[[6]](#footnote-6) arată că:

* 79% din utilizatorii de smartphone-uri l-au folosit cel puțin o dată pentru a face o achiziție online
* 40% din toate achizițiile de comerț electronic în perioada sărbătorilor din 2018 au fost efectuate pe un dispozitiv mobil
* 80% dintre cumpărători au folosit un smartphone în interiorul unui magazin fizic pentru a consulta recenziile produselor sau pentru a compara prețurile

Figura 6. Numărul de smartphone-uri vândute în lume din 2007 până în 2020



Sursă: <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>

Conform acestei statistici, se poate observa un trend ascendent al utilizarii internetului de pe smartphone, iar procentele cresc de la an la an.

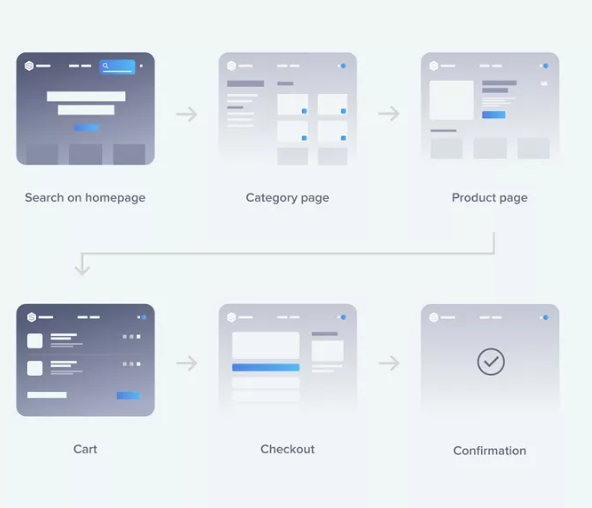
Astfel, optimizarea site-ului și a conținutului pentru mobil devine o necesitate, chiar dacă poate părea o altă sarcină de îndeplinit. Mai jos regăsim câteva modalități de optimizare a site-ului pentru mobil:

* Instalarea unei teme adaptabile
* Simplificarea meniului
* Păstrarea formularelor cât mai scurte
* Afișarea clară a butoanelor CTA( Call to Action )
* Inserarea unei funcții de căutare
* Adăugarea informațiilor cu privire la asistență tehnică
* Crearea butoanelor astfel încât să fie accesibile, atât ca plasare în site, cât și ca dimensiune
* Eliminarea pop-up-urilor
* Evitarea blocurilor mari de text
* Alegerea fontului potrivit
* Prioritizarea vitezei de încărcarea a site-ului

**1.6. Optimizarea procesului de vânzare**

Un site bun de comerț electronic trebuie să permită unui client nou care are pentru prima dată interacțiune cu site-ul respectiv, ușurarea procesului de a trece de la un pas la altul, până la cumpărarea produsului.

Figura 7. Model clasic de structură a unui site de comerț electronic



Sursă: <https://snipcart.com/ecommerce-website-design>

**1.6.1. Pagina de pornire**

Pagina de pornire este locul potrivit pentru a vinde brand-ul companiei noilor veniți, astfel încât aceștia ar trebui sa poată înțelege nucleul afacerii în câteva secunde, dar nici să facem din pagina de pornire o listă a produselor comercializate. Astfel, pagina de pornire trebuie să conțină:

* **Un titlu atrăgător**

Poate fi reprezentat de o frază bine structurată care să cuprindă esența produsului. Este, de asemenea, locul unde se pot expune campaniile de vânzări și promoțiile.

* **Butoane call-to-action vizibile**

Titlul trebuie să fie urmat de un call-to-action vizibil și relevant pentru produsele comercializate, care să nu fie nici prea generic și să exprime identitatea brand-ului.

* **Imagini cu produsele comercializate**

Imaginile de înaltă calitate arată profesionalism și inspiră încredere în mintea clienților. O imagine valorează o mie de cuvinte și, din moment ce cumpărătorii online nu au niciodată timpul pentru o mie de cuvinte, site-urile le oferă poze, multe dintre ele.

* **Includerea unei prezentări a celor mai bine vândute produse**

Chiar dacă pagina de pornire nu ar trebui să fie o listă de produse, asta nu înseamnă că nu se pot prezenta cele mai bune articole pe ea. Așa cum intr-un magazin fizic, se expun în vitrine cele mai bine vândute produse sau cele de prezentare pentru sărbători, evenimente sau anotimpuri, așa trebuie să conțină și pagina de pornire, reprezentată intr-un magazin fizic de vitrină, aceste produse.

**1.6.2. Pagina de listare produse**

Utilizatorii ajung în general la o listă (sau un catalog) prin bara de căutare sau selectând o categorie de produse prin navigare. Acolo vor găsi o listă a fiecărui produs asociat categoriei aproapiate nevoilor lor actuale.

Desigur, dacă site-ul comercializează doar câteva produse, este posibil să nu fie necesară defalcarea pe categorii. În acest caz, pagina de listare va include toate articolele comercializate. Acest tip de pagină, trebuie să conțină câteva elemente de design, astfel:

* **Introducerea pe categorii**

O scurtă introducere în pagina de catalog ar putea ajuta clienții să știe imediat dacă sunt la locul potrivit pentru a găsi ceea ce caută. Deoarece este foarte important ca utilizatorul să nu își piardă timpul căutând fără succes, descrierea categoriei trebuie să fie clară și transparentă.

* **Activarea funcțiilor de filtrare și sortare**

Chiar și în interiorul unei singure categorii, ar trebui să activăm filtrarea și sortarea. Cu cât sunt mai multe produse disponibile, cu atât este mai important acest aspect. A nu se evita folosirea mai multe variabile de sortare, cum ar fi culorile, gama de prețuri, mărimile, mărcile, etc.

Un alt aspect foarte important este că în cazul în care nu există rezultate pentru filtrele aplicate, nu trebuie să returnăm doar o pagină goală care să anunțe utilizatorul că nu există produse pentru filtrele aplicate, ci trebuie să folosim această oportunitate pentru a păstra clientul pe site, evidențiându-i , spre exemplu, cele mai bine vândute produse și recenziile acestora.

* **Afișarea disponibilității stocului**

Aceasta este mai mult o măsură preventivă, astfel în cazul în care un articol este epuizat, clientul să nu afle abia când dorește să finalizeze comanda, evitând astfel frustrarea acestuia sau chiar abandonarea cumpărăturilor. Mai mult decât atât, în cazul unor versiuni limitate sau cantități, aceasta poate reprezenta un stimulent excelent pentru a cumpăra rapid.

Un alt aspect important este alegerea modului de afișare a produselor, unde remarcăm două categorii: de tip listă și de tip grilă.

O vizualizare a listei este mai potrivită pentru produsele care au nevoie de descrieri mai lungi și informații extinse. Acestea pot fi produse tehnice sau de înaltă calitate.

Pe de altă parte, o vedere grilă este o potrivire bună pentru produsele care necesită mai mult o prezentare vizuală, fără prea multe descrieri. Acest format permite, de asemenea, compararea rapidă a produselor.

**1.6.3. Pagina de produs**

Pentru a facilita trecerea la pasul următor, această pagină trebuie să fie cât mai aproape de o experiență reală de cumpărături. Aceasta înseamnă că fotografiile, descrierile și specificațiile ar trebui să fie suficient de detaliate, astfel încât un client să se simtă încrezător să-și investească banii și să obțină ceea ce plătește.

Mai jos regăsim elementele esențiale ale unei pagini de produse și modul în care acestea trebuie concepute corect:

**1.6.3.1. Butonul „Adaugă în coș”**

Acesta este elementul central al paginii de produs, fiind un pas necesar către o tranzacție viitoare.

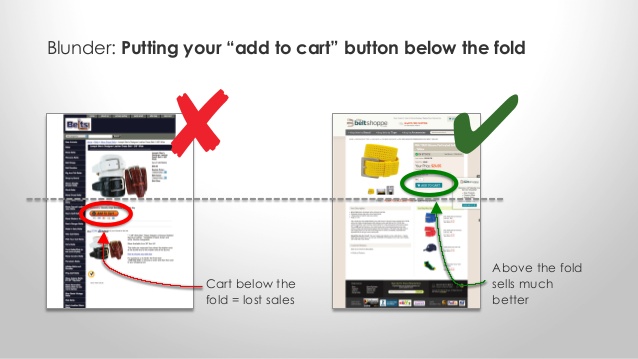
Un buton de cumpărare poate părea un element destul de simplu de proiectat, dar există câteva reguli și elemente de luat în vedere la proiectarea acestui buton:

* **Să fie vizibil și proeminent**

Pentru a respecta această regulă butonul trenbuie să fie plasat în cel mai evident loc posibil, să fie la îndemâna clienților, în orice moment.

Un loc bun de plasare al butonului este deasupra marginii de jos a ferestrei paginii web, ca în imaginea de mai jos.

Figura 8. Plasarea butonului „Adaugă în coș”



Sursă: <https://www.slideshare.net/HubSpot/conversion-blunders-etailers-should-avoid-at-all-costs-webinar-slides>

De asemenea, butonul trebuie să respire, iar pentru ca acesta să iasă în evidență, nu trebuiesc înghesuite prea multe elemente în jurul său.

O altă practică bună este dublarea butonului de cumpărare și plasarea lui și în partea de jos a paginii. În consecință, un client care derulează pe întreaga pagină este încă invitat să cumpere și în partea de jos a acesteia.

* **Să fie contrastat**

Pentru această regulă putem folosi o culoare care nu a mai fost folosită în altă parte a paginii.

* **Să fie mare**

**1.6.3.2. Imaginile**

Imaginile reprezintă cea mai bună oportunitate de a prezenta produsul către potențialii clienți. Oamenii vor să știe și să vadă ce cumpără. Scopul în această etapă este de a face experiența cât mai tangibilă pentru o platformă digitală. Există câteva modalități de a realiza acest lucru:

* **Folosirea imaginilor realizate profesional din unghiuri diferite**

Fiecare caracteristică semnificativă a produsului trebuie să fie reprezentată vizual, așa că se recomandă utilizarea a cât mai multor unghiuri necesare pentru a realiza acest lucru.

* **Utilizarea modalităților interactive pentru a afișa imaginile**

O prezentare de diapozitive foto cu o funcție de zoom este un exemplu de experiență interactivă pentru utilizatori atunci când vine vorba de imagini.

* **Folosirea demonstrațiilor video**

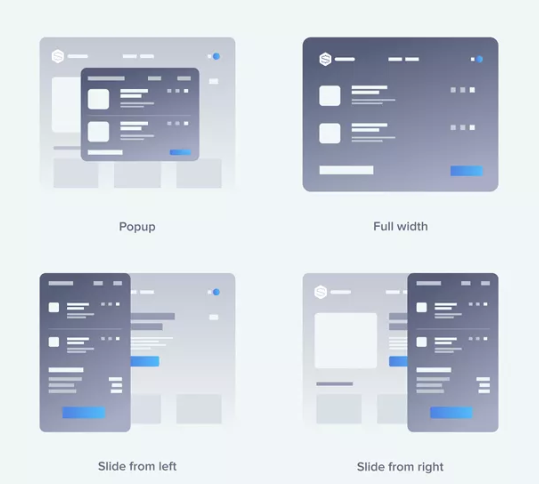
Demonstrațiile video devin din ce în ce mai populare, astfel dacă produsul este mai greu de vândut doar cu imagini, ar trebui luată în considerare și acestă posibilitate.

