

Universitatea “Politehnica” din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

*Aplicație web destinată vizualizării, cumpărării și rezervării unor componente
auto*

Proiect de diplomă

prezentat ca cerință parțială pentru obținerea titlului de
Inginer în domeniul Calculatoare și Tehnologia Informației
programul de studii de licență *Ingineria Informației*

Conducător științific
Ș.l. dr. ing. Radu BADEA

Absolvent
Radu Cristian SIMIONESCU

2021

Universitatea "Politehnica" din Bucuresti
Facultatea de Electronica, Telecomunicatii si Tehnologia Informatiei
Program de studiu INF

Anexa 1

TEMA PROIECTULUI DE DIPLOMĂ
a studentului **SIMIONESCU M.F. Radu-Cristian, 444A**

1. Titlul temei: Aplicatie web destinata vizualizării, cumpărării si rezervării unor componente auto

2. Descrierea temei si a contributiei personale a studentului (in afara partii de documentare):

Obiectivul proiectului de diplomă este proiectarea si dezvoltarea unei aplicatii web pentru vizualizarea, cumpărarea si rezervarea unor produse/piese auto. Această aplicatie va contine o parte de interfatare cu utilizatorul (front-end) ce urmează a fi implementată în HTML / CSS / JavaScript. Se va genera o listă de produse oferite spre vânzare împreună cu informatii despre acestea (pret, tip piesă, cod identificare, descriere produs, marcă compatibilă, etc.). Un client va putea selecta obiectele care se vor adăuga în cosul de cumpărături sau se vor rezerva pentru ridicare ulterioară. Unde este necesar va exista si o galerie foto pentru detalii despre piesa respectivă. Partea de back-end va fi implementată în Java (prelucrare comenzi piese, implementarea accesului la baza de date, etc.) precum si o bază de date de tip MySQL în vederea stocării informatiilor relevante pentru procesarea comenzilor si afisarea articolelor oferite spre vânzare pe site. Optional, se va implementa si un sistem de autentificare al utilizatorilor. Modulele software rezultante vor fi testate, urmând la final integrarea acestora în vederea realizării aplicatiei complete.

3. Discipline necesare pt. proiect:

Programarea calculatoarelor, Programarea orientata obiect, Tehnologii de programare in Internet, Proiectarea bazelor de date, Inginerie Software

4. Data înregistrării temei: 2020-11-26 13:06:20

Conducător(i) lucrare,
Ș. L. Dr. Ing. Radu BADEA

Student,
SIMIONESCU M.F. Radu-Cristian

Director departament,
Ș.L. dr. ing Bogdan FLOREA

Decan,
Prof. dr. ing. Mihnea UDREA

Cod Validare: **6728063a80**

Declarație de onestitate academică

Prin prezenta declar că lucrarea cu titlul “ *Aplicație web destinată vizualizării, cumpărării și rezervării unor componente auto*”, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității “Politehnica” din București ca cerință parțială pentru obținerea titlului de *Inginer* în domeniul *Calculatoare și Tehnologia Informației*, programul de studii *Ingineria Informației* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmentele de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurărilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 29.06.2021

Absolvent *Radu Cristian SIMIONESCU*



(semnătura în original)

CUPRINS

Lista figurilor utilizate în lucrare.....	9
Lista tabelelor utilizate în lucrare.....	11
Lista acronimelor utilizate în lucrare.....	13
INTRODUCERE.....	95
CAPITOLUL 1 PREZENTAREA GENERALĂ A SITE-URILOR.....	17
1.1 Tipuri de site-uri web	18
1.2 E-Commerce sau Electronics Commerce.....	18
1.3 Dezvoltarea site-urilor web	21
1.4 Publicarea paginilor web	23
1.5 Înregistrarea URL-urilor web.....	24
1.6 Site-uri Web Hosting.....	25
CAPITOLUL 2 INSTRUMENTELE SOFT FOLOSITE PENTRU REALIZAREA APLICAȚIEI.....	27
2.1 Limbajul PHP	27
2.1.1. Introducere în PHP	27
2.1.2. Câteva referiri la limbajul PHP	28
2.2 Limbajul MySQL	31
2.3 CSS.....	33
2.4 HTML.....	35
2.5 JavaScript.....	39
CAPITOLUL 3 PROIECTAREA DE DETALIU A APLICAȚIEI INFORMATICE... 41	
3.1. Definirea obiectivelor aplicației informatice.....	41
3.2 Proiectarea bazei de date	43
3.3 Schema de sistem a aplicației	52
CAPITOLUL 4 PREZENTAREA PRODUSULUI SOFTWARE	59
3.1. Descrierea funcțiilor aplicației	59
3.2 Manual de utilizare.....	61
CONCLUZII.....	75
BIBLIOGRAFIE.....	77
ANEXA 1	81

Lista figurilor utilizate în lucrare

Figura 1.1 Comunicarea client pagină Web [1].....	17
Figura 1.2 Modalități de schimb de informații [2].....	19
Figura 1.3 Dezvoltarea site-urilor web [4].....	21
Figura 1.4 Pașii în procesul de dezvoltare web [4].....	22
Figura 2.2 Exemplu tabel de date MySQL [22].....	32
Figura 2.3 Tipuri de selectori [25].....	34
Figura 2.4 Exemplu simplu cod HTML [26].....	35
Figura 3.1 Tabele conținute în baza de date “piese auto”.....	43
Figura 3.2 Tipul de date al atributelor din tabelul tadmins.....	43
Figura 3.3 Valorile introduse în atributele aflate în tadmins.....	43
Figura 3.4 Tipul de date al atributelor din tabelul tcategorii.....	44
Figura 3.5 Valorile introduse în atributele aflate în tcategorii.....	44
Figura 3.6 Tipul de date al atributelor din tabelul tcomenzi.....	45
Figura 3.7 Tipul de date al atributelor din tabelul tcos.....	45
Figura 3.8 Valorile introduse în atributele aflate în tproducatori.....	46
Figura 3.9 Tipul de date al atributelor din tabelul tproduse.....	47
Figura 3.10 Valorile introduse în atributele aflate în tproduse.....	47
Figura 3.11 Valorile generate după efectuarea unor comenzi.....	48
Figura 3.12 Tipul de date al atributelor din tabelul tsubcategorii.....	48
Figura 3.13 Valorile introduse în atributele aflate în tsubcategorii.....	49
Figura 3.14 Tipul de date al atributelor din tabelul tutilizatori.....	49
Figura 3.15 Utilizatorii înregistrați până acum pe site și datele acestora.....	50
Figura 3.16 Voturile înregistrate pe site.....	50
Figura 3.17 Schema conceptuală a bazei de date.....	51
Figura 3.18 Arhitectura client-server pe trei nivele [34].....	52
Figura 3.19 Arhitectura generală a sistemului.....	52
Figura 3.20 Reprezentare grafică propusă a nivelelor arhitecturii cu tehnologiile aferente.....	53
Figura 3.21 Funcția de conectare la baza de date.....	54
Figura 3.22 Meniul din stânga.....	54
Figura 3.23 Funcția pentru validarea unui formular.....	55
Figura 3.24 Meniul orizontal.....	55
Figura 3.25 Meniul din stânga în pagina de administrator.....	56
Figura 3.26 Meniul de logare.....	57
Figura 4.1 Pagina Home a aplicației.....	61
Figura 4.2 Formularul pentru crearea unui cont nou.....	62
Figura 4.3 Atenționare completare incorectă.....	62
Figura 4.4 Formular de contact al firmei.....	63
Figura 4.5 Pagina cu toate modelele disponibile.....	64

Figura 4.6 Mesaj de eroare, necesită autentificare în cont.....	64
Figura 4.7 Mesaj care informează de lipsa produselor din coș.....	64
Figura 4.8 Modul de afișare al produselor în coș.....	65
Figura 4.9 Formularul pentru datele de comandă.....	65
Figura 4.10 Confirmarea comenzii.....	66
Figura 4.11 Mesajul de finalizare a comenzii.....	66
Figura 4.12 Meniul de administrare a produselor.....	66
Figura 4.13 Implementarea meniului de administrare a produselor.....	67
Figura 4.14 Meniul din stânga al administratorului.....	67
Figura 4.15 Tabul de modificare a parolei pentru admin.....	68
Figura 4.16 Opțiunile pentru gestionarea site-ului.....	68
Figura 4.17 Meniul pentru adăugarea unui producător.....	69
Figura 4.18 Meniul pentru modificarea unei categorii.....	69
Figura 4.19 Mesaj de atenționare neselectare categorie.....	70
Figura 4.20 Meniul pentru adăugare produs.....	70
Figura 4.21 Implementarea meniului de adăugare a unui produs.....	71
Figura 4.22 Opțiunile de adminstrare a utilizatorilor și a comenzilor.....	71
Figura 4.23 Lista cu utilizatorii înregistrați.....	72
Figura 4.24 Meniul de adminstrare a comenzilor.....	72
Figura 4.25 Detaliile unei comenzi finalizate.....	72
Figura 4.26 Meniul de vânzări totale.....	73
Figura 4.27 Meniul de vânzări pe o anumită perioadă calendaristică.....	73
Figura 4.28 Meniul de vânzări pe clienți.....	73
Figura 4.29 Meniul de vânzări pe produse.....	74

Lista tabelelor utilizate în lucrare

Tabelul 1.1 Domenii generice de nivel superior [8].....	24
Tabelul 1.2 Tipuri de găzduire [10].....	25

Lista acronimelor utilizate în lucrare

API - Application Programming Interface/ Interfață de Programare a Aplicațiilor
CGI - Common Gateway Interface
CMS - Content management system/ Sistem de administrare a conținutului
COM - Common Object Model
CORBA - Common Object Request Broker Architecture
CPU - Unitate centrală de prelucrare/ Central Processing Unit
CSS - Cascading Style Sheets
DOM - Document Object Model
GIF - Graphics Interchange Format
HMTL5 - HyperText Markup Language5
HTML - HyperText Markup Language
ID - Identification/Identity/Identifier
IIS - Internet Information Services
IMAP - Internet Message Access Protocol
JPG - Joint Photographic Experts Group
LDAP - Lightweight Directory Access Protocol
PHP - Hypertext Preprocessor
PHP4 - Hypertext Preprocessor4
POP3 - Post Office Protocol/ Protocolul Post Office
RDBMS - Relational Database Management System
SVG - Scalable Vector Graphics
UI - User Interface
UX - User experience
VPS - Virtual Private Server/ Server privat virtual
WEB - World Wide Web
WWW - World Wide Web
XHTML - EXtensible HyperText Markup Language
XML - Extensible Markup Language

INTRODUCERE

Comerțul electronic câștigă rapid teren față de modul clasic de a face cumpărăturile, cel fizic. Tot mai multe afaceri implementează site-uri web care oferă funcționalități pentru efectuarea de tranzacții comerciale pe web. Este rezonabil să spunem că procesul de cumpărături pe web devine banal.

Obiectivul acestui proiect este de a dezvolta un magazin de comerț electronic de uz general, în care produsele precum piesele auto să poată fi cumpărate din confortul casei prin internet. Cu toate acestea, în scopul implementării, această lucrare se va ocupa de cumpărături online de piese auto.

Un magazin online este un magazin virtual de pe internet unde clienții pot naviga în catalog și selecta produse de interes. Articolele selectate pot fi colectate într-un coș de cumpărături. La checkout, articolele din coșul de cumpărături vor fi prezentate ca o comandă. În acel moment, vor fi necesare mai multe informații pentru a finaliza tranzacția. De obicei, clientului i se va cere să completeze sau să selecteze o adresă de facturare, o adresă de expediere, o opțiune de expediere și informații de plată, cum ar fi numărul cardului de credit. Aplicația își propune să vină și în ajutarea administratorului prin oferirea de funcționalități foarte importante, cum ar fi, ușurința în adăugarea/modificarea/ștergerea produselor și modelelor de mașini din site, dar și posibilitatea analizării unor raporte detaliate și diverse în privința vânzărilor. Capitolul unu face o descriere a tipurilor de site-uri web, dezvoltarea și publicarea acestora. În capitolul doi sunt prezentate software-urile informatice și tehnologiile care au fost folosite în realizarea acestui proiect. În capitolul trei este descris procesul de proiectare al aplicației. În cadrul acestui capitol sunt descrise arhitectura sistemului, proiectarea bazei de date, proiectarea datelor de intrare și ieșire.

În cadrul capitolului patru este prezentată aplicația web. Începând cu modulul utilizatorului la care vor avea acces clienții, trecând prin implementarea conexiunii la baza de date și a interfeței interactive, prin implementarea interfeței și funcționalităților site-ului pentru administrator până la vizualizarea comenzilor efectuate pe o anumită perioadă, care sunt cei mai fideli clienți sau cele mai bine vândute piese auto.

La finalul lucrării vor fi prezentate concluziile despre modul în care a decurs realizarea acestui proiect și de asemenea și perspectivele viitoare de dezvoltare în ceea ce privește comerțul online.

CAPITOLUL 1 PREZENTAREA GENERALĂ A SITE-URILOR

Site-ul Web este o locație web și este găzduit pe un server web. Este format dintr-un set de pagini web asociate. Este accesat utilizând adresa de Internet cunoscută sub numele de Uniform Resource Locator.

Site-urile web statice sunt, de asemenea, cunoscute ca site-uri plate sau staționare. Acestea sunt încărcate în browser-ul clientului exact așa cum sunt stocate pe serverul web. Aceste site-uri conțin doar informații statice. Utilizatorul poate citi numai informațiile, dar nu poate modifica sau interacționa cu informațiile.

Site-urile statice sunt create folosind doar cod HTML. Site-urile web statice sunt utilizate numai atunci când informațiile nu trebuie modificate. [1]

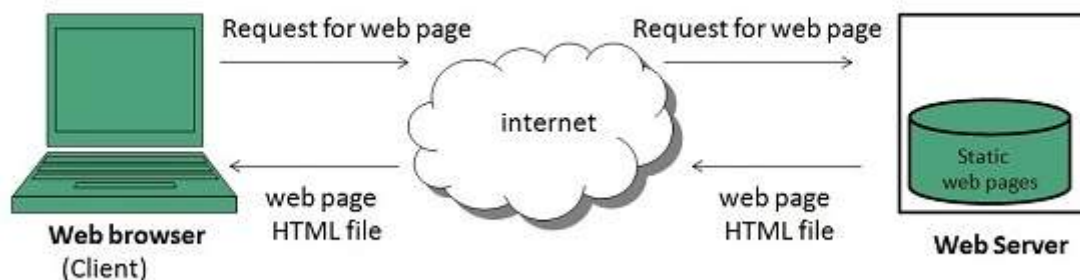


Figura 1.1 Comunicarea client pagină Web [1]

Site-urile dinamice afișează informații diferite în momente diferite. Prin utilizarea tehnologiei **Ajax** este posibilă modificarea unei porțiuni a unei pagini web fără a încărca întreaga pagină web.

Pagina dinamică a serverului

Acesta este creată folosind script-ul de pe server. Există parametri de scripting din partea serverului care determină modul de asamblare a unei pagini web noi, care include, de asemenea, configurarea procesării pe mai multe direcții client.

Pagina dinamică a clientului

Acesta este procesată folosind scripting din zona clientului, cum ar fi javascript. Apoi este trecut în **Document Object Model (DOM Object Model)**. [1]

1.1 Tipuri de site-uri web

Un forum pe internet este un forum de discuții unde oamenii pot purta conversații prin postarea mesajelor.

Puncte cheie

- Un forum poate conține mai multe sub-forumuri.
- Fiecare din sub-forumuri poate conține o serie de subiecte.
- În cadrul unui forum, fiecare nouă discuție postată se numește fir.
- La acest thread pot răspunde mai mulți oameni. [2]

Blog

Termenul **Blog** este luat de la **b**(browser) **log**(logare). Este un site web care este actualizat în mod regulat, cu conținut despre aproape orice. Cu alte cuvinte, blogul este un sistem de management al conținutului (CMS), o modalitate ușoară de a publica articole pe internet.

Blogging -terminologie

BLOG

Un tip de site web folosit pentru a publica conținut pe internet.

BLOGGER

O persoană care scrie pentru un blog.

BLOGGING

Scrierea pentru bloguri este denumită blogging.

BLOGOSFEREI

Un termen este folosit pentru a face referire la toate blogurile de pe web.

Ce să scriem pe blog

- Scriem ceea ce știm despre un subiect. De exemplu, dacă avem cunoștințe bune despre computer, putem scrie despre acest subiect.
- Putem împărtăși din experiența noastră și a altora. Putem scrie, de asemenea, ceea ce am câștigat din această experiență, ceea ce am învățat.
- Detaliile cercetării personale. [2]

1.2 E-Commerce sau Electronics Commerce

Este o metodologie a afacerilor moderne care abordează nevoia organizațiilor de afaceri, a furnizorilor și a clienților de a reduce costurile și de a îmbunătăți calitatea produselor și serviciilor, sporind în același timp viteza de livrare. Comerțul electronic se referă la schimbul de informații în afaceri fără hârtie prin următoarele modalități.

- Schimbul electronic de date (EDI)
- Poșta electronică (e-mail)

- Tabele electronice cu buletine
- Transfer electronic de fonduri (EFT)
- Alte tehnologii bazate pe rețea [2]

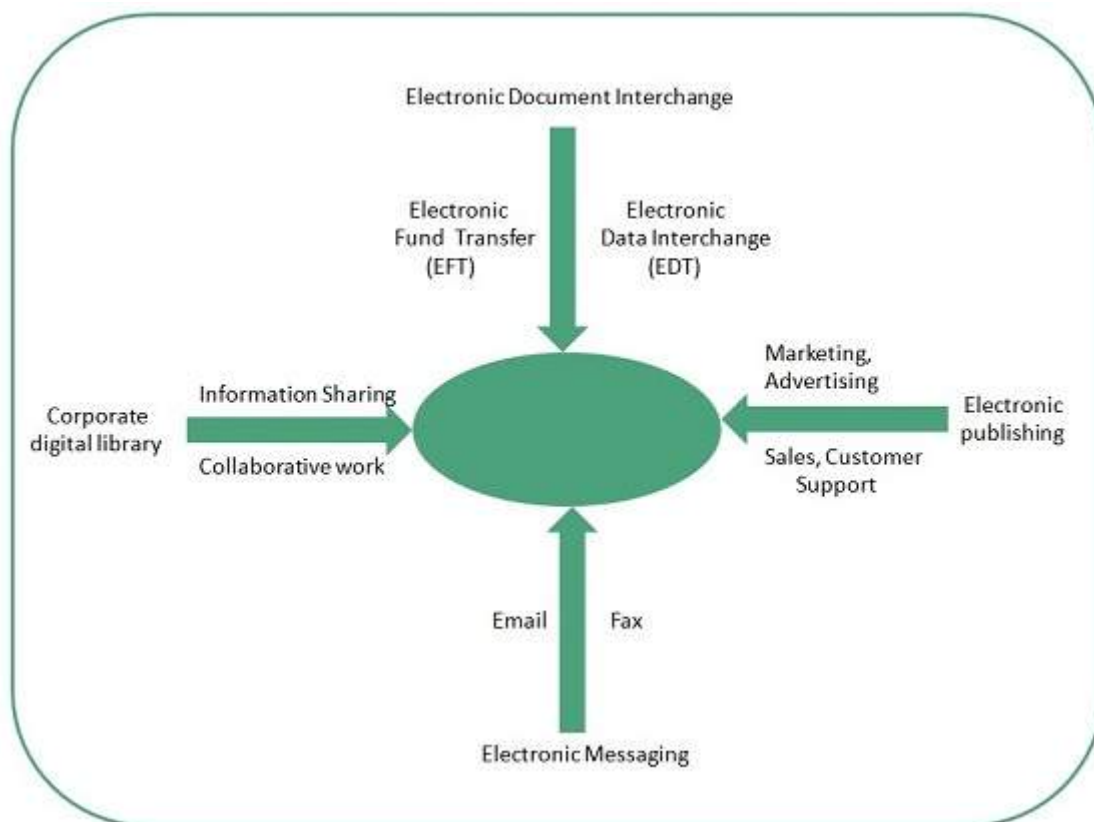


Figura 1.2 Modalități de schimb de informații [2]

Comerțul electronic oferă următoarele caracteristici:

PLATA NON-NUMERAR

Comerțul electronic permite utilizarea cardurilor de credit, cardurilor de debit, cardurilor inteligente, transferul electronic de fonduri prin intermediul site-ului băncii și altor modalități de plată electronică.

DISPONIBILITATEA SERVICIULUI 24X7

Comerțul electronic automatizează afacerile întreprinderilor iar serviciile pe care le oferă clienților sunt disponibile oricând, oriunde. În acest domeniu, 24x7 se referă la 24 de ore din fiecare șapte zile dintr-o săptămână.

PUBLICITATE / MARKETING

Comerțul electronic mărește vizibilitatea publicității produselor și serviciilor întreprinderilor. Ajută la o mai bună gestionare a produselor / serviciilor de marketing.

VÂNZĂRI ÎMBUNĂTĂȚITE

Utilizând Comerțul electronic, comenzile pentru produse pot fi făcute oricând, oriunde, fără intervenția omului. În acest fel cresc vânzările.

