

UNIVERSITATEA DIN BUCUREȘTI



**FACULTATEA
DE
MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA CALCULATOARE ȘI TEHNOLOGIA
INFORMATIEI**

Proiect de diplomă

Aplicație web de vânzare produse pentru casă

Absolvent

Student Oblu Ionuț Adrian

Coordonator științific

Lect. dr. Carmen Chiriță

București, septembrie 2020

Rezumatul lucrării

În vederea susținerii licenței am elaborat lucrarea cu titlul „Aplicație web de vânzare produse pentru casă”. Obiectivul acestui proiect este de a implementa o aplicație web, magazin ce cuprinde produse folosite cu precădere pentru îmbunătățirea casei. Motivația pentru acest proiect vine din dorința mea de a învăța din ce în ce mai mult despre domeniul de proiectare Javascript, baza de date din fișier(ul) json cu serverul json-server, proiectarea site-urilor web și popularitatea lor din ce în ce mai mare prin prezentarea acestei lucrări.

În cadrul unei aplicații există cuvântul „design” web, care poate avea mai multe semnificații. Poate cea mai cunoscută dintre acestea se referă la designul de interfață vizuală și de utilizator. Inițialele “UI ”ale unui site web provin din limba engleză de la cele două cuvinte “user interface”. Acest aspect este important pentru că, cel mai adesea, aspectul unui site web și ușurința cu care poate fi manipulat atrag atenția persoanei care îl vizualizează înaintea tehnologiilor folosite în frontend și a modului în care sunt operate datele pe serverul web. Mulți programatori fără experiență subestimează importanța modului în care este implementată componenta invizibilă a aplicației, formată din cod și baza de date. Constituentul vizual al unei aplicații determină inițial interesul vizitatorilor, dar funcționalitatea acesteia îi îndeamnă să revină de fiecare dată.

O aplicație poate fi implementată foarte repede pe baza anumitor cerințe inițiale. Dacă ideea de ansamblu nu este corect concepută, poate deveni dificil, dacă nu imposibil, de modificat și început de la zero. Performanța este unul dintre principalele motive pentru care utilizatorii sunt atrași de aplicație.

Această aplicație, numită „Aplicație web de vânzare produse pentru casă” este realizată folosind HTML, JAVASCRIPT, CSS. Acest proiect acoperă următoarele implementări:

1) O pagină principală cu informații și promoții

Lucrarea începe cu un meniu ce cuprinde: catalogul de produse, promoțiile magazinului online, secțiunea pentru solicitarea sprijinului specializat, un blog cu mai multe detalii despre produse și coșul de cumpărături.

În corpul acestei pagini este afișat un set de imagini în continuă derulare. În plus, există o secțiune de testimonial în care clienții și deținătorul magazinului pot formula păreri despre produse.

2) Coșul de cumpărături

Este implementat un coș de cumpărături personalizat, care stochează datele sale în baza de date locală cu ajutorul serverului json-server. De asemenea, este creat un coș care arată la fiecare adăugare a unui obiect numărul lor.

3) Administrarea produselor

Administrarea produselor se face direct din baza de date, adăugând noi câmpuri pentru produsele din acest magazin online.

Acest proiect este o experiență interesantă. Am înțeles mai bine ce presupune lucrul la un proiect ambițios, deși acesta este realizat la nivel local. Am mai lucrat în echipe când am făcut proiecte mari și nu am crezut că voi termina niciodată un proiect ambițios precum acesta, am învățat cum să-mi asum anumite sarcini sub presiune. Dezvoltarea aplicației software încurajează o bună organizare. Documentele produse mi-au oferit o perspectivă asupra necesității unei documentări mai exacte. De asemenea, în ceea ce privește proiectul aplicației web a fost o oportunitate excelentă și mă va ajuta în proiectele viitoare.

CUPRINS

I. Introducere	5
I.1 Tipul lucrării și subdomeniul specific în care se încadrează tema	5
I.2 Prezentarea generală a temei.....	5
I.3 Scopul și motivația alegerii temei.....	8
II. Preliminarii.....	9
II.1 Contribuția proprie în realizarea lucrării (pe scurt)	9
II.1.1 Aplicația web	10
II.2 Structura lucrării, cu descrierea succintă a fiecărui capitol	11
II.2.1 Programarea funcțională	11
II.2.2 Arhitectura	12
II.2.3 Dezvoltare	13
II.2.4 Folosirea protocoalelor HTML	17
II.3 Câteva repere istorice relativ la temă și rezultate cunoscute	18
II.4 Noțiuni științifice sau tehnologice care stau la baza temei	19
II.5 Obiectivele lucrării prin raportate la context.	25
III. Contribuția propriu-zisă a candidatului.....	26
III.1 Alegerea tehnologiei	26
III.2 Funcționalități.....	27
III.3 Aplicația în practică	30
IV. Concluzii.....	35
IV.1 Concluzii cu referire la modul de realizare a temei	35
IV.2 Posibile dezvoltări ulterioare ale temei	36
IV.3 Aprecierile personale privind relevanța rezultatelor obținute	36
IV.4 Eventuale aplicații practice sau interferențe cu alte domenii	37
Bibliografie	38
ANEXE.....	39
Dicționar de termeni și acronime	39

I. Introducere

I.1 Tipul lucrării și subdomeniul specific în care se încadrează tema

Comerțul electronic cuprinde orice tranzacție comercială care implică transferul informațiilor pe internet. Aplicația caută să descopere aspecte legate de:

- prețurile atractive pentru clienți.
- îmbunătățirea aspectului casei în interior cât și pe exterior.
- produse noi apărute pe piață pentru utilizatori interesați de achiziționare.

La aflarea informațiilor utilizatorii se pot ghida pe 2 dintre căile următoare:

- alegerea magazinului online .
- alegerea produsului în funcție de marcă.

Aplicația poate fi extinsă prin deschiderea unor magazine comerciale în vederea îmbunătățirii serviciilor oferite, a extinderii magazinului pe piață, adăugarea mai multor produse din categoria pe care o vizează clienții ș.a.

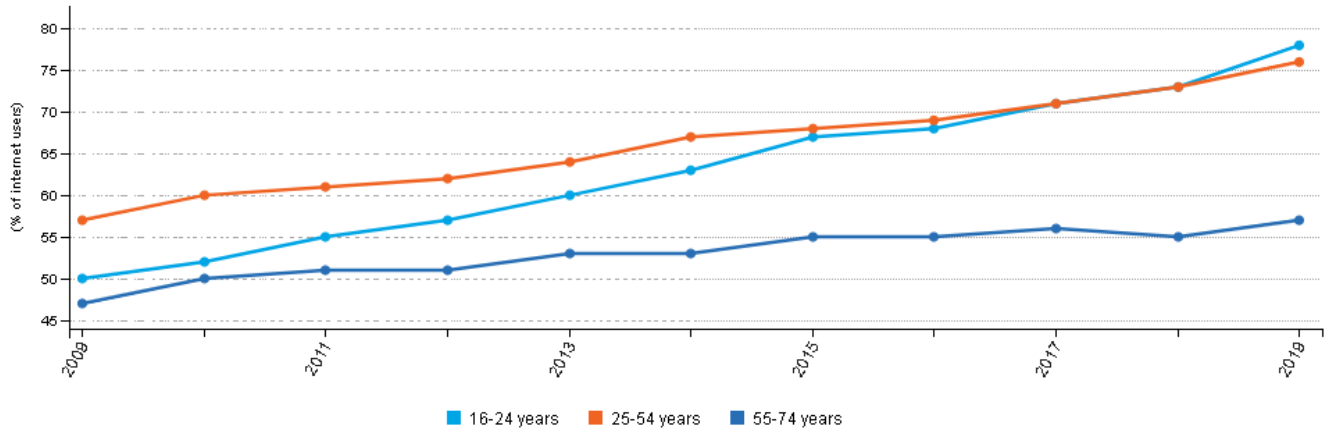
I.2 Prezentarea generală a temei

Deși există multe modalități de a vinde consumatorilor pe internet, principalul avantaj al comerțului online este că realizarea actului comercial nu este condiționată de spațiu și timp. Este irelevant dacă stau acasă pe canapea, la prânzul lor la serviciu sau chiar folosesc aplicația într-un magazin fizic, oriunde pot finaliza o achiziție online.

De aceea, nu există multe date publice exacte despre vânzările online. În urma cercetării îndelungate, am găsit următoarele:

→ pe site-ul EUROSTAT ce cuprinde informații despre achizițiile online, în europa, în perioada 2009-2019 în funcție de categoria de vârstă (<https://ec.europa.eu/>). Se constată că 71%dintre utilizatorii cu acces la internet au cumpărat online în 2019.