**1.6.3. Încheierea tranzacţiei**

Un aspect foarte important care poate face diferența între o comandă finalizată și un coș abandonat este procesul de checkout.

Mai jos avem mai multe modalităţi de afişare a coşului de cumpărături, niciuna din ele nefiind greşită. Important este că deschiderea coșului nu ar trebui să introducă o pauză în experiența de cumpărături. Ar trebui să fie ușor să arunci o privire rapidă asupra conținutului coșului și apoi să te întorci la cumpărături, dacă este nevoie.

Figura 9. Modalități de afișare a coșului de cumpărături



Sursă: <https://snipcart.com/ecommerce-website-design>

Dacă coșul este gol, se poate lansa o invitaţie către utilizatori, să adauge produse în coş, prin butoane call-to-action sau prin sugestii despre produs. Dacă există produse în coș, utilizatorii ar trebui să vadă rapid aceste elemente prezentate într-un mod clar, organizat:

* Inventar de produse (cu imagini şi scurte descrieri)
* Cantitate
* Opţiuni ale produsului (cu câmpuri personalizate)
* Preţ individual
* Preţ total

Am menționat deja că întregul site web ar trebui optimizat pentru mobil, dar este deosebit de crucial atunci când vine vorba de coșul de cumpărături. În timp ce vânzările pe mobil ar trebui să reprezinte aproximativ 52% din toate vânzările de comerț electronic până în 2021, rata medie de abandonare a coșurilor de cumpărături mobile este încă incredibil de mare. Vorbim de o proporție de 85%.

Mai jos avem o statistică a motivelor de abandonare a cumpărăturilor online, astfel putem să observăm că procesul de optimizare (mai ales al coşului de cumpărături) joacă un rol crucial.

Figura 10. Motive de abandon ale cumpărăturilor online

Sursă: <https://www.quicksprout.com/mobile-friendly-website-guide/>

Există câteva elemente pe care trebuie să le luăm în considerare la proiectarea design-ului pe mobil, din punctul de vedere al experienţei utilizatorului:

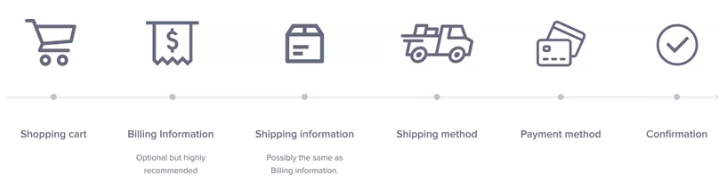
* Utilizarea unui buton call-to-action proeminent („Finalizare” sau „Comandă completă”) în partea de sus a paginii care nu se mișcă în timp ce utilizatorul derulează în jos. În acest fel, rămâne mereu vizibil și la îndemâna utilizatorului pentru a da click.
* Utilizarea unor butoane clare și ușor de atins pentru toate acțiunile, de la scăderea cantității produsului până la avansarea până la finalizare și ascunderea coșului. Degetele sunt mult mai puţin precise decât un mouse, astfel este foarte importantă dimensiunea textului şi a butoanelor.

**1.6.4. Finalizarea**

Fluxul de finalizare al comenzii ar trebui să conţină următoarele elemente şi în următoarea ordine

* Coşul de cumpărături
* Date de facturare (opţional, dar recomandat)
* Informaţii despre expediere
* Metodă de livrare
* Metodă de plată
* Confirmare

Figura 11. Fluxul de finalizare al comenzii



Sursă: <https://snipcart.com/ecommerce-website-design>

**2. Tehnologii folosite în crearea site-urilor**

**2.1. HTML - (HyperText Markup Language)**

HTML este un standard folosit pentru construirea paginilor web și are abilitatea de a aduce împreună text, imagini, sunete și video. Documentele HTML sunt fișiere cu text simplu, de aceea ele pot fi editate folosind editoare standard ca NotePad și WordPad. Acestea pot fi salvate cu extensia .html, .htm sau .shtml și afișate folosind browsere ca Mozilla Firefox, Chrome, Safari, Opera, Internet Explorer și altele.

HyperText Markup Language este instrumentul de formatare standard pentru World Wide Web, deoarece are proprietatea de multi-platformă, ceea ce înseamnă că este compatibil și poate rula pe diferite tipuri de computere și dispozitive mobile cu diferite sisteme de operare.

**2.1.1. Scurtă istorie**

La sfârşitul anului 1991, cercetătorul britanic Tim Berners-Lee publică prima scriere, numită „HTML Tags”, publicaţie ce descrie 18 elemente care conțin designul inițial, relativ simplu de HTML. Cu excepția etichetei hyperlink, acestea au fost puternic influențate de SGML (Standard Generalized Markup Language) , un standard pentru definirea limbajelor de marcare generalizate pentru documente, folosit la CERN, acolo unde Tim Berners-Lee lucra.. Unsprezece dintre aceste elemente există încă în HTML.

Ca urmare a apariţiei HTML, Berners-Lee a dezvoltat un protocol de acces, prin hyperlink-uri, numit HTTP (HyperText Transfer Protocol)[[7]](#footnote-7).

Fiind într-un continuu proces de dezvoltare, cu modificări aduse constant, HTML a evoluat în mai multe versiuni, astfel:

* HTML 2.0 (Noiembrie 1995)
* HTML 3.2 (Ianuarie 1997)
* HTML 4.1 (Decembrie 1999)
* HTML5 (în curs de dezvoltare din octombrie 2011 şi până în prezent)[[8]](#footnote-8)

O altă descoperire importantă o reprezintă apariţia CSS (Cascading Style Sheets), alături de HTML 4.0, deoarece a permis separarea opţiunilor de formatare şi stilizare a paginilor web de scopul principal al HTML, de definire a structurii paginii web. Astfel, odată cu introducerea CSS, s-a putut realiza separarea noţiunilor de formatare de documentul HTML într-un alt fişier, ce poate fi accesat şi inclus în document, folosind o etichetă de legătură (link tag).

**2.1.2. Tag-uri HTML**

Tag-urile HTML sunt cuvinte și simboluri text incluse în paranteze unghiulare, <tag>, descriind de obicei începutul și finalul unei informații sau comenzi. Paginile web sunt vizualizate pe Internet cu ajutorul browserelor web, care citesc codurile HTML și le afișează astfel încât să poată fi citite de vizitatori. Practic, browserele web afișează informația conținută între tag-uri într-un anumit mod specificat de tag-urile corespunzătoare.

**2.1.2.1. Tag-uri principale**

|  |  |
| --- | --- |
| <!DOCTYPE> | Definește tipul de document |
| <html> | Definește un document HTML |
| <head> | Cuprinde informații despre document |
| <title> | Setează titlul |
| <body> | Definește cuprinsul documentului |
| <h1> to <h6> | Setează rubricile documentului |
| <p> | Setează apariția unui paragraf |
| <br> | Inserează o pauză de linie |
| <hr> | Definește o schimbare tematică a conținutului |
| <!--...--> | Definește un comentariu |

Tabelul 1.1.Tag-uri principale

**2.1.2.2. Tag-uri de formatare**

Procedura corectă pentru formatarea textului și stilizarea lui este utlizarea CSS, însă în situații rare, se pot folosi și tag-uri HTML.

|  |  |
| --- | --- |
| <abbr> | Definește o abreviere sau un acronim |
| <address> | Definește informațiile de contact ale autorului |
| <b> | Setează îngroșarea textului |
| <blockquote> | Marchează o secțiune care este un citat din altă sursă |
| <mark> | Marchează textul prin evidențierea lui |
| <template> | Definește un șablon |
| <time> | Setează data/ceasul |
| <u> | Marchează textul într-un stil diferit față de cel normal |

Tabelul 1.2.Tag-uri de formatare

Tag-uri ca <big>, <center>, <font>, <strike>, <tt> nu mai sunt compatibile în HTML5, recomandându-se folosirea CSS pentru suplinirea lor.

**2.1.2.3. Tag-uri pentru formulare și introducere**

|  |  |
| --- | --- |
| <form> | Definește un formular pentru introducere date |
| <input> | Marchează posibilitatea introducerii datelor |
| <textarea> | Definește mai multe linii pentru introducerea datelor |
| <button> | Definește un buton |
| <select> | Definește o listă |
| <optgroup> | Definește un grup de acțiuni dintr-o listă |
| <option> | Definește o opțiune dintr-o listă |
| <label> | Definește o etichetă pentru elementul introdus |
| <fieldset> | Grupează elemente asemănătoare într-un formular |
| <datalist> | Specifică o listă de opțiuni predefinite |

Tabelul 1.3.Tag-uri pentru formulare și introducere

**2.1.2.4. Tag-uri pentru imagini**

|  |  |
| --- | --- |
| <img> | Definește o imagine |
| <map> | Definește o imagine-hartă |
| <area> | Definește o zonă într-o imagine-hartă |
| <canvas> | Este folosit pentru grafică, prin scripturi |
| <picture> | Definește un container pentru mai multe imagini |
| <svg> | Definește un container pentru grafice cu vectori scalabili |

Tabelul 1.4.Tag-uri pentru imagini

**2.1.2.5. Tag-uri Audio/Video[[9]](#footnote-9)**

|  |  |
| --- | --- |
| <audio> | Definește conținut audio |
| <source> | Definește conținut media (<video>,<audio>,<picture>) |
| <track> | Definește text vizibil atunci când rulează conținut media |
| <video> | Definește conținut video |

Tabelul 1.5.Tag-uri Audio/Video

**2.1.2.6. Tag-uri de legătură**

|  |  |
| --- | --- |
| <a> | Definește un hyperlink |
| <link> | Definește relația cu o resursă externă |
| <nav> | Definește legăturile în bara de navigare |

Tabelul 1.6.Tag-uri de legătură

**2.1.2.7. Tag-uri pentru liste**

|  |  |
| --- | --- |
| <ul> | Definește o listă neordonată |
| <ol> | Definește o listă ordonată |
| <li> | Definește un element al listei |
| <dl> | Definește o listă de descrieri |
| <dt> | Definește un termen într-o listă de descrieri |
| <dd> | Definește o descriere a unui termen într-o listă |

Tabelul 1.7.Tag-uri pentru liste

**2.1.2.8. Tag-uri semantice și de stil[[10]](#footnote-10)**

|  |  |
| --- | --- |
| <style> | Definește informații stilistice |
| <div> | Definește o secțiune |
| <span> | Definește o secțiune |
| <header> | Definește antetul unui document/secțiune |
| <footer> | Definește subsolul unui document/secțiune |
| <main> | Definește conținutul principal dintr-un document |
| <section> | Definește o secțiune |
| <article> | Definește un articol |
| <details> | Definește detalii care se pot vizualiza/ascunde |
| <dialog> | Marchează o fereastră de dialog |
| <data> | Asignează conținutul unei traduceri automate |

Tabelul 1.8.Tag-uri semantice și de stil

**2.1.2.9. Tag-uri meta**

|  |  |
| --- | --- |
| <head> | Definește informații despre document |
| <meta> | Definește metadate despre document |
| <base> | Specifică adresa URL de bază |

Tabelul 1.9.Tag-uri meta

**2.1.2.10. Tag-uri pentru programare[[11]](#footnote-11)**

|  |  |
| --- | --- |
| <script> | Definește un script |
| <embed> | Definește un container pentru o aplicație externă |
| <object> | Definește un obiect |
| <param> | Definește un parametru pentru un obiect |

Tabelul 1.10.Tag-uri pentru programare

**2.1.3. Atribute HTML**

Elementele HTML pot avea atribute în deschiderea etichetei. Atributul oferă informații suplimentare despre acel element. Atributele HTML sunt adăugate în eticheta de deschidere și pot avea o anumită valoare. Cele mai multe dintre atributele unui element sunt perechi nume-valoare, separate prin „=” și scrise în eticheta de start a unui element, după numele elementului. Valoarea poate fi între ghilimele simple sau duble, cu toate că valorile compuse din anumite caractere pot fi lăsate necotate în HTML. Lăsând valorile atributelor necotate este considerată o acțiune cu rezultate nesigure.