SUSTINERE

E-Commerce oferă diverse modalități de asistență prealabilă vânzărilor și post-vânzare pentru a oferi servicii mai bune clienților. [2]

PORTOFOLIU

Portofoliul online este o colecție de imagini, videoclipuri, e-mailuri, bloguri și hyperlink-uri care sunt gestionate online. Acesta poate fi văzut ca un fel de înregistrare care oferă dovezi reale despre realizări.

Există trei tipuri de portofolii online:

1. Dezvoltare (de exemplu, de lucru)
2. Reflecție (de exemplu, învățare)
3. Prezentare (de ex. Vitrină)

Portofoliul de **dezvoltare** conține toate lucrurile pe care o persoană le-a făcut într-o perioadă de timp.

Portofoliul **de reflecție** conține ideile personale asupra conținutului.

Portofoliul online **de prezentare** se referă la ce am realizat într-o anumită activitate. [2]

Proiectarea site-urilor web

Proiectarea web are legătură directă cu aspectul vizual al unui site Web. Designul web eficient este necesar pentru a comunica ideile în mod eficient. Acesta este un subset în dezvoltarea web.

Puncte cheie

Planul de proiectare trebuie să includă următoarele:

- Detalii despre arhitectura informațiilor.
- Structura planificată a site-ului.
- O hartă a paginilor [3]

Wireframe se referă la un ghid vizual pentru paginile web. Ajută la definirea structurii site-ului web, prin conectarea paginilor web și aspectul elementelor vizuale.

Următoarele lucruri sunt incluse într-o rețea wireframe:

- Elemente de grafică primară;
- Plasarea titlurilor și subtitlurilor;
- Cum sunt structurate structurile simple;
- Apeluri la acțiune;
- Blocuri de text.

Wireframe poate fi creat folosind un program precum Visio, dar putem utiliza și un stilou și o hârtie. [3]

1.3 Dezvoltarea site-urilor web

Dezvoltarea web se referă la construirea site-ului Web și implementarea pe web. Dezvoltarea web necesită utilizarea limbajelor de scripting atât la server, cât și la client.

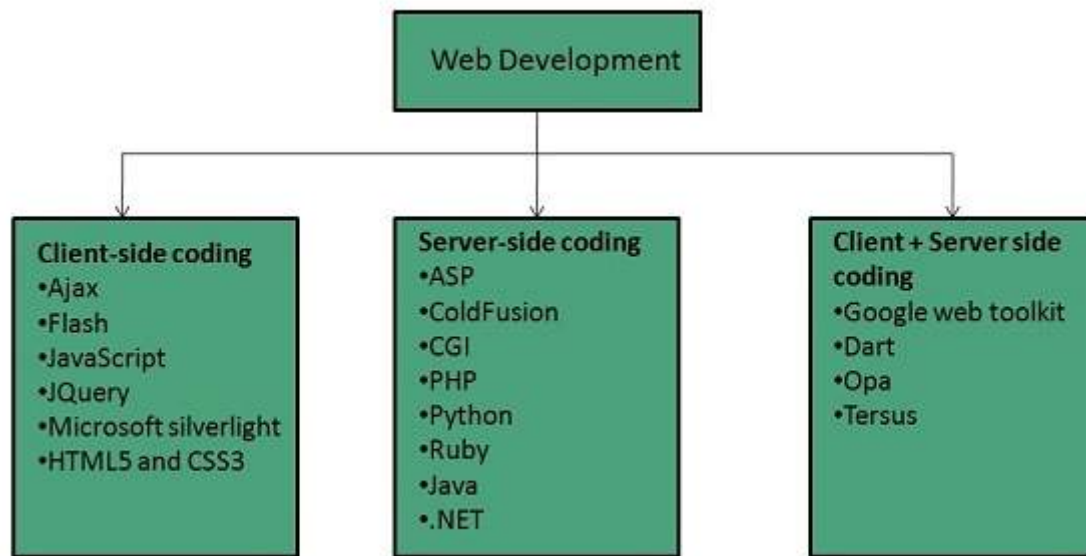


Figura 1.3 Dezvoltarea site-urilor web [4]

Înainte de a dezvolta un site web trebuie să ținem cont de câteva aspecte, cum ar fi:

- Ce conținut să pun pe site-ul web?
- Cine o să-l găzduiască?
- Cum fac să fie interactiv?
- Cum să creez un site web prietenos pentru motoarele de căutare?
- Cum să securizez frecvent codul sursă?
- Cum va fi afișat designul site-ului web în diferite browsere?
- Meniurile de navigare vor fi ușor de utilizat?
- Dacă se va încărca rapid site-ul web?
- Cât de ușor îi va fi vizitatorului să găsească detalii importante despre site-ul web?
- Cât de eficiente sunt foile de stil folosite pe site-urile mele web? [4]

Procesul de dezvoltare web

Procesul de dezvoltare web include toți pașii care trebuie făcuți pentru a construi un site atractiv, eficient și receptiv. Acești pași sunt prezentați în următoarea diagramă:

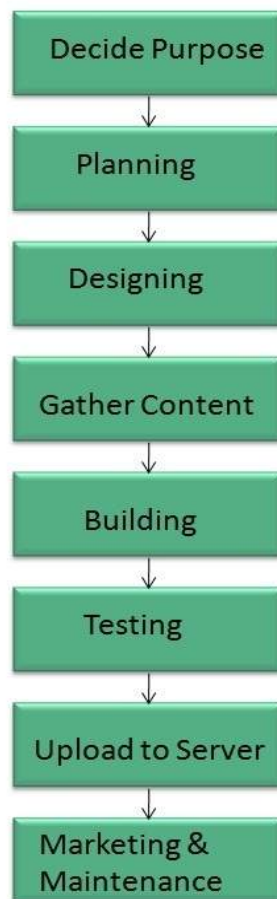


Figura 1.4 Pașii în procesul de dezvoltare web [4]

Instrumentele de dezvoltare web ajută dezvoltatorul să testeze și să depaneze site-urile Web. Acum instrumentul de dezvoltare web vine cu browserele web ca add-on-uri. Toate browserele web au construit instrumente pentru acest scop.

Aceste instrumente permit dezvoltatorului web să utilizeze HTML, CSS și JavaScript etc. Acestea sunt accesate prin plasarea cursorului peste un element dintr-o pagină web și selectarea "Inspect Element" din meniul contextual. [4]

Caracteristici

În continuare prezentăm aspectele comune ale fiecărui instrument de dezvoltare web:

HTML ȘI DOM

Vizualizatorul HTML și DOM vă permite să vedeți DOM așa cum este redat. De asemenea, permite modificarea HTML și DOM și pot fi văzute modificările în pagină imediat după efectuarea acestora.

ASPECTE DE PAGINĂ WEB, RESURSE ȘI INFORMAȚII DESPRE REȚEA

Instrumentele de dezvoltare web ajută de asemenea la inspectarea resurselor încărcate și disponibile pe pagina web.

PROFILARE ȘI AUDIT

Profilarea se referă la obținerea de informații despre performanța unei pagini web sau a unei aplicații web, iar auditarea oferă sugestii dezvoltatorilor, după analizarea paginii, pentru optimizări asupra îmbunătățirii timpului de încărcare a paginii.

Abilități cerute

Pentru a fi un dezvoltator web de succes, trebuie să avem următoarele abilități:

- Înțelegerea scripting-ului client - server.
- Crearea, editarea și modificarea de șabloane pentru un CMS sau un cadru de dezvoltare web.
- Realizarea testelor utilizatorilor.
- Testarea conformității cu standardele specificate.
- Accesibilitate în zona clientului.
- Programarea cu JavaScript, PHP și JQuery etc. [4]

1.4 Publicarea paginilor web

Publicarea pe site este procesul de încărcare a conținutului pe internet.

Include:

- încărcarea fișierelor
- actualizarea paginilor web
- postarea de bloguri

Site-ul web este publicat prin încărcarea fișierelor de pe serverul furnizat de compania de găzduire.

Pentru a publica site-ul avem nevoie de următoarele lucruri:

- Software de dezvoltare web
- Conexiune internet
- Server Web [6]

Software de dezvoltare web

Este folosit pentru construirea de pagini web pentru site-ul nostru. Dreamweaver și WordPress sunt două exemple de software de dezvoltare web.

Conexiune internet

Conexiunea la internet este necesară pentru a ne conecta la un server web localizat la distanță.

Server Web

Serverul Web este locația reală în care se află site-ul nostru.

Un server web poate găzdui site-uri unice sau multiple, în funcție de serviciul de găzduire pe care l-am plătit. [6]

1.5 Înregistrarea URL-urilor web

Numele de domeniu este partea din adresa noastră de internet care vine după "**www**". De exemplu, în **www.site.com** numele de domeniu este **site.com**.

Un nume de domeniu devine adresa noastră, astfel încât trebuie să avem grijă cum alegem numele pentru domeniu. Numele de domeniu trebuie să fie ușor de memorat și ușor de scris.

Extensii de domenii

Partea finală de la sfârșitul adresei de internet este cunoscută sub numele de nume de domeniu de nivel superior. Partea de după punct este cea mai înaltă în ierarhie.

Tabelul 1.1 Domenii **generice** de nivel superior [8]

Domeniu	Sens
.com	Afaceri Comerciale
.edu	Educație
.gov	Agenția guvernamentală americană
.int	Entitatea internațională
.mil	Armata americană
.net	Organizarea de rețele
.org	Organizație non profit

Înregistrarea unui nume de domeniu este foarte simplă. Trebuie să parcurgem următorii pași pentru a înregistra numele de domeniu dorit:

- Trebuie să alegem un nume care să reprezinte afacerea. Pentru a afla numele disponibile, introducem un nume la registratorul de nume de domenii comercial, cum ar fi GoDaddy.
- Dacă numele de domeniu introdus este disponibil, selectăm numele respectiv pentru domeniu.
- Se vor cere informații personale stocate în baza de date WHOIS.
- Apoi se cer informații de plată.

- Odată ce am terminat toți pașii de mai sus, utilizăm instrumentele lor pentru a încărca materialele pe site-ul nostru. [8]

1.6 Site-uri Web Hosting

Găzduirea web este un serviciu de furnizare a spațiului online pentru stocarea paginilor web. Aceste pagini web sunt disponibile prin **World Wide Web**. Companiile care oferă găzduirea site-ului web sunt cunoscute ca **gazde web**.

Serverele pe care se găsește site-ul web rămân pornite 24 x 7. Aceste servere sunt gestionate de companiile de gazduire web. Fiecare server are o adresă IP proprie. Deoarece adresele IP sunt greu de memorat, webmaster-ul își arată numele de domeniu la adresa IP a serverului pe care este stocat site-ul.

Nu putem găzdui site-ul pe computerul local deoarece, asta ar însemna să lăsăm PC- ul deschis mereu, din acest motiv se folosesc companiile de găzduire web. [10]

Tipuri de hosting

Tabelul 1.2 Tipuri de găzduire [10]

SN	Descriere de gazduire
1	<p>Gazduire comună</p> <p>În gazduirea comună, compania de găzduire pune mii de site-uri pe același server fizic. Fiecare client are propria alocare a spațiului web fizic și un set de limite de lățime de bandă. Deoarece toate site-urile web partajează aceeași memorie fizică, un site web de pe server care se confruntă cu mai mult trafic va afecta performanța tuturor site-urilor de pe server.</p>
2	<p>Virtual Private Server (VPS)</p> <p>Este, de asemenea, cunoscut ca Virtual Dedicated Server. Este un server care este împărțit în servere mai mici. În acesta clientului îi este dat partiție proprie, care este instalată cu propriul sistem de operare. Spre deosebire de hostingul partajat, VPS nu împarte memoria sau procesorul, ci mai degrabă alocă o anumită zonă de memorie și CPU, ceea ce înseamnă că orice problemă de pe o partiție VPS de pe aceeași unitate nu va afecta alți clienți VPS.</p>
3	<p>Server dedicat</p>

	<p>În acest tip de găzduire, un server dedicat este setat pentru un singur client. Este utilizat în mod obișnuit de companiile care au nevoie de putere, control și securitate.</p>
4	<p>Reseller Hosting</p> <p>Un revânzător acționează ca un om de mijloc și vinde spațiu de găzduire altcuiva.</p>
5	<p>Grid Hosting</p> <p>În loc să utilizeze un singur server, Grid Hosting trimite resursele pe un număr mare de servere. Este o modalitate destul de stabilă și flexibilă. Serverele pot fi adăugate sau scoase din rețea fără a se prăbuși sistemul.</p>

CAPITOLUL 2 INSTRUMENTELE SOFT FOLOSITE PENTRU REALIZAREA APLICAȚIEI

2.1 Limbajul PHP

2.1.1. Introducere în PHP

PHP este open source și a evoluat continuu. Prima versiune a fost scoasă de Rasmus Lerdorf în 1994.

"PHP este un acronim recursiv pentru PHP: Hypertext Preprocessor". [11]

Utilizări uzuale ale PHP [11]

- îndeplinește funcțiile sistemului;
- gestionează formularele;
- Adăugare, ștergere, modificarea elementelor din bazele de date cu ajutorul PHP;
- Accesarea variabilelor cookie-urilor și setarea acestor;
- restricționarea utilizatorilor să nu poată accesa anumite pagini Web;
- Poate cripta datele.

PHP este un limbaj de script pe partea serverului, care este încorporat în HTML. Este utilizat pentru a gestiona conținut dinamic, baze de date, urmărirea sesiunilor, și chiar pentru a construi site-uri întregi de comerț electronic.

Acesta vine integrat cu o serie de baze de date populare, inclusiv MySQL, PostgreSQL, Oracle, Sybase, Informix și Microsoft SQL Server.

PHP este plăcut în execuția sa, mai ales atunci când este compilat ca un modul Apache pe partea Unix. Serverul MySQL, odată pornit, execută chiar și interogări foarte complexe cu seturi uriașe de rezultate, oferite într-un timp foarte scurt.

PHP acceptă un număr mare de protocoale majore, cum ar fi POP3, IMAP și LDAP. PHP4 a adăugat suport pentru Java și arhitecturi de obiecte distribuite (COM și CORBA), făcând ca dezvoltarea pe n niveluri să fie posibilă pentru prima dată.

Cele mai importante caracteristici ale PHP, putem spune că sunt: simplitatea, eficiența, securitatea, flexibilitatea și familiaritatea.

Pentru a dezvolta și rula pagini web PHP, trebuie să fie instalate trei componente vitale pe sistemul computerului:

Server Web - PHP va funcționa cu aproape toate software-urile Web Server, inclusiv Internet Information Server (IIS), dar cel mai des utilizat este Apache Server care este disponibil gratuit.

Baza de date - PHP funcționează practic cu toate software-urile bazei de date, inclusiv Oracle și Sybase, dar cel mai frecvent utilizată este baza de date MySQL disponibilă gratuit.

Parser PHP - Pentru a procesa instrucțiunile scriptului PHP, trebuie să fie instalat un parser pentru a genera ieșire HTML care poate fi trimisă browserului Web. [12]

2.1.2. Câteva referiri la limbajul PHP

Variabile în PHP

Domeniul de aplicare poate fi definit ca fiind intervalul de disponibilitate al unei variabile pentru programul în care este declarată. Variabilele PHP pot fi una dintre cele patru tipuri:

- Variabile locale;
- Parametrii funcției;
- Variabile globale;
- Variabile statice. [13]

Denumirea unei variabile

Regulile pentru denumirea unei variabile sunt:

- Numele variabilei trebuie să înceapă cu un caracter litera sau underscore.
- Un nume de variabilă poate fi format din numere, litere, underscore(_), dar nu se pot folosi caractere ca +, -, %, (,), &, etc

Constante

O constantă este un identificator pentru o valoare simplă. O valoare constantă nu se poate modifica în timpul executării scriptului. În mod implicit, o constantă este sensibilă la minuscule. Prin convenție, identicatorii constanți sunt întotdeauna formați din litere mari. Un nume constant începe cu o literă sau un underscore, urmată de un număr de litere, numere sau _. Dacă definim o constantă, ea nu poate fi niciodată schimbată. [14]

Pentru a defini o constantă, trebuie să utilizăm funcția `define()` și pentru a extrage valoarea unei constante, trebuie doar să specificăm numele acesteia. Spre deosebire de variabile, nu este nevoie să avem o constantă cu un \$. De asemenea, putem utiliza funcția `constant()` pentru a citi valoarea unei constante dacă dorim să obținem numele constantei dinamic.

Constantă ()

După cum indică numele, această funcție va returna valoarea constantei.

Acest lucru este util atunci când dorim să preluăm valoarea unei constante, dar nu-i cunoaștem numele, adică este stocată într-o variabilă sau returnată de o funcție.

```
<?php
define("MINSIZE", 50);
echo MINSIZE;
echo constant("MINSIZE"); // același lucru ca și linia precedentă
?>
```

Doar datele scalare (boolean, integer, float și string) pot fi conținute în constante.

Operatori

- Operatori aritmetici
- Operatorii de comparare
- Operatori logici (sau relaționali)
- Operatori de alocare
- Operatori subordonați (sau ternari) [15]

Un **tablou** (array) este o structură de date care stochează unul sau mai multe tipuri similare de valori într-o singură variabilă. De exemplu, dacă dorim să stocăm 100 de numere, în loc să definim 100 de variabile, este mult mai ușor să definim un tablou de lungimea 100.

Există trei tipuri diferite de tablouri și fiecare valoare a tabloului este accesată utilizând un ID care se numește index de matrice.

Matrice numerică - Un tablou cu un index numeric. Valorile sunt stocate și accesate în mod liniar.

Matrice asociativă - Un tablou cu șiruri de caractere ca index. Aceasta stochează valorile elementelor în asociere cu valorile cheie, mai degrabă decât într-o ordine strictă a indexului liniar.

Matrice multidimensională - O matrice care conține una sau mai multe linii și valori este accesată utilizând mai mulți indicatori. [16]

Exemplu

```
<html>
  <body>

    <?php
      /* prima metoda de creare a tabloului.. */
      $numere = array( 1, 2, 3, 4, 5);
      foreach( $numere as $value ) {
        echo "Value is $value <br />";      }
      /* a doua metoda de creare a tabloului. */
      $numere[0] = "one";
      $numere[1] = "two";
      $numere[2] = "three";
      $numere[3] = "four";
      $numere[4] = "five";
      foreach( $numere as $value ) {
        echo "Value is $value <br />";      }
    ?>

  </body>
</html>
```

PHP oferă un număr mare de variabile predefinite pentru orice script pe care rulează. Acesta oferă un set suplimentar de tablouri predefinite care conțin variabile de la serverul web, mediul și intrarea utilizatorilor. Aceste noi rețele se numesc superglobals. [17]

Toate variabilele următoare sunt disponibile automat în fiecare domeniu de aplicare.

Variabile - descriere

1.\$ GLOBALS

Conține o referință la fiecare variabilă disponibilă în prezent în cadrul domeniului global al scenariului. Cheile acestei matrice sunt numele variabilelor globale.

2.\$ _SERVER

Acesta este o matrice care conține informații precum anteturi, căi și locații de script-uri. Intrările din această matrice sunt create de serverul web. Nu există nicio garanție că fiecare server web va furniza oricare dintre acestea.

3.\$ _GET

O matrice asociată de variabile ce a trecut prin scriptul curent prin metoda HTTP GET.

4. \$ _POST

O matrice asociată de variabile ce a trecut prin scriptul curent prin metoda HTTP POST.

5. \$ _FILES

O matrice asociată de elemente încărcate în scriptul curent prin metoda HTTP POST.

6.\$ _REQUEST

O matrice asociată constând din conținutul \$ _GET, \$ _POST și \$ _COOKIE.

7.\$ _COOKIE

O matrice asociată de variabile ce a trecut prin scriptul curent prin intermediul modulelor cookie HTTP.

8.\$ _SESSION

O matrice asociată care conține variabilele de sesiune disponibile pentru scriptul curent.

9.\$ _PHP_SELF

Un șir care conține numele fișierului script PHP în care este apelat.

10.\$ php_errormsg

\$ php_errormsg este o variabilă care conține textul ultimului mesaj de eroare generat de PHP.

[17]

2.2 Limbajul MySQL

MySQL este cel mai popular sistem de gestionare a bazelor de date SQL relațional Open Source. Acesta este unul dintre cele mai bune RDBMS utilizate pentru dezvoltarea diverselor aplicații software bazate pe web. MySQL este dezvoltat, comercializat și susținut de MySQL AB, din Suedia.

O bază de date este o aplicație separată care stochează o colecție de date. Fiecare bază de date are una sau mai multe API-uri distincte pentru crearea, accesarea, gestionarea, căutarea și reproducerea datelor pe care le deține.

În prezent, folosim sisteme de gestionare a bazelor de date relaționale (RDBMS) pentru a stoca și gestiona un volum imens de date. Aceasta se numește bază de date relațională, deoarece toate datele sunt stocate în tabele diferite și relațiile sunt stabilite folosind chei primare sau alte chei cunoscute sub numele de chei străine. [18]

MySQL funcționează foarte bine în combinație cu diverse limbaje de programare, cum ar fi PERL, C, C ++, JAVA și PHP. Dintre acestea, PHP este cel mai popular datorită capacităților sale de dezvoltare a aplicațiilor web. [19]

2.2.1 Crearea și utilizarea unei baze de date

După ce știm cum să introducem instrucțiuni SQL, suntem gata să accesăm o bază de date.