Internet users who bought or ordered goods or services for private use in the previous 12 months, by age group, EU-28, 2009-2019



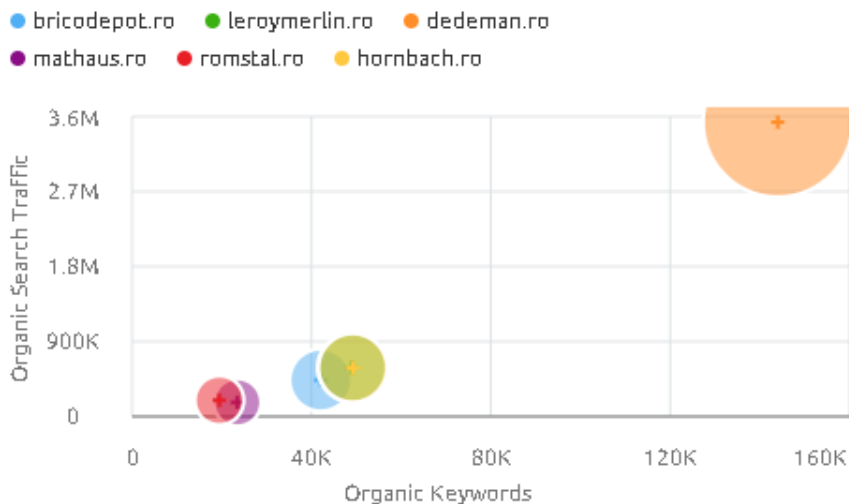
Source: Eurostat (online data code: isoc_ec_ibuy)

eurostat





(Figura I.2.1; Titlu: Grafic situație; Utilizatorii de internet ce au cumpărat bunuri de pe internet)

→ site-ul (<https://www.semrush.com/>) prezintă date despre accesările magazinelor online și nu numai cu o comparație între numărul persoanelor care îl vizitează. Se observă o creștere a vânzărilor pe un site accesat mai des.

Competitive Positioning Map



(Figura I.2.2; Titlu:Comparație magazine; Comparația unor magazine online din România conform semrush.com)

Domain	Com. Level 	Common Keywords 	SE Keywords	Traffic	Costs (USD)	Paid Keywords
hornbach.ro 	<div><div></div></div> 45%	17.8K	49.3K	583.2K	61.3K	2
leroymerlin.ro 	<div><div></div></div> 35%	12.5K	49K	586.6K	37.6K	407
bricodepot.ro 	<div><div></div></div> 29%	11.2K	42.1K	437.3K	41.8K	184
compari.ro 	<div><div></div></div> 19%	15K	169.3K	1.3M	121.4K	2.6K
okazii.ro 	<div><div></div></div> 19%	19K	297.5K	986.5K	114.3K	177
cel.ro 	<div><div></div></div> 18%	13.5K	169.3K	711.7K	70.2K	879
emag.ro 	<div><div></div></div> 16%	41.2K	726.6K	20.8M	2.7M	2.4K
shopmania.ro 	<div><div></div></div> 16%	9.5K	109.4K	418.4K	45.3K	24
altex.ro 	<div><div></div></div> 16%	10.1K	121.9K	3M	204.3K	2.4K
carrefour.ro 	<div><div></div></div> 15%	9K	102.8K	835.1K	156.8K	239

(Figura I.2.3; Titlu: Datele din România; Situația site-urilor web de vânzări pentru casă în funcție de trafic conform semrush.com)

→ știri de actualitate conform căutărilor în articole online – se concentrează mai mult pe veniturile magazinelor online cu o cifră de afaceri rotunjită decât pe date concrete despre situația vânzărilor online în România.

În aceste condiții concurența pe piață este acerbă. Cei ce s-au distanțat în competiție vor avea monopol pe piață. Acest monopol se datorează tendinței oamenilor de a cumpăra fiindcă cunosc brandul, în ciuda prețurilor mai ridicate. Astfel, apare nevoia unei aplicații unde prețurile sunt reduse, pentru a intra pe piața competitivă și a întreprinde un nou brand de produse.

→ aplicația își propune să mulțumească toți clienții indiferent de categoria de vârstă. Pentru acest lucru, este pusă la dispoziție și o gamă largă de produse din care ei pot face achiziția în meniul aplicației, la secțiunea de produse.

I.3 Scopul și motivația alegerii temei

Prin intermediul acestei lucrări de licență, urmăresc să dovedesc că sunt capabil să gestionez o aplicație software.

Această lucrare sprijină deprinderea unor abilități prin efectuarea sarcinilor precum :

- Prelucrarea și stocarea datelor:
 - Crearea unei baze de date.
 - Extragerea datelor din cadrul aplicației.
 - Prelucrarea și salvarea datelor.

Mai mult decât atât, am ales să folosesc tehnologia JavaScript deoarece am mai lucrat cu aceasta. Am căutat o multitudine de soluții pentru dezvoltarea unei astfel de aplicații care să conducă la ceea ce doream. Paginile de teorie pe care le-am întocmit vor arăta motivul pentru care am făcut această alegere.

NodeJS este o soluție intrucât cuprinde atât dezvoltarea serverului, cât și ceea ce accesează clienții. Prin reținerea simultană a backend-ului și frontend-ului unei aplicații, JavaScript contribuie la sincronizarea datelor în timp real. Tehnologia NodeJS totodată oferă dezvoltatorilor oportunitatea de a construi aplicații mai interactive. Instalarea unui IDE (un mediu de rulare) nu este necesară în scopul rulării codului sursă JavaScript, deoarece aproape fiecare browser și sistem îl poate compila direct și asta facilitează interacțiunea cu orice utilizator.

Accesul rapid la produse și funcția de căutare ușor de folosit sunt câteva constituențe importante ale aplicației web de comerț electronic. NodeJS reprezintă o platformă bine echipată pentru gestionarea eficientă a produselor din pagină. Datorită bazei de date locale json-server ce poate fi utilizată, diferite produse pot fi găsite cu ușurință. Magazinele de comerț electronic construite cu NodeJS pot livra cele mai sofisticate pagini de produse fiindcă dispun de un design surprinzător și ușor de accesat.

În concluzie, NodeJS este în prezent cea mai potrivită tehnologie pentru dezvoltarea web de comerț electronic. Datorită ușurinței cu care poate fi utilizat, este preferat de multe startup-uri care vor să intre pe piața comerțului electronic.

II. Preliminarii

II.1 Contribuția proprie în realizarea lucrării (pe scurt)

Aplicația web: un program ce rulează printr-o arhitectură client-server, pagini web cu un conținut oarecare de exemplu texte, grafice cu informații despre produsele căutate.

Utilizatorul trebuie să efectueze o comandă, apoi ea este încărcată în fișierul json.

II.1.1 Aplicația web

Această aplicație este alcătuită din:

- O bază cu date, locală unde sunt salvate date despre obiecte si comenzi.
- Platforma pe care sunt afișate diverse informații pe baza interacțiunii utilizatorului cu aplicația. Permite gestionarea produselor în coșul de cumpărături, contactul cu un reprezentant al site-ului în aflarea de informații noi și conectarea directă cu alte interfețe (spre exemplu alte pagini ale aplicației)

Tehnologiile folosite:

- O bază cu date: *locală, RESTful⁽¹⁾ database cu json-server.*
- O platformă web: aplicație folosind *Javascript.*

¹ RESTful definește protocoalele HTTP(GET, PUT, POST, DELETE) și prin HTTP se indică acțiunile dorite care trebuie efectuate pentru o anumită resursă.

² Express.js (open-source și gratuit) este conceput pentru crearea aplicațiilor web și API.

II.2 Structura lucrării, cu descrierea succintă a fiecărui capitol

În continuare, în capitolele acestei lucrări sunt prezentate următoarele noțiuni: programarea funcțională, arhitectura aplicației, procesul de dezvoltare și descrierea protocoalelor HTTP. În arhitectura aplicației sunt expuse moduri variate de construire a acesteia și interacțiunea dintre ele cu detalierea căilor din backend direct implicate în interchimbarea datelor dintre server și client. Capitolul privind procesul de dezvoltare include metodele de programare folosite și exemple ale unor linii de cod explicate. În cadrul capitolului de descriere al protocoalelor HTTP voi ilustra întrebuințarea și importanța tipurilor de URLuri folosite pentru accesul la resurse.

II.2.1 Programarea funcțională

Gradul de dezvoltare al aplicațiilor enterprise este direct proporțional cu gradul de complexitate al acestora, iar pentru a adăuga noi funcții în sistem, sunt necesare plugin-uri sau programe suplimentare. Apariția unei disfuncționalități în cadrul acestei aplicații este foarte dificil de înțeles și de remediat. Complexitatea programelor cu care se lucrează trebuie să respecte nivelul de cunoștințe propriu.

Programarea funcțională (numită și FP) definește modalitatea de abordare a unei probleme de programare de către dezvoltator. FP este un model pentru dezvoltarea de programe care utilizează funcții pure și evită starea comună. În scopul înțelegerii acestui concept, funcțiile pure sunt esențiale. Ele trebuie să îndeplinească anumite reguli: ⁽³⁾

- pentru același set de argumente trimise către funcție, ea va returna același rezultat.
- această funcție nu va avea alte efecte secundare. ⁽³⁾

Programarea funcțională este utilă în situații ca:

- apariția necesității efectuării mai multor operații simultan
- existența unei mulțimi de operații sau activități ce trebuie să fie) efectuate pe un set fix de date⁽³⁾

II.2.2 Arhitectura

O aplicație poate fi implementată foarte repede pe baza anumitor cerințe inițiale. Dacă ideea de ansamblu nu este corect concepută, poate deveni dificil, dacă nu imposibil, de modificat și început de la zero.