**2.1.3.1. Atribute comune**

Există mai multe atribute comune care pot apărea în mai multe elemente:

* Atributul „id” oferă un identificator unic pentru un element. Acest lucru este folosit pentru a identifica elementul astfel încât foile de stil pot modifica proprietățile sale de prezentare, iar script-urile pot modifica, anima sau șterge conținutul sau prezentarea acestuia. Anexat la URL-ul paginii, acesta oferă un identificator unic global pentru element, de obicei o sub-secțiune a paginii.
* Atributul „class” oferă o modalitate de clasificare a elementelor similare. Acest lucru poate fi utilizat în scopuri semantice sau de prezentare. De exemplu, un document HTML ar putea folosi semantic desemnarea class=„notație” pentru a identifica faptul că toate elementele cu această valoare de clasă sunt subordonate textului principal al documentului. În prezentare, astfel de elemente ar putea fi adunate împreună și prezentate ca note de subsol pe o pagină, în loc de a apărea în locul în care apar în sursa HTML. Atributele de clasă sunt utilizate semantic în microformate și pot avea valori multiple specificate.
* Atributul „style” este folosit pentru a atribui proprietăți de pezentare unui anumit element. Se consideră mai bună practica de a folosi un atribut „id” sau „class” al unui element pentru a selecta elementul din interiorul unei foi de stil.
* Atributul „title” este folosit pentru a atașa explicație subtextuală unui element. În cele mai multe browsere acest atribut este afișat ca o notă informativă.
* Atributul „lang” identifică limbajul natural al conținuturilor elementului, care pot fi diferite de cele din restul documentului[[12]](#footnote-12).

**2.1.3.2. Atribute boolean**

O serie de atribute sunt atribute booleane. Prezența unui atribut boolean pe un element reprezintă valoarea adevărată, iar absența atributului reprezintă valoarea falsă.

Dacă atributul este prezent, valoarea acestuia trebuie să fie șirul gol sau o valoare care este o potrivire nesemnificativă ASCII pentru numele canonic al atributului, fără niciun spațiu alb sau principal. Valorile „adevărat” și „fals” nu sunt permise atributelor booleane. Pentru a reprezenta o valoare falsă, atributul trebuie omis în totalitate.

|  |  |
| --- | --- |
| Checked | Indică selecția într-o căsuță de verificare |
| Selected | Indică opțiunea selectată dintr-o listă de opțiuni |
| Disabled | Dezactivează butoane, opțiuni, selecții, etc. |
| Readonly | Împiedică utilizatorul să scrie text |

Tabelul 1.11.Atribute boolean

**2.1.3.3. Atribute prezentaționale[[13]](#footnote-13)**

Atributele prezentaționale au ca utilitate stilizarea elementelor HTML.

|  |  |
| --- | --- |
| Style | Definește stilurile CSS |
| Size | Definește lungimea unui element |
| Color | Definește culoarea unui element |
| Border | Setează liniile de frontieră într-un tabel |
| Cellspacing | Schimbă spațiul între celulele tabelului |
| Align | Aliniază stânga, dreapta și centru |

Tabelul 1.12.Atribute prezentaționale

Dintre toate atributele prezentaționale doar „style” și „size” se mai folosesc în HTML5, toate celelalte fiind înlocuite de atribute CSS.

**2.1.4. HTML5**

HTML5 este un limbaj utilizat pentru structurarea și prezentarea conținutului pentru World Wide Web, reprezentând o tehnologie nucleu pentru Internet.

Propus inițial pentru software-ul Opera, HTML5 este a cincea revizuire a standardului HTML și din octombrie 2011 este în curs de dezvoltare. Obiectivele sale principale au fost acelea de a îmbunătăți limbajul cu un suport pentru cele mai recente apariții multimedia, menținându-l în același timp, ușor de citit de oameni și bine înțeles de computere și device-uri.

HTML5 este destinat să subsumeze nu numai HTML4, dar și XHTML1 și DOM2HTML.  
Urmărind predecesorii săi imediați HTML 4.01 și XHTML 1.1, HTML5 este un răspuns la observația că HTML și XHTML, utilizate în comun pe World Wide Web, sunt un amestec de caracteristici introduse de specificații diferite. Împreună cu acestea mai sunt și caracteristicile diferite aduse de software, de browsere, și multe erori de sintaxă în documentele web existente. Astfel, HTML5 devine o încercare de a defini un singur limbaj de marcare, care poate fi scris în oricare dintre sintaxele HTML sau/și XHTML. Acesta include modele detaliate de prelucrare pentru a încuraja mai multe implementări interoperabile, extinde, îmbunătățește, raționalizează disponibilitățile pentru documentele web și introduce marcaje și interfețe de programare a aplicațiilor (API) pentru aplicații web complexe.

Din aceste motive, HTML5 este un bun candidat pentru aplicațiile de platforme mobile. Multe caracteristici ale HTML5 au fost create din considerarea că va trebui să devină capabil să ruleze pe dispozitive cum ar fi smartphone-urile sau tabletele.

În special, HTML5 aduce multe noi caracteristici sintactice. Acestea cuprind elemente ca <video>, <audio> și <canvas>,  precum și integrarea de conținut cu grafică vectorială scalabilă (SVG), care a înlocuit utilizarea tag-ului generic <object>. Aceste noutăți sunt proiectate pentru a facilita includerea și manipularea în web a conținuturilor multimedia și grafice fără a fi nevoie să se recurgă la proprietățile de plugin și API. Alte noi elemente ca <section>, <article>, <header> și <nav> sunt proiectate să îmbunătățească conținutul semantic al documentelor. Noi atribute au fost introduse în același scop dar, în același timp, unele elemente și atribute au fost îndepărtate. Elemente ca <a>, <cite> și <menu> au fost schimbate, redefinite și standardizate. API-urile și DOM-urile (Document Object Model) nu mai sunt lăsate pe planul secund și devin părți fundamentale în specificațiile HTML5.

 HTML5, de asemenea, definește in câteva detalii, prelucrările necesare pentru documentele invalide, astfel încât sintaxa erorilor va fi tratată uniform de toate browserele cunoscute[[14]](#footnote-14).

**2.1.4.1.Caracteristicile HTML5**

**2.1.4.1.1. Marcarea**

HTML5 introduce elemente și atribute care reflectă utilizarea tipică pe site-urile web moderne. Unele dintre ele sunt înlocuiri semantice pentru utilizări comune ale elementelor de bloc generic (<div>) și inline (<span>), de exemplu <nav> (bloc de navigare site), <footer> (de obicei referindu-se la partea de jos a paginii web sau la ultimele linii de cod HTML), sau <audio> și <video> în loc de <object>. Unele elemente depreciate din HTML 4.01 au fost retrase, inclusiv elemente pur formale, cum ar fi <font> și <center>, ale căror efecte au fost de mult timp înlocuite cu mai capabilul CSS. Există, de asemenea, un accent reînnoit pe importanța de scripting DOM (de exemplu, JavaScript) în comportamentul Web.

Sintaxa HTML5 nu se mai bazează pe SGML, în ciuda similitudinii marcării sale.

Aceasta este dotată cu o noua linie introductivă care arată ca o declarație de tip document SGML, <!DOCTYPE html>, care declanșează modul de redare care respectă standardele.

**2.1.4.1.2. Noi API-uri**

Față de specificarea marcării, HTML5 specifică și scripting pentru interfețe de programare a aplicațiilor (API-uri), care pot fi utilizate cu JavaScript.

Se introduc, astfel, noi API- uri, cum ar fi:

* Elementul canvas pentru modul direct de proiectare 2D
* Redare media cronometrată
* Aplicații web offline
* Editarea de documente
* Drag-and-drop
* Mesagerie cross-document
* Managementul istoricului browser-ului
* Înregistrare manipulant de protocol și tip MIME
* Microdate
* Stocare web, un cadru de stocare pereche de chei-valoare care oferă un comportament similar cu cookie-uri, dar cu o capacitate de stocare mai mare

Nu toate tehnologiile de mai sus sunt incluse în specificația W3C HTML5, deși ele sunt în specificația HTML WHATWG. Unele tehnologii conexe, care nu fac parte nici din HTML W3C, nici din specificația WHATWG HTML, sunt după cum urmează:

* *Geolocation*, folosit pentru localizare
* *Web SQL Database*, o bază de date SQL localp (nu mai este menținută)
* *Indexed Database API*, o stocare de chei-valori indexate ierarhic (fosta WebSimpleDB)
* *HTML5 API File*, se ocupă de încărcarea fișierelor și manipularea lor
* *Directories and System*, un API destinat să satisfacă cazuri de utilizare stocare

pe parte de client, care nu sunt deservite de bazele de date

* *File Writer*, un API pentru scrierea de fișiere din aplicații web
* *Web Audio API*, un API JavaScript de nivel înalt pentru prelucrarea și sintetizarea sunetelor în aplicațiile web
* *ClassList API*, folosit pentru a adăuga, elimina și comuta clasele unui element

HTML5 nu poate oferi animație în pagini web. Este necesară o funcționalitate suplimentară JavaScript sau CSS3 pentru animarea elementelor HTML[[15]](#footnote-15).

**2.1.5. Editoare HTML**

Codul HTML al unei pagini web poate fi scris cu oricare editor de text, însă editoarele HTML specializate pot oferi conveniență și funcționalitate suplimentară. De exemplu, multe editoare HTML nu lucrează numai cu HTML, ci și cu tehnologii corespunzătoare precum CSS, XML și JavaScript sau ECMAScript. În unele cazuri, pot administra și comunicații cu servere la distanță via FTP și WebDAV, dar și sisteme de management al versiunii precum CVS sau Subversion.

**2.1.5.1. Tipuri de editoare HTML**

Există diferite tipuri de editoare HTML: text, obiect și WYSIWYG (What You See Is What You Get).

**2.1.5.1.1. Editoarele de text**

Editoarele de text (sursă) folosite pentru HTML oferă de obicei, posibilitatea evidențierii sintaxei. Template-uri, bare de utilități și accesul rapid al tastaturii pot introduce ușor elemente și structuri HTML. Wizard, bulele informative și autocompletarea pot ajuta în sarcinile obișnuite.