Să presupunem că avem diverse animale de companie în casa și dorim să urmărim diferite tipuri de informații despre ele. Putem face acest lucru prin crearea de tabele pentru a menține informațiile și pentru a le încărca cu informațiile dorite. Apoi putem răspunde la diverse feluri de întrebări despre animale prin preluarea datelor din tabele. Această secțiune arată cum să efectuăm următoarele operații:

- Creăm o bază de date
- Creăm un tabel
- Încercăm datele în tabel
- Preluăm datele din tabel în diferite moduri
- Utilizăm mai multe tabele [20]

Baza de date a animalelor este simplă, dar nu este dificil să ne gândim la situațiile din lumea reală în care ar putea fi folosit un tip asemănător de bază de date. De exemplu, o astfel de bază de date ar putea fi utilizată de către un fermier care să țină evidența animalelor, sau de către un medic veterinar care să țină evidența dosarelor pacienților.

Crearea unei tabele

```
mysql> CREATE DATABASE nume;
```

Afișarea bazei de date

```
SHOW TABLES;
```

```
mysql> SHOW TABLES;
```

Partea mai dificilă este să decidem cum ar trebui să fie structura bazei dvs. de date: de ce tabele avem nevoie și ce coloane ar trebui să conțină fiecare dintre ele.

Avem nevoie de un tabel care conține o înregistrare pentru fiecare dintre animalele de companie. Acest lucru poate fi numit tabel de animale, și ar trebui să conțină, ca minim, numele fiecărui animal. Deoarece numele în sine nu este foarte interesant, tabelul ar trebui să conțină și alte informații. De exemplu, în cazul în care mai mult de o persoană din familie păstrează animalele de companie, poate dorim să enumerăm proprietarul fiecărui animal. S-ar putea dori, de asemenea, să înregistrăm unele informații descriptive de bază, cum ar fi speciile și sexul.

Ce spunem de vârsta? Acest lucru ar putea fi de interes, dar nu este bine să o stocăm într-o bază de date. Vârsta se modifică cu timpul, ceea ce înseamnă că va trebui să actualizăm înregistrările de multe ori. În schimb, este mai indicat să stocăm de exemplu data nașterii. Apoi, ori de câte ori avem nevoie de vârstă, putem să o calculăm ca diferența dintre data curentă și data nașterii. MySQL oferă funcții pentru efectuarea de calcule aritmetice, deci nu este așa dificil.

Păstrarea datei de naștere mai degrabă decât vârsta are și alte avantaje:

- Putem utiliza baza de date pentru activități cum ar fi generarea de mementouri pentru viitoarele zile de naștere.
- Putem calcula vârsta în raport cu alte date decât data curentă. Dacă stocăm data când a murit, putem calcula cu ușurință cât a trăit un animal de companie.

Putem să introducem și alte tipuri de informații care ar fi utile în tabelul pentru animalele de companie, dar cele identificate până acum sunt suficiente: numele, proprietarul, specia, sexul, nașterea și data când a murit. [21]

Utilizăm o instrucțiune CREATE TABLE pentru a specifica aspectul tabelului:

Ex: `mysql> CREATE TABLE animal (nume VARCHAR(20), stapan VARCHAR(20), specie VARCHAR(20), sex CHAR(1), nastere DATE, moarte DATE);`

name	owner	species	sex	birth	death
Fluffy	Harold	cat	f	1993-02-04	
Claws	Gwen	cat	m	1994-03-17	
Buffy	Harold	dog	f	1989-05-13	
Fang	Benny	dog	m	1990-08-27	
Bowser	Diane	dog	m	1979-08-31	1995-07-29
Chirpy	Gwen	bird	f	1998-09-11	
Whistler	Gwen	bird		1997-12-09	
Slim	Benny	snake	m	1996-04-29	

Figura 2.2 Exemplu tabel de date MySQL [22]

2.3 CSS

CSS (Cascading Style Sheets) permite să cream pagini web minunate, acesta este un limbaj pentru specificarea modului în care documentele sunt prezentate utilizatorilor - cum sunt acestea desenate, prezentate, etc.

Un document este, de obicei, un fișier text structurat folosind un limbaj de markup - HTML este cel mai obișnuit limbaj de markup, dar vom întâlni și alte limbaje de marcaje, cum ar fi SVG sau XML.

Prezentarea unui document unui utilizator înseamnă convertirea acestuia într-o formă utilizabilă pentru publicul nostru. Browserele, cum ar fi Firefox, Chrome sau Internet Explorer, sunt concepute pentru a prezenta documente vizual, de exemplu, pe un ecran de computer, un proiector sau o imprimantă. [23]

Un set de proprietăți, care au valori setate să actualizeze modul în care este afișat conținutul HTML, de exemplu, vreau ca lățimea elementului meu să fie de 50% din elementul părinte și fundalul să fie roșu.

Selector care selectează elementele pe care dorim să aplicăm valorile actualizate ale proprietăților. De exemplu, doresc să aplic regulă CSS tuturor paragrafelor din documentul HTML.

Ex:

```
<html>
  <head>
    <meta charset="utf-8">
    <title>primul meu css</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Buna ziua!</h1>
    <p>acesta este primul meu paragraf</p>
  </body>
</html> [24]
```

La nivelul său cel mai de bază, CSS este format din două blocuri:

Proprietăți: Identificatorii care pot fi citați de către oameni care indică caracteristicile stilistice (de ex. Font, lățime, culoarea de fundal) pe care dorim să le modificăm.

Valori: Fiecare proprietate specificată are o valoare care indică modul în care dorim să schimbăm aceste caracteristici stilistice (de ex. Ce dorim să modificăm la font, lățimea sau culoarea de fundal).

O proprietate asociată cu o valoare se numește o declarație CSS. Declarațiile CSS sunt plasate în blocurile declarației CSS. În final, blocurile de declarații CSS sunt asociate cu selectori pentru a produce seturi de reguli CSS (sau Reguli CSS). [24]

În CSS, selectorii sunt folosiți pentru a direcționa elementele HTML pe paginile noastre web pe care dorim să le stilizăm. Există o gamă largă de selectori CSS, permițând o precizie cu granulație fină la selectarea stilului elementelor. În următoarele rânduri vom trece prin tipuri diferite în detaliu, văzând cum funcționează.

A CSS ruleset (or rule):

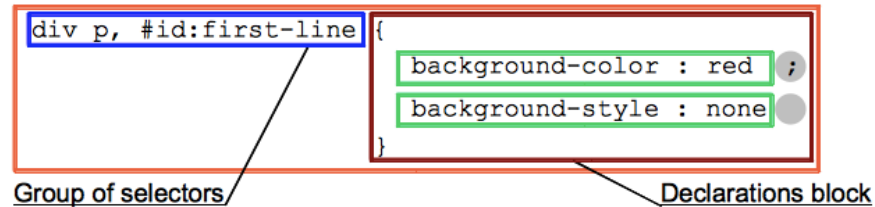


Figura 2.3 Tipuri de selectori [25]

Selectorii pot fi împărțiți în următoarele categorii:

Selectorii simpli: poziționăm unul sau mai multe elemente pe baza tipului, clasei sau id-ului elementului.

Selector de atribute: Se poziționează unul sau mai multe elemente bazate pe valorile acestor attribute / atributului.

Pseudo-clase: se poziționează unul sau mai multe elemente care sunt într-o stare identică, cum ar fi un element care asigură plutirea indicatorului mouse-ului sau o casetă de selectare care este dezactivată sau un element care este primul copil al acesteia ca părinte în arborele DOM.

Pseudo-elemente: potrivim una sau mai multe părți din conținut care se găsește într-o poziție stabilită în raport cu un element, ex. primul cuvânt din fiecare paragraf sau conținutul generat care apare chiar înaintea unui element.

Combinatii: Acestea nu sunt ele însele selectori, ci modalități de a combina doi sau mai mulți selectori în moduri utile pentru selecții foarte specifice. De exemplu, am putea selecta numai paragrafele care sunt descendenți direcți ai div-urilor sau paragrafele care sunt poziționate imediat după titluri.

Selectorii multipli: Din nou, aceștia nu sunt selectori separați; ideea este că putem pune selectori multipli pe aceeași foaie CSS, separate prin virgule, putem aplica un singur set de declarații elementelor selectate de aceștia.

Textul fundamental și stilul fontului

Listele de stil

Listele se comportă ca orice alt text în cea mai mare parte, dar există câteva proprietăți CSS specifice listelor despre care trebuie să știm și câteva bune practici pe care trebuie să le luăm în considerare.

Link-uri de styling

Atunci când facem legătura de stil, este important să înțelegem cum să utilizăm în mod eficient clasele cu pseudo-clase pentru a stabili legăturile de stil și cum să creăm linkuri de stil pentru utilizarea caracteristicilor variate pentru interfețe variate, cum ar fi meniurile de navigare și filele.

Fonturi Web

Acestea permit să descărcăm fonturi personalizate pentru pagina web, pentru a permite un stil de text personalizat și mai variat. [24]

2.4 HTML

HTML este locul unde începe magia proiectării paginilor web. Nu este prezentativ - pentru asta este CSS - HyperText Markup Language este un mod simplu și elegant de a structura conținutul.

Structura de bază a unui document HTML include etichete care înconjoară conținutul și aplică semnificația acestuia.

Ex

```
<!DOCTYPE html>
<html>
<body>
  This is my first web page
</body>
</html>
```

Figura 2.4 Exemplu simplu cod HTML [26]

Prima linie din partea de sus, `<!DOCTYPE html>`, este o declarație de tip document și permite browserului să cunoască HTML. Este foarte important să reținem acest lucru. Dacă nu, navigatorii vor presupune că nu știți cu adevărat ce faceți și că acționați într-un mod foarte ciudat.

Pentru a reveni la acest punct, `<html>` este eticheta de deschidere care spune browserului conținutul dintre el și eticheta de închidere `</html>`. Ce este între `<body>` și `</body>` reprezintă conținutul principal al documentului care va apărea în fereastra browserului. [26]

Etichete de închidere

Nu toate etichetele au etichete de închidere ca aceasta (`<html>` `</html>`). Tag-ul break-break, de exemplu, arată astfel: `
` - o pauză de linie nu are continut, așa ca tag-ul stă singur. Ce trebuie să știm este că toate etichetele cu conținut între ele ar trebui să fie închise, sub formă de etichetă de deschidere → conținut → etichetă de închidere.

S-ar putea să întâlniți etichete "de auto-închidere", prin care o etichetă `br`, de exemplu, va arăta ca `
` în loc de simplu `
`. Aceasta este o rămășiță a XHTML, o formă de HTML bazată pe un alt limbaj de markup numită XML. HTML5 este mult mai permisiv decât XHTML și va fi mulțumit de oricare format. Unii dezvoltatori preferă într-un fel, alții îl preferă pe celălalt. [26]

Atribute

Etichetele pot avea, de asemenea, atribute. Atributele apar în interiorul etichetei de deschidere, iar valorile lor se sflă între ghilimele. Arată ceva de genul `<tag attribute = "value"> Conținut</tag>`.

Încă o dată, ghilimele nu sunt întotdeauna esențiale, dar este o convenție de bună practică pe care alegem să o folosim pentru consistență și claritate. [26]

Elemente

Elementele sunt secvențele care alcătuiesc pagini web. Ați spune, de exemplu, că tot ce este între (și include) etichetele <body> și </ body> este corpul paginii web. Ca un alt exemplu, în timp ce "<title>" și "</ title>" sunt etichete, "<title> Rumble Stiltskin </ title>" este un element de titlu. [26]

Liste

Listele neordonate și listele ordonate funcționează în același mod, cu excepția faptului că prima este utilizată pentru liste non-secvențiale cu elemente de listă, de obicei precedate de simboluri, iar ultima este pentru liste succesive, care sunt în mod normal reprezentate de numere incrementate.

Eticheta "ul" este folosită pentru a defini liste neordonate, iar **eticheta "ol"** este utilizată pentru a defini liste ordonate. În interiorul listelor, eticheta "li" este utilizată pentru a defini fiecare element din listă. [27]

Ex:

```
<ul>
  <li>To learn HTML</li>
  <li>To show off</li>
  <li>Because I've fallen in love with my computer and want to give her some HTML
loving.</li>
</ul> [27]
```

"H" și "T" din "HTML" reprezintă "hipertext", ceea ce înseamnă în principiu un sistem de text legat.

O etichetă de **ancoră** (a) este utilizată pentru a defini o legătură, dar trebuie să adăugăm și ceva la eticheta de ancoră - destinația linkului.

```
<h2>unde ai gasit tutorialul</h2>
<p><a href="http://www.tutoriale.ro">Tutoriale HTML</a></p>
```

Destinația linkului este definită în atributul href al etichetei. Linkul poate fi absolut, cum ar fi "http://www.tutoriale.ro", sau poate fi relativ la pagina curentă.

Deci, dacă, de exemplu, am avut un alt fișier numit "flyingmoss.html" în același director, atunci linia de cod ar fi pur și simplu text . [28]

Un link nu trebuie să facă legătura cu alt fișier HTML, ci poate să facă legătura cu orice fișier de oriunde pe web.

Un link poate trimite, de asemenea, un utilizator către o altă parte a aceleiași pagini pe care se află. Putem adăuga un atribut "id" la aproape orice etichetă, de exemplu, <h2 id = "moss"> text </ h2>, apoi conectați-l folosind ceva similar: mergeți la text . Dacă selectați acest link, veți derula pagina direct la elementul cu acel cod. [28]

Eticheta img este folosită pentru a plasa o imagine într-un document HTML și arată astfel:

Ex:

```

```

Atributul src indică browserului unde să găsească imaginea. Ca o etichetă, aceasta poate fi absolută, așa cum demonstrează exemplul de mai sus, dar este de obicei relativă. De exemplu, dacă creăm

propria imagine și o salvăm ca "poză.jpg" într-un director numit "imagini", atunci codul va fi `
 <tr>
 <td>Rândul 1, celula 1</td>
 <td>Rândul 1, celula 2</td>
 </tr>
 <tr>
 <td>Rândul 2, celula 1</td>
 <td>Rândul 2, celula 2</td>
 </tr>
</table>
```

Elementul "table" definește tabelul, elementul "tr" definește un rând de table, iar "td" definește o celulă de date. Acestea trebuie să fie închise în etichetele "tr", după cum se arată mai sus.

Dacă vă imaginați o masă 3x4, care este de 12 celule, ar trebui să existe patru elemente "tr" pentru a defini rândurile și trei elemente "td" din fiecare rând, făcând un total de 12 elemente "td". [30]

**Formularele** sunt utilizate pentru colectarea datelor introduse de un utilizator. Ele pot fi folosite ca interfață pentru o aplicație web, de exemplu, sau pentru a trimite date pe web.

Singure, formularele nu sunt de obicei utile. Acestea tind să fie utilizate împreună cu un limbaj de programare pentru a procesa informațiile introduse de utilizator. Etichetele de bază utilizate în HTML-ul real al formularelor sunt formularul, intrarea, textarea, selectarea și opțiunea.

Dacă utilizăm un formular pentru ca un utilizator să trimită informații, este necesar un atribut de acțiune pentru a indica forma în care conținutul acestuia va fi trimis. [31]

Atributul "method" indică formularului modul în care vor fi transmise datele din acesta pentru a obține valoarea dorită, prin metoda post, care trimite în mod invizibil informațiile despre formular și metoda "get" este utilizată pentru fragmente mai scurte de informații nesensibile - putem vedea, de exemplu, informațiile pe care le-am trimis într-o căutare a unui site web pentru a apărea în adresa web a paginii cu rezultatele căutării. Metoda "post" este folosită pentru trimiteri mai lungi și mai sigure, cum ar fi formularele de contact.

Exemplu de formular:

```
<form action = "processingscript.php" method = "post">
</Form> [31]
```

## Input

Eticheta de intrare este partea cea mai importantă din formular.

<input type = "text"> sau pur și simplu <input> este o casetă de text standard. Acest lucru poate avea, de asemenea, un atribut de valoare, care stabilește textul inițial în caseta de text.

<input type = "password"> este similar cu caseta de text, dar caracterele introduse de utilizator vor fi ascunse.

<input type = "checkbox"> este o casetă de selectare, care poate fi activată și dezactivată de către utilizator. Acest lucru poate avea, de asemenea, un atribut verificat (<input type = "checkbox" checked> - atributul nu necesită o valoare) și face ca starea inițială a casetei de control să fie activă.

<input type = "radio"> este similar cu o casetă de selectare, utilizatorul selectază doar un singur buton dintr-un grup. Acest lucru poate avea și un atribut verificat.

<input type = "submit"> este butonul care atunci când va fi selectat va trimite formularul. Putem controla textul ce va apărea pe butonul de trimitere cu atributul de valori, de exemplu, <input type = "submit" value = "Text pe un buton.">. [31]

## Textarea

Textarea este, în principiu, o casetă de text mare, cu mai multe linii. Numărul de rânduri și coloane poate fi definit cu atributele rândurilor și coloanelor, deși putem manipula dimensiunea conținutului folosind CSS.

```
<textarea rows = "5" cols = "20"> O încărcătură mare de text </ textarea> [31]
```

## Selectare

Eticheta selectare funcționează împreună cu eticheta de opțiuni pentru a face o selectare derulantă.

```
<select>
 <option> Opțiunea 1 </ option>
 <option> Opțiunea 2 </ option>
 <option value = "a treia opțiune"> Opțiunea 3 </ option>
</ select>
```

Când se trimite formularul, se va trimite valoarea opțiunii selectate. Această valoare va fi textul dintre eticheta opțiunii de deschidere și închidere selectată, cu excepția cazului în care este specificată o valoare explicită cu atributul "value", caz în care va fi trimisă aceasta. Deci, în exemplul de mai sus, dacă primul element este selectat, se va trimite "Opțiunea 1", dacă este selectat cel de-al treilea element, va fi trimisă "a treia opțiune". [31]

Similar cu atributul verificat al casetelor de selectare și al butoanelor radio, o etichetă de opțiune poate avea și un atribut selectat, pentru a porni cu unul dintre elemente deja selectat, de ex. <option selected> Dacia </ option> ar selecta în prealabil "Dacia" din opțiunile pe care le avem.

## 2.5 JavaScript

JavaScript este un limbaj de programare ușor de interpretat. Este conceput pentru a crea aplicații centrate pe rețea. Este gratuit și integrat cu Java. Acesta este foarte ușor de implementat, deoarece este integrat cu HTML. Javascript se găsește pretutindeni, este instalat pe fiecare browser web modern și astfel, nu aveți nevoie de nicio configurare de mediu specială. De exemplu, Chrome, Mozilla Firefox, Safari și fiecare browser pe care îl cunoașteți de astăzi, acceptă Javascript.

Mecanismul JavaScript din partea clientului oferă multe avantaje față de scripturile tradiționale din partea serverului CGI. De exemplu, se poate utiliza JavaScript pentru a verifica dacă utilizatorul a introdus o adresă de e-mail validă într-un câmp de formular, lucru pe care îl folosesc și eu în realizarea aplicației mele. Codul JavaScript este executat atunci când utilizatorul trimite formularul și numai dacă toate intrările sunt valide, acestea vor fi trimise la serverul web. [32]

JavaScript poate fi folosit pentru a captura evenimentele inițiate de utilizator, cum ar fi clicurile pe butoane, navigarea pe linkuri și alte acțiuni pe care utilizatorul le inițiază în mod explicit sau implicit.

### **Avantajele JavaScript**

Mai puțină interacțiune cu serverul – Se poate valida introducerea utilizatorului înainte de a trimite pagina pe server. Acest lucru economisește traficul serverului, ceea ce înseamnă o încărcare mai mică pe server și un timp de răspuns îmbunătățit.

Feedback imediat pentru vizitatori - Nu trebuie să aștepte reîncărcarea paginii pentru a vedea dacă au uitat să introducă ceva.

Interactivitate crescută - Se pot crea interfețe care reacționează atunci când utilizatorul trece deasupra lor cu un mouse sau le activează prin intermediul tastaturii.

Interfețe mai bogate – Acesta ajută la obținerea de elemente vizuale interesante, precum glisare și plasare de componente și afișarea de ferestre glisante pentru a oferi o interfață bogată vizitatorilor site-ului. [32]

Limbajul de programare poate fi implementat folosind instrucțiuni JavaScript care sunt plasate în etichetele `<script> ... </script>` HTML dintr-o pagină web. Etichetele `<script>`, care conțin JavaScript, se pot plasa oriunde în pagina web, dar în mod normal este recomandat să fie păstrate în etichetele de început `<head>`.

Eticheta script are două atribute importante: [33]

Limbă - Acest atribut specifică ce limbaj de scriptare utilizați. De obicei, valoarea acestuia va fi javascript.

Tipul - Acest atribut este ceea ce este recomandat acum pentru a indica limbajul de scriptare utilizat și valoarea acestuia trebuie setată la „text / javascript”.

Deci, bucata de cod JavaScript va arăta ca –