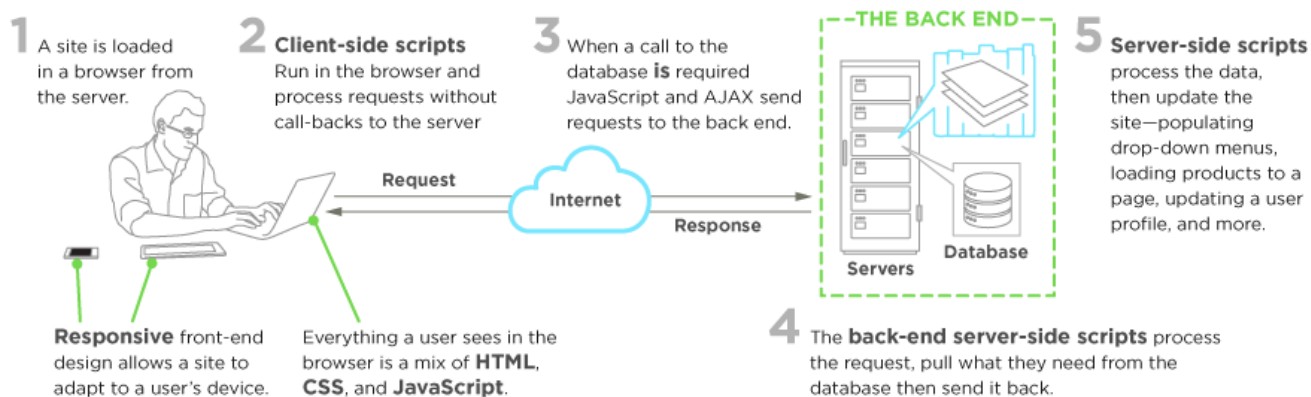
Arhitectura unei aplicații este similară cu arhitectura unei clădiri, deoarece primul pas îl constituie schițarea unui model ce trebuie urmat în vederea obținerii rezultatului dorit.

Este un proces pe care aplicația trebuie să-l întreprindă indiferent dacă este proiectată pentru piața sau dacă este construită pentru un alt spațiu de desfășurare. Spre exemplu, electricianul cunoaște locul unde prizele trebuie montate. Acest lucru este foarte similar cu programarea. În cazul în care priza nu este montată la locul potrivit, modificările sunt mai ușor de făcut pe hârtie decât în realitate.

Serverul din aplicație are rolul de a salva toate resursele comune. Acestea pot fi statice, cum este cazul unui document PDF, sau pot fi dinamice, cum e cazul elementelor de input pentru a trimite date către server. În cele două cazuri menționate anterior, serverul deține controlul în a direcționa datele și de a le pune la dispoziția utilizatorului, prin adrese URL.

În cazul celei din urmă categorii și anume resursele dinamice, serverul are rol de stocare, dar poate avea și rol de prelucrare, astfel încât orice client să poată accesa resursă, sau, cum ar fi în cazul elementelor introduse de utilizator. Aceste resurse sunt refăcute la primirea oricărui element, astfel încât răspunsul la o cerere trimisă să conțină anumite date.

³ Referințe din cartea Abhilasha Sinha, Ranjit Battewad, Alok Ranjan. *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript*, Chapter 5, Aprilie 2020.



(Figura II.2.2.1; Titlu: Relația client-server; Imagine ce exemplifică cum sunt manipulate datele din aplicație de către browser si server)

II.2.3 Dezvoltare

În utilizarea limbajului JavaScript presupune mult efort, iar cerințele consumatorilor au crescut. Un produs final bun determină succesul aplicației, prin feedback-ul pozitiv.

Ce este JSON? ⁽⁴⁾

Notarea obiectului JSON sau JavaScript este un format pentru structurarea datelor.

Pentru ce se folosește?

Ca și XML, este unul dintre modurile de formatare a datelor. Un astfel de format de date este utilizat de aplicațiile web pentru a comunica între ele.

Caracteristicile JSON

- Poate fi scris și citit de oameni
- Este un format de schimb de date bazat pe text, care înseamnă că este mai simplu de citit și de scris decât XML.
- Deși provine dintr-un subset de JavaScript, totuși este independent de limbă. Astfel, codul pentru generarea și analizarea datelor JSON poate fi scris în orice alt limbaj de programare.⁽⁴⁾

Obiectul JSON de test:

```
{
  "users": [
    { "id": 1,
      "email": "rob23@gmail.com" },
    { "id": 2,
      "email": "lucyb56@gmail.com" },
    { "id": 3,
      "email": "annasmith23@gmail.com" },
    { "id": 4,
      "email": "bobbrown432@yahoo.com" },
    { "id": 5,
      "email": "rogerbacon12@yahoo.com" }
  ]
}
```

Exemplu HTML:

În frontend, vom crea un formular care să ia adresa de e-mail ca intrare și să-l transforme în obiect JSON folosind javascript și să-l trimită către server.

După ce facem click pe butonul de trimitere este apelat un `trimiteObiectJson()`, care este definit mai jos.

⁴ Referințe Wazeed17: JavaScript JSON, site-ul <https://www.geeksforgeeks.org>

Metodele de tip GET si POST fac parte din corpul funcției trimiteObiectJson()

GET

```
fetch('http://localhost:3000/users')
  .then(function (response) {
    response.json().then(function (elements) {
      data = response.data;
      data.forEach(e => {
        console.log(`${e.id},${e.email}`)
      });
    }).catch(error => {
      console.log(error);
    });
  });
```

(Figura II.2.3.1; Titlu:Model GET; Imagine care arată o metodă de tip GET)

POST

```
const postObject = {
  id: id.value,
  mail: mail.value
}
fetch('http://localhost:3000/posts', {
  method: 'post',
  headers: {
    "Content-type": "application/json"
  },
  body: JSON.stringify(postObject)
}).then(function () {
  });
```

(Figura II.2.3.2; Titlu:Model POST; Imagine care arată o metodă de tip POST)

II.2.4 Folosirea protocoalelor HTML

Următoarele adrese HTTP finale sunt create automat de un server JSON:

```
GET /users
GET /users/{id}
POST /users
DELETE /users/{id}
```

(Figura II.2.4.1; Titlu: HTTP; Imagine cu metode HTTP)

Dacă sunt făcute solicitări POST sau DELETE, modificările vor fi salvate automat pe db.json. O cerere POST ar trebui să includă un antet de tip „content: Application / json” pentru a utiliza JSON în corpul de solicitare. În caz contrar, va rezulta un 200 OK, dar fără modificări ale datelor. ⁽⁵⁾

Este posibilă extinderea adreselor URL cu un parametru suplimentar. De exemplu poate fi aplicat filtrarea folosind parametrii URL, cum se poate vedea în următoarele: ⁽⁵⁾

http: // localhost: 3000 / users?mail = Sebastian

Astfel, se returnează doar un obiect al user-ului. Sau se efectuează efectuăm un text complet pentru toate proprietățile: ⁽⁵⁾

http: // localhost: 3000 / user? q = yahoo.com

Pentru a exclude o resursă adăugăm _ne:

http: // localhost: 3000 / user?id_ne=1

Adăugăm _like pentru a filtra:

http: // localhost: 3000 / user?id_like=1⁽⁵⁾

⁽⁵⁾Referințe Sebastian Eschweiler: Create A REST API With JSON Server, site-ul <https://medium.com/>

II.3 Câteva repere istorice relativ la temă și rezultate cunoscute

În 2019, în comerțul electronic au fost înregistrate vânzări la nivel mondial având suma de 3,53 trilioane de dolari americani. Aceste venituri, din comerțul electronic vor fi preconizate să crească la 6,54 miliarde de dolari americani în 2022. Una dintre activitățile populare pe internet sunt cumpărăturile online, la nivel mondial. Mulți comercianți cu amănuntul mai mari sunt capabili să mențină o prezență offline și online, prin conectarea ofertelor fizice și online. ⁽⁶⁾

Comerțul electronic le permite clienților să depășească barierele geografice și le permite să achiziționeze produse oricând și de oriunde. Piețele online și tradiționale au diferite strategii de desfășurare a activității. Retailerii tradiționali oferă mai puține sortimente de produse din cauza spațiului în care rafturile, retailerii online nu dețin adesea niciun inventar, dar trimit clienții comenzi direct la fabricație. Strategiile de stabilire a prețurilor sunt diferite și pentru comercianții cu amănuntul tradițional și online. Retailerii tradiționali își bazează prețurile pe traficul magazinelor și pe costurile pentru păstrarea stocurilor. Retailerii online bazează prețurile pe viteza de livrare. ⁽⁷⁾ ⁽⁸⁾

Există două modalități prin care marketerii pot desfășura activități prin e-commerce: complet online. Marketerii online pot oferi prețuri mai mici, o selecție mai mare a produselor și rate de eficiență ridicate. Mulți clienți preferă piețele online dacă produsele pot fi livrate rapid la un preț relativ mic. Cu toate acestea, retailerii online nu pot oferi experiența fizică pe care o pot oferi retaileri tradiționali. Poate fi dificil să judeci calitatea unui produs fără experiența fizică, ceea ce poate determina clienții să experimenteze incertitudinea produsului sau a vânzătorului. Un alt dezavantaj al pieței online este securitatea tranzacțiilor online. Mulți clienți rămân loiali retailerilor cunoscuți. ⁽⁷⁾ ⁽⁸⁾

Securitatea este o problemă importantă în cadrul comerțului electronic în țările dezvoltate și în curs de dezvoltare. Securitatea comerțului electronic protejează site-urile și clienții companiei împotriva accesului, utilizării, modificării sau distrugerii neautorizate. În mediul online există amenințări precum: coduri rău intenționate, programe nedorite (reclame, spyware), phishing, hacking și vandalism cibernetic. Site-urile de comerț electronic folosesc diferite instrumente pentru a evita amenințările la adresa securității. Aceste instrumente includ firewall-uri, software de criptare, certificate digitale și parole. ⁽⁷⁾ ⁽⁸⁾

II.4 Noțiuni științifice sau tehnologice care stau la baza temei

E foarte dificilă dezvoltarea unei aplicații web în totalitate cu ajutorul unei singure tehnologii. Această situație presupune folosirea unor clase de utilități și a altor stiluri de programare.