Editoarele de text HTML includ de obicei, funcțiuni integrate sau integrarea cu instrumente externe pentru astfel de sarcini precum sursa și controlul versiunii, verificare linkuri, verificare cod și validare, curățire cod și formatare, verificare gramatică, încărcare prin FTP sau WebDAV și structurare ca un proiect.

Editoarele de text necesită înțelegerea de către utilizator a HTML și a oricărei alte tehnologii web pe care designerul vrea să o folosească, precum CSS, JavaScript și limbaje scripting pentru server.

**2.1.5.1.2. Editoare de obiecte**

Unele editoare permit editarea alternativă a textului sursă al obiectului într-un mod vizual mai organizat decât simpla evidențiere prin colorare, dar fără WYSIWYG. Unele editoare WYSIWYG includ opțiunea de folosire a ferestrei de paletă care permite editarea parametrilor în câmpuri pentru fiecare parametru, sau ferestre de text pentru a edita întregul grup de text sursă pentru obiectul selectat. Ele pot include widget-uri pentru a prezenta și selecta opțiuni când sunt editați parametrii.

**2.1.5.1.3. Editoare WYSIWYG**

Editoarele HTML WYSIWYG oferă o interfață de editare care arată cum va fi afișată pagina într-un browser web. Unele editoare, precum cele sub forma extensiilor de browser, permit editarea într-un browser web. Întrucât folosirea unui editor WYSIWYG nu necesită cunoștințe HTML, sunt ușor de folosit de un utilizator, mediu ca nivel, de calculator.

Vizualizarea WYSIWYG este realizată prin inserarea unui dispozitiv asemănător cu cel folosit într-un browser web. Acest dispozitiv a fost îmbunătățit considerabil de dezvoltatorii editoarelor pentru a permite tipărirea, copierea, ștergerea și mutarea conținutului. Scopul este ca oricând pe perioada editării, rezultatul sa fie văzut așa cum se va vedea el într-un browser web.

Editoarele WYSIWYG fac web designul mai rapid și mai ușor, dar mulți profesioniști încă folosesc editoarele text, în ciuda faptului că cele mai multe editoare WYSIWYG au și un mod de editare manual pentru codul HTML. Web-ul nu a fost creat inițial ca un mediu vizual, iar încercările de a avea mai mult control, precum prin CSS, au fost foarte puțin sprijinite de majoritatea browserelor web. Din această cauză, codul generat automat de editoarele WYSIWYG dăunau adesea mărimii fișierelor și compatibilității cu browserele[[16]](#footnote-16).

**2.2. CSS - (Cascading Style Sheets)**

Cascading Style Sheets este un standard pentru formatarea și afișarea elementelor unui document HTML, permițând definirea construcțiilor de stil ca fonturi, culori și poziționare.

Odată cu introducerea HTML 3.2 au fost introduse și atributele de personalizare a tag-urilor precum "font", "color" etc. A fost atunci cand limbajul de programare HTML a devenit greoi. Fiecare pagină a website-ului trebuia luată separat și modificate proprietățile elementelor principale. Aceasta problemă a fost rezolvată în versiunea 4.0 a HTML-ului. Toate atributele de personalizare au fost scoase și salvate într-un fișier extern cu extensia ".css".

În felul acesta, modificând un singur fișier, putem schimba forma în care sunt afișate toate paginile unui website[[17]](#footnote-17).

**2.2.1. Inserarea foilor de stil**

O foaie de stil CSS poate fi inserată prin mai multe modalități, astfel:

a)**Ca fișier extern**

O foaie de stil externă este ideală atunci când stilul respectiv se aplică mai multor pagini. Cu o foaie CSS externă, puteți modifica aspectul întregului site modificând un singur fișier.

Fiecare pagină trebuie să se lege la foaia de stil folosind tagul <link> în interiorul secțiunii head:

<html>

<head>

<link rel=”stylesheet” type=”text/css” href=”style.css”/>

</head>

<body>

.......

</body>

</html>

unde:

- atributul **rel** specifică relația dintre documentul apelant și documentul legat. Are numeroase valori posibile, dar în cazul de față valoarea sa trebuie să fie **stylesheet**;

- atributul **type** specifică tipul documentului legat, adică definește ce fel de resursă este (extensia fișierului nu este suficientă pentru a determina această) - în cazul nostru trebuie să aibă valoarea **text/css;**

- atributul **href** cum am văzut și la elementul link, specifică adresa (URL) de unde se introduce fișierul foaie de stil.

O foaie externă de stil poate fi scrisă în **Notepad++,** nu trebuie să conțină taguri html și trebuie salvată cu extensia .css. Un exemplu de foaie de stil externă, memorată în fișierul style.css, este prezentată mai jos:

hr {color:sienna;}

p {margin-left:20px;}

body {background-image:url("roses.gif");}

b) **Ca foaie internă de stil**

O foaie internă de stil trebuie folosită când un singur document are un anumit stil. Foaia internă de stil se definește în secțiunea **head**a paginii HTML folosind tagul **<style>**, ca în exemplul următor:

<head>

<style type="text/css">

hr {color:#424242;}

p {margin-left:20px;}

body {background-image:url("roses.gif");}

</style>

</head>

c) **Ca stiluri inline**

Un stil inline pierde multe din avantajele foilor de stiluri amestecând conținutul cu prezentarea. Se recomandă folosirea acestei metode cât mai rar. Pentru a utiliza un stil inline, se folosește atributul style în tagul respectiv. Atributul style poate conține orice proprietăți CSS. Exemplul următor ilustrează cum se poate schimba culoarea și marginea stângă a unui paragraf:

<p style="color:#424242; margin-left:20px">Acesta este un paragraf.</p>

d) **Ca foi multiple de stiluri**

Dacă anumite proprietăți ale unui selector au fost setate în foi de stiluri diferite, valorile vor fi moștenite din cea mai specifică foaie de stil. Spre exemplu,într-o foaie de stil externă, selectorul h3 are setate următoarele trei proprietăți:

h3 {

color:red;

text-align:left;

font-size:8pt;

}

Și foaia de stil intern setează pentru **h3** următoarele proprietăți:

h3 {

text-align:right;

font-size:20pt;

}

Dacă pagina care conține această foaie internă de stil se leagă și la foaia de stil externă, atunci proprietățile pentru h3 vor fi:

color:red;

text-align:right;

font-size:20pt;

Culoarea este moștenită din foaia externă, iar alinierea și dimensiunea fontului sunt înlocuite cu valorile din foaia internă de stil.

**2.2.2. Ordinea de aplicare a foilor de stil**

Termenul „Cascading” se referă la maniera în care fiecare regulă de stil se aplică elementelor într-un document HTML. Într-o foaie de stiluri, acestea formează o ierarhie în care stilurile mai specifice înlocuiesc stilurile mai generale. Este responsabilitatea CSS să determine prioritatea regulilor de stil conform acestei ierarhii, care stabilește, astfel un efect de cascadă.

La modul general, putem spune că stilurile se vor mixa într-o foaie de stil virtuală după următoarele reguli:

* setările implicite ale browserului
* foi de stiluri externe (din fișiere CSS)
* foi de stiluri interne (din secțiunea head)
* stiluri inline (din tagul elementului HTML)[[18]](#footnote-18)

**2.2.3. Selectori**

Odată cu adăugarea elementelor într-o pagină web, acestea pot fi stilizate folosind CSS. Un selector definește exact cărui element sau căror elemente din documentul HTML li se vor aplica aceste stiluri ca mărime, culoare sau poziționare.

Cei mai comuni selectori folosiți sunt:

* **selectorii de tip element** – vizează elemente în funcție de numele acestora;

De exemplu, având codul HTML:

<body>  
 <h1> Headline 1 </h1>  
 <h1> Headline 2 </h1>

<p> Paragraph 1 </p>  
 <p> Paragraph 2 </p>  
 </body>

Dorim să selectăm pentru stilizare doar elementele <p> din documentul HTML. Astfel, va fi nevoie doar să specificăm numele elementului și să definim regulile de formatare.

Sintaxa CSS: **p** {  
 color: red;  
 font-size: 20px;  
 }

* **selectorii de tip id** - vizează elementul care este definit printr-un id specificat;

Exemplu HTML: <h1 **id**="first-headline"> Headline 1 </h1>

Sintaxa CSS: **#first-headline** {  
 color: purple;  
 font-size: 30px;  
 }

* **selectorii de tip clasă** – permit selectarea unui element pe baza atributului de clasă, indiferent de numele etichetei lui;

Exemplu: a) HTML:

<h1 **class**="my-first-class"> Headline 1 </h1>

b) Sintaxa CSS:

**.my-first-class**{  
 color:green;  
 }

* **selectorii de tip atribute** – vizează elemente cu un atribut și o valoare anume; acest tip de selectori poate fi util în cazul câmpurilor de intrare;

De exemplu, având codul HTML: <body>  
 <input type="text" />  
 <input type="number" />  
 </body>

Pentru a specifica reguli doar pentru intrările de numere, de exemplu, trebuie să definim un selector de atribute (după numele etichetei din paranteze, scriind numele atributului și valoarea acestuia) ca mai jos:

**input[type=number]** {  
 background: gray;  
 }

Astfel regula se va aplica doar pentru câmpurile de introducere a numărului.

* **selectorul universal[[19]](#footnote-19)** – vizează toate elementele din pagină, aplicându-le simultan aceleași reguli.

Sintaxa: **\*** {  
 font-size: 20px;  
 }

**2.3. JavaScript**

JavaScript (JS) este un limbaj de programare orientat obiect, folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către browser. Limbajul este binecunoscut pentru folosirea sa în construirea site-urilor web, dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații.

JavaScript este un limbaj de scriptare (scripting language), reprzentând unul dintre cele trei limbi de bază utilizate pentru a dezvolta site-uri web. În timp ce HTML și CSS oferă structura și stilul unui site web, JavaScript permite adăugarea unor noi funcționalități și comportamente pe site-ul web, permițând vizitatorilor site-ului web să interacționeze cu conținutul în mai multe moduri. Astfel, folosit pentru a controla conținutul dinamic al site-ului web, JavaScript introduce caracteristici precum:

* grafică animată
* diapozitive foto
* sugestii de completare automată a textului
* forme interactive

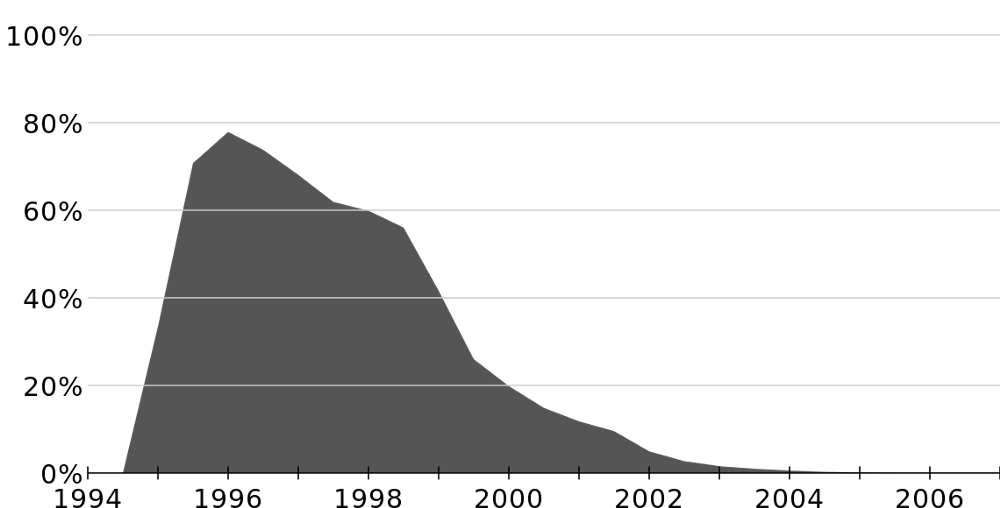
**2.3.1. Scurtă istorie[[20]](#footnote-20)**

Începutul și mijlocul anilor 1990 a fost o perioadă importantă pentru Internet. Jucătorii cheie precum Netscape și Microsoft se aflau în mijlocul „războiului browserelor”, astfel avem pe un flanc Navigator-ul Netscape (cu JavaScript), iar pe celălalt flanc, Internet Explorer-ul Microsoft (cu Jscript).