```
<script language = "javascript" type = "text/javascript">
 codul JavaScript
</script> [33]
```





## CAPITOLUL 3 PROIECTAREA DE DETALIU A APLICAȚIEI INFORMATICE

### 3.1. Definirea obiectivelor aplicației informatice

Prin proiectul *Proiectarea și implementarea unui magazin online de piese auto*, îmi propun să dezvolt o aplicație software - un magazin online de *piese auto* al unei firme care vinde *componente auto*. Firma în această perioadă de pandemie luptă să supraviețuiască în piață și pentru a obține mai multe comenzi au dorit să dezvolte un site web ca să facă cât mai ușor posibil vânzarea produselor. Aplicația a fost creată și pentru automatizarea vânzărilor de *piese auto*. Scopul creării site-ului este creșterea vânzărilor la toate categoriile de mașini, prin atragerea de noi clienți prin oferte periodice și prin fidelizarea vechilor clienți. Acest lucru presupune un design cât mai atractiv și un meniu ușor de folosit în navigarea pe site astfel încât utilizatorii să nu se rătăcească în căutarea produselor dorite. Clienții vor găsi un site securizat, cu referințe bune și o buna imagine creată pe piață prin care vor face comenzi.

Pentru a stabili care sunt obiectivele ce pot fi realizate cu ajutorul aplicației, trebuie stabilit un plan pas cu pas pentru a sprijini dezvoltarea normală a acestei afaceri. Primul obiectiv este oferirea clienților de informații despre firmă prin intermediul site-ului web și unul din modulele oferite este crearea de conturi personale cu ajutorul cărora vor face cumpărături. La nivelul bazei de date, acest pas se realizează cu crearea unui tabel care reține datele legate de utilizatori, și tabelul care păstrează datele legate de produsele prezentate.

Al doilea obiectiv care trebuie atins este de a atrage cât mai mulți vizitatori posibil printr-un design atragător adaptat pentru toate tipurile de dispozitive: laptop, telefon, tabletă. Astfel de sarcini implică introducerea în baza de date a unor imagini atractive pentru noile produse. La nivel de site, se recomandă promovarea site-ului prin adăugarea de produse noi care atrag clienți prin oferte și servicii de fidelizare a vechilor clienți.

Al treilea obiectiv este de a realiza suficiente vânzări online pentru a face din magazin un back office. Pentru a face acest lucru, actualizăm baza de date adăugând noi tabele sau alte câmpuri la tabelele existente. Acest lucru va facilita utilizatorilor să găsească tipul de produs pe care îl doresc.

#### Metodologie

Aplicația va ajuta firma în automatizarea vânzărilor, deoarece produsele vor fi distribuite automat, iar stocul fiecărui produs va fi înregistrat permanent. În plus, aplicația va fi foarte utilă pentru gestionarea *pieselor auto*, deoarece va înregistra strict produsele vândute și cele aflate în stoc. Un alt avantaj al acestei aplicații web este că putem vedea ce produse au fost vândute și care nu se mai vând, așa că vom ști ce produse pot fi stocate în număr mai mare și care nu merită vândute.

Limitările proiectului sunt cauzate de lipsa mărcilor de top cu care să coopereze compania și oferă orientări limitate clienților. Produsele afișate pe site trebuie să fie competitive în ceea ce privește prețul pentru a obține cât mai mult profit posibil, dar numărul și tipurile de piese auto pentru clienți sunt încă foarte mici. În ideea dezvoltării în continuare a proiectului, încercăm să găsim câteva soluții în viitorul apropiat. Un studiu privind răspândirea fenomenului de cumpărare și vânzare online arată că anumite lucruri pot fi realizate în continuare cu un cost minim și o sarcină de lucru moderată. Primul lucru care merită implementat pe site-ul web este o metodă de plată printr-un cont online ce poate fi utilizat la nivel global (de exemplu, PayPal).

Principalele funcționalități ale aplicației sunt:

- Posibilitatea de a utiliza un coș de cumpărături, în care clienții pot adăuga articole din oferta propusă;

- Posibilitatea personalizării contului;

- Afișarea unei liste cu toate piesele auto;

- Căutarea de piese după modelul mașinii, motor și producător;

- Acordarea de calificative/voturi pentru fiecare produs în parte;

- Utilizatorii pot comanda piese auto online și pot plăti la livrare;

- Posibilitatea contactării firmei printr-un formular de contact.

Baza de date

Datele și informațiile solicitate de aplicația auto sunt stocate în baza de date. Deoarece sql oferă posibilitatea de a construi orice tip de date sau informații într-un mod logic, acestea pot fi procesate cu ușurință. Mai mult, aceste documente sunt ușor de proiectat și pot fi înțelese de orice utilizator. Aplicația auto utilizează documente php și o bază de date auto cu 11 tabele pentru procesare. În cele ce urmează, voi introduce structura tabelor utilizate pentru stocarea datelor sau stocarea diferitelor setări specifice ale aplicației în aplicația automatizată.

## 3.2 Proiectarea bazei de date

Proiectul utilizează o bază de date MySQL pentru a stoca informații. Baza de date conține 10 tabele:

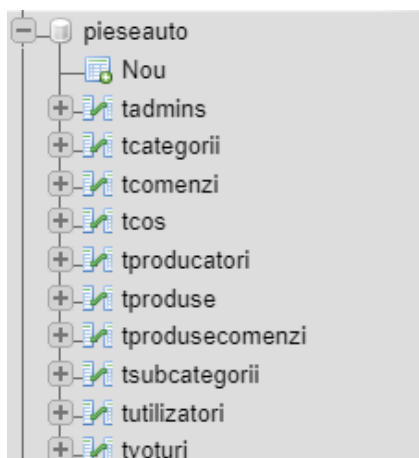


Figura 3.1 Tabele conținute în baza de date “piese auto”

### 1. Tabelul tadmins:

Este folosit pentru a stoca administratorii site-ului.





Name	Type	NULL	Default	Extras
 PRIMARY	fIdAdmin			unique
 fIdAdmin	int(11) unsigned	No		auto_increment
 fUsername	varchar(32)	Yes		
 fPassword	varchar(32)	Yes		

Figura 3.2 Tipul de date al atributelor din tabelul tadmins

Object Browser   Data Browser   SQL Editor		
fIdAdmin	fUsername	fPassword
1	admin	admin

Figura 3.3 Valorile introduse în attributele aflate în tadmins

Câmpuri:

- fIdAdmin: Id-ul înregistrării;
- fUsername: numele utilizator al administratorului;
- fPassword: parola administratorului.

### 2. Tabelul tcategorii:

Este folosit pentru stocarea mașinilor pentru care sunt pisele auto.

Exemple de mașini:

BMV

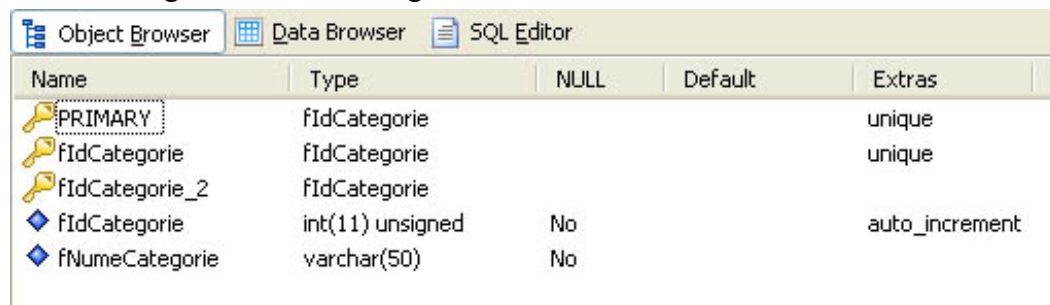
AUDI

DACIA

FORD etc

Câmpuri:

- fIdCategorie: Id-ul înregistrării;
- fNumeCategorie: Numele categoriei.



Name	Type	NULL	Default	Extras
PRIMARY	fIdCategorie			unique
fIdCategorie	fIdCategorie			unique
fIdCategorie_2	fIdCategorie			
fIdCategorie	int(11) unsigned	No		auto_increment
fNumeCategorie	varchar(50)	No		

Figura 3.4 Tipul de date al atributelor din tabelul tcategorii



		fIdCategorie	fNumeCategorie
<input type="checkbox"/>		1	BMV
<input type="checkbox"/>		2	AUDI
<input type="checkbox"/>		3	DACIA
<input type="checkbox"/>		4	CITROEN
<input type="checkbox"/>		5	FORD

Figura 3.5 Valorile introduse în atributele aflate în tcategorii

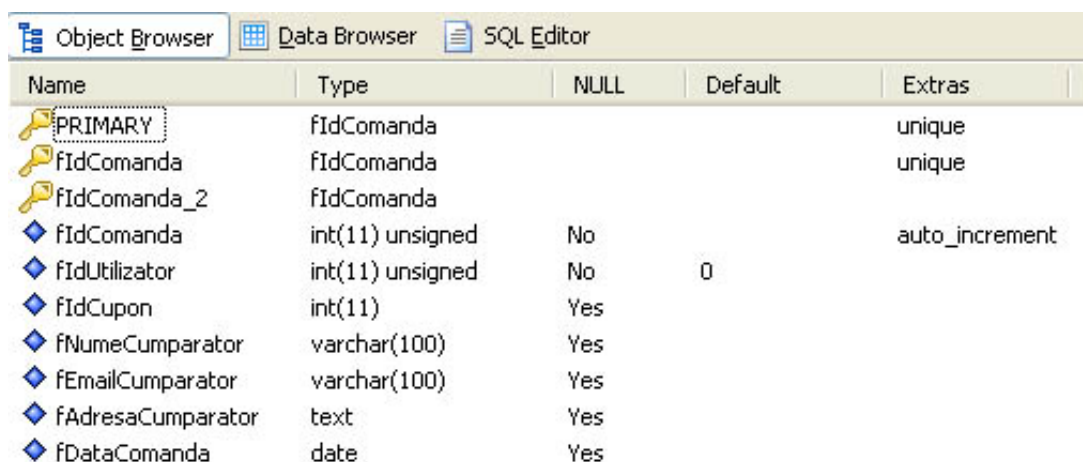
### 3. Tabelul tcomenzi:

Este folosit pentru a stoca informații despre comenzile plasate de utilizatori.

Câmpuri:

- fIdComanda: Id-ul înregistrării;
- fIdUtilizator: Id-ul utilizatorului ce a efectuat comandă;
- fIdCupon: Id-ul cuponului de reduceri. Dacă Id-ul este egal cu 0 cuponul nu este luat în considerare;
- fNumeCumparator: Numele cumpărătorului. Poate fi diferit de numele utilizatorului care a efectuat comanda;
- fEmailCumparator: Adresa email a cumpărătorului. Poate fi diferită de adresa email a utilizatorului care a efectuat comandă;

- fAdresaCumparator: Adresa cumpărătorului. Poate fi diferită de adresa utilizatorului care a efectuat comandă;
- fDataComanda: Dată la care a fost efectuată comandă.



Name	Type	NULL	Default	Extras
PRIMARY	fIdComanda			unique
fIdComanda	fIdComanda			unique
fIdComanda_2	fIdComanda			
fIdComanda	int(11) unsigned	No		auto_increment
fIdUtilizator	int(11) unsigned	No	0	
fIdCupon	int(11)	Yes		
fNumeCumparator	varchar(100)	Yes		
fEmailCumparator	varchar(100)	Yes		
fAdresaCumparator	text	Yes		
fDataComanda	date	Yes		

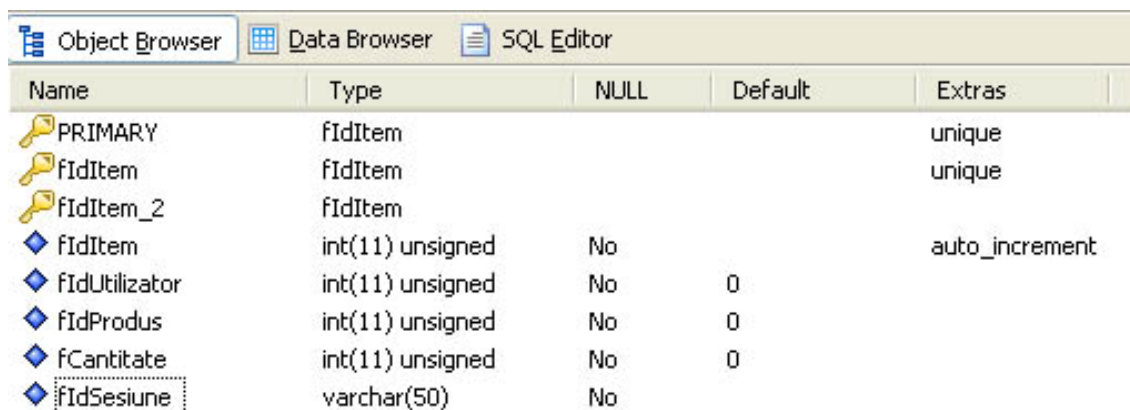
Figura 3.6 Tipul de date al atributelor din tabelul tcomenzi

#### 4. Tabelul tcos:

Este folosit pentru a stoca produse în coșul de cumpărături. După conectarea sau deconectarea din cont, conținutul coșului de cumpărături este golit. În mod similar, după plasarea unei comenzi, coșul de cumpărături este golit.

Câmpuri:

- fIdItem: Id-ul înregistrării;
- fIdUtilizator: Id-ul utilizatorului care efectuează cumpărăturile;
- fIdProdus: Id-ul produsului adăugat în coș;
- fCantitate: Cantitatea comandată;
- fIdSesiune: Id-ul sesiunii utilizatorului. Acesta este asignat de server și este unic pentru fiecare vizită.



Name	Type	NULL	Default	Extras
PRIMARY	fIdItem			unique
fIdItem	fIdItem			unique
fIdItem_2	fIdItem			
fIdItem	int(11) unsigned	No		auto_increment
fIdUtilizator	int(11) unsigned	No	0	
fIdProdus	int(11) unsigned	No	0	
fCantitate	int(11) unsigned	No	0	
fIdSesiune	varchar(50)	No		

Figura 3.7 Tipul de date al atributelor din tabelul tcos

## 5. Tabelul tproducatori:

Este folosit pentru a stoca informații despre producător.

Câmpuri:

- fIdProducator: Id-ul înregistrării;
- fNumeProducator: Numele producătorului;
- fUrl: Adresa web a paginii producătorului;
- fDescriere: Descrierea producătorului.





















+ Opțiuni			fIdProducator	fNumeProducator	fUrl	fDescriere
<input type="checkbox"/>			1	LAUBER	lauber.com	[BLOB - 0octeți]
<input type="checkbox"/>			2	Delphy	delphi.com	[BLOB - 0octeți]
<input type="checkbox"/>			3	Sofima	sofima.com	[BLOB - 0octeți]
<input type="checkbox"/>			4	SASIC	sasic.com	[BLOB - 0octeți]
<input type="checkbox"/>			5	HEPU	hepu.com	[BLOB - 0octeți]
<input type="checkbox"/>			6	OPTIMAL	optimal.com	[BLOB - 0octeți]
<input type="checkbox"/>			7	ASAM	asam.com	[BLOB - 0octeți]
<input type="checkbox"/>			8	ATE	ATE.COM	[BLOB - 0octeți]
<input type="checkbox"/>			9	RUVILLE	ruville.com	[BLOB - 0octeți]
<input type="checkbox"/>			10	FILTRON	filtron.com	[BLOB - 0octeți]

Figura 3.8 Valorile introduse în atributele aflate în tproducatori

## 6. Tabelul tproduse:

Este folosit pentru a stoca informații despre produse.

Câmpuri:

- fIdProdus: Id-ul înregistrării;
- fIdSubcategorie: Id-ul modelului de mașină din care face parte produsul;
- fIdProducator: Id-ul producătorului de care aparține acest produs;
- fNumeProdus: Numele produsului;
- fCodProdus: Codul produsului;
- fImagine: Calea către imaginea produsului. Imaginile sunt stocate în directorul “images/produse”;
- fDescriere: Descrierea detaliată a produsului;
- fSpecificatii: Specificațiile tehnice ale produsului;
- fPret: Prețul produsului, exprimat în RON, fără TVA;
- fLuniGarantie: Numărul lunilor de garanție oferite de producător/vânzător.

Object Browser   Data Browser   SQL Editor				
Name	Type	NULL	Default	Extras
PRIMARY	fIdProdus			unique
fIdProdus	fIdProdus			unique
fIdProdus_2	fIdProdus			
fIdProdus	int(11) unsigned	No		auto_increment
fIdSubcategorie	int(11) unsigned	No	0	
fIdProducator	int(11) unsigned	No	0	
fNumProdus	varchar(255)	No		
fCodProdus	varchar(50)	Yes		
fImagine	varchar(100)	Yes		
fDescriere	varchar(255)	Yes		
fSpecificatii	blob	Yes		
fPret	varchar(50)	No	0	
fLuniGarantie	varchar(25)	No	0	

Figura 3.9 Tipul de date al atributelor din tabelul tproduse

Sortare după cheie: Nici unul(a)										
+ Opțiuni										
	fIdProdus	fIdSubcategorie	fIdProducator	fNumProdus	fCodProdus	fImagine	fDescriere	fSpecificatii	fPret	fLuniGar
<input type="checkbox"/>	1	1	1	Alternator	e1	c1.jpg	Generator/alternator LAUBER 11.0379 Cod produs: 11...	[BLOB - 219octet]	106	24
<input type="checkbox"/>	2	1	2	Saboti frana	e1	c2.jpg	Set saboti frana DELPHI LS1283	[BLOB - 93octet]	13	1
<input type="checkbox"/>	3	2	3	Filtru aer	e2	c3.jpg	Filtru aer habitacul BMW Seria 6 E63 (E63) 630 i ...	[BLOB - 200octet]	70	3
<input type="checkbox"/>	4	2	4	Curea transmisie cu caneluri	e2	c4.jpg	Curea transmisie cu caneluri	[BLOB - 362octet]	150	3
<input type="checkbox"/>	5	3	5	Pompa apa	e3	c5.jpg	Pompa apa AUDI Q7 (4L) 3.0 TDI (150KW / 204CP) HEP...	[BLOB - 264octet]	200	3
<input type="checkbox"/>	6	3	1	Set placute frana	e3	c6.jpg	Set placute frana, frana disc AUDI Q7	[BLOB - 338octet]	120	3
<input type="checkbox"/>	7	4	1	Starter	e4	c7.jpg	Starter LAUBER 22.0513	[BLOB - 326octet]	18	0
<input type="checkbox"/>	8	4	6	Saiba arc	e4	c8.jpg	Saiba arc OPTIMAL F8-7372	[BLOB - 100octet]	12	0
<input type="checkbox"/>	9	5	7	Pompa ulei	e5	c9.jpg	Pompa ulei DACIA Logan (LS_) 1.5 dCi (LS04) (55KW ...	[BLOB - 87octet]	100	3
<input type="checkbox"/>	10	5	1	Starter	e5	c10.jpg	Starter LAUBER 22.5310	[BLOB - 131octet]	75	24
<input type="checkbox"/>	11	6	4	Suport motor	e6	c11.jpg	Suport motor DACIA Duster 1.6	[BLOB - 181octet]	75	0
<input type="checkbox"/>	12	6	8	Set placute frana	E6	c12.jpg	Set placute frana, frana disc DACIA Duster 1.6	[BLOB - 471octet]	87	0
<input type="checkbox"/>	13	7	1	Starter	e7	c13.jpg	Starter LAUBER	[BLOB - 334octet]	57	24
<input type="checkbox"/>	14	8	9	Set pompa apa+curea dintata	e8	c14.jpg	Set pompa apa+curea dintata CITROEN Xsara (N0) 1.8...	[BLOB - 163octet]	450	24
<input type="checkbox"/>	15	9	10	Filtru aer	e9	c15.jpg	Filtru aer FORD Focus Turnier 1.6	[BLOB - 228octet]	17	24

Figura 3.10 Valorile introduse în attributele aflate în tproduse

## 7. Tabelul tprodusecomenzi:

Conține articolele unei comenzi deja efectuate de către un utilizator. Este asemănător tabelului tcos.

Câmpuri:

- fIdArticol: Id-ul înregistrării;
- fIdComanda: Id-ul comenzii din care face parte articolul;
- fIdProdus: Id-ul produsului comandat;
- fCantitate: Cantitatea în care a fost comandat produsul.

fIdArticol	fIdComanda	fIdProdus	fCantitate
1	1	9	1
2	1	6	2
3	2	7	1
4	3	15	1
5	4	9	1

Figura 3.11 Valorile generate după efectuarea unor comenzi

## 8. Tabelul tsubcategorii:

Este folosit pentru a stoca informații despre modelul mașinii și despre motorizările acestora.

Exemple de motorizări:

Seria 5/518g(85 KW/115 CP)

Q7 3.0TDI (15KW/204CP)

LOGAN 1.5dCi (L504) ( 55KW/75CP )

A3 1.4TFSI (90KW/122CP) etc

Câmpuri:

- fIdSubcategorie: Id-ul înregistrării;
- fIdCategorie: Id-ul mașinei de care aparține modelul;
- fNumeSubcategorie: Numele modelului de mașină.

Object Browser   Data Browser   SQL Editor					
Name	Type	NULL	Default	Extras	
PRIMARY	fIdSubcategorie			unique	
fIdSubcategorie	fIdSubcategorie			unique	
fIdSubcategorie_2	fIdSubcategorie				
fIdSubcategorie	int(11) unsigned	No		auto_increment	
fIdCategorie	int(11) unsigned	No	0		
fNumeSubcategorie	varchar(100)	No			

Figura 3.12 Tipul de date al atributelor din tabelul tsubcategorii



fIdSubcategorie	fIdCategorie	fNumeSubcategorie
1	1	Seria 5/518g(85 KW/115 CP)
2	1	seria 6/ 630i (190 KW/258 CP)
3	2	Q7 3.0TDI (15KW/204CP)
4	2	A3 1.4TFSI (90KW/122CP)
5	3	LOGAN 1.5dCi (L504) ( 55KW/75CP )
6	3	DUSTER 1.6 16VLPG (77KW/105 CP)
7	4	AX 1.4 4x4948KW/65CP)
8	4	XSARA 1.8 VTS (85KW/116CP)
9	5	FOCUS 1.6 EcoBoost (134KW/102CP)
10	5	ACTIVA 1.8 GLX(84KW/114 CP)

Figura 3.13 Valorile introduse în attributele aflate în tsubcategorii

## 9. Tabelul tutilizatori:

Este folosit pentru a stoca informații despre utilizatorii înregistrați pe site. Trebuie să vă înregistrați pentru a comanda / vota / adăuga în cos produse.

Câmpuri:

- fIdUtilizator: Id-ul înregistrării;
- fNumeUtilizator: Numele utilizatorului;
- fParola: Parola utilizatorului;
- fEmail: Adresa email a utilizatorului;
- fAdresa: Adresa utilizatorului;
- fDataInregistrare: Dată la care utilizatorul s-a înregistrat pe website.

Object Browser   Data Browser   SQL Editor				
Name	Type	NULL	Default	Extras
PRIMARY	fIdUtilizator			unique
fIdUser	fIdUtilizator			unique
fIdUser_2	fIdUtilizator			
fIdUtilizator	int(11) unsigned	No		auto_increment
fNumeUtilizator	varchar(32)	No		
fParola	varchar(32)	No		
fEmail	varchar(150)	No		
fAdresa	text	Yes		
fDataInregistrare	date	Yes		

Figura 3.14 Tipul de date al atributelor din tabelul tutilizatori

fIdUtilizator	fNumeUtilizator	fParola	fEmail	fAdresa	fDataInregistrare
4	cristian	cristian	cristian@yahoo.com	loc Craiova	2021-02-09
5	mihai	mihai	mihai@yahoo.com	Str Eminesci n5/C/5, Targu Mures	2021-02-12
1	claudiu	claudiu	claudiu@yahoo.com	Bucuresti sector 6 Str Grigorescu	2021-03-13
2	viorel	viorel	viorel@yahoo.com	Bucuresti sector 3 Str Livezi nr 6	2021-03-23
8	radus	radus	zjo17365@zwoho.com	dr taberei	2021-06-07

Figura 3.15 Utilizatorii înregistrați până acum pe site și datele acestora

### 10. Tabelul tvoturi:

Este folosit pentru a înregistra voturile utilizatorilor înregistrați pe site-ul web pentru produse. Utilizatorii pot vota o singură dată pentru un produs.

Câmpuri:

- fIdVot: Id-ul înregistrării;
- fIdProdusL Id-ul produsului căruia i se aplică votul;
- fIdUtilizator: Id-ul utilizatorului care a votat produsul;
- fValoareVot: Valoarea votului (un număr între 1 și 10).

fIdVot	fIdProdus	fIdUtilizator	fValoareVot
1	14	4	10
2	5	4	10
3	1	4	6

Figura 3.16 Voturile înregistrate pe site

Tabela tadmins are atributul fIdAdmin ca atribut principal și ca primary key, acesta fiind id-ul pe care îl obține automat fiecare administrator în parte. Pe lângă acesta, tabela mai conține username-ul și parola aleasă de admin.

Tcategorii conține id-ul primit de fiecare categorie (care este și cheie primară) și numele ales de administrator pentru fiecare categorie în parte.

Tabela tproducatori are ca primary key id-ul producătorului care se auto incrementează și pe lângă mai conține numele, site-ul web și o mică descriere a producătorului de piese auto.

Utilizatorii sunt reținuți în tabela tutilizatori care este compusă din id-ul generat automat (cheie primară), numele ales de aceștia pentru cont, parola, email-ul, adresa și data la care s-au înregistrat pe site.

Tabela tsubcategorii conține id-ul subcategoriei ca primary key (auto\_increment), numele subcategoriei ales de admin și un foreign key IdCategorie legat de tabela tcategorii.

Tproduse conține id-ul produsului ca cheie primară, numele produsului, codul, imaginea, descrierea, specificațiile, prețul și lunile de garanție, toate introduse de administrator și două chei străine fIdSubcategorie citit din tabela cu același nume și iIdProducator din tabela tproducatori.

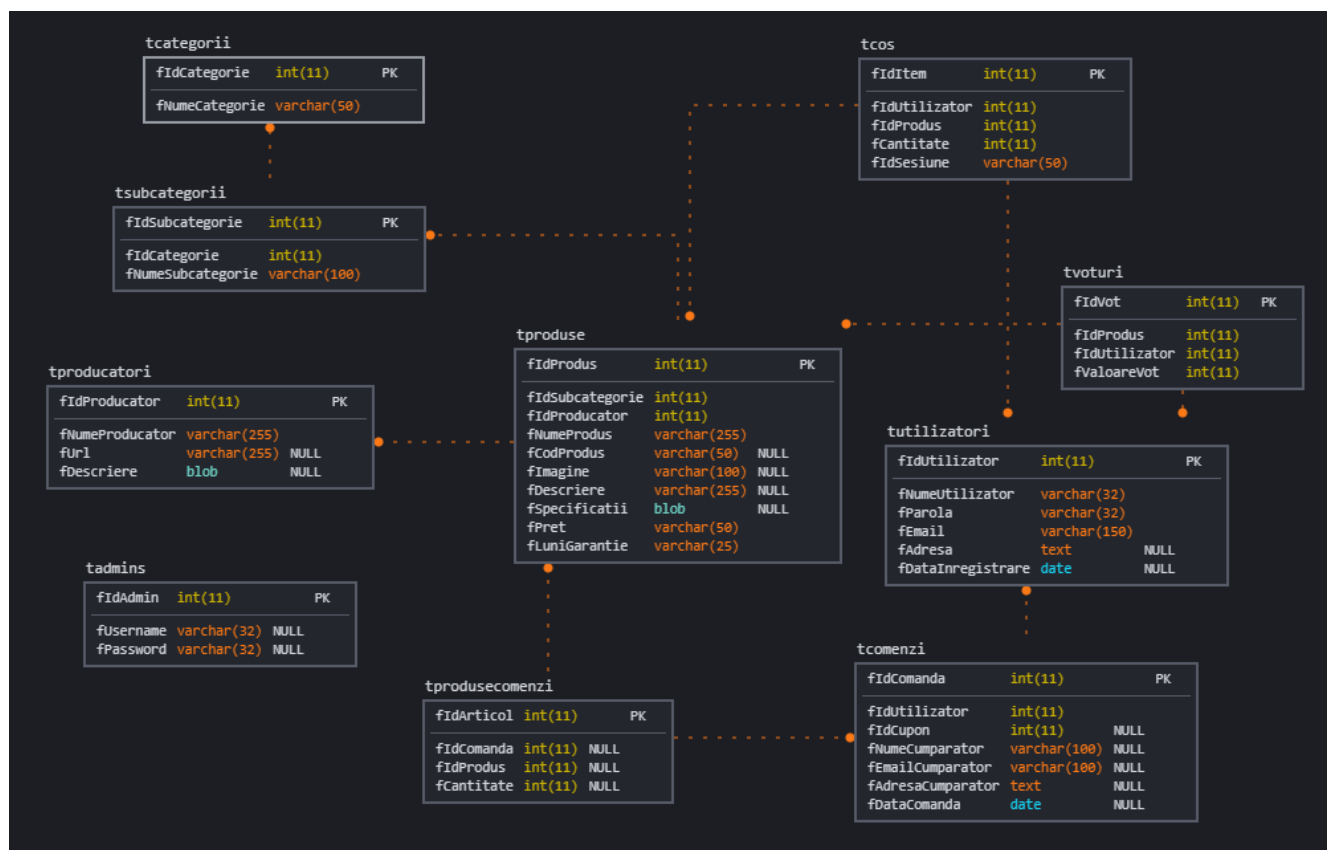


Figura 3.17 Schema conceptuală a bazei de date

Comenzile sunt înregistrate de tabela tcomenzi care este compusă din id ca cheie primară, numele, email-ul și adresa cumpărătorului, data la care s-a plasat comanda, iar id-ul utilizatorului este foreign key.

Tabela tprodusecomenzi are primary key atributul fIdArticol. fIdComanda, fIdProdus și fCantitate fiind trei chei străine care sunt citite din tabelele tcomenzi, tproduse, respectiv tcas.

Tcas are cheie primară fIdItem, id-ul utilizatorului și id-ul produsului fiind două foreign key-uri asociate tabelor tutilizatori și tproduse, fCantitate este citit în funcție de cantitatea introdusă de utilizator pentru fiecare componentă în parte, iar fIdSesiune este generat automat fiind activ până când comanda este plasată, user-ul golește coșul sau se deloghează.

Voturile sunt păstrate în tabela cu același nume, id-ul votului se auto incrementează, id-ul produsului și id-utilizatorului sunt două chei străine, iar valoarea votului este salvată în funcție de nota oferită de utilizator.

### 3.3 Schema de sistem a aplicației

#### Arhitectura sistemului

Arhitectura sistemului pe care am ales-o este arhitectura client-server pe trei nivele (three-tier architecture) deoarece este o arhitectură lightweight ce oferă performanțe bune în ceea ce privește aplicațiile web și nu numai.

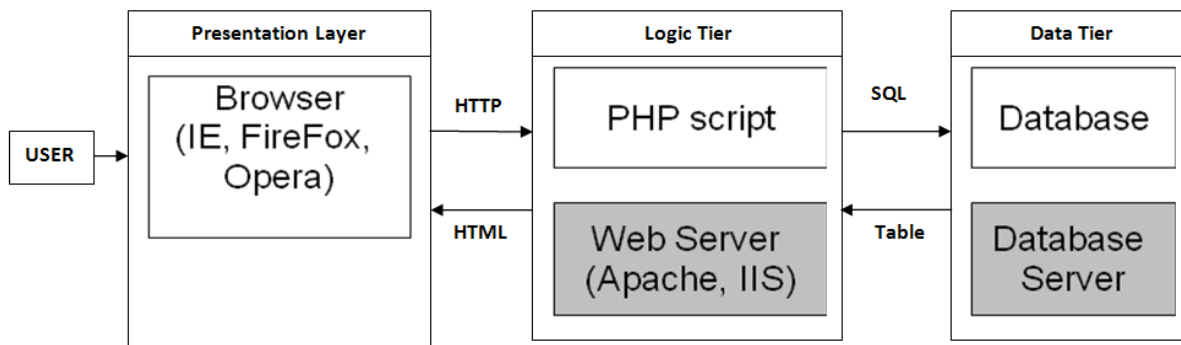


Figura 3.18 Arhitectura client-server pe trei nivele [34]

Arhitectura aleasă este o arhitectură client-server alcătuită din nivelul de prezentare, nivelul logic și nivelul de date. Nivelul de prezentare sau nivelul client reprezintă cel mai înalt nivel al arhitecturii și este responsabil cu UI (User Interface) și UX (User Experience). În cadrul nivelului de prezentare vor fi afișate funcționalitățile aplicației precum: căutare, vizualizare produse, cumpărare și conținut coș cumpărături. Nivelul logic reprezintă un nivel intermediar care se ocupă cu funcționalitatea aplicației. În cadrul acestui nivel se procesează comenzi, au loc decizii și evaluări logice. Nivelul de date reprezintă serverul de baze de date. Toate informațiile sunt stocate și extrase în cadrul acestui nivel. După ce datele sunt preluate vor fi trimise la nivelul logic la serverul web pentru procesare iar ulterior vor fi trimise la utilizator.

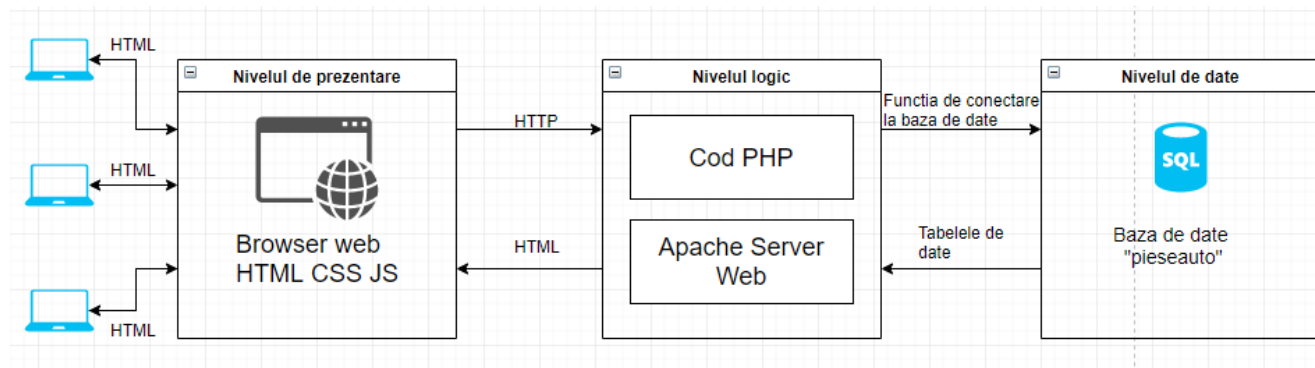


Figura 3.19 Arhitectura generală a sistemului



Figura 3.20 Reprezentare grafică propusă a nivelelor arhitecturii cu tehnologiile aferente

Alte aspecte luate în considerare la alegerea acestei arhitecturi au fost flexibilitatea și reutilizarea. Este o arhitectură flexibilă deoarece nivelele sunt independente și management-ul informațiilor se poate realiza fără a accesa nivelul de date. Menținerea nivelului logic este mai rapidă iar dacă sunt schimbări minore la acest nivel nu trebuie reînștalat sistemul în totalitate. Doar componenta logică este dezvoltată și testată se poate folosi la mai multe proiecte pe viitor ceea ce o face reutilizabilă pentru diferite nivele de prezentare. Arhitectura pe trei nivele reprezintă o arhitectură destul de sigură deoarece clientul nu poate accesa baza de date direct.