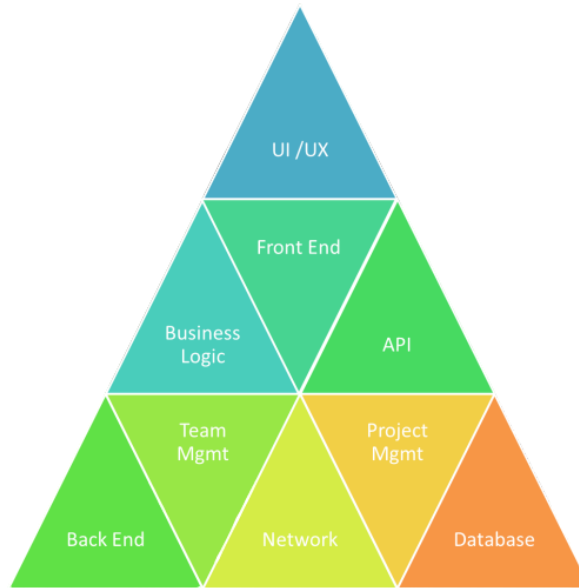
- Compunerea standard de utilități și stiluri este denumită “stivă” , dar o aplicație web constă în câte o stivă pentru cele două elemente, client si server.⁽⁹⁾
- “Full stack” se numește compunerea de tehnologii pentru progresul unei aplicații care se poate traduce “ de la început până la final”. Un programator expert care are experiență, cunoștințe și este foarte priceput în dezvoltarea software se numește “dezvoltator full stack”. Asta înseamnă că pot aborda proiecte care implică baze de date, construind site-uri web orientate către utilizatori sau chiar lucrează cu clienții în faza de planificare a proiectelor.⁽⁹⁾

⁶ Referințe J. Clement: "Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. dollars)". www.statista.com.

⁷ Referințe Li, Zhaolin: "Online Versus Bricks-And-Mortar Retailing: A Comparison of Price, Assortment and Delivery Time". <https://www.tandfonline.com/>

⁸ Referințe Dimkota, Angelika: "On Product Uncertainty in Online Markets: Theory and Evidence" (PDF), 395–A15

⁽⁹⁾ Referințe Scott Morris, “FULL STACK, FRONT END, BACK END—WHAT DOES IT ALL MEAN?” <https://skillcrush.com/>



(Figura II.4.1; Titlu: Stiva; Stiva cu o multitudine de module folosite de dezvoltator full-stack)

Oamenii experimentați ce cunosc nivele de dezvoltare și sunt buni în programare, cu o mulțime de tehnologii avansate și distincte, sunt foarte de puțini. În concluzie, grație transmiterii de informații din ziua de astăzi, este foarte cunoscută utilizarea JavaScript. Cele trei limbaje de programare pe care dezvoltatorii web sunt nevoiți să le cunoască sunt: ⁽¹⁰⁾

1. HTML este folosit pentru definirea paginilor aplicațiilor.
2. CSS este folosit pentru a producerea design-ul web al paginilor.
3. JavaScript este folosit cu scopul de a produce acțiunea în spațiul static construit de celălalte două limbaje de programare.

Limbajul JavaScript este folosit în programele de server și de desktop. Node.js este mai cunoscut decât JavaScript, fiind un mediu de rulare pentru JavaScript ce deține multe librării necesare dezvoltării software.⁽¹⁰⁾

⁽¹⁰⁾ Referințe <https://www.w3schools.com/js/>

JavaScript(pe scurt)

- JavaScript a fost folosit inițial pentru ca paginile să fie aduse la viață. Acesta folosește extensia “.js” și se numește un limbaj de script.. Este conceput pentru a oferi oamenilor acces facil la componente preconstruite. JavaScript este orientat pe obiecte, dar, contrar așteptărilor, nu utilizează clase. Folosește prototipuri de obiecte la moștenire. În cadrul moștenirii, JavaScript are o singură construcție: obiecte. Fiecare obiect are o proprietate privată care deține o legătură cu un alt obiect, numit prototipul său. Acest obiect prototip are un prototip propriu și așa mai departe până când un obiect este atins cu null ca prototip. Aici, în JavaScript, funcțiile au posibilitatea de a transmite un număr distinct de argumente, dar și ele sunt tot obiecte. Aproape toate obiectele în JavaScript sunt instanțe din “Object” ce este în prototipul principal. ⁽¹¹⁾

Totuși, JavaScript este un limbaj complex.

- Unul dintre cei mai importanți factori care contribuie la complexitatea crescută a limbajului este implementarea diferită a limbajului într-un browser care îl acceptă pe piață, ca Internet Explorer. Internet Explorer acceptă indirect JavaScript, oferind suport pentru JScript, propriul său limbaj compatibil cu JavaScript. ⁽¹¹⁾

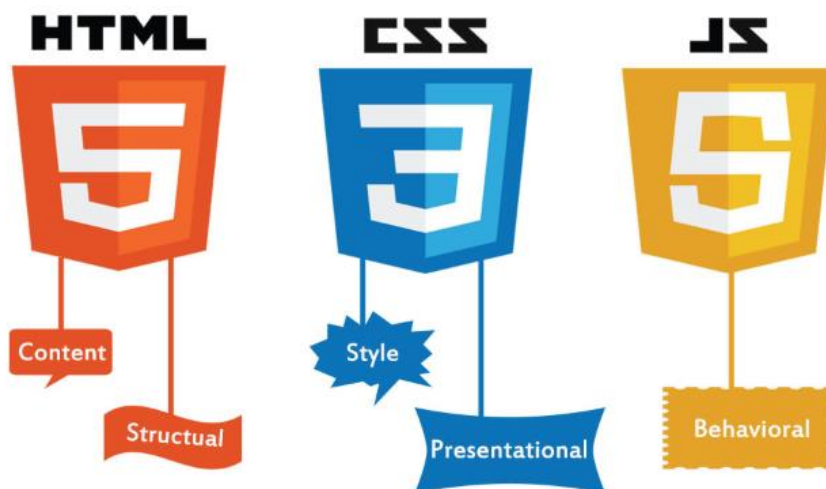
La ușurința cu care este folosit limbajul JavaScript, se adaugă rapiditatea sa. Ca majoritatea limbajelor de script, este interpretat (fără compilare). În cadrul programării cu ajutorul limbajului compilat, cum ar fi C ++, reformatarea sau întocmirea fișierului de cod este necesară, înainte de a putea rula. Limbajul JavaScript permite efectuarea modificărilor în cod cu testare imediată, fără compilație în prealabil. Omiterea compilării economisește mult timp în etapă de debugging a dezvoltării paginilor Web. ⁽¹¹⁾

⁽¹¹⁾ Referințe Chris Minnick, Eva Holland: "The Best Things about JavaScript" <https://www.dummies.com/>

Un alt lucru grozav despre utilizarea unui limbaj interpretat precum JavaScript este că testarea unui script interpretat nu va conduce la o eroare totală, așa cum este cu un limbaj compilat. De exemplu, dacă linia 10 a unui script de 20 de linii conține o eroare de sintaxă, prima jumătate a scriptului dvs. poate fi rulată și puteți primi rezultatele imediat. Aceeași eroare într-un program compilat C/ C++ poate împiedica rularea programului. ⁽¹¹⁾

Mai departe voi prezenta câteva exemple HTML, CSS și JavaScript simple și ușor de folosit și de manevrat:

- De exemplu, dacă doriți să declanșați un fel de eveniment când o persoană face clic pe un buton, accesați gestionarul de eveniment onClick asociat obiectului buton; dacă doriți să declanșați un eveniment când este trimis un formular HTML, accesați evenimentul onSubmit asociat cu obiectul.
- Când încărcați o pagină web interesantă în browserul dvs. și vă întrebați cum autorul a creat efectul în JavaScript, de cele mai multe ori tot ce trebuie să faceți pentru a vă satisface curiozitatea este să faceți clic pentru a vizualiza codul sursă (F12 sau Developer Tools -> Network și acolo puteți încerca să vedeți cu Pretty Print anumite fișiere de cod, de exemplu JavaScript sau HTML). Acest cod sursă gratuit, care este pur și simplu imposibil de compilat cu limbaje de programare precum Java, vă ajută să descifrați programarea JavaScript. ⁽¹¹⁾



(Figura II.4.2; Titlu:Tehnologii; Tehnologii folosite pentru front-end si back-end)

Hyper Text Markup Language (HTML) este un limbaj basic utilizat la crearea paginilor web. Orice browser poate citi fișiere HTML și le poate face conversia în aplicații web vizibile. HTML are ca scop prezentarea informațiilor (paragrafe, font, tabele). Această caracteristică îl face un limbaj de marcare și nu unul de programare.⁽¹²⁾