În septembrie 1995, un programator Netscape, pe nume Brandan Eich, a dezvoltat un nou limbaj de scriptare în doar 10 zile. Inițial a fost numit Mocha, dar a devenit rapid cunoscut sub numele de LiveScript și, mai târziu, JavaScript.

Netscape, pentru o perioadă, a făcut cel mai bun browser din lume și s-a bucurat de dominarea pieței.

Figura 12. Cota de piață Netscape Navigator

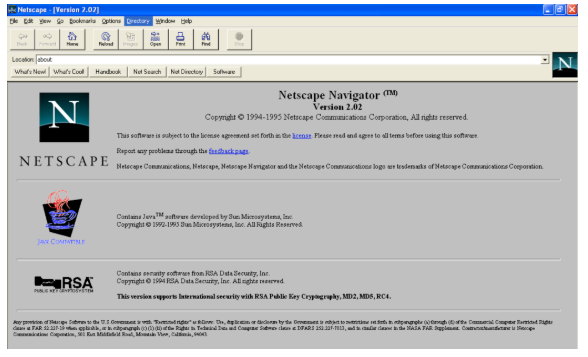


Sursă: <https://medium.com/@_bjma/lesson-1a-the-history-of-javascript-8c1ce3bffb17>

La sfârșitul anului 1995, proiectul Internet Explorer a fost demarat într-o încercare completă de combatere a controlului asupra platformei emergente din Netscape.

Până când lumea browserului s-a schimbat și Internet Explorer a devenit browserul dominant, JavaScript a evoluat în standardul aprobat pentru scrierea procesării interactive rulate într-un browser web.

Figura 13. Netscape Navigator 2



Sursă:<https://www.springboard.com/blog/history-of-javascript/>

**2.3.1.1.JavaScript vs. Java**

Există deseori o oarecare confuzie în legătură cu cele două noțiuni, însă JavaScript și Java nu au aproape nimic în comun. Numele JavaScript a provenit din suportul Netscape al applet-urilor Java din browserul său. Mulți spun că a fost, de asemenea, o tactică de marketing pentru a devia o oarecare atenție de la Java, care a fost cel mai zgomotos limbaj din acea vreme. Pentru a rula programe Java, codul trebuie mai întâi compilat într-o formă executabilă. Pe de altă parte, JavaScript a fost creat pentru a fi interpretat în timpul rulării, ceea ce îl face mult mai dinamic.

JavaScript nu a reușit să înceapă cel mai bine. Nu s-a comportat la fel de bine, iar cei care se dezvoltă în Java au considerat JavaScript mai mult ca un „lipici interfață-utilizator” pentru a fi folosit mai ales de designeri și alți non-ingineri. Însă realitatea este că a avea un limbaj „lipici” a permis Internetului să înflorească cu adevărat. Programatorii ar putea utiliza evenimente și pot compune componente interactive. Și datorită acestui fapt, JavaScript s-a răspândit ca un incendiu și a devenit foarte repede „lingua franca” a web-ului.

**2.3.1.2.Ajax: O nouă abordare a aplicațiilor web**

Anul 2005 s-a dovedit a fi unul mare pentru JavaScript. O lucrare lansată de Jesse James Garrett a prezentat Ajax, o suită revoluționară de tehnologii care includea JavaScript[[21]](#footnote-21). Ajax a îmbunătățit mult experiența utilizatorului, permițând paginilor web să se simtă mai mult ca niște aplicații desktop native. Aceasta a împins cu adevărat JavaScript în centrul atenției ca limbaj de programare profesional.

Această lucrare este considerată a fi una dintre elementele de bază fondatoare ale comunității JavaScript. La vremea respectivă, JavaScript a avut multe provocări, inclusiv natura sa verbală atunci când face lucruri simple și probleme de incompatibilitate între browsere. Comunitatea a răspuns cu cadre și biblioteci JavaScript mari și populare, cum ar fi Dojo și Mootools și jQuery.

Ajax reunește mai multe tehnologii, fiecare înflorind la propriu:

• prezentare bazată pe standarde folosind XHTML și CSS;

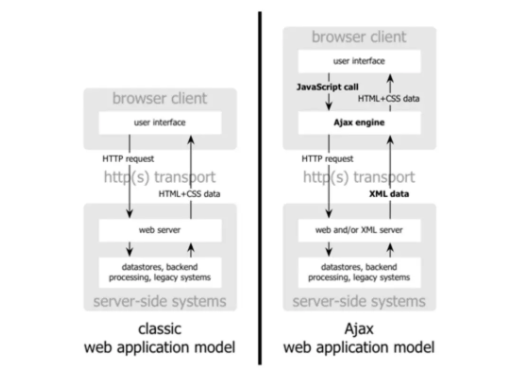
• afișare dinamică și interacțiune cu ajutorul Document Object Model;

• schimbul și manipularea datelor folosind XML și XSLT;

• recuperare asincronă de date utilizând XMLHttpRequest;

• JavaScript, având rolul de a leaga totul împreună.

Figura 14.   
Modelul tradițional pentru aplicații web (stânga) comparativ cu modelul Ajax (dreapta)

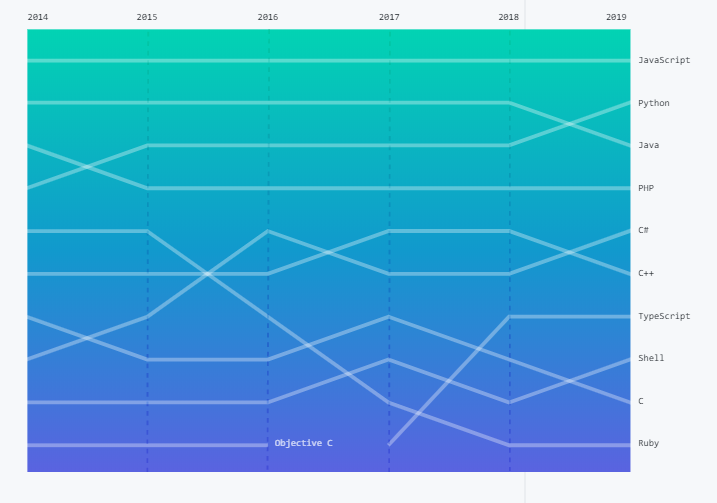


Sursă: <https://immagic.com/eLibrary/ARCHIVES/GENERAL/ADTVPATH/A050218G.pdf>

**2.3.1.3. Ascensiunea JavaScript**

De la începutul său ușor stângaci, JavaScript a ajuns să fie cel mai popular limbaj de programare din lume. Conform unui raport al octoverse.github.com, JavaScript este cel mai folosit limbaj de programare, din 2014 până în 2019. O serie de cadre și biblioteci JavaScript, cum ar fi Ember, Angular, React și Vue, au fost dezvoltate pentru a permite scrierea aplicațiilor web puternice și complicate folosind echipe mici în perioade scurte de timp. Alături de software-ul client și server, acum este posibil scrierea aplicațiilor mobile native folosind JavaScript. Nu este surprinzător, acest lucru devine din ce în ce mai popular datorită capacității de a partaja cod între lumile de telefonie mobilă și web.

Figura 15. Topul limbajelor de programare folosite în lume



Sursă: <https://octoverse.github.com/>

Ascensiunea JavaScript către cel mai popular limbaj de programare din lume este sinonimă cu creșterea Internet în sine.

Creat din necesitate, este utilizat pentru a construi 95,2% (1,52 miliarde) de site-uri web astăzi, inclusiv unele dintre cele mai mari din lume, cum ar fi Facebook și YouTube. Fără aceasta, nu am avea aplicații web populare și utile, cum ar fi Google Maps și eBay.

**2.3.2. Biblioteci și framework-uri JavaScript**

Cazurile de utilizare în care JavaScript a demonstrat că înlocuiește platformele tradiționale sunt în continuă creștere. Pe măsură ce domeniul de utilizare crește, devine extrem de important să se utilizeze o metodă de distribuție a codului, care nu numai că aduce simplitate în dezvoltarea ulterioară, ci permite și reutilizarea codului de către mase. Bibliotecile JavaScript rezolvă foarte bine această problemă și furnizează cod JavaScript pre-scris care face ușor de făcut sarcinile comune sau complexe. Bibliotecile JavaScript vizează adesea sarcini specifice, cum ar fi manipularea DOM, configurarea cadrului și gestionarea AJAX.

Diferența cheie între bibliotecile JavaScript și framework-uri este că bibliotecile constau din funcții pe care o aplicație le poate apela pentru a îndeplini o sarcină, în timp ce un framework definește modul în care un dezvoltator proiectează o aplicație. Cu alte cuvinte, framework-ul apelează la codul aplicației, mai degrabă decât invers. Natura acestor instrumente s-a schimbat în ultimii ani. În 2015, de exemplu, jQuery era cea mai bună bibliotecă JavaScript. Astăzi, potrivit JavaScript Scene, bibliotecile JavaScript tradiționale rămân importante, dar se confruntă cu o concurență sporită alcătuită din cadrele JavaScript, cum ar fi React, Angular, Node.js și Ember.js[[22]](#footnote-22).

**2.3.2.1. Biblioteci JavaScript**

* **D3.js**

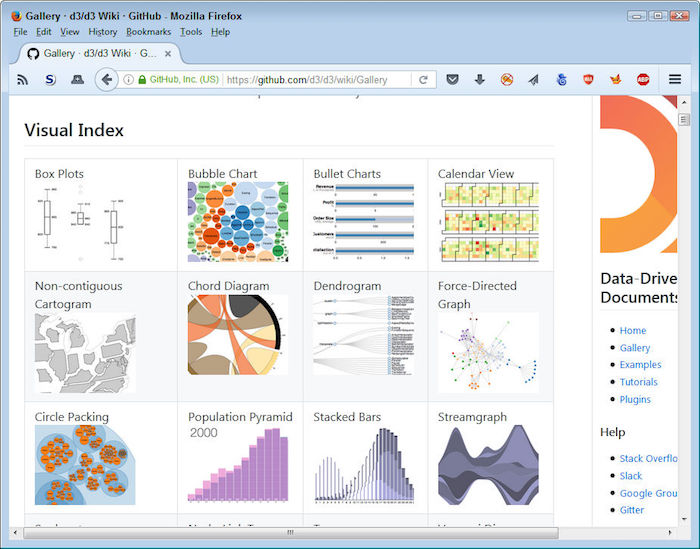
Multe site-uri web moderne sunt bazate pe date. Site-urile de știri, de exemplu, trebuie să-și reîmprospăteze conținutul în mod constant, ceea ce face imposibilă reconfigurarea la fiecare câteva secunde pentru a susține aceste modificări. Biblioteca D3.js este unică prin faptul că este axată pe date. Pe lângă flexibilitate extremă, prezentarea relativ simplificată a D3.js ajută la îmbunătățirea vitezei, eliminând o mulțime de bagaje pe care alte biblioteci le folosesc.