```

1 <?php
2 require_once("config.inc.php");
3 switch ($connType) {
4 case "local" : {
5 $server_name = "localhost";
6 $db_name = "pieseauto";
7 $db_user = "root";
8 $db_pass = "";
9 break;
10 }
11 }
12 //-----
13 function fDBConnect() {
14 global $server_name;
15 global $db_user;
16 global $db_pass;
17 global $db_name;
18 $myConn = mysql_connect($server_name, $db_user, $db_pass) or
19 die("Error while connecting to db: " . mysql_error());
20 $res = mysql_select_db($db_name);
21 if (!$res)
22 die ("Error while opening the db");
23 return $myConn;
24 }
25 //-----
26 function fDBClose($link) {
27 mysql_close ($link);
28 }
29 ?>

```

Figura 3.21 Funcția de conectare la baza de date

Funcția fDBConnect() realizează conectarea între nivelul logic și nivelul de date pentru aplicația mea. Prin acest script PHP se face conexiunea la baza de date “pieseauto”. \$server\_name va fi gazda pe care rulează serverul nostru, și este de obicei “localhost”, \$db\_name este numele bazei de date pe care am creat-o și pe care o folosim, \$db\_user este numele de utilizator, iar \$db\_pass parola pe care o alegem noi, în acest caz nu este setată niciuna. Funcția face conexiunea către baza de date și în același timp verifică dacă se poate realiza conectarea, în caz contrar un mesaj de eroare o să fie afișat.

```

function finsertCategLeftMenu() {
 echo "<tr>\n";
 echo "<td style='text-align: center;' valign='top'>\n";
 //echo "<div id='divcategmenu'>\n";

 echo "<div class='btn-group'>\n";
 //echo "<div class='item'>\n";
 //echo "<div class='leftmenu'>\n";
 echo "<div class='btn-group'>\n";
 echo "<center><button class='button' style='width: 200px;'>PIESE MASINI:</button></center></div>\n";
 $strSQL = "SELECT * FROM tcategorii ORDER BY fIdCategorie";
 $result = mysql_query($strSQL);
 if (mysql_num_rows($result) > 0) { // categories fou

 while ($row = mysql_fetch_array($result)) {

 echo "<div class='dropdown'>\n";

 $intIdCategorie = $row['fIdCategorie'];
 echo " <button class='button' style='width: 200px;'> " . $row['fNumCategorie'] . "</button>\n";

 $strSQLSub = "SELECT * FROM taubcategorii WHERE fIdCategorie=$intIdCategorie";
 $resultSub = mysql_query($strSQLSub);

 echo "<div class='dropdown-content'>\n";

 if (mysql_num_rows($resultSub) > 0) { // exista subcategorii

 while ($rowSub = mysql_fetch_array($resultSub)) {
 $intIdSubcategorie = $rowSub['fIdSubcategorie'];
 $strNumSubcategorie = $rowSub['fNumSubcategorie'];

```

Figura 3.22 Meniul din stânga

Secvențe de cod pentru realizarea meniului vertical din stânga, vizibil în meniul utilizatorului, care este creat cu funcția `fInsertCategLeftMenu()`. Acesta creează un tabel vertical în care introduce butonul **PIESE MASINI** fără vreo referință, iar apoi se afișează modelele de producători auto, prin selectarea din tabela `tcategorii` a acestora și ordonarea lor în funcție de id. Apoi am implementat un meniu de dropdown pentru categoriile de mașini care au subcategorii, acesta afișând modelele pentru fiecare producător.

```
//-----
function fValidareAdaugareCategorie() {
 if (document.getElementById("txtNume").value == "") {
 alert ("Completati campul 'Nume Categorie'.");
 document.getElementById("txtNume").focus();
 return false;
 }
 return true;
}

//-----
function fValidareAdaugareSubcategorie() {
 if (document.getElementById("selCategorie").value == "select") {
 alert ("Selectati o categorie din lista.");
 document.getElementById("selCategorie").focus();
 return false;
 }
 if (document.getElementById("txtNume").value == "") {
 alert ("Completati campul 'Nume Categorie'.");
 document.getElementById("txtNume").focus();
 return false;
 }
 return true;
}
```

Figura 3.23 Funcția pentru validarea unui formular

Acest tip de funcții JavaScript pentru validare ajută să nu introducem mașini sau modele de mașini cu date nevalide sau incomplete, caz în care un mesaj de avertizare o să fie afișat. Fără aceste funcții aplicația nu ar putea să lucreze corect deoarece, ar însemna apariția multor erori, text incomplet sau a apăsărilor accidentale de butoane care nu ar mai fii oprite de un mesaj de confirmare. Aceste operații ajută și la împiedicarea de creare a unor parole slabe pentru că, obligă la un număr minim de caractere pentru parolă.

```
//-----
function fInsertTopMenu() {

 echo "<div class=\"btn-group\">\n";

 echo " <button class=\"button\">Home</button>\n";
 echo "<button class=\"button\">Administrare</button>\n";
 echo "<button class=\"button\">Alege masina</button>\n";
 echo "<button class=\"button\">Contact</button>\n";
 echo "</div>\n";
}

//-----
```

Figura 3.24 Meniul orizontal

Funcția `fInsertTopMenu()` introduce meniul de sus, cel orizontal, din paginile disponibile utilizatorilor. Acesta a fost făcut cât mai simplu și intuitiv pentru orice tip de vizitatori ai site-ului, fie ei mai experimentați cu comerțul online, fie nu. Opțiunile disponibile sunt Home, Administrare, Alege mașina și Contact. Prima opțiune întoarce mereu la pagina principală, a doua este destinată administratorilor site-ului și o să deschidă pagina în care este necesară logarea pentru admin, următoarea o să ducă către o listă cu toate modelele de mașini disponibile, iar ultima este cea pentru formularul de contact.