Cascading Style Sheets (CSS) descrie layout-ul, aspectul, dar și formatarea elementelor apărute dintr-un document ce este scris în HTML sau alt limbaj de marcare. Deși, adesea este folosit cu rolul de a schimba stilul aplicațiilor web și al interfețelor utilizatorilor, se poate aplica la orice fel de document. Alături de HTML și JavaScript, CSS este o tehnologie de puternică ce stă la baza multor site-uri web. Acestea au rolul de a crea aplicații web implicate vizual și interfețe de utilizator pentru multe aplicații web și mobile.⁽¹³⁾

Hyper Text Transfer Protocol se folosește pentru transmiterea informațiilor precum documente hipermedia. HTTP urmează un model client-server în care clientul deschide conexiunea și așteaptă răspunsul serverului.⁽¹⁴⁾

Uniform Resource Locator (URL) face referire la o resursă care specifică locația pe o rețea strâns legată de pc-uri și o tehnică pentru preluarea acesteia. Acesta mai este numit și adresă web. URL-urile apar cel mai frecvent la paginile web de referință (http), dar sunt de asemenea utilizate pentru transfer de fișiere (ftp), e-mail (mailto), acces la baze de date și multe alte aplicații.⁽¹⁵⁾

Document Object Model - o convenție fără atașamente de platformă și limbaj obiectele pentru reprezentarea și interacțiunea cu obiectele din documentele XML, HTML. Reprezintă arborele structurat pentru fiecare document și modalitatea în care acesta este deschis și manevrat.⁽¹⁶⁾

^{(12), (13), (14), (15)} Referințe HTML, DOM, URL, HTTP, CSS <https://en.wikipedia.org/>

⁽¹⁶⁾ Referințe www.w3.org

Advantages & Disadvantages of Using



(Figura II.4.3; Titlu: Node.js; Avantaje si Dezavantaje Node js)

Avantaje Node.js:

- Node.js propune dezvoltatorilor luxul de a coda aplicațiile din server în limbajul JavaScript. Faptul ăsta permite ca în Node.js să se scrie atât aplicația web front-end cât și cea de back-end în JavaScript folosind un mediu de rulare.
- Mediul de rulare open-source al Node.js oferă posibilitatea de a memora în cache module. Ori de câte ori există vreo solicitare pentru primul modul, acesta este memorat în memoria aplicației.
- Npm: este un manager pentru pachetele node. Oferă posibilitatea de a instala și gestiona modulele unei aplicații mai ușor .
- Pentru a fi constant îmbunătățit acesta este sprijinit de o masă largă de oameni fiind atât de popular. Asta rezultă prin adăugarea noilor module. ⁽¹⁷⁾

Dezavantaje Node.js:

- JavaScript nu are un sistem de bibliotecă bine echipat. Rezultatul este că utilizatorii sunt nevoiți să ia suportul bibliotecii pentru executarea diverselor sarcini, cum ar fi Object-Relational Mapping (ORM), procesarea imaginilor, gestionarea operațiunilor bazei de date și analiză XML ș.a. ⁽¹⁷⁾

⁽¹⁷⁾ Referințe Malvi Kiran: "The Positive and Negative Aspects of Node.js Web App Development"

II.5 Obiectivele lucrării prin raportate la context.

Succesul unui site web constă într-un instrument conceput inteligent fiindcă întreprinderile mici aflate la început de drum pătrund pe piața de desfacere cu un dezavantaj față de competitori.

➤ **Educație**

Consider că educația joacă un rol important în formarea unui viitor întreprinzător de afaceri. Cunoștințele dobândite și experiența reprezintă elementele esențiale ce se află în spatele unui om de afaceri de succes. Drumul spre o carieră reușită este presărat cu diverse obstacole, iar secretul pentru victorie îl reprezintă depășire lor.

Pe de altă parte, educația poate face selecție în rândul publicului care accesează aplicația web. Transparența este un concept însemnat, deoarece clienții care au la dispoziție mai multe informații sunt sprijiniți să facă alegeri corecte, ceea ce poate rezulta în creșterea vânzărilor.

➤ **Publicitate**

Publicitatea are ca scop atragerea publicului larg, iar în cadrul aplicațiilor Web iau forma notificărilor de pe email sau a anunțurilor pe site-uri des accesate. Însă această noțiune este mai complexă decât pare.

Aplicațiile web diferă semnificativ de publicitatea tradițională. Spre deosebire de un anunț afișat, un panou publicitar sau un loc de difuzare, un site Web nu este atât de ușor observabil.

➤ **Vânzări directe**

Cumpărăturile online reprezintă viitorul și vom asista la o evoluție uriașă în acest sens. Pentru început, afacerea se poate lansa pe internet până la obținerea unui profit considerabil care să permită deschiderea unui magazin fizic în viitor.

➤ *Relații Clienți*

Formularele online de feedback sunt o sursă eficientă de conștientizare a îmbunătățirilor care pot fi aduse unui site web. Perfecționarea sistematică pot face diferența și propulsa site-ul, având ca urmare creșterea cifrei de afaceri.

III. Contribuția propriu-zisă a candidatului

III.1 Alegerea tehnologiei

În vederea construirii unei aplicații web eficiente, o etapă semnificativă este alegerea stivei tehnologice potrivite, deoarece dezvoltarea produsului software nu constă doar în proiectarea unui UX (User Experience) flexibil și a unei interfețe de utilizare potrivite. Siguranța și stabilitatea unei aplicații web nu pot fi neglijate, întrucât au potențialul de a avea un impact pozitiv asupra clienților și de a aduce companiei mai mult profit.

Atunci când sunt alese tehnologiile la includerea unui produs de aplicație nou pe piață, trebuie să fie luate în considerare aspecte precum:

- **Stabilitatea tehnologiei:** dacă tehnologia există de mult timp pe piață, soluțiile la problemele ce pot apărea pot fi rapid identificate.
- **Experți în domeniu:** În vederea rezolvării posibilelor inconveniente cât mai rapid, este necesară existența unor experți ce cunosc domeniul de interes.
- **Viitorul tehnologiei:** Viitorul unei tehnologii nu poate fi preconizat, dar acesta depinde în mod direct de tehnologiile concurente ulterioare. De asemenea, tehnologiile trebuie să se afle în proces de continuă schimbare, fiindcă cele care stagnează nu pot trece testul timpului.

Luând în considerare aspectele mai sus menționate, am ales următoarele:

JavaScript împreună cu JQuery si Bootstrap

- Framework Express JS de la Node.JS peste care este folosit json-server
- HTML
- CSS

III.2 Funcționalități

Mi-am propus implementarea funcționalităților următoare:

→ *header & footer:*

- ◆ header: afișare meniu cu taburi diferite pentru fiecare pagină
- ◆ footer: adăugare pagini pentru contact, informații livrare, și politica datelor personale.

→ *Blog:*

- ◆ anunțuri despre produse.
- ◆ afișare.

→ *Contact:*

- ◆ trimitere mesaj.
- ◆ afișare formular.

→ *Produse:*

- ◆ afișare produse.

→ *Coș de cumpărături:*

- ◆ afișare produse.
- ◆ achiziționare produse.
- ◆ adăugare produse în coșul de cumpărături .
- ◆ ștergere produse.

→Promoții:

- ◆ afișare produse.

Aplicația este creată cu ajutorul unor comenzi: „npm init” care creează package.json, unde sunt atribuite setări pentru compilarea aplicației.

Dintr-un fișier denumit „server.js” se încarcă module și se creează un server json de pe care vor fi luate datele atribuite din baza de date, acesta fiind fișierul „db.json”.

```
const jsonServer = require('json-server'),
    app = jsonServer.create(),
    router = jsonServer.router('db.json'),
    middlewares = jsonServer.defaults();

var port = process.env.PORT || 3000;
app.use(middlewares);
app.use(router);
```

Pentru datele primite din „db.json”, va fi pornit default pe portul 3000 si va asculta request-uri cu urmatoarele linii de cod.

```
app.listen(port, function(){
    console.log('GULP ruleaza pe portul: ', port);
})
```

GULP este folosit pentru a accelera lucrurile împreună cu modulul nodemon, astfel încât la orice modificare a oricărui fișier, serverul rămâne deschis, funcționând în continuare, chiar dacă codul va rula cu erori.

Pentru configurarea interfeței grafice, vor fi folosite mai multe module bootstrap, care vor fi încărcate în fiecare pagină. „Header.js” încarcă stylesheet-urile mini cu bootstrap, jquery, fontawesome, sweetalert2.

În directorul views am adăugat urmatoarele pagini:

- index.html prima pagină, statică; va cuprinde un carusel pentru afișarea unor imagini sugestive, a unor produse și testimoniale;
- date.html statică, va conține un PDF despre politica datelor personale;
- about.html: pagină statică pe care sunt afișate informații despre acest magazin online
- prod.html: o pagină ce trimite o interogare HTTP POST cu anumite produse selectate din meniul principal.

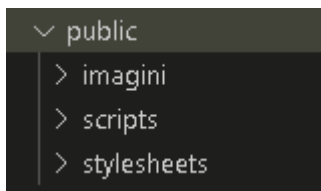
- `cart.html`: conține un formular ce trimite o interogare HTTP POST ce cuprinde actualizări ale câmpurilor produselor din coșul de cumpărături și HTTP DELETE pentru ștergerea itemelor din cos. De asemenea, conține HTTP POST și HTTP DELETE, pentru a efectua o comandă și a șterge toate itemele comandate din interiorul cosului.