Principalul avantaj al acestei biblioteci este manipularea sa extrem de flexibilă a conținutului.

Pentru un site de comerț electronic însă, este posibil ca această bibliotecă să nu fie cea mai bună alegere, pentru a evita transformarea site-ului într-o prezentare de date și informații.

Imaginea de mai jos arată doar câteva dintre numeroasele prezentări de date care se găsesc în indexul vizual al D3.js.

Figura 16. Indexul vizual al D3.js



Sursă: <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/>

* **jQuery**

Biblioteca **jQuery** a fost concepută pentru a ușura și îmbunătăți procese precum traversarea arborelui DOM în HTML, managementul inter-browser al evenimentelor, animații și cereri tip AJAX. jQuery și-a câștigat statutul de conducător al paginilor web. Multe site-uri utilizează jQuery din trei motive esențiale:

- este ușor de învățat, oferind mai multe exemple decât majoritatea bibliotecilor

- este rapid

- are o bază uriașă instalată, bucurându-se de o cantitate semnificativă de sprijin comunitar și contribuitori la proiect.

jQuery se poate folosi pentru a rezolva următoarele probleme specifice programării web:

- selecții de elemente în arborele DOM, folosind propriul motor de selecții open source Sizzle, un proiect născut din jQuery

- parcurgerea și modificarea arborelui DOM (incluzând suport pentru selectori CSS3 și XPath simpli)

- înregistrarea și modificarea evenimentelor din browser

- manipularea elementelor CSS

- efecte și animații

- cereri tip AJAX

- extensii (plugin-uri)

* **jQuery UI**

jQuery UI este doar unul dintre numeroasele suplimente pentru jQuery, dar este cel mai des utilizat. Este folosit pentru a adăuga elemente grafice de bază în aplicație după ce este încorporat jQuery. Biblioteca jQuery UI vine cu o mulțime de widget-uri interesante, care fac posibilă configurarea aplicației pentru a oferi servicii utile.

Această bibliotecă iese în evidență din mai multe motive. Cel mai important, JQuery UI funcționează cu aproape orice browser.

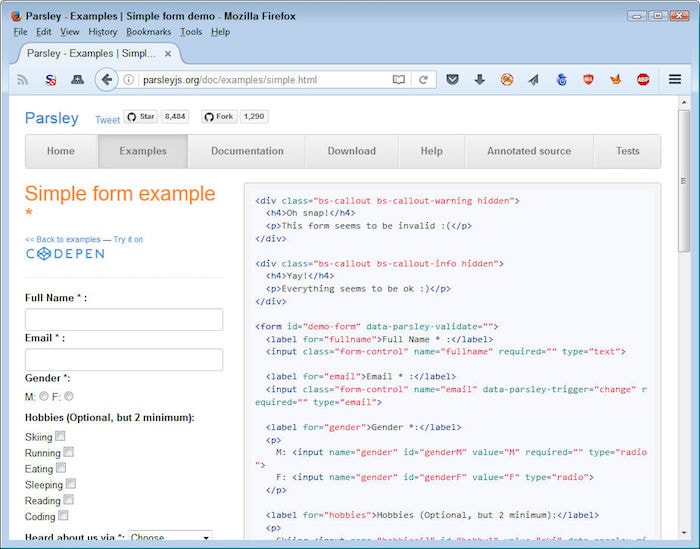
Problema cu grupul de biblioteci jQuery este că au devenit mari. Dimensiunea bibliotecilor JQuery le poate face să se încarce lent pe dispozitive mai mici. Bibliotecile terțe părți alternative, cum ar fi Granim.js și Multiple.js, au apărut pentru a oferi un subset de funcții jQuery, uneori cu un set focalizat de funcții suplimentare, cu timp de încărcare semnificativ mai rapid.

* **Parsley**

Validarea formularului într-o pagină web este o sarcină esențială. Deoarece datele sunt acum analizate de mașini, acuratețea acestora este mai esențială ca niciodată. Obținerea unor date „curate” necesită mult timp și nu este niciodată la fel de exactă ca obținerea unor informații corecte oferite de utilizator.

Parsley oferă mai multe tehnici de validare a formularului decât în ​​orice altă bibliotecă JavaScript. Chiar și validarea presupus simplă prezentată mai jos conține multe tipuri de validare diferite, împreună cu codul necesar pentru a efectua sarcina.

Figura 17. Exemplu de cod pentru mai multe tipuri de validare



Sursă: <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/>

* **QUnit**

Chiar dacă este utilizat un script, testarea manuală a aplicațiilor complexe, dacă poate găsi chiar toate erorile, consumă mult timp și în principiu, este obligatorie.

Funcționalitatea de testare oferită de multe biblioteci însă, lasă mult de dorit.

QUnit este una dintre multele biblioteci de testare a unităților disponibile pentru JavaScript. În programare, testarea unităților este o metodă de testare software prin care unitățile individuale de cod sursă, seturi de unul sau mai multe module de programe, împreună cu datele de control asociate, procedurile de utilizare și procedurile de operare, sunt testate pentru a determina dacă sunt potrivite pentru utilizare.

Scopul testării unităților este de a izola fiecare parte a programului și de a arăta care părți individuale sunt corecte și care nu.

* **React**

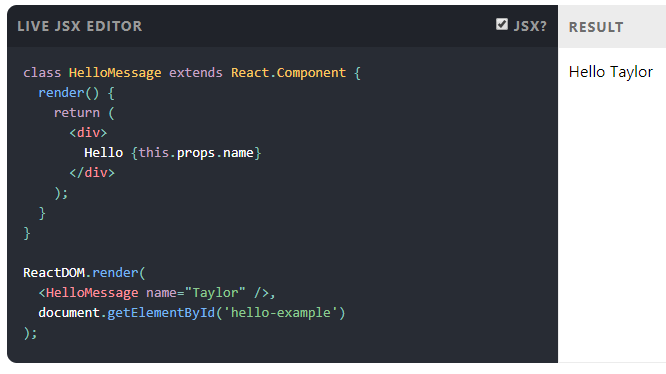
Considerată de obicei bibliotecă, React este uneori denumit cadru. În contextul abordării Model-View-Controller (MVC) pentru crearea de aplicații la scară largă, React oferă posibilitatea de vizualizare.

React nu se bazează pe DOM-ul (Document Object Model) browserului, ci folosește în schimb un DOM virtual care afișează rapid informațiile. Unul dintre cele mai bune lucruri despre React este că este declarativ, ceea ce înseamnă că îi spui cadrului ce vrei să faci, nu cum să faci.

Prețul plătit însă pentru viteza și capacitatea de utilizare a MVC, este un anumit grad de complexitate. Chiar și o componentă mică necesită un pic de cod (așa cum demonstrează exemplele de pe site-ul React). Se câștigă însă, flexibilitate și viteză atunci când proiectele sunt cu adevărat mari.

Componentele React implementează o metodă de randare care preia date de intrare și returnează ceea ce trebuie afișat. Exemplul de mai jos utilizează o sintaxă asemănătoare XML numită JSX. Datele de intrare care sunt transmise către componentă pot fi accesate de render() prin this.props[[23]](#footnote-23).

Figura 18. Exemplu cod React



Sursă: <https://reactjs.org/>

**2.3.2.2. Framework-uri JavaScript**

* **Angular**

Angular face posibilă extinderea HTML-ului, acționând ca porțiunea de control a MVC. Controlerele sunt comportamentul din spatele componentelor DOM. Angular face posibilă crearea de noi comportamente într-o manieră naturală și simplă. Extensiile sfârșesc prin a fi adaosuri la HTML, mai degrabă decât ceva fixat în loc. Motivele esențiale pentru utilizarea acestui produs sunt viteza și performanța și setul de instrumente pentru dezvoltarea aplicațiilor de care dispune. Este complet extensibil și funcționează bine cu alte biblioteci. Fiecare caracteristică poate fi modificată sau înlocuită pentru a se potrivi cu fluxul de lucru unic de dezvoltare și nevoile funcției.

Cu toate acestea, codificarea poate deveni complexă, iar acest cadru poate fi mai potrivit pentru întreprinderi decât startup-urile mici. Este important de reținut că Angular se bazează și pe TypeScript în loc de JavaScript pur, ceea ce crește curba de învățare, dar oferă scalabilitate.

Din fericire, Angular vine cu un tutorial extins care ar trebui să faciliteze utilizarea lui de către dezvoltatorii cu experiență.

* **Ember.js**

Companii cunoscute precum Microsoft, Netflix și LinkedIn folosesc Ember.js, deoarece utilizează modelul Model-View-View-Model (MVVM) și încorporează cele mai bune practici ca parte a cadrului. Cel mai bun dintre toate este scalabilitatea. Interesant este că, în timp ce Ember.js se concentrează pe dezvoltarea web, se poate utiliza și pentru a construi aplicații mobile și desktop - a fost folosit pentru a construi Apple Music.

Spre deosebire de multe alte cadre, utilizatorul poate folosi suita completă de instrumente bazate pe Ember pentru a crea un mediu de dezvoltare compatibil. CLI Ember (interfața liniei de comandă) oferă acces la o multitudine de instrumente care se pot utiliza folosind scripturi. Ember Data, de exemplu, oferă acces la date web utilizând maparea relațională cu obiecte (ORM). Ember Inspector, între timp, este un plugin pentru browserele Firefox și Chrome, concepute pentru a facilita depanarea.

* **Node.js**

Node.js este un mediu de rulare JavaScript asincron, bazat pe evenimente, cu unele caracteristici unice, care execută cod JavaScript în afara unui browser web.

Acest produs seamănă mai mult cu jQuery decât cu jQueryUI, prin faptul că nu este folosit pentru a crea interfețe-utilizator, ci pentru a crea cod care reacționează la evenimente - un client generează evenimente, iar serverul răspunde la acele evenimente.

Cu Node.js, dezvoltatorii pot construi:

- aplicații backend

- blog-uri

- sisteme de management al clienților

- servicii în timp real cum ar fi aplicații de chat și jocuri

- API-uri REST

- aplicații de rețele sociale

- utilități și instrumente

* **Vue.js**

Scalând între o bibliotecă și un cadru complet, Vue utilizează un model de dezvoltare bazat pe componente.

Funcțiile Key Vue includ componente, șabloane, tranziții și legarea datelor în două sensuri, dar poate cea mai distinctivă caracteristică este sistemul său de „reactivitate”, prin care actualizarea unui obiect JavaScript în Vue actualizează automat și discret șabloanele Vue.

Vue a fost creat pentru a fi utilizat de oricine, cu cunoștințe de lucru HTML, CSS și JavaScript[[24]](#footnote-24).