```
function fPrintLoginAreaAdmin($bIsLoggedIn, $strError) {
 if ($bIsLoggedIn) {
 // the admin is logged in, say HI and display link to cart and link to logout
 $strUsername = $_SESSION['strUsername'];
 echo "<table class='modelspecs'><tr><td>\n";
 echo "
Bine ai venit $strUsername

";
 echo "Modificare parola
";

 echo "Administrare producatori";
 echo "Administrare masini";
 echo "Administrare modele masini+motorizari";
 echo "Administrare piese auto
";

 echo "Administrare utilizatori";
 echo "Administrare comenzi";

 echo "Vanzari totale";
 echo "Vanzari pe clienti";
 echo "Vanzari pe o perioada";
 echo "Vanzari produse";

 echo "Iesire";
 echo "</td></tr></table>\n";
 } else {
 // the admin is not logged in, display the login/register form
 echo "<form name='frmLogin' method='post' action='execute.php?action=login' onsubmit='return fValidateLogin();'>\n";
 if ($strError == "loginerror") {
 echo "<center>Eroare autentificare!</center>\n";
 }
 echo "<table class='modelspecs'><tr><td>Utilizator:
\n";
 echo "<input class='text' type='text' style='width: 150px;' name='txtUsername' id='txtUsername' maxlength='32'>
\n";
 echo "Parola:
\n";
 echo "<input class='text' type='password' style='width: 150px;' name='txtPassword' id='txtPassword' maxlength='32'>

\n";
 echo "<input class='buttons' type='submit' style='width: 150px;' name='cmdLogin' id='cmdLogin' value='Autentificare'>

\n";
 echo "</td></tr></table></form>\n";
 }
}
```

Figura 3.25 Meniul din stânga în pagina de administrator

Funcția `fPrintLoginAreaAdmin($bIsLoggedIn, $strError)` introduce meniul din stânga din pagina de administrare. Acest meniu este destinat exclusiv adminilor site-ului și poate fi accesat doar introducând un username și o parolă validă. Opțiunile disponibile sunt destul de numeroase oferind deținătorului aplicației web posibilitatea de a manageria categoriile, subcategoriile și componentele auto disponibile în site. Acesta poate să vadă utilizatorii cu conturi create, lista comenzilor finalizate, dar și niste rapoarte detaliate despre vânzarile totale sau pe o anumită perioadă.



```

function fInsertLoginArea() {
 echo "<div id=\"loginarea\">\n";
 $intIdUtilizator =isset($_SESSION['intIdUtilizator']) ? $_SESSION['intIdUtilizator'] : '';
 if ($intIdUtilizator > 0) { // user is authenticated, display name & logout link
 echo "Bine ai venit " . $_SESSION['strNumeUtilizator'] . " <a class=\"TopMyAccountLink\" href="
 } else { // user is not authenticated, display login form
 echo "<form action=\"execute.php?action=login\" method=\"post\" style=\"display: inline;\" onsubmit="
 echo "<fieldset style=\"padding: 0px; border: 0px;\">\n";
 echo "<label class=\"loginlabel\" for=\"txtNumeUtilizator\">Utilizator:</label>\n";
 echo "<input class=\"logininput\" type=\"text\" name=\"txtNumeUtilizator\" id=\"txtNumeUtilizator\" m
 echo "<label class=\"loginlabel\" for=\"txtParola\">Parola:</label>\n";
 echo "<input class=\"logininput\" type=\"password\" name=\"txtParola\" id=\"txtParola\" maxlength=\"3
 echo "<input type=\"submit\" alt=\"Autentificare\" class=\"loginbutton\" name=\"cmdLogin\" id=\"cmdLo
 echo "<input type=\"button\" alt=\"Cont nou\" class=\"loginbutton\" onclick=\"location.href='inregist
 echo "</fieldset>\n";
 echo "</form>\n";
 }
 echo "</div>\n";
}

```

Figura 3.26 Meniul de logare

Funcția `fInsertLoginArea()` introduce meniul de logare în interfața utilizatorului și verifică dacă utilizatorul este autentificat sau nu. Partea de logare pentru utilizatori a fost păstrată cât mai ușor de înțeles și utilizat, aceștia au un câmp unde să introducă username-ul, un altul pentru parolă și chiar lângă butonul de login. Dacă username-ul și/sau parola sunt incorecte un mesaj de eroare o să fie afișat, în schimb dacă acestea au fost introduse corect o să apară un mesaj de bun venit alături de două noi opțiuni, unul pentru accesarea contului personal de utilizator și opțiunea de ieșire din cont. Pentru vizitatorii noi, există opțiunea de cont nou care duce către formularul de creare a contului.



## CAPITOLUL 4 PREZENTAREA PRODUSULUI SOFTWARE

### 4.1. Descrierea funcțiilor aplicației

Pentru a ajuta utilizatorii, aplicația își propune să îndeplinească următoarele funcții:

- Posibilitatea de a utiliza un coș de cumpărături, unde clienții pot adăuga articole din oferta propusă;

- Posibilitatea de a personaliza contul;
- Afișarea unei liste de motoare pentru anumite modele de mașini;
- Puteți căuta piese după producător, mașină, model;
- Postarea de note / voturi pentru fiecare produs în parte;
- Utilizatorii pot comanda piese auto online plătind în momentul livrării.

Pentru a adăuga aceste funcționalități, aplicația a fost împărțită în mai multe module, care vor fi discutate în detaliu. Aplicația Auto este dezvoltată pe următoarele module:

- Autentificare client
- Comanda on-line
- Lista de mașini după model și motorizare
- Administrarea site-ului

Fiecare dintre aceste module va fi descris mai jos și vor fi introduse multe aspecte legate de implementarea eficientă a acestor module. Autentificarea clientului în aplicație se referă la capacitatea clientului de a crea de conturi personale pentru efectuarea achizițiilor.

Serviciile oferite de acest modul sunt următoarele:

- Crearea unui cont nou;
- Modificare datele cu caracter personal;
- Selectare domeniu și subdomeniu preferat;
- Vizualizarea comenzii;
- Schimbarea parolei și resetare parola.

Pentru a crea un cont nou, trebuie să completați un formular care conține date personale, apoi să alegeți un nume de cont și o parolă. Trebuie să completați informații valide în toate câmpurile. Dacă datele introduse sunt nevalide, nu puteți adăuga un cont nou.

Modificarea datelor cu caracter personal este similară cu procesul de creare a unui cont nou. Pentru fiecare utilizator, puteți schimba numele și prenumele, precum și orașul sau adresa de e-mail. Efectuați modificări utilizând butonul de editare a profilului. Ca și în cazul creării unui cont nou, toate câmpurile trebuie completate cu informații valide. Dacă datele introduse sunt nevalide, nu pot fi modificate.

Prin apăsarea butonului de comandă a produsului pentru a selecta modelul și motorul preferat pe pagina principală, puteți selecta modelul preferat sau puteți selecta toate modelele disponibile în mod implicit. Conform selecției de pe pagina de pornire a site-ului, vor fi afișate numai produsele din modelul

sau motorul selectat. Vizualizarea comenzii de cumpărături este o altă opțiune care se face la apăsarea butonului coșului de cumpărături..

Când apăsați butonul, va apărea o fereastră nouă, care conține o listă a tuturor produselor și poate afișa informații despre fiecare în parte. Comanda online în aplicația auto se referă la posibilitatea clienților de a comanda componente auto prin intermediul site-ului. Administratorii aplicației pot, de asemenea, să vizualizeze și să urmărească aceste comenzi. Cumpărăturile se fac cu ajutorul coșului de cumpărături, care este un modul mai complex, care permite clienților să cumpere mai multe produse. Pentru a explica modulul, considerăm că doriți să achiziționați o piesă. Selectați o piesă din modelul sau motorul selectat, apoi furnizați „Adaugă în coș”. După selectare, produsul va fi adăugat în coșul de cumpărături virtual disponibil pentru fiecare utilizator.

Sunt disponibile următoarele opțiuni: modificați cantitatea fiecărui articol sau utilizați butonul „Ștergeți” pentru a șterge articolele care nu mai sunt necesare din coșul de cumpărături. Acest exemplu arată cum să creați un coș de cumpărături virtual și ce puteți face. Lista mașinilor clasificate după model și motor este un modul care afișează o listă completă a tuturor mașinilor și o listă completă a motoarelor. Modulul obține modelele de mașini și motoare disponibile din baza de date și afișează o listă completă de înregistrări sub forma a două tabele. Selectând o anumită mașină sau un anumit model, va apărea o nouă pagină, care va conține o listă cu toate piesele de mașină aparținând mașinii corespunzătoare și modelului selectat.

**Administrarea site-ului** este cel mai important modul, iar de eficiența acestuia depinde succesul oricărui magazin electronic. Acest modul va permite administratorilor să modifice conținutul site-ului și să vizualizeze anumite informații. Acest modul este destinat doar administratorilor de site-uri.

Serviciile oferite de acest modul sunt următoarele:

- Adăugare piese auto;
- Modificare piese auto, utilizatori;
- Ștergere piese auto, utilizatori sau comenzi;
- Vizualizare comenzi efectuate;
- Vizualizare rapoarte detaliate despre vânzări și clienți.

Adăugarea este esențială pentru această aplicație, pornind de la ideea că este un magazin online, adăugarea de piese auto noi este esențială. Codul piesei auto trebuie adăugat manual, iar imaginea produsului trebuie să aibă următoarele câmpuri: cod piesă auto și extensie. Trebuie să completați informații valide în toate câmpurile. Dacă datele introduse sunt nevalide, nu se vor adăuga. Modificările făcute sunt similare cu cele de adăugare și, pentru fiecare produs sau utilizator, puteți schimba numele și funcția. Efectuați modificări utilizând butonul de modificare de lângă fiecare înregistrare. Ca și în cazul adăugării, toate câmpurile trebuie să fie completate cu informații valide. Dacă datele introduse sunt nevalide, nu pot fi modificate. Ștergerea unei înregistrări se face prin selectarea piesei de mașină sau a utilizatorului dorit și folosind butonul „Șterge” de lângă înregistrare.

Apăsând butonul „Șterge” va apărea o nouă pagină cu informații detaliate despre înregistrarea selectată și un buton „Confirmă ștergerea”. Dacă ștergerea este confirmată, înregistrarea va fi ștearsă definitiv din baza de date. Comanda de vizualizare a comenzilor finalizate este o altă opțiune disponibilă

administratorului. Această opțiune poate fi utilizată în meniul principal apăsând butonul „Vânzări totale”. Când apăsați butonul, va apărea o fereastră nouă, care listează toate comenzile realizate și poate afișa informații detaliate despre fiecare comandă. Modulul de administrator permite, pe lângă vizualizarea numărului total de comenzi, și observarea unor rapoarte detaliate despre vânzările pe clienți, vânzările pe produse și pe o anumită perioadă de timp. Astfel putem extrage foarte ușor care este cel mai fidel client, care este cel mai bine vândut produs sau care este cea mai productivă perioadă pentru vânzări.

Modulul de comandă permite utilizatorilor să comande piese auto online plătind în momentul livrării. Acest modul permite utilizatorilor să selecteze diverse informații despre producător, lista pieselor, lista mașinilor, informațiile de contact, posibilitatea de a găsi piese și de a comanda piese din meniul afișat pe ecran. Utilizatorul va completa formularul cu datele personale: nume, prenume, adresă, adresă de e-mail, localitate, județ/sector și cod poștal.

## 4.2 Manual de utilizare

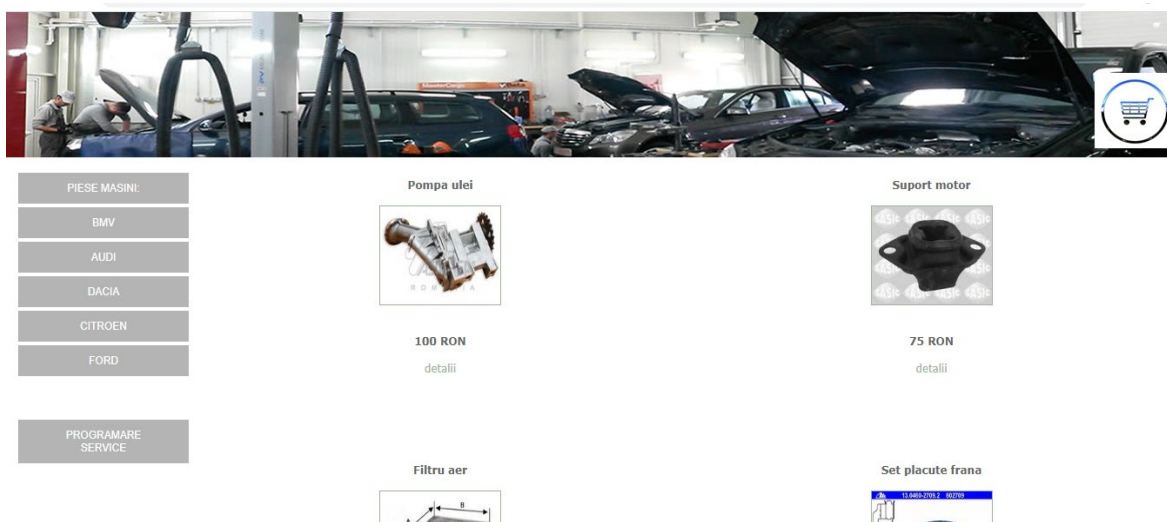


Figura 4.1 Pagina Home a aplicației

Structura principală a interfeței este următoarea:

Titlul aplicației este afișat în partea de sus. De asemenea, în colțul din dreapta sus este un link către coșul de cumpărături. Un utilizator cu un cont creat în aplicație va introduce numele contului și parola într-un formular, iar după apăsarea butonului „autentificare”, aplicația va verifica dacă există un astfel de cont și dacă parola introdusă este corectă. În conformitate cu rezultatele testelor efectuate de aplicație, utilizatorul va fi permis să acceseze funcțiile modulului client, în caz contrar va fi afișat mesajul de eroare corespunzător. Trebuie remarcat faptul că formularul este verificat și dacă este valid se verifică dacă utilizatorul a specificat un cont și o parolă (formularul nu poate fi gol).

După autentificare, secțiunea va afișa un mesaj de întâmpinare cu două opțiuni:

- contul meu;
- ieșire.

Utilizați opțiunea “Contul meu” pentru a actualiza datele personale ale utilizatorului, cu excepția numelui de utilizator selectat în timpul înregistrării, care nu poate fi modificat ulterior. Scopul accesării opțiunii “Ieșire” este de a anula autentificarea utilizatorului și a șterge informațiile aferente acestuia (de

exemplu, ID-ul utilizatorului păstrat în variabila sesiune). Reveniți la această opțiune pentru a crea un cont nou, așa cum se arată mai jos:



Cont nou

Nume Utilizator:

Parola:

Confirmare Parola:

Adresa Email:

Confirmare Adresa Email:

Adresa:

Figura 4.2 Formularul pentru crearea unui cont nou

Câmpurile necesare pentru crearea unui cont nou sunt:

1. Numele de utilizator nu poate fi schimbat în viitor.
2. Parola necesară pentru autentificare. Parola trebuie confirmată deoarece caracterele sale sunt invizibile și ascunse de caractere \*.
3. O adresă de e-mail pe care administratorul o poate folosi pentru a comunica cu utilizatorii. În plus, adresa de e-mail trebuie confirmată pentru a evita introducerea unei adrese greșite, ceea ce poate împiedica administratorul site-ului să comunice cu utilizatorul.
4. Adresa utilizatorului. Aceasta o sa fie considerată adresa principală a utilizatorului. Cu excepția numelui de utilizator, câmpurile cu informații personale pot fi modificate după înregistrare și autentificare.

Acestea sunt, de asemenea, utilizate în procesul de comandare a produselor, dar sunt recomandate doar utilizatorului care au cont, dar acesta poate actualiza înainte de a confirma comanda iar expedierea produsului poate fi la o adresă diferită. Cu toate acestea, în majoritatea cazurilor, utilizatorii folosesc aceleași informații introduse atunci când creează un cont și o comandă. Butonul „Anulează modificări” resetează starea formularului la starea inițială, ștergând astfel toate câmpurile formularului. Butonul “Înregistrare” trimite datele către scriptul ce le va însera în baza de date. În cazul în care utilizatorul nu a completat toate câmpurile, acesta este atenționat prin mesaje, cu ajutorul funcțiilor de validare JavaScript.

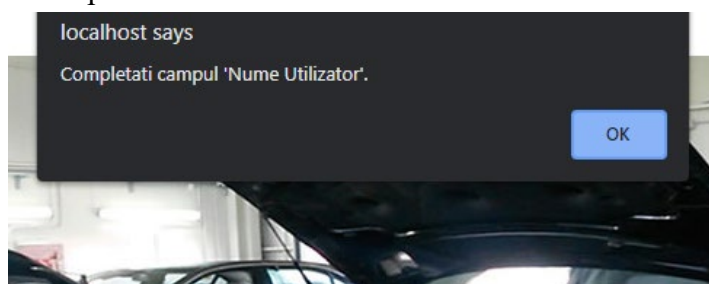


Figura 4.3 Atenționare completare incorectă

După conectare, utilizatorii își pot schimba informațiile personale accesând opțiunea „Contul meu”: Aici puteți introduce informații pentru actualizarea datelor personale. Aplicația memorează și data înregistrării pe site. Ca și în ecranul de înregistrare, parola și adresa de e-mail trebuie confirmate pentru a evita utilizarea datelor incorecte, ceea ce poate duce la imposibilitatea de a efectua autentificarea ulterioară sau de a comunica cu administratorul site-ului. Butonul „Anulează modificările” restabilește formularul la starea inițială, adică va afișa informațiile din baza de date fără a reîncărca pagina. Butonul de actualizare salvează modificările în baza de date. Efectul opțiunii de deconectare este de a anula autentificarea utilizatorului și de a-l redirectiona către pagina principală a site-ului.

În funcție de selecția făcută, numai piesele auto sau piesele modelului selectat vor fi afișate pe pagina de pornire a site-ului. Afișarea trebuie făcută aleatoriu, adică de fiecare dată când pagina (index.php) este încărcată, conținutul tabelului trebuie să afișeze detaliile altor piese. Dacă utilizați butonul “Adaugă în coș”, produsul va fi adăugat în coșul de cumpărături cu o cantitate egală cu 1, cu excepția cazului în care acesta există deja în coșul de cumpărături, caz în care cantitatea deja existentă va crește cu 1. Utilizați modulul de gestionare pentru a afișa categoriile de produse definite de administrator în baza de date. Accesând opțiunile din acest meniu, produsele din categoria selectată sunt afișate într-un mod similar cu rezultatele căutării prezentate mai sus.

Al doilea set de informații include opțiuni care conțin detalii de contact, hărți ale site-ului și formulare de înregistrare.

Pagina de contact afișează informații despre contactarea companiei, cum ar fi:

- Numele companiei;
- Adresa;
- Număr de telefon;
- Numar de fax;
- Adresa de e-mail.

De asemenea, pe această pagină, pot trimite un formular rapid fără a crea un cont de utilizator.

**INFORMATII DE CONTACT**  
**ASCAR NET SRL**  
**Str. Stefan Velovan 11 B, Craiova**  
**Tel: +40-21.034.00.70**  
**Fax: +40-21.024.10.28**  
**Mob:+40-726.024.100**  
**Email: office@ascar.ro**

**Formular de contact**  
\* lipsesc datele  
Nume:  \*  
E-mail:  \*  
Telefon:   
Mesaj:

[Acasa](#) | [Masini](#) | [Harta Site](#) | [Contact](#)

Figura 4.4 Formular de contact al firmei

Pagina „Harta site-ului” afișează linkuri către părți ale mașinilor și modelelor existente și motoarelor acestora:

Home | Administrare | Harta | Contact

Utilizator:  Parola:  [Autentificare](#) [Cont nou](#)

[Căsuța de cumpărături](#)

PIESE MASINI:

- BMV
- AUDI
- DACIA
- CITROEN
- FORD

PROGRAMARE SERVICE

Categorii produse : BMV, AUDI, DACIA, CITROEN, FORD

BMV	Seria 5/518g(85 KW/115 CP) seria 6/ 630i (190 KW/258 CP)
AUDI	Q7 3.0TDI (15KW/204CP) A3 1.4TFSI (90KW/122CP)
DACIA	LOGAN 1.5dCi (L504) ( 55KW/75CP ) DUSTER 1.6 16VLPG (77KW/105 CP)
CITROEN	AX 1.4 4x4948KW/65CP) XSARA 1.8 VTS (85KW/116CP)
FORD	FOCUS 1.6 EcoBoost (134KW/102CP) ACTIVA 1.8 GLX(84KW/114 CP)

Acasa | Masini | Harta Site | Contact

Activate Windows  
Go to PC settings to activate Windows.

Figura 4.5 Pagina cu toate modelele disponibile

Dacă mașina nu are un model definit de administrator, va fi afișat mesajul „Fără modele”, altfel va afișa un link către modelul existent.

În partea dreaptă sus se afla un link către coșul de cumpărături, care poate fi accesat doar de către clienții autentificați. Accesul neautentificat la coșul de cumpărături va afișa următorul mesaj:

**Continutul cosului de cumparaturi**

**Eroare ! Pentru a accesa cosul de cumparaturi trebuie sa va autentificati.**

Figura 4.6 Mesaj de eroare, necesită autentificare în cont

După autentificare, conținutul coșului de cumpărături poate fi afișat în două moduri. Dacă coșul de cumpărături nu conține produse, este afișat primul mod:

**Continutul cosului de cumparaturi**

**Cosul dvs. de cumparaturi este gol.**

Figura 4.7 Mesaj care informează de lipsa produselor din coș

Dacă există produse în coșul de cumpărături, aceste produse și următoarele informații vor fi afișate:

- Numele produsului;
- Cantitate;



- Prețul unității (pe bucată și fără TVA)
  - Preț total (per produs cu TVA);
- Acesta este prețul unitar înmulțit cu numărul de piese și adunat TVA-ul;
- Prețul final al tuturor produselor și cantităților.

Continutul cosului de cumparaturi			
Nume produs	Cantitate	Pret unitar	Total
Pompa ulei	1 <input type="button" value="Sterge"/>	100 RON	119 RON
Set pompa apa+curea dintata	1 <input type="button" value="Sterge"/>	450 RON	535.5 RON
<input type="button" value="Actualizeaza"/>		RON fara TVA	RON cu TVA
<input type="button" value="Goleste cosul"/>			<b>Total (cu TVA): 654.5 RON</b>
			<input type="button" value="Comanda produsele"/>

Figura 4.8 Modul de afișare al produselor în coș

Alte opțiuni disponibile pe pagina coșului de cumpărături:

- Stergeți produse;
- Actualizați cantitatea;
- Goliți coșul;
- Comanda Produsele.

Golirea este opțiunea din coșul de cumpărături, dacă utilizatorul confirmă, de a șterge toate produsele din coșul de cumpărături. Utilizatorul poate modifica cantitatea fiecărui produs folosind butonul „Actualizare” pentru a salva noua cantitate.

Dacă introduceți cantitatea 0, produsul va fi șters automat din coșul de cumpărături. Dacă utilizatorul introduce un caracter non-numeric în caseta cantității, va fi afișat un mesaj de eroare și se va păstra vechea valoare. Dacă un produs existent este adăugat în coșul de cumpărături, cantitatea acestuia va crește cu 1.

Continutul cosului de cumparaturi			
Nume produs	Cantitate	Pret unitar	Total
Pompa ulei	1 <input type="button" value="Sterge"/>	100 RON	119 RON
Set placute frana	2 <input type="button" value="Sterge"/>	120 RON	285.6 RON
<input type="button" value="Actualizeaza"/>		RON fara TVA	RON cu TVA
<input type="button" value="Goleste cosul"/>			<b>Total (cu TVA): 404.6 RON</b>

Nume Cumparator:	<input type="text" value="claudiu"/>
Adresa Email:	<input type="text" value="claudiu@yahoo.com"/>
Confirmare Adresa Email:	<input type="text" value="claudiu@yahoo.com"/>
Adresa:	<input type="text" value="Bucuresti sector 6 Str Grigorescu"/>

Activate Windows

Figura 4.9 Formularul pentru datele de comandă

Butonul “Comandă produsele” afișează pagina următoare cu informații despre cumpărător, ce sunt completate din profilul personal al utilizatorului, pe care acesta le poate modifica.

De exemplu, utilizatorul poate alege o adresă de expediere diferită de adresa personală.

Această pagină conține un rezumat al comenzii și anume:

- Produse din coșul de cumpărături;

## - Informații utilizator.

Confirmare Comanda			
Nume produs	Cantitate	Pret unitar	Total
Set placute frana	1	120 RON	142.8 RON
			<b>Total (cu TVA): 142.8 RON</b>

Nume Cumparator: cristian  
 Adresa Email: cristian@yahoo.com  
 Adresa: loc Craiova

[Comanda produsele](#)

Figura 4.10 Confirmarea comenzii

După confirmarea comenzii, utilizatorul va primi următorul mesaj:

**Comanda finalizata**

Felicitari,  
 Comanda dvs. a fost finalizata. Va multumim pentru alegerea facuta.

Figura 4.11 Mesajul de finalizare a comenzii

## Secțiunea de administrare

Administratorul poate configura categoria de produse, subcategoriile sale, produsele, producătorii, comenzi, utilizatori, recenzii de produse. Pagina principală este următoarea:

AutoMag :: Secțiunea de administrare :: Administrare produse

Bine ai venit **admin**

[Adauga produs](#)

Nume produs	Producator	Categorie	Subcategorie	Detalii	Editare	Stergere
Alternator	LAUBER	BMV	Seria 5/518g(85 KW/115 CP)			
Saboti frana	Delphy	BMV	Seria 5/518g(85 KW/115 CP)			
Filtru aer	Sofima	BMV	seria 6/ 630i (190 KW/258 CP)			
Curea transmisie cu caneluri	SASIC	BMV	seria 6/ 630i (190 KW/258 CP)			
Pompa apa	HEPU	AUDI	Q7 3.0TDI (15KW/204CP)			
Set placute frana	LAUBER	AUDI	Q7 3.0TDI (15KW/204CP)			
Starter	LAUBER	AUDI	A3 1.4TFSI (90KW/122CP)			
Saiba arc	OPTIMAL	AUDI	A3 1.4TFSI (90KW/122CP)			
Pompa ulei	ASAH	DACIA	LOGAN 1.5dCi (L504) ( 55KW/75CP )			
Starter	LAUBER	DACIA	LOGAN 1.5dCi (L504) ( 55KW/75CP )			
Suport motor	SASIC	DACIA	DUSTER 1.6 16VLPG (77KW/105 CP)			
Set placute frana	ATE	DACIA	DUSTER 1.6 16VLPG (77KW/105 CP)			
Starter	LAUBER	CITROEN	AX 1.4 4x4948KW/65CP)			
Set pompa apa+curea dintata	RUVILLE	CITROEN	XSARA 1.8 VTS (85KW/116CP)			
Filtru aer	FILTRON	FORD	FOCUS 1.6 EcoBoost (134KW/102CP)			

Figura 4.12 Meniul de administrare a produselor