- `contact.html` : o pagină statică ce conține un formular care trimite date despre clienți și feedback-ul acestora în legătură cu magazinul online pe care l-au vizitat.

- `blog.html`: conține mai multe date despre produsele noi care vor fi adăugate în magazinul online.

- `livrare.html`: este o pagină cu datele livrării și costurile aferente.

În directorul public am adăugat următoarele subfoldere:



- Imagini: cuprinde mai multe imagini statice pentru interfața statică.
- Scripts este un folder în care am adăugat toate scripturile folosite în cadrul aplicației: `cart.js`, unde se fac request-uri de tip GET, POST, DELETE pentru produsele din coș și alte funcții ajutătoare pentru acțiunile corespunzătoare funcționării normale.

`contact.js`: pentru a obține mesajele utilizatorilor, este folosită o aplicație web numită „formspring.io”, în cadrul căreia sunt reținute toate mesajele trimise prin intermediul aplicației.

`headers.js` este folosit pentru încărcarea tuturor fișierelor de îmbunătățire a interfeței utilizatorului: bootstrap, fontawesome și codul CSS scris de mine. De asemenea, se va încărca doar acest fișier, în locul a 4 fișiere CSS.

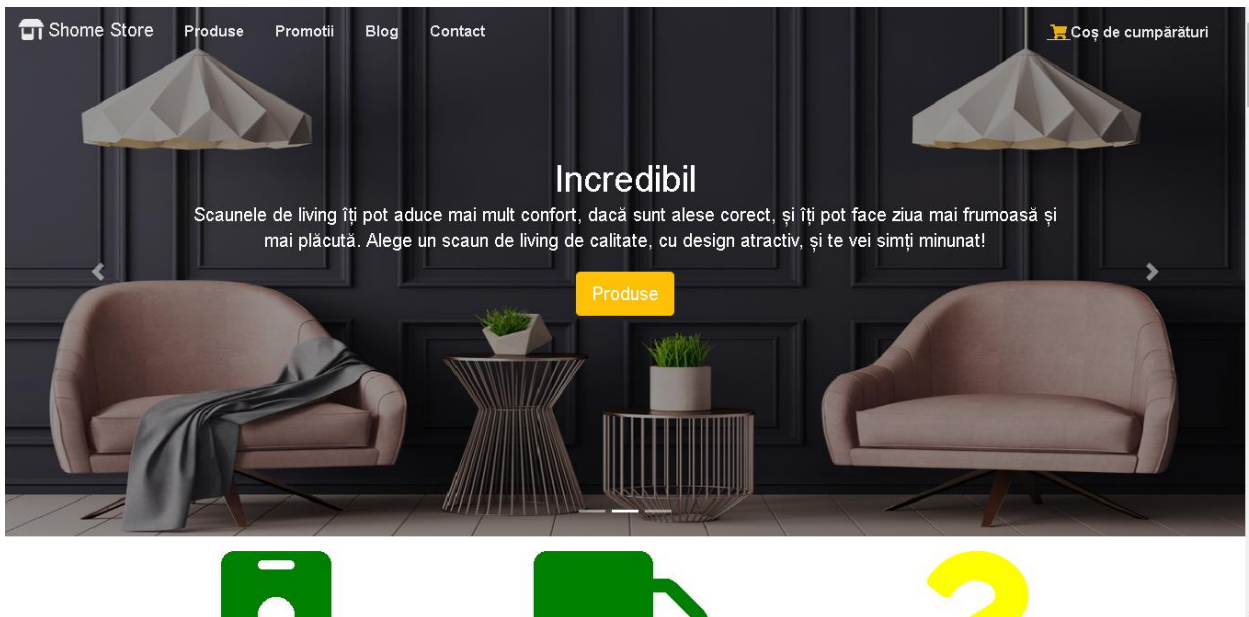
`index.js` cuprinde toate funcțiile necesare paginii de pornire și a tuturor celorlalte pagini dintre care se menționează metodele de GET, PUT, DELETE, coșul de cumpărături, GET și POST pentru promoții, trimiterea de mail-uri de verificare pentru abonarea la newsletter.

- Stylesheets cuprinde 2 fișiere „`carousel.css`” și „`style.css`”, în care sunt atribuite proprietăți tag-urilor din fișierele html. Fișierul „`carousel.css`” este creat special pentru derularea imaginilor de pe prima pagină, în „`style.css`” aflându-se restul îmbunătățirilor aduse întregii aplicații.

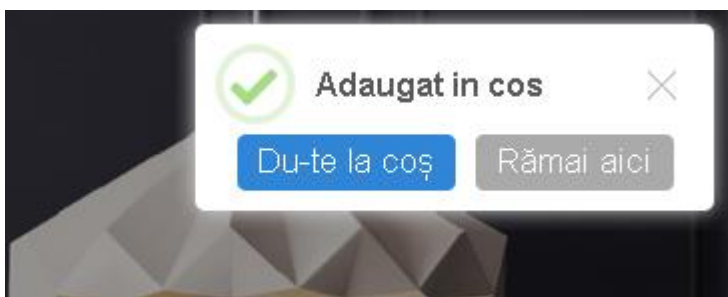
III.3 Aplicația în practică

În stadiul de față, aplicația este cam greoaie și codul nu este tocmai descriptiv și este de asemenea foarte învețată, dar funcționează.

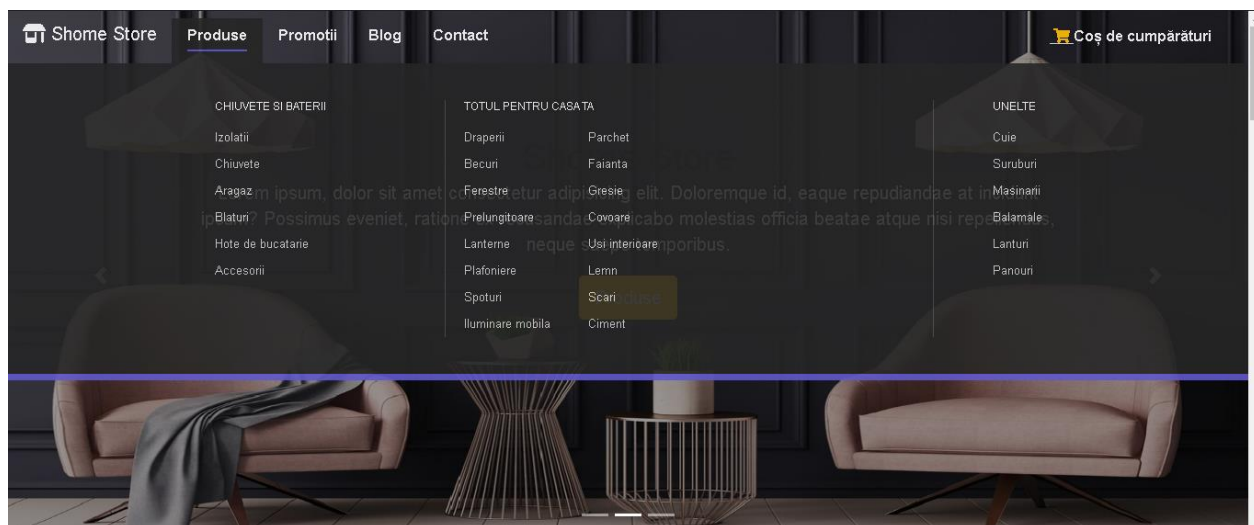
Aplicația poate fi accesată local prin fișierul index.html. Pagina principală are legături către toate celelalte pagini, dar și celelalte pagini, legături înapoi.



Când un produs este adăugat în coș, următorul font este accesat cu un modul folosit din sweetalert2:

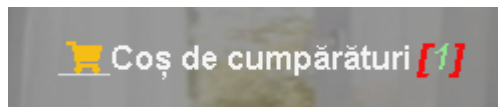


Există opțiunea de a accesa coșul de cumpărături și opțiunea de a rămâne pe pagină.