**3. Prezentarea site-ului**

**3.1. Aspecte generale ale proiectării site-ului**

Interfața grafică a unui sistem computerizat trebuie să ofere utilizatorului metafore de interacțiune, astfel încât conceptele folosite să transmită atât funcționalitate, cât și logică și înțeles semantic.

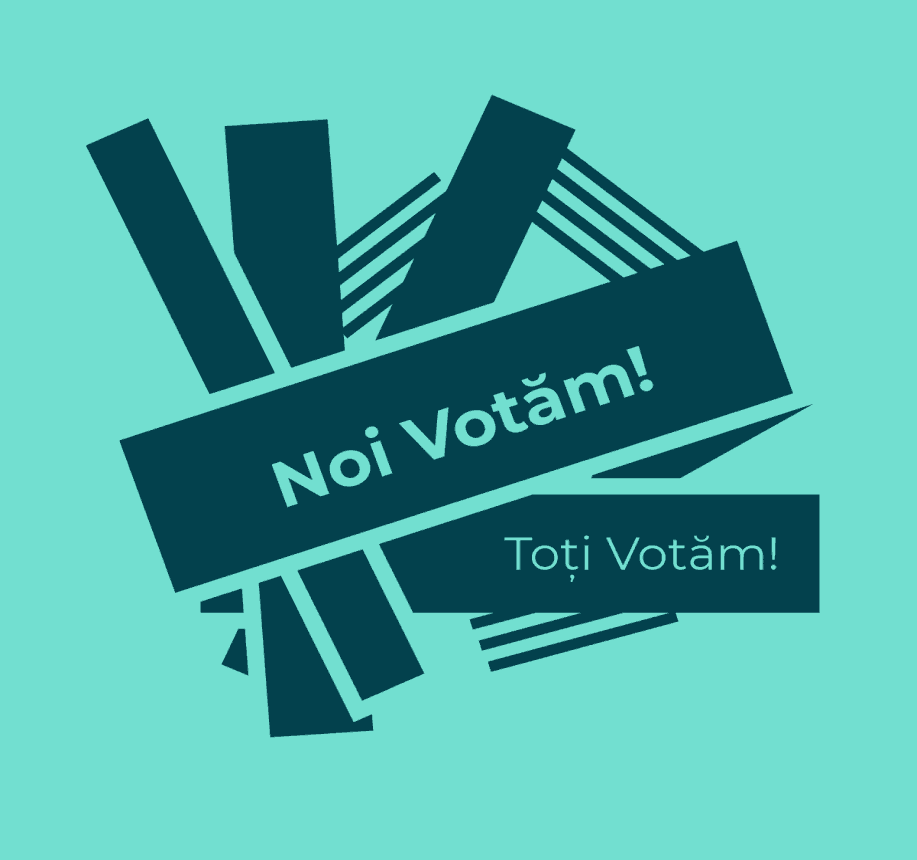
Designul și funcționalitatea site-ului sunt proiectate în jurul utilizatorului, astfel ca interacțiunea lui cu site-ul să fie una intuitivă, ușor de folosit și să existe o soluție vizibilă și ușor de detectat pentru fiecare acțiune dorită.

Navigarea prin site trebuie să poată asigura o privire de ansamblu asupra conținutului site-ului, iar acesta nu trebuie să fie unul complicat, utilizatorii noi, apreciând mai degrabă simplitatea în pofida unui site greu de utilizat. Am reușit să ofer utilizatorului această privire de ansamblu prin folosirea unor pictograme clare, a imaginilor sugestive și a unui meniu, simplu, clar, bine organizat.

Pentru a asigura posibilitatea întoarcerii la pagina principală, am folosit atât imaginea cu logo-ul site-ului, cît și componenta „Home” a meniului.

**3.1.1. Imagini și pictograme**

Figura 19. Logo-ul site-ului



Sursă: <https://hatchful.shopify.com/>

Figura 20. Componente care asigură întoarcerea la pagina principală



Sursă: Creație proprie

Figura 21. Alte imagini sugestive folosite

Sursă: <https://www.google.com/>

Cu ajutorul BootstrapCDN am realizat pictograme și butoane sugestive aferente diverselor acțiuni ale site-ului.

Bootstrap este un framework CSS gratuit și opend-source, direcționat către dezvoltarea web front-end receptivă și mobilă. Conține șabloane de design bazate pe CSS și JavaScript pentru tipografie, formulare, butoane, navigare și alte componente de interfață[[25]](#footnote-25).

Font Awesome este un set de instrumente pentru fonturi și pictograme bazat pe CSS și LESS. Acesta a fost realizat de Dave Gandy pentru utilizare cu Bootstrap, iar ulterior a fost încorporat în BootstrapCDN[[26]](#footnote-26).

Figura 22. Pictograme Bootstrap folosite



Sursă: Creație proprie

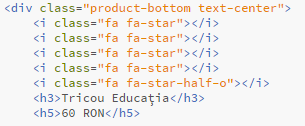
Pictogramele Bootstrap sunt proiectate pentru a lucra cu componente Bootstrap, de la formulare la navigare. Pictogramele Bootstrap sunt SVG-uri, astfel încât se scalează rapid și ușor și pot fi stilizate cu CSS. Deși sunt construite pentru Bootstrap, vor funcționa în orice proiect[[27]](#footnote-27).

Pentru a folosi aceste elemente, am definit legăturile cu scripturile, astfel:



De asemenea, pentru fiecare element folosit, am definit câte o clasă, pe care am personalizat-o în foaia de stiluri CSS.

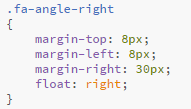
Astfel, pentru pictograma aferentă gradului de popularitate al produsului am folosit câte o clasă, prin codul următor:



Un alt exemplu este folosirea pictogramei care împarte un meniu într-un submeniu, definită prin clasa „fa fa-angle-right” astfel:



Personalizarea stilului am făcut-o prin următoarele atribute: aliniere margine de sus la 8 pixeli, aliniere margine stînga la 8 pixeli și aliniere margine dreapta la 30 pixeli.



**3.1.2. Meniul lateral**

În ceea ce privește design-ul site-ului, din punct de vedere al poziționării meniului principal, am ales o navigare laterală, care să faciliteze accesarea paginilor componente.

Acest tip de poziționare a meniului conferă o valoare estetică, un plus de transparență a site-ului prin afișarea tuturor elementelor, dar și un grad ridicat de vizibilitate.

Figura 23. Meniul lateral



Sursă: Creație proprie

În containerul de mai jos am folosit atributul „href” pentru a crea legături cu celelalte pagini ale site-ului (componente ale meniului) și tag-urile <ul> și <li> pentru a ordona meniul sub forma unei liste.



În foaia de stiluri CSS, proprietățile clasei „side-menu”, cea care tranduce aceste propretăți în elemente vizuale, setează lățimea și înălțimea containerului, mărimea fontului, culoarea de fundal, culoarea fontului.



**3.1.3. Subsolul site-ului**

Subsolul unui site are un rol foarte important, deoarece în majoritatea cazurilor, ar trebui să conțină informații de ordin juridic, mai ales în cazul celor de comerț electronic.

Pe lângă acestea, aici putem include datele de contact, astfel încât să fie vizibile, la îndemăna utilizatorului. Eu am ales sa afișez în partea de subsol a site-ului o secțiune cu informațiile legate de politica de confidențialitate, termenii și condițiile de utilizare și polita de retur, dar și o secțiune cu datele de contact și canalele de socializare, unde comunitatea este vizibilă.

Figura 24. Subsolul (footer-ul)



Sursă: Creație proprie

La accesarea link-ului despre politica de confidențialitate, utilizatorul va găsi detalii despre informațiile pe care site-ul le colectează și modul de păstrare și utilizare al acestora, în conformitate cu legea în viguare.

Secțiunea „Termeni și condiții” descrie condițiile pe care utilizatorul trebuie să le îndeplinească pentru vizitarea site-ului și acordurile oferite la accesarea acestuia.

De asemenea, următoarea opțiune, descrie procesul de returnare al produselor achiziționate, informații despre condițiile pe care produsul trebuie să le întrunească și perioda de retur.

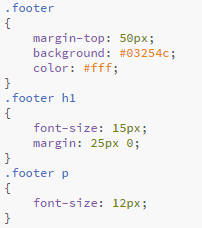
Și nu în ultimul rând, secțiunea dedicată rețelelor de socializare, permite utilizatorului accesarea paginilor de Facebook și Instagram ale comunității.

Toate aceste funcționalități au fost implementate cu ajutorul elementului <a> și a atributului său „href” , care realizează legăturile cu fișierele pdf, în care am definit politica de confidențialitate, de retur și de utilizare a site-ului, respectiv paginile către canalele de socializare.

Figura 25. Codul HTML folosit la crearea footer-ului



Proprietățile au fost definite pentru a seta culoarea de fundal, culoarea fontului, pentru a seta poziționarea, iar pentru elementele conținute de tag-urile <h1> și <p> se stabilește mărimea fontului.



**3.2. Paginile de nivel doi de navigare**

Paginile de nivel doi de navigare sunt paginile adăugate ca sub-pagini în meniu.

După cum se poate observa în figura 23, meniul pe care l-am dezvoltat, cuprinde 6 pagini de nivel doi de navigare, deoarece la secțiunea „Magazin”, am creat un sub-meniu de tip listă, care să împartă produsele în două categorii: pentru femei și pentru bărbați.

Așa cum sugerează și terminologia folosită în meniu, pagina „Despre proiect”, conține o prezentare a ceea ce doresc să transmit ca mesaj prin existența site-ului, a informațiilor prezentate în secțiunea „Despre vot” și bineînțeles, prin comercializarea tricourilor.

Pagina „Despre vot” este tot de tip informațională și în ea se regăsesc „curiozități” legate de viața politică, menite să tragă un sistem de alarmă cu privire la neparticiparea la vot a cetățenilor.

**3.2.1. Magazinul**

Magazinul este format din două categorii, fiecare având listate produsele disponibile prin imagini, denumiri, gradul de popularitate și preț.

Figura 26. Conținutul sub-meniului „Femei”

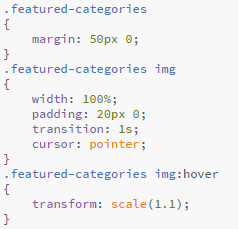


Sursă: Creație proprie

Pentru realizarea acestui conținut am folosit o secțiune cu trei containere, iar in fiecare dintre acestea, am setat imaginea prin tag-ul <img>, gradul de popularitate prin pictograme Bootstrap și textul prin tag-urile <h3>, respectiv <h5>.

Secțiunea este stilizată în CSS prin proprietatea „margin”, setând distanța la 50 de pixeli față de elementul anterior, iar proprietatea „transition” setează un efect de tranziție la o secundă atunci când cursorul se află pe imaginea produsului.