```


 Adauga produs

<?php
 $strSQL = "SELECT tproduse.*, tsubcategorii.fNumSubcategorie, tproducatori.fNumProdicator, tcategorii.fIdCategorie,
 if ($intIdProdicator > 0) {
 $strSQL .= " AND tproduse.fIdProdicator=$intIdProdicator ";
 }
 if ($intIdSubcategorie > 0) {
 $strSQL .= " AND tproduse.fIdSubcategorie=$intIdSubcategorie ";
 }
 $result = mysql_query($strSQL);
 echo "<table cellpadding=\"3\" cellspacing=\"0\" border=\"0\" width=\"1100\" class=\"modelspecs\">\<n\>";
 $intCount = 0;
 echo "<tr style=\"font-weight: bold;\">";
 echo "<td width=\"200\">Nume produs</td>";
 echo "<td width=\"200\">Prodicator</td>";
 echo "<td width=\"200\">Categorie</td>";
 echo "<td width=\"200\">Subcategorie</td>";
 echo "<td align=\"center\" width=\"100\">Detalii</td>";
 echo "<td align=\"center\" width=\"100\">Editare</td>";
 echo "<td align=\"center\" width=\"100\">Stergere</td>";
 echo "</tr>";
 while ($row = mysql_fetch_array($result)) {
 $intIdProdus = $row['fIdProdus'];
 $intIdProdicator = $row['fIdProdicator'];
 $intIdCategorie = $row['fIdCategorie'];
 $intIdSubcategorie = $row['fIdSubcategorie'];
 $strNumProdus = $row['fNumProdus'];
 $strNumCategorie = $row['fNumCategorie'];
 $strNumSubcategorie = $row['fNumSubcategorie'];
 $strNumProdicator = $row['fNumProdicator'];

 if ($intCount % 2 == 0) {
 echo "<tr bgcolor=\"#F0E7D8\">";

```

Figura 4.13 Implementarea meniului de administrare a produselor

Administratorii trebuie să se conecteze aici pentru a accesa opțiunile de gestionare. După autentificare, administratorul va avea următorul meniu:

Bine ai venit <b>admin</b>
<b>Modificare parola</b>
<b>Administrare producatori</b>
<b>Administrare masini</b>
<b>Administrare modele masini+motorizari</b>
<b>Administrare piese auto</b>
<b>Administrare utilizatori</b>
<b>Administrare comenzi</b>
<b>Vanzari totale</b>
<b>Vanzari pe clienti</b>
<b>Vanzari pe o perioada</b>
<b>Vanzari produse</b>
<b>Iesire</b>

Figura 4.14 Meniul din stânga al administratorului

Prima opțiune este schimbarea parolei de acces. La fel ca utilizatorii standard, numele de utilizator nu poate fi schimbat. Ecranul de modificare a parolei este după cum urmează:

**Modificare parola**

Parola noua:	<input type="text"/>
Confirmați parola noua:	<input type="text"/>
<input type="button" value="Anuleaza modificari"/> <input type="button" value="Modifica parola"/>	

Figura 4.15 Tabul de modificare a parolei pentru admin

Parola trebuie confirmată pentru a preveni schimbarea parolei cu o altă parolă greșită, ceea ce va împiedica accesul viitor la opțiunile de gestionare. Următoarele grupuri de opțiuni sunt utilizate pentru a defini producătorul, categoria și subcategoria mașinii:

- Gestionarea producătorului;
- Gestionarea autoturismelor;
- Gestionarea modelului de mașină;
- Managementul produselor.

Modificare parola
Administrare producatori
Administrare masini
Administrare modele masini+motorizari
Administrare piese auto

Figura 4.16 Opțiunile pentru gestionarea site-ului

Opțiunea de gestionare a producătorului oferă următoarele posibilități:

- Afișare informațiile detaliate ale producătorului;
- Editare Producător;
- Stergere producător;
- Afișarea articolelor producătorului;
- Adăugare producător nou.

Pentru o navigare mai ușoară, producătorul este afișat cu un fundal alternativ. Pagina suplimentară a producătorului vă permite să inserați o nouă înregistrare completând următoarele câmpuri:

- Numele producătorului;
- Site-ul web al acestuia;
- Descrierea.

**Adaugare producator**

Nume Producator:

Site Web:

Descriere:

Anuleaza modificari

Adauga

Figura 4.17 Meniul pentru adăugarea unui producător

Pagina cu opțiunea de editare este similară cu pagina adăugare, care declară că valoarea câmpului a fost completată, iar butonul „Adaugă” va fi înlocuit cu butonul „Modifică”. Următoarele opțiuni sunt utilizate pentru a defini piesele auto în funcție de tipul de mașină.

Opțiunile sale sunt:

- Afișați numele categoriei;
- Afișați subcategoriile incluse;
- Schimbă informații despre categorie;
- Sterge categoria;
- Adăugați o nouă categorie.

Adăugarea unei noi categorii presupune completarea numelui categoriei. La fel, schimbarea unei categorii implică schimbarea numelui categoriei.

**Modificare categorie**

Nume Categorie:

Anuleaza modificari

Modifica

Figura 4.18 Meniul pentru modificarea unei categorii

Următoarea opțiune din acest grup este utilizată pentru a defini subcategoriile. Funcționează similar cu definiția unei categorii și menționează că, în acest din urmă caz, trebuie selectată și categoria căreia îi aparține subcategoria.

În acest caz, opțiunile includ:

- Afișați numele subcategoriei;
- Afișați produsele conținute;
- Schimbați informații despre subcategorie;
- Sterge subcategoriile;
- Adăugați subcategorii noi.

Adăugarea sau editarea unei subcategorii implică selectarea categoriei părinte și apoi completarea numelui subcategoriei. Nerespectarea acestor câmpuri va duce la un mesaj de eroare:

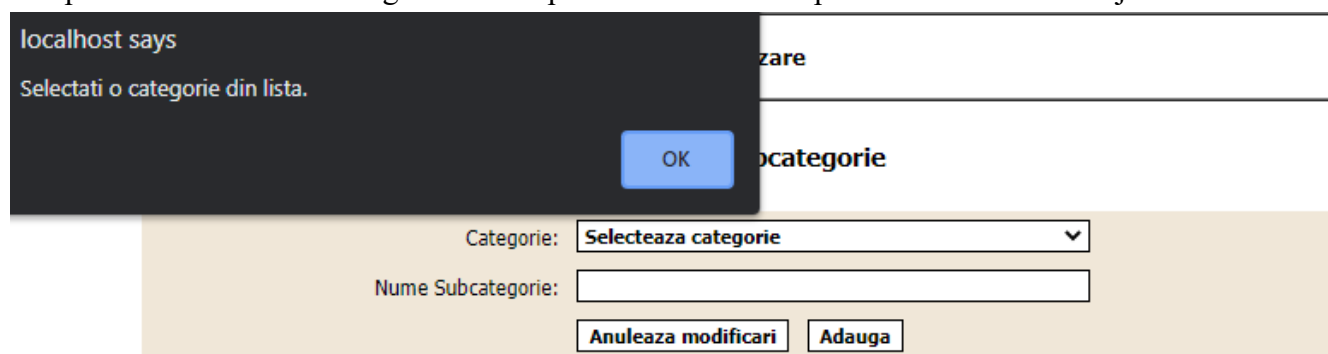


Figura 4.19 Mesaj de atenționare neselectare categorie

Ultima opțiune din acest grup este utilizată pentru a defini articolul. Se afișează: numele produsului; numele producătorului; numele categoriei; numele subcategoriei; detaliile produsului; editarea produsului; ștergerea produsului; adăugarea de produse noi.

Adăugarea de produse noi și editarea produselor trebuie să completeze următoarele câmpuri: categorie; subcategorie, nume produs, producător; cod produs, imagine produs; descriere produs; specificații tehnice produs; preț produs; garanție.

Imaginile produsului trebuie să respecte următoarele reguli:

- Fișiere acceptate: GIF / JPG
- Mărime maximă: 500kb
- Dimensiune maximă: 200 pixeli / 200 pixeli

**Adaugare produs**

Categorie:	<input type="text" value="Selecteaza categorie"/>
Subcategorie:	<input type="text" value="Selecteaza subcategorie"/>
Producator:	<input type="text" value="Selecteaza producator"/>
Nume Produs:	<input type="text"/>
Cod Produs:	<input type="text"/>
Imagine:	<input type="text"/> <input type="button" value="Browse..."/>
<small>Fisiere acceptate: GIF/JPG Marime maxima: 500kb Dimensiuni maxime: 200px/200px</small>	
Descriere Produs:	<input type="text"/>
Specificatii:	<div></div>
Pret (RON):	<input type="text"/>
Garantie (nr. luni):	<input type="text"/>
<input type="button" value="Anuleaza modificari"/> <input type="button" value="Adauga"/>	

Figura 4.20 Meniul pentru adăugare produs

```

<?php
 $strSQL = "SELECT * FROM tcategoriai";
 $result = mysql_query($strSQL);
 while ($row = mysql_fetch_array($result)) {
 $intIdCateg = $row['fIdCategorie'];
 $strNumeCategorie = $row['fNumeCategorie'];
 echo "<option value=\"\$intIdCateg\"";
 if ($intIdCategorie == $intIdCateg) {
 echo " selected ";
 }
 echo "> $strNumeCategorie</option>";
 }
?>
</select>
</td>
</tr>
<tr bgcolor="#F0E7D8">
 <td align="right" width="250">Subcategorii:</td>
 <td width="450">
 <select name="selSubcategorii" id="selSubcategorii" style="width: 300px;" class="text">
 <option value="select">Selecteaza subcategorii</option>
 <?php
 $strSQL = "SELECT * FROM tsubcategorii WHERE fIdCategorie=$intIdCategorie";
 $result = mysql_query($strSQL);
 while ($row = mysql_fetch_array($result)) {
 $intIdSubcateg = $row['fIdSubcategorii'];
 $strNumeSubcategorii = $row['fNumeSubcategorii'];
 echo "<option value=\"\$intIdSubcateg\"";
 if ($intIdSubcategorii == $intIdSubcateg) {
 echo " selected ";
 }
 echo "> $strNumeSubcategorii</option>";
 }
 ?>
 </td>
</tr>

```

Figura 4.21 Implementarea meniului de adăugare a unui produs

Următoarele grupuri de opțiuni permit să gestionați utilizatorii, comenzile de produse și planurile de servicii:

<b>Administrare utilizatori</b>
<b>Administrare comenzi</b>
<b>Vanzari totale</b>
<b>Vanzari pe clienti</b>
<b>Vanzari pe o perioada</b>
<b>Vanzari produse</b>
<b>Iesire</b>

Figura 4.22 Opțiunile de administrare a utilizatorilor și a comenzilor

Gestionarea utilizatorilor ne permite doar să îi ștergem, dar nu și să adăugăm sau să modificăm datele acestora.













Nume Utilizator	Detalii	Stergere
cristian		
mihai		
claudiu		
viorel		
radus		
george		

Figura 4.23 Lista cu utilizatorii înregistrați

De asemenea, puteți afișa detalii despre utilizator, iar adresa de e-mail a utilizatorului este afișată ca o legătură mailto, astfel încât administratorul să poată contacta rapid utilizatorul. Gestionarea comenzii afișează informații despre utilizatorul care a efectuat comanda și data comenzii. Informații detaliate despre produsul comandat, adresa de livrare a produsului, prețul și cantitatea totală pot fi găsite pe pagina de detalii a comenzii:

Secțiunea de administrare :: Administrare comenzi













Utilizator	Data Comenzii	Detalii	Stergere
mihai	2021-05-30		
mihai	2021-05-30		
claudiu	2021-05-30		
claudiu	2021-05-30		
claudiu	2021-05-30		
viorel	2021-05-30		
viorel	2021-05-30		

Figura 4.24 Meniul de administrare a comenzilor

Detalii comanda

Nume Utilizator:	cristian
Data Comanda:	2021-05-23
Nume Cumparator:	cristian
Email Cumparator:	<input type="text" value="cristian@yahoo.com"/>
Adresa Cumparator:	Ioc Craiova

Nume produs	Cantitate	Pret unitar	Total
Filtru aer	1	70 RON	83.3 RON
Curea transmisie cu caneluri	1	150 RON	178.5 RON
			<b>Total (cu TVA): 261.8 RON</b>

Figura 4.25 Detaliile unei comenzi finalizate

Administratorul poate utiliza comanda de afișare a adresei de e-mail pentru a contacta direct utilizatorul care a efectuat comanda. Secțiunea de gestionare a comenzilor afișează numele și informațiile detaliate ale comenzii: producătorul mașinii, locația, data.

În opțiunea comenzi totale sunt afișate comenzile pe utilizator, totalul vânzărilor înregistrate pe aplicație.



#### Administrare rapoarte

Utilizator	Data Comenzii	Nume produs/td>	Pret unitar	Total
cristian	2021-06-06	Pompa ulei	100 Lei	119 Lei
cristian	2021-06-07	Set placute frana	120 Lei	142.8 Lei
cristian	2021-06-18	Set placute frana	120 Lei	142.8 Lei
cristian	2021-06-24	Alternator	106 Lei	126.14 Lei
cristian	2021-06-26	Suport motor	75 Lei	89.25 Lei
mihai	2021-05-30	Starter	57 Lei	67.83 Lei
mihai	2021-05-30	Filtru aer	17 Lei	20.23 Lei
claudiu	2021-05-30	Starter	18 Lei	21.42 Lei
claudiu	2021-05-30	Filtru aer	17 Lei	20.23 Lei
viorel	2021-05-30	Pompa ulei	100 Lei	119 Lei
viorel	2021-05-30	Alternator	106 Lei	126.14 Lei
viorel	2021-05-10	Suport motor	75 Lei	89.25 Lei
radus	2021-06-07	Pompa apa	200 Lei	238 Lei
george	2021-06-30	Set pompa apa+curea dintata	450 Lei	535.5 Lei

NUMAR TOTAL DE FACTURI	14
TOTAL VALOARE CU TVA	1857.59

Figura 4.26 Meniul de vânzări totale

Următoarea opțiune ne poate arăta vânzările pe o anumită perioadă calendaristică, trebuie doar să introducem intervalul pentru care dorim să afișăm comenzile.

#### Administrare rapoarte

VANZARILE PENTRU O PERIOADA CALENDARISTICA( introduceri data in format 2021-05-22)

Data de inceput:

Data de sfarsit:

Afiseaza

Utilizator	Data Comenzii	Nume produs	Pret unitar	Total
cristian	2021-06-06	Pompa ulei	100 Lei	119 Lei
cristian	2021-06-07	Set placute frana	120 Lei	142.8 Lei
cristian	2021-06-18	Set placute frana	120 Lei	142.8 Lei
cristian	2021-06-24	Alternator	106 Lei	126.14 Lei
cristian	2021-06-26	Suport motor	75 Lei	89.25 Lei
radus	2021-06-07	Pompa apa	200 Lei	238 Lei
george	2021-06-30	Set pompa apa+curea dintata	450 Lei	535.5 Lei
NUMAR TOTAL DE FACTURI			7	
TOTAL VALOARE CU TVA			1393.49	

Figura 4.27 Meniul de vânzări pe o anumită perioadă calendaristică

Rapoartele cu vânzări pe clienți, conțin numărul de produse cumpărate și suma totală plătită pentru achiziționarea produselor. Cel mai bun client este cel care a realizat cea mai bună vânzare și ne poate ajuta să răsplătim fidelitatea acestuia.

#### Administrare rapoarte

Utilizator	Nr produse	Total
mihai	2	88.06
claudiu	5	446.25
viorel	2	245.14

#### CEL MAI FIDEL CLIENT

NUMAR TOTAL PRODUSE CUMPARATE	5
NUME CLIENT	claudiu
VANZARE REALIZATA	446.25

Figura 4.28 Meniul de vânzări pe clienți

Rapoartele cu vânzări pe produse, conțin tipul, numărul de produse cumpărate și suma totală plătită pentru achiziționarea lor. Cu ajutorul acestui raport puteam observa care produse se vând cel mai bine.

Bine ai venit **admin**

Modificare parola

Administrare producatori

Administrare masini

Administrare modele masini+motorizari

Administrare piese auto

Administrare utilizatori

Administrare comenzi

Vanzari totale

Vanzari pe clienti

Vanzari pe o perioada

Vanzari produse

Iesire

Nume produs	Nr produse vandute	Total suma vanzare
Alternator	1	126.14
Set placute frana	2	285.6
Starter	1	21.42
Pompa ulei	2	238
Starter	1	67.83
Filtru aer	2	40.46

CEL MAI VANDUT PRODUS

NUMAR TOTAL MASINI VANDUTE	2
NUME MASINA	Set placute frana
VANZARE REALIZATA	285.6

Figura 4.29 Meniul de vânzări pe produse

Utilizând opțiunea de deconectare, administratorul poate anula autentificarea pentru a preveni accesul neautorizat la panoul de gestionare a site-ului.

## CONCLUZII

Această lucrare este o aplicație web utilizată pentru organizarea magazinelor online de piese auto, care are caracteristicile unui site de comerț electronic, oferind clienților posibilitatea de a comanda bunuri publicate în magazinul online. Aplicația Auto este scrisă în PHP și SQL și oferă o interfață clasică pentru plasarea comenzilor de piese auto. Particularitatea aplicației este de a gestiona preferințele clienților pentru a oferi recomandări și instrumente de mesagerie internă pentru a comunica cu clienții care au aceleași preferințe atunci când cumpără piese auto.

Aplicația oferă un set de instrumente de gestionare a site-ului online și poate vizualiza cantitatea și informațiile despre comenzile plasate, precum și o listă a tuturor clienților care au achiziționat piese auto. Din punctul de vedere al clientului, aplicația dezvoltată este o alternativă mai ușor de utilizat și simplifică procesul de cumpărare a pieselor auto, iar din punctul de vedere al antreprenorului, simplifică și procesul de vânzare a pieselor auto. În dezvoltarea aplicației, este luată în considerare și experiența de cumpărare a consumatorilor digitali și se speră ca designul aplicației ar trebui să fie, de asemenea, atractiv și să răspundă la funcțiile de pe dispozitivul mobil. În plus, la dezvoltarea CMS (Content Management System) pentru site-ul administratorului, au fost luate în considerare și experiența utilizatorului și ușurința gestionării aplicației.

Ca perspective viitoare ale micilor afaceri online, antreprenorii care doresc să migreze sau să deschidă afaceri online pot folosi platforme de comerț electronic open source, precum Magento Community Edition, Drupal Commerce, PrestaShop, OpenCart, Zen Cart, Spree Commerce, simpleCart (js) , osCommerce, WooCommerce, WP e-commerce sau Jigoshop. În viitor, utilizatorilor li se pot oferi unele facilități, cum ar fi sistemele de plată online pentru achiziționarea articolelor (cupoane de cumpărare) și diverse informații statistice generate dinamic sub formă de diagrame.



## BIBLIOGRAFIE

- [1] TutorialsPoint, Websites Overview  
[https://www.tutorialspoint.com/internet\\_technologies/websites\\_overview.htm](https://www.tutorialspoint.com/internet_technologies/websites_overview.htm), accesat la data: 09.05.2021
- [2] TutorialsPoint, Websites Types  
[https://www.tutorialspoint.com/internet\\_technologies/websites\\_types.htm](https://www.tutorialspoint.com/internet_technologies/websites_types.htm), accesat la data: 09.05.2021
- [3] TutorialsPoint, Website Designing  
[https://www.tutorialspoint.com/internet\\_technologies/website\\_designing.htm](https://www.tutorialspoint.com/internet_technologies/website_designing.htm), accesat la data: 09.05.2021
- [4] TutorialsPoint, Website Development  
[https://www.tutorialspoint.com/internet\\_technologies/websites\\_development.htm](https://www.tutorialspoint.com/internet_technologies/websites_development.htm), accesat la data: 10.05.2021
- [5] TutorialsPoint, E-Commerce - Business Models  
[http://www.tutorialspoint.com/e\\_commerce/e\\_commerce\\_business\\_models.htm](http://www.tutorialspoint.com/e_commerce/e_commerce_business_models.htm), accesat la data: 10.05.2021
- [6] TutorialsPoint, Websites Publishing  
[https://www.tutorialspoint.com/internet\\_technologies/website\\_publishing.htm](https://www.tutorialspoint.com/internet_technologies/website_publishing.htm), accesat la data: 10.05.2021
- [7] Netcraft, PHP just grows & grows <http://news.netcraft.com/archives/2013/01/31/php-just-grows-grows.html>, accesat la data: 11.05.2021
- [8] TutorialsPoint, Websites URL Registration  
[https://www.tutorialspoint.com/internet\\_technologies/website\\_url\\_registration.htm](https://www.tutorialspoint.com/internet_technologies/website_url_registration.htm), accesat la data: 11.05.2021
- [9] Webopedia, Windows Operating System History & Versions  
[http://www.webopedia.com/DidYouKnow/Hardware\\_Software/history\\_of\\_microsoft\\_windows\\_operating\\_system.html](http://www.webopedia.com/DidYouKnow/Hardware_Software/history_of_microsoft_windows_operating_system.html), accesat la data: 11.05.2021
- [10] TutorialsPoint, Websites Hosting  
[https://www.tutorialspoint.com/internet\\_technologies/website\\_hosting.htm](https://www.tutorialspoint.com/internet_technologies/website_hosting.htm), accesat la data: 12.05.2021
- [11] TutorialsPoint, PHP - Introduction  
[https://www.tutorialspoint.com/php/php\\_introduction.htm](https://www.tutorialspoint.com/php/php_introduction.htm), accesat la data: 16.05.2021
- [12] TutorialsPoint, PHP - Environment Setup  
[https://www.tutorialspoint.com/php/php\\_environment.htm](https://www.tutorialspoint.com/php/php_environment.htm), accesat la data: 16.05.2021
- [13] TutorialsPoint, PHP - Variable Types  
[https://www.tutorialspoint.com/php/php\\_variable\\_types.htm](https://www.tutorialspoint.com/php/php_variable_types.htm), accesat la data: 16.05.2021
- [14] TutorialsPoint, PHP - Constants Types  
[https://www.tutorialspoint.com/php/php\\_constants.htm](https://www.tutorialspoint.com/php/php_constants.htm), accesat la data: 17.05.2021

- [15] TutorialsPoint, PHP - Operator Types  
[https://www.tutorialspoint.com/php/php\\_operator\\_types.htm](https://www.tutorialspoint.com/php/php_operator_types.htm), accesat la data: 17.05.2021
- [16] TutorialsPoint, PHP - Arrays  
[https://www.tutorialspoint.com/php/php\\_arrays.htm](https://www.tutorialspoint.com/php/php_arrays.htm), accesat la data: 17.05.2021
- [17] TutorialsPoint, PHP - Predefined Variables  
[https://www.tutorialspoint.com/php/php\\_predefined\\_variables.htm](https://www.tutorialspoint.com/php/php_predefined_variables.htm), accesat la data: 18.05.2021
- [18] TutorialsPoint, MySQL - Introduction  
<https://www.tutorialspoint.com/mysql/mysql-introduction.htm>, accesat la data: 18.05.2021
- [19] TutorialsPoint, MySQL - PHP Syntax  
<https://www.tutorialspoint.com/mysql/mysql-php-syntax.htm>, accesat la data: 18.05.2021
- [20] DevMySQL, Chapter 4 Creating and Using a Database  
<https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/database-use.html>, accesat la data: 19.05.2021
- [21] DevMySQL, 4.2 Creating a Table <https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/creating-tables.html>, accesat la data: 19.05.2021
- [22] DevMySQL, 4.3 Loading Data into a Table <https://dev.mysql.com/doc/mysql-tutorial-excerpt/8.0/en/loading-tables.html>, accesat la data: 19.05.2021
- [23] MDN Web Docs, What is CSS? [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS), accesat la data: 21.05.2021
- [24] MDN Web Docs, How CSS is structured [https://developer.mozilla.org/en-US/docs/Learn/CSS/First\\_steps/How\\_CSS\\_is\\_structured](https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_is_structured), accesat la data: 21.05.2021
- [25] Fela JS, Rules <https://fela.js.org/docs/latest/basics/rules>, accesat la data: 21.05.2021
- [26] HTML Dog, Tags, Attributes, and Elements  
<https://htmldog.com/guides/html/beginner/tags/>, accesat la data: 24.05.2021
- [27] HTML Dog, Lists <https://htmldog.com/guides/html/beginner/lists/>, accesat la data: 24.05.2021
- [28] HTML Dog, Links <https://htmldog.com/guides/html/beginner/links/>, accesat la data: 24.05.2021
- [29] HTML Dog, Images <https://htmldog.com/guides/html/beginner/images/>, accesat la data: 24.05.2021
- [30] HTML Dog, Tables <https://htmldog.com/guides/html/beginner/tables/>, accesat la data: 25.05.2021
- [31] HTML Dog, Forms <https://htmldog.com/guides/html/beginner/forms/>, accesat la data: 25.05.2021

- [32] TutorialsPoint, JavaScript – Overview  
[https://www.tutorialspoint.com/javascript/javascript\\_overview.htm](https://www.tutorialspoint.com/javascript/javascript_overview.htm), accesat la data: 26.05.2021
- [33] TutorialsPoint, JavaScript - Syntax  
[https://www.tutorialspoint.com/javascript/javascript\\_syntax.htm](https://www.tutorialspoint.com/javascript/javascript_syntax.htm), accesat la data: 26.05.2021
- [34] Ali Tarhini, Concepts of Three-Tier Architecture  
<https://alitarhini.wordpress.com/2011/01/22/concepts-of-three-tier-architecture/>, accesat la data: 30.05.2021
- [35] James Gillies și Robert Cailliau, (2000), How the Web Was Born: The Story of the World Wide Web
- [36] Stig Saether Bakken, Alexander Aulbach, Egon Schmidt, PHP Manual
- [37] Bootstrap, CSS <https://getbootstrap.com/docs/3.3/css/>, accesat la data: 20.05.2021
- [38] HTML Dog, Paragraphs <https://htmldog.com/guides/html/beginner/paragraphs/>, accesat la data: 26.05.2021
- [39] Chindea M. E. – Proiectarea sistemelor informatice economice, București, 2009
- [40] W3, CSS tutorial [www.w3.org/Style/Examples/011/firstcss.en.html](http://www.w3.org/Style/Examples/011/firstcss.en.html), accesat la data: 19.05.2021
- [41] W3schools, CSS Tutorial [www.w3schools.com/css/default.asp](http://www.w3schools.com/css/default.asp), accesat la data: 18.05.2021
- [42] Anghel, T. (2005), Dezvoltarea aplicațiilor WEB folosind XHTML, PHP și MySQL, Polirom, București.
- [43] Castro, E., Hyslop B. (2012), HTML5 and CSS3, Seventh Edition: Visual QuicksStart Guide, Peachpit Press, Berkeley.
- [44] Chindea M. E.(2009) Proiectarea sistemelor informatice economice, București
- [45] Doyle, M. (2010), Beginning PHP 5.3, Wiley Publishing, Inc., Indianapolis.
- [46] Duckett, J. (2011), HTML & CSS Design and Build Websites, John Wiley & Sons, Inc., Indianapolis.
- [47] Dașu, Narcisa Gabriela,.(2011), Crearea, proiectarea, optimizarea, publicarea și promovarea site-urilor Web , Buzău : Teocora;
- [48] Jay Greenspan si Brad Bulger- MySQL / PHP Database Application
- [49] Kevin Yank , Building a Database – Driven Web Site using PHP and MySQL
- [50] Liciniu A. Kovács ; colab.: Cormos Monica Lavinia, Pop Gabriel Marius , Ghid avansat de comerț electronic (2011), ed.: Colin Holcombe. - Cluj Napoca : Editura Fundației pentru Studii Europene,
- [51] Lambrescu, Ionuț. Analiza și prelucrarea imaginilor, (2011), Ploiești : Editura Universității Petrol-Gaze din Ploiești
- [52] Oprea D. (2009) Analiza și proiectarea sistemelor informaționale economice, Ed. POLIROM, Iași.

- [53] Stanciu V. (2017), Proiectarea sistemelor informatice de gestiune, Ed. Cison, București



## ANEXA 1 – Codul sursă

### mysql.inc.php

```
<?php
require_once("config.inc.php");
switch ($connType) {
 case "local" : {
 $server_name =
"localhost";

 $db_name = "pieseauto";
 $db_user = "root";
 $db_pass = "";
 break;
 }
}
//-----
function fDBConnect() {
 global $server_name;
 global $db_user;
 global $db_pass;
 global $db_name;
 $myConn = mysql_connect($server_name,
$db_user, $db_pass) or
 die("Error while connecting to
db: " . mysql_error());
 $res = mysql_select_db($db_name);
 if (!$res)
 die ("Error while opening the
db");
 return $myConn;
}
//-----
function fDBClose($link) {
 mysql_close ($link);
}
?>

functions.inc.php
<?php
require_once("mysql.inc.php");
function fInsertCSS() {
 echo "<link type=\"text/css\"
rel=\"stylesheet\" href=\"../css/style.css\"
/>\n";
 echo "<link type=\"text/css\" rel=\"stylesheet\"
href=\"../css/style1.css\" />\n";
}
function fInsertJS() {
 echo "<script type=\"text/javascript\"
src=\"../js/functions.js\"></script>\n";
 echo "<script type=\"text/javascript\"
src=\"../js/login.js\"></script>\n";
}
function fInsertLoginArea() {
 echo "<div id=\"loginarea\">\n";
 $intIdUtilizator
=isset($_SESSION['intIdUtilizator']) ?
$_SESSION['intIdUtilizator'] : '';
```

```
 if ($intIdUtilizator > 0) { // user is
authenticated, display name & logout link
 echo "Bine ai venit " .
$_SESSION['strNumeUtilizator'] . " <a
class=\"TopMyAccountLink\"
href=\"contul_meu.php\">:: Contul meu
| <a class=\"LogoutLink\"
href=\"execute.php?action=logout\">::
Iesire\n";
 } else { // user is not authenticated,
display login form
 echo "<form
action=\"execute.php?action=login\"
method=\"post\" style=\"display: inline;\"
onsubmit=\"return fValidateLogin();\">\n";
 echo "<fieldset style=\"padding:
0px; border: 0px;\">\n";
 echo "<label class=\"loginlabel\"
for=\"txtNumeUtilizator\">Utilizator:</label>\n"
;
 echo "<input class=\"logininput\"
type=\"text\" name=\"txtNumeUtilizator\"
id=\"txtNumeUtilizator\" maxlength=\"32\" />\n";
 echo "<label class=\"loginlabel\"
for=\"txtParola\">Parola:</label>\n";
 echo "<input class=\"logininput\"
type=\"password\" name=\"txtParola\"
id=\"txtParola\" maxlength=\"32\" />\n";
 echo "<input type=\"submit\"
alt=\"Autentificare\" class=\"loginbutton\"
name=\"cmdLogin\" id=\"cmdLogin\"
value=\"Autentificare\" />\n";
 echo "<input type=\"button\"
alt=\"Cont nou\" class=\"loginbutton\"
onclick=\"location.href='inregistrare.php'\"
name=\"cmdRegister\" id=\"cmdRegister\"
value=\"Cont nou\" />\n";
 echo "</fieldset>\n";
 echo "</form>\n";
 }
 }
 echo "</div>\n";
}
//-----
function fInsertTopMenu() {
 echo "<div class=\"btn-group\">\n";

 echo " <button
class=\"button\">Home</button>\n";
 echo "<button
class=\"button\">Administrare</button>\n";
 echo "<button
class=\"button\">Alege masina</button>\n";
```

```

 echo "<button
class=\"button\">Contact</button>\n";
 echo "</div>\n";
 }
//-----

function fInsertCategLeftMenu() {
 echo "<tr>\n";
 echo "<td style=\"text-align: center;\n
valign=\"top\">\n";
 //echo "<div id=\"divcategmenu\">\n";

 echo "<div class=\"btn-group1\">\n";
 //echo "<div class=\"item\">\n";
 //echo "<div class=\"leftmenu\">\n";
 echo "<div class=\"btn-group\">\n";
 echo "<center><button
class=\"button\" style=\"width: 200px;\n>PIESE
MASINI:</button></center></div>\n";
 $strSQL = "SELECT * FROM tcategorii
ORDER BY fIdCategorie";
 $result = mysql_query($strSQL);
 if (mysql_num_rows($result) > 0) { //
categories fou

 while ($row =
mysql_fetch_array($result)) {

 echo "<div class=\"dropdown\">\n";

 $intIdCategorie = $row['fIdCategorie'];
 echo "<a
href=\"categorii.php?intIdCategorie=" .
$row['fIdCategorie'] . "\"> <button
class=\"button1\" style=\"width: 200px;\n> " .
$row['fNumeCategorie'] . "</button>\n";

 $strSQLSub = "SELECT * FROM tsubcategorii WHERE
fIdCategorie=$intIdCategorie";
 $resultSub =
mysql_query($strSQLSub);

 echo "<div class=\"dropdown-content\">\n";

 if (mysql_num_rows($resultSub) > 0) { // exista
subcategorii

 while ($rowSub =
mysql_fetch_array($resultSub)) {

 $intIdSubcategorie =
$rowSub['fIdSubcategorie'];

 $strNumeSubcategorie =
$rowSub['fNumeSubcategorie'];
 echo "<a
href=\"subcategorii.php?intIdSubcategorie=$intId
Subcategorie\">$strNumeSubcategorie\n";

```

```

 }
 echo "</div>\n";
 echo "</div>\n";
}

 } else { // no category found
 echo "<div style=\"padding: 10px;
text-align: center;\n>Nici un tip produs
.</div>\n";
 }
 echo "</div>\n";
 //echo "</div>\n";
 //echo "</div>\n";
 echo "</td>\n";
 echo "</tr>\n";
}
//-----
function fInsertOthersLeftMenu() {
 echo "<tr>\n";
 echo "<td style=\"text-align:
center;\n>\n";

 echo "<div class=\"btn-group1\">\n";

 echo "<div class=\"btn-group\">\n";

 echo "
\n";echo "
\n";
 echo " <button
class=\"button\" style=\"width: 200px;\n>
CONTACT</button>\n";

 echo "</div>\n";
 echo "</div>\n";

 echo "</td>\n";
 echo "</tr>\n";
}
//-----
function fInsertBottomMenu() {
 echo "<div id=\"bottommenu\">\n";
 echo "<a class=\"BottomLink\"
href=\"index.php\">Acasa |
\n";
 echo "<a class=\"BottomLink\"
href=\"categorii.php?mode=main\">Masini
 | \n";

 echo "<a class=\"BottomLink\"
href=\"harta_site.php\">Harta Site |
 \n";
 echo "<a class=\"BottomLink\"
href=\"contact.php\">Contact\n";
 echo "</div>\n";
}
//-----
function fInsertProdLeft($intIdProdus) {
 $strSQL = "SELECT tproduse.* FROM
tproduse WHERE fIdProdus=$intIdProdus";

```

```

$result = mysql_query($strSQL);
$row = mysql_fetch_array($result);
$strName = $row['fNumeProdus'];
$strImage = $row['fImagine'];
$strSpecs = $row['fSpecificatii'];
$strPrice = $row['fPret'];

echo "<td style=\"width: 33%; text-align: center; vertical-align: top;\">\n";
echo "<table class=\"prodsumtable\">\n";
echo "<tr><td
class=\"prodsumtop\"></td></tr>\n";
echo "<tr><td class=\"prodsummain\">\n";
echo "<a
href=\"detalii_produs.php?intIdProdus=$intIdProdus\"
class=\"ProdNameLink\">$strName

\n";
echo "<img alt=\"$strName\"
src=\"./images/produse/$strImage\"
style=\"width: 140px; height: 110px; border: 1px #9EB79E solid;\" />
\n";
echo "<p style=\"text-align: left; padding: 4px;\">\n";
//echo nl2br($strSpecs);
echo "</p>\n";
echo "$strPrice RON

\n";
echo "<a
href=\"detalii_produs.php?intIdProdus=$intIdProdus\"
class=\"DetailsLink\">detalii\n";
echo "</td></tr>\n";
echo "<tr><td
class=\"prodsumbottom\"></td></tr>\n";
echo "</table>\n";
echo "</td>\n";
}
//-----
function fInsertProdRight($intIdProdus) {
 $strSQL = "SELECT tproduse.* FROM
tproduse WHERE fIdProdus=$intIdProdus";
 $result = mysql_query($strSQL);
 $row = mysql_fetch_array($result);
 $strName = $row['fNumeProdus'];
 $strImage = $row['fImagine'];
 $strSpecs = $row['fSpecificatii'];
 $strPrice = $row['fPret'];

 echo "<td style=\"width:33%; text-align: center; vertical-align: top;\">\n";
 echo "<table class=\"prodsumtable\">\n";
 echo "<tr><td
class=\"prodsumtop\"></td></tr>\n";
 echo "<tr><td class=\"prodsummain\">\n";
 echo "<a
href=\"detalii