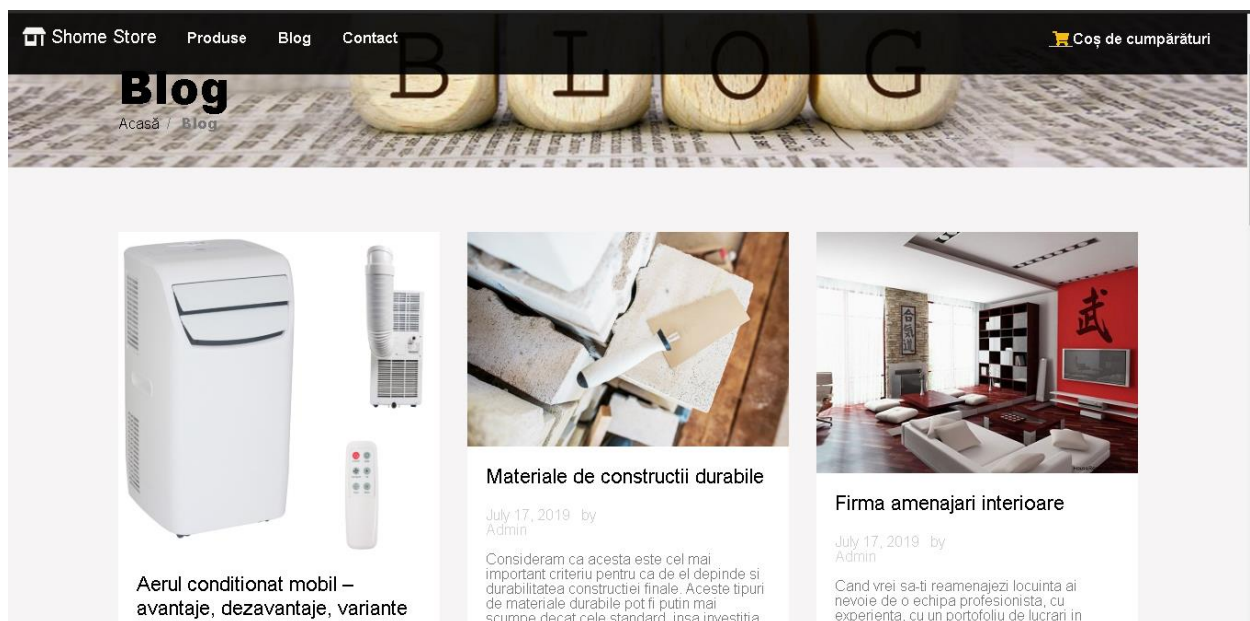


Utilizatorii pot lua legătura cu deținătorul aplicației prin completarea formularului de contact, aflat în header/footer.

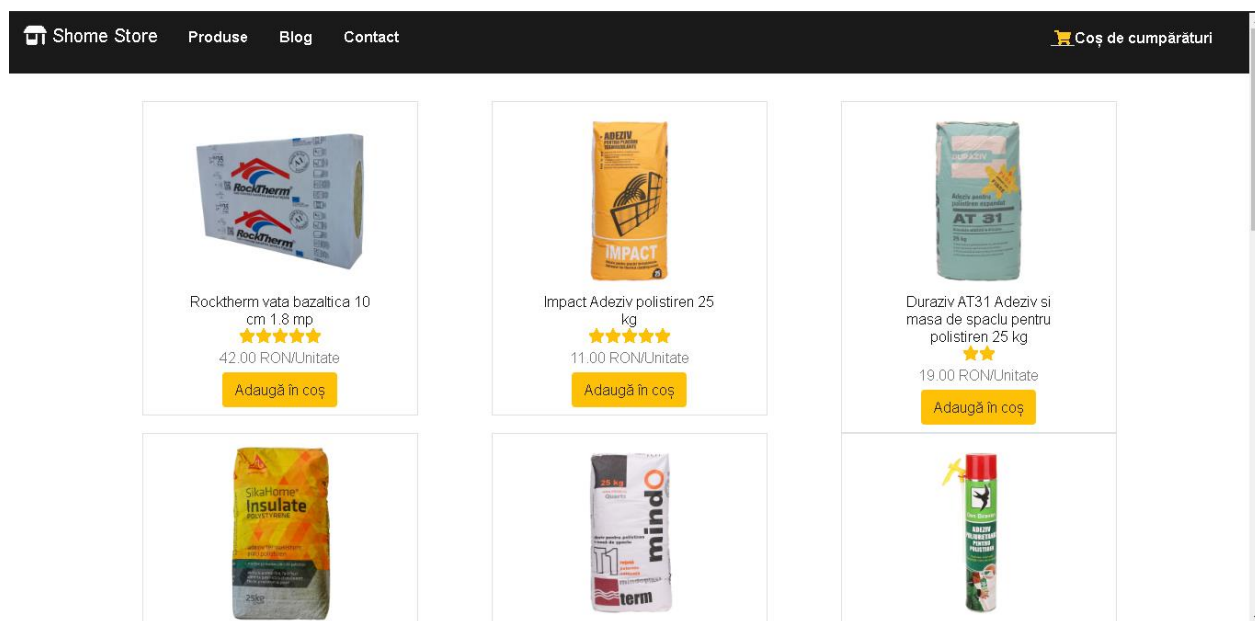
În plus, când este adăugat un produs în coșul de cumpărături va apărea o notificare, iar numărul de iteme prezente în coș va fi afișat pe toată durata accesării aplicației.



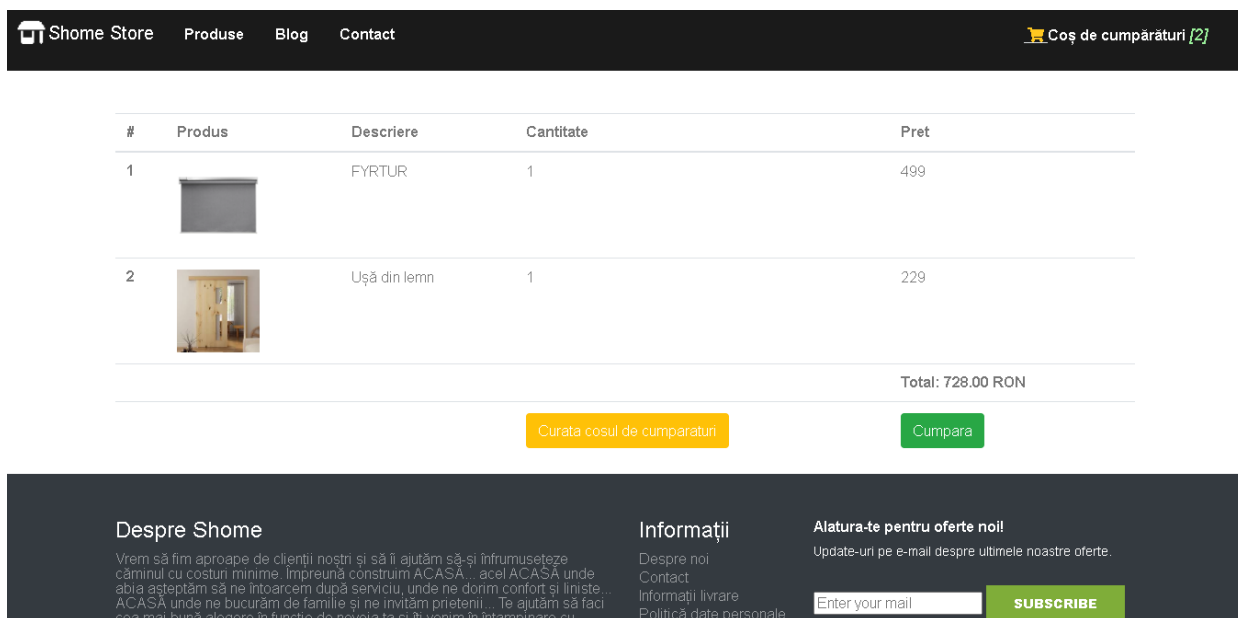
Pentru accesarea blog-ului disponibil în meniul principal, utilizatorul va apăsa butonul “Blog” din meniul superior al aplicației.



Clienții pot selecta din meniul principal o categorie de produse, fiind conduși către pagina aferentă, ca în exemplul de mai jos :

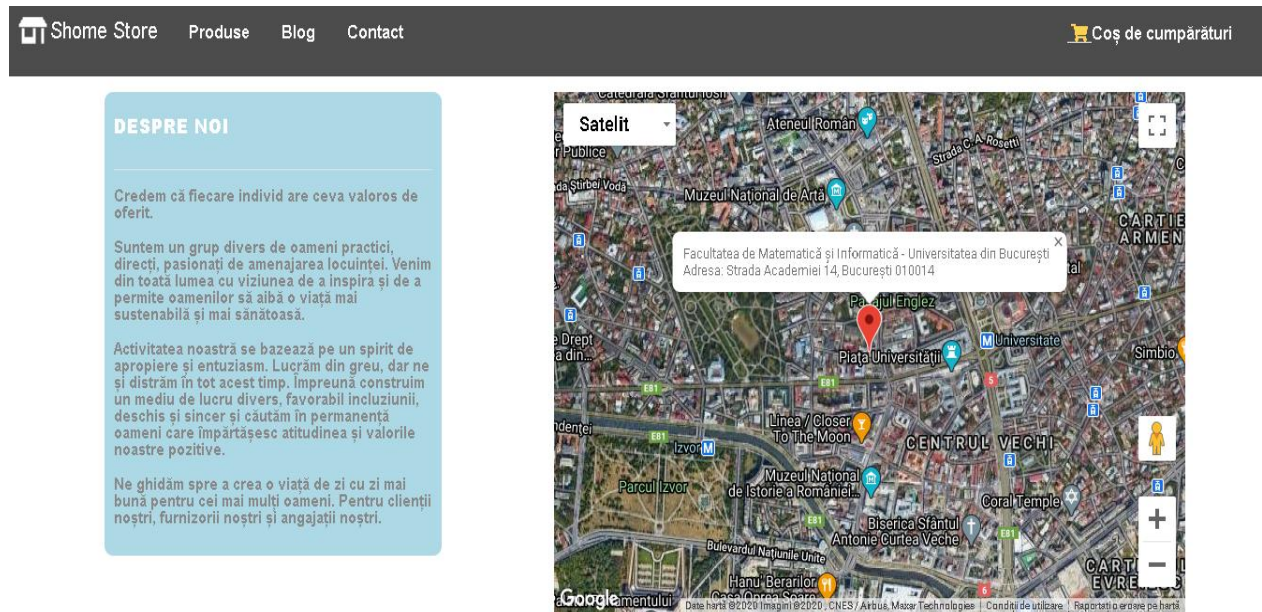


În coșul de cumpărături se află obiectele adăugate la apăsarea pe butonul de adaugă în coș :



Am adăugat și o pagină numită „despre noi”, în care sunt evidențiate informații despre

locația magazinului printr-un API Google Maps și alte câteva date.