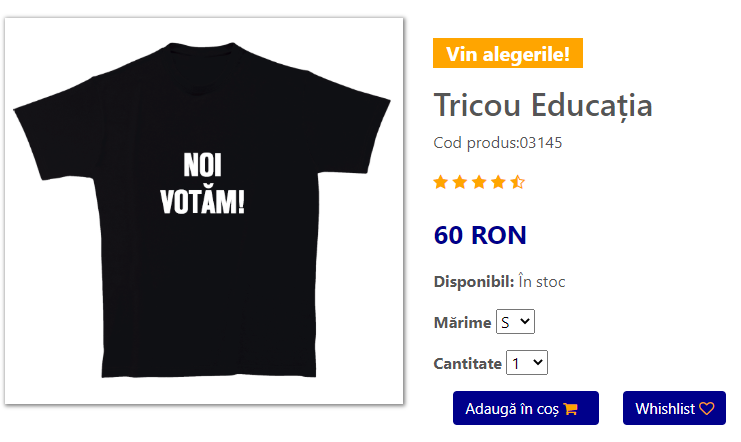
Efectul generat este descris de proprietatea „transform” prin metoda de transformare 2D scale(), care mărește dimensiunea elementului.



**3.2.2. Pagina de produs**

Fiind un site de comerț electronic, pagina de produs prezintă un grad ridicat de vizibilitate și accesabilitate. Pentru a crește acest grad, am introdus atât în pagina de start o secțiune cu produsele disponibile, cât și secțiuni cu produse similare în pagina de produs.

Figura 27. Parte din conținutul paginii de produs



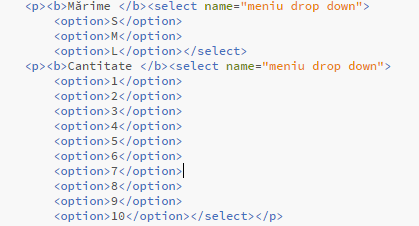
Sursă: Creație proprie

Prezentarea grafică a produsului este realizată cu ajutorul clasei „carousel slide carousel-fade” – instrument de prezentare de diapozitive, construit cu transformări CSS 3D și JavaScript.



Atributul „data-ride = carousel” este folosit pentru a seta momentul de start al animației imediat la încărcarea paginii HTML, iar „d-block w-100” împiedică alinierea implicită a browser-ului. Prezentarea conține și butoane de control, implementate prin clasele „carousel-control-prev”, pentru înapoi, respectiv „carousel-control-next” pentru înainte.

Pentru a selecta mărimea și cantitatea am folosit butoane de tip radio.

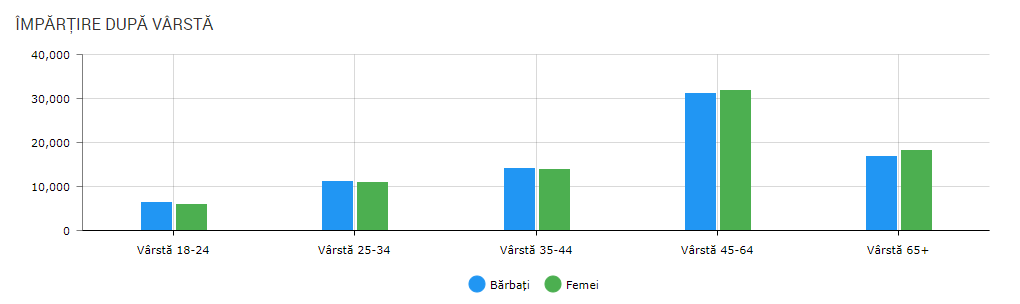


**4. Concluzii**

Am realizat proiectarea și implementarea site-ului ca urmare a unei probleme sociale a societății românești, din dorința de a promova spiritul civic, mai ales în rândul tinerilor.

Conform statisticilor BEC ( Biroul Electoral Central ), categoria de vârstă cu cea mai mică rată de participare la vot este 18 -24 ani.

Figura 28. Prezență la alegerile prezidențiale, noiembrie 2019, turul 1



Sursă: <https://prezenta.bec.ro/prezidentiale10112019/romania-stats>

Am considerat că prin dezvoltarea acestui proiect voi încuraja această categorie de vârstă să fie mai implicată în viața politică și să învețe să se documenteze asupra acestui subiect.

Reprezentând totodată și un business, am proiectat site-ul în conformitate cu reguli și politici ale site-urilor de comerț electronic, urmărind mai multe puncte cheie în dezvoltarea lui: simplitate, coerență, alegerea culorilor în funcție de publicul țintă prin studierea psihologiei culorilor. Astfel, am ales un meniu simplu, care să nu determine abandonarea site-ului, iar utilizatorul să aibe variante de continuare a acțiunii pe care dorește să o facă la pasul următor, dând astfel, coerență. În ceea ce privește alegerea culorilor și având în vedere tema site-ului, am ales culoarea dominantă albastru pentru a inspira încredere, fiabilitate, calitate.

De asemenea, pentru a inspira dezvoltarea simțului civic și a naționalismului, am hotărât să folosesc pentru produse, culorile tricolorului.

Consider, de asemenea, că un alt punct cheie în lansarea unui magazin online, îl reprezintă încrederea pe care acesta o inspiră utilizatorilor prin politicile de confidențialitate, acordarea unui grad ridicat de atenție asupra datelor personale și diversitatea termenilor și condițiilor. Având în vedere creșterea utilizatorilor de smartphone-uri din ultimii ani, a fost foarte important pentru mine să realizez un site care să fie optimizat pentru mobil. Pentru a ajuta acest proces, am utilizat butoane „call-to-action” vizibile, eliminarea pop-up-urilor și evitarea blocurilor mari de text, pe lângă componenta tehnică folosită.

Tehnologiile pe care le-am folosit majoritar la dezvoltarea site-ului sunt HTML și CSS, iar pentru editarea codului sursă am folosit Brackets, un software gratuit și open-source licențiat de MIT (Massachusetts Institute of Technology). Fișierul sursă este index.html, iar foaia de stil, style.css.

Elementele principale ale site-ului au fost create prin div-uri, având aplicate clase, astfel:

* Header-ul, prin div-ul cu id-ul "top-nav-bar"
* Meniul lateral, prin div-ul cu id-ul "side-menu"
* Footer-ul, prin section-ul "footer"
* Prezentarea și descrierea produsului, prin section-ul "single-product", respectiv "product-description"

În concluzie, consider că site-ul pe care l-am creat întrunește toate condițiile pentru a ajunge o platformă cu impact social prin tema aleasă și o soluție de comerț online, care se poate extinde pe mai multe ramuri.

**5. Bibliografie**

1. **Sfetcu**, Nicolae, *Proiectarea, dezvoltarea și întreținerea siturilor web*, MultiMedia Publishing, 2014
2. **Shenoy**, Aravind, *Thinking in HTML*, Packt Publishing, 2014
3. <https://snipcart.com/ecommerce-website-design>
4. <https://www.webdesignerdepot.com/2014/06/the-ultimate-guide-to-designing-ecommerce-websites/>
5. <https://www.printful.com/blog/color-psychology-in-ecommerce-and-branding>
6. <https://www.quicksprout.com/how-colors-affect-conversion-rate/>
7. <https://tatedesign.net/does-age-affect-color-preferences/>
8. <https://www.colormatters.com/color-and-design/basic-color-theory>
9. <https://www.weforum.org/agenda/2019/03/five-ways-to-restore-trust-in-e-commerce/>
10. <https://www.clarity-ventures.com/articles/the-importance-of-transparency-in-ecommerce-clarity>
11. <https://www.fluid-digital.com/blog/why-transparency-is-crucial-for-e-commerce-success/1162/>
12. <https://www.quicksprout.com/mobile-friendly-website-guide/>
13. <https://html.spec.whatwg.org/multipage/common-microsyntaxes.html>
14. <https://tutorial.techaltum.com/attribute.html>
15. <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>
16. <https://tutorialehtml.com/ro/tutoriale-css-introducere-in-css/>
17. <https://ro.wikipedia.org/wiki/Cascading_Style_Sheets>
18. <https://levelup.gitconnected.com/introduction-to-css-selectors-3993ff9b5f92>
19. <https://web.ceiti.md/lesson.php?id=2>
20. <https://immagic.com/eLibrary/ARCHIVES/GENERAL/ADTVPATH/A050218G.pdf>
21. <https://www.springboard.com/blog/history-of-javascript/>
22. <https://medium.com/@benastontweet/lesson-1a-the-history-of-javascript-8c1ce3bffb17>
23. <https://ro.wikipedia.org/wiki/JavaScript>
24. <https://www.checkmarx.com/blog/javascript-history-infographic/>
25. <https://webplatform.github.io/docs/concepts/programming/javascript/libraries/>
26. <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/>
27. <https://reactjs.org/>

1. <https://www.quicksprout.com/how-colors-affect-conversion-rate/> [↑](#footnote-ref-1)
2. <https://www.colormatters.com/color-and-design/basic-color-theory> [↑](#footnote-ref-2)
3. <https://www.printful.com/blog/color-psychology-in-ecommerce-and-branding/> [↑](#footnote-ref-3)
4. <https://www.weforum.org/agenda/2019/03/five-ways-to-restore-trust-in-e-commerce/> [↑](#footnote-ref-4)
5. <https://www.clarity-ventures.com/articles/the-importance-of-transparency-in-ecommerce-clarity> [↑](#footnote-ref-5)
6. <https://snipcart.com/ecommerce-website-design> [↑](#footnote-ref-6)
7. Aravind Shenoy, *Thinking in HTML,* Packt Publishing, 2014, pag. 15 [↑](#footnote-ref-7)
8. <https://ro.wikipedia.org/wiki/HyperText_Markup_Language> [↑](#footnote-ref-8)
9. <https://www.w3schools.com/TAGS/default.ASP> [↑](#footnote-ref-9)
10. <https://www.w3schools.com/TAGS/default.ASP> [↑](#footnote-ref-10)
11. <https://www.w3schools.com/TAGS/default.ASP> [↑](#footnote-ref-11)
12. Nicolae Sfetcu, *Proiectarea, dezvoltarea și întreținerea siturilor web*, 2014, pag. 28 [↑](#footnote-ref-12)
13. <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes> [↑](#footnote-ref-13)
14. <https://ro.wikipedia.org/wiki/HTML5> [↑](#footnote-ref-14)
15. Nicolae Sfetcu, *Proiectarea, dezvoltarea și întreținerea siturilor web*, 2014, pag. 37-39 [↑](#footnote-ref-15)
16. Nicolae Sfetcu, *Proiectarea, dezvoltarea și întreținerea siturilor web*, 2014, pag. 42-43 [↑](#footnote-ref-16)
17. <https://tutorialehtml.com/ro/tutoriale-css-introducere-in-css/> [↑](#footnote-ref-17)
18. <https://web.ceiti.md/lesson.php?id=2> [↑](#footnote-ref-18)
19. <https://levelup.gitconnected.com/introduction-to-css-selectors-3993ff9b5f92> [↑](#footnote-ref-19)
20. <https://www.checkmarx.com/blog/javascript-history-infographic/> [↑](#footnote-ref-20)
21. <https://www.springboard.com/blog/history-of-javascript/> [↑](#footnote-ref-21)
22. <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/> [↑](#footnote-ref-22)
23. <https://reactjs.org/> [↑](#footnote-ref-23)
24. <https://blog.newrelic.com/engineering/best-javascript-libraries-frameworks/> [↑](#footnote-ref-24)
25. <https://ro.wikipedia.org/wiki/Bootstrap> [↑](#footnote-ref-25)
26. <https://en.wikipedia.org/wiki/Font_Awesome> [↑](#footnote-ref-26)
27. <https://getbootstrap.com/> [↑](#footnote-ref-27)