Aplicația dispune de o pagină cu detalii despre livrare.



Te ajutam cu livrarea cumparaturilor realizate in magazin, in 24 ore*

Consulti catalogul online, alegi marfa si lasi transportul in seama noastra.

Te contactam telefonic in prealabil.

Livrăm la tine acasă sau acolo unde ai nevoie.

Costul unei livrări este calculat pentru 1.5 Tone sau 6 m³ și variază în funcție de distanța până la locația ta.

IV. Concluzii

IV.1 Concluzii cu referire la modul de realizare a temei

Pentru mine o aplicație folosind cele 3 tehnologii nu a fost atât de complicată și am realizat-o rapid. Folosirea celor 3 tehnologii pentru realizarea aplicației a fost ușoară, necesitând puțin timp. Am reușit în câteva zile să îi pun, apoi mi-am îndreptat atenția către funcționalități.

Am făcut o cercetare îndelungată în căutarea produselor și a informațiilor pe care să le adaug în aplicația mea însă nu a reprezentat un obstacol major, datorită accesului la internet și a multitudinii de informații de care am dispus. Având în vedere scopul lucrării, nu am pus accentul pe modalitățile prin care să măresc numărul clienților în aplicație.

Făcând o comparație între aplicațiile web pentru lucruri casnice ale unor magazine de pe piață actuală și aplicația creată de mine, am făcut următoarele constatări:

- magazinele online care sunt pe piață de mult timp sunt de referință în acest domeniu și au avantajul brand-ului creat.
- aplicația mea este în stadiul de proiect local, nu are făcut deploy și nu va avea succes până nu îi voi implementa funcționalități ce se ridică la standardele magazinelor online în trend.

IV.2 Posibile dezvoltări ulterioare ale temei

Publicitatea este o modalitate excelentă de a lansa o afacere de comerț electronic aflată în stadiul de început, deoarece conduce tot traficul pe site-ul companiei. Dar publicitatea, prin metode precum pay-per-click, reprezintă o strategie slabă pe termen lung pentru generarea de trafic, întrucât oamenii nu sunt tentați să acceseze același site de 2 ori.

Cele mai eficiente metode de publicitate, cum ar fi anunțurile Google, nu sunt profitabile din cauza costurilor mari. Implicarea activă periodică a companiei în evenimente caritabile și oferirea serviciilor gratuite vor influența pozitiv traficul pe termen și vor aduce o creștere a profitului.

O altă alternativă viabilă ar fi atragerea investitorilor, ceea ce presupune o idee stabilă, flexibilă și fiabilă.

Alegerea uneia dintre opțiunile prezentate mai sus este condiționată de o investiție inițială în vederea obținerii de bunuri și servicii care să fie ulterior livrate.

IV.3 Aprecierile personale privind relevanța rezultatelor obținute

Deși nu mă caracterizează spiritul de lider sau cel de antreprenor, apreciez că am lucrat și am încercat să aduc această aplicație în stadiul de proiect final.

Mă consider destul de răbdător și eficient când îmi fixez un obiectiv. Astfel, am depus mult efort pentru a obține elemente de funcționalitate și design în cadrul aplicației mele. Diferența dintre ideea inițială ce viza aplicația web și rezultatul obținut în prezent nu este semnificativă, deoarece am urmat aceeași traiectorie. Am fost oarecum organizat în a duce sarcina la bun sfârșit și m-am întrecut pe mine însămi în construirea unor elemente de design și funcționalități. Mă simt încrezător în stadiul acesta și realizez că întotdeauna este loc de evoluție.

IV.4 Eventuale aplicații practice sau interferențe cu alte domenii

Partea practică a acestei aplicații este obținerea profitului prin vânzarea unor produse online publicului interesat. Cu toate acestea, consumatorii dispun de o spectru larg de aplicații pentru comerț online. Atunci când prețul este factorul competitiv, supraviețuirea unui start-up este o luptă grea, pentru că nu își permit micșorarea prețurilor, precum site-urile de renume. Brand-urile consacrate în acest domeniu au la bază o plajă mare de clienți fideli, ceea ce reduce semnificativ rata de succes a afacerilor aflate la început.

Pentru un nou vânzător nu există posibilitatea de reducere a costurilor și nici nu dispune de marcă consacrată. Astfel, șansa pentru un nou magazin este promovarea abilităților unice ale produselor. Oamenii decid cu rapiditate dacă o aplicație web le place sau nu. Ca interesul lor să nu se piardă, prima pagină ar trebui să fie cât mai atractivă. Ofertele de cumpărături sunt cele mai atractive. Unii clienți sunt atrași de transportul gratuit, însă aproape toți sunt interesați de promoții. Așadar ofertele, reducerile și transportul gratuit sunt exclusive și de obicei sunt primele lucruri pe care le caută vizitatorii site-ului.

Interesul oamenilor pentru produse noi este în creștere, iar calitatea lor poate determina succesul unei afaceri.

Bibliografie

1. Abhilasha Sinha, Ranjit Battewad, Alok Ranjan. *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript*, Chapter 5, Aprilie 2020.
2. Dimkota, Angelika: "On Product Uncertainty in Online Markets: Theory and Evidence" (PDF). 36 (2): 395–A15

Resurse online

- <https://ec.europa.eu>, 25 iunie.
- <https://semrush.com>, 25 iunie.
- <https://www.w3schools.com/js>, 27 iunie.
- <https://en.wikipedia.org/>, 27 iunie.
- Chris Minnick, Eva Holland: "The Best Things about JavaScript"
<https://www.dummies.com/>, 28 iunie.
- J. Clement: "Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. dollars)". www.statista.com, 26 iunie.
- Kiran Malvi: "The Positive and Negative Aspects of Node.js Web App Development" www.mindinventory.com, 28 iunie.
- Li, Zhaolin: "Online Versus Bricks-And-Mortar Retailing: A Comparison of Price, Assortment and Delivery Time". <https://www.tandfonline.com/>, 26 iunie.
- Scott Morris, "FULL STACK, FRONT END, BACK END—WHAT DOES IT ALL MEAN?" <https://skillcrush.com/>
- Sebastian Eschweiler: Create A REST API With JSON Server, site-ul <https://medium.com/>, 26 iunie.
- Wazeed17: JavaScript JSON, site-ul <https://www.geeksforgeeks.org/>, 27 iunie.

ANEXE

Dicționar de termeni și acronime

Am folosit mai multi termeni si tabelul de jos contine mai multe definitii pentru toti termenii folositi in aceasta lucrare.

NPM	<i>node package manager</i> , este o utilitate din linia de comandă pentru interacțiunea cu un repository, care ajută la instalarea pachetelor
JSON	<i>JavaScript Object Notation</i> , este un format bazat pe text pentru interschimbarea datelor între aplicații.
RESTful	<i>REpresentational State Transfer</i> , fiind un stil arhitectural, impune anumite limite pentru serviciile web.
HTTP	<i>HyperText Transfer Protocol</i> . Se folosește pentru transmiterea informațiilor precum documentele hipermedia.
UI	<i>User Interface</i> . Este procesul pe care designerii îl utilizează pentru a construi interfețe în software sau dispozitive computerizate, cu accent pe aspect sau stil.
UX	<i>User Experience</i> . Aceasta înseamnă experiența unui utilizator într-o aplicație cu privire procesul de design pe care l-au săvârșit dezvoltatorii să livreze acel produs.
URL	<i>Uniform Resource Locator</i> . Face referire la o resursă care specifică locația pe o rețea strâns legată de pc-uri și o tehnică pentru preluarea acesteia
PDF	<i>Portable Document Format</i> . Este un format de fișier pentru a prezenta documente, inclusiv formatarea textului și imaginile.

DOM	<p><i>Document Object Model</i> este o convenție pentru reprezentarea și interacțiunea cu obiectele din documentele HTML.</p> <p>DOM este un model de obiect standard si programarea interfetei pentru HTML.</p> <p>Definește :</p> <p>Elementele HTML ca obiecte.</p> <p>Proprietățile, metodele și acțiunile tuturor elementelor.</p>
ORM(O/RM)	<p><i>Object relational mapping</i>. Poate fi utilizată din limbajul de programare si creează o „bază de date de obiecte virtuale”.</p>
BOOTSTRAP	<p>Bootstrap este un mediu de dezvoltare CSS, open-source, către dezvoltarea de aplicații web pentru front-end. Are în componență template-uri de design din CSS și componente JavaScript.</p>
JQUERY	<p>jQuery este o bibliotecă JavaScript, destinată simplificării folosirii arborelui HTML anume DOM, explicat mai sus și gestiunii animațiilor CSS și evenimentelor.</p>
GULP	<p>Este un task runner pentru a automatiza sarcinile consumatoare de timp si repetitive implicate in dezvoltarea aplicatiei web.</p>
NODEMON	<p>Este un tool care ajuta la dezvoltarea aplicatiilor web bazate pe node.js prin repornirea automata a aplicatiei atunci cand sunt aduse modificari detectate.</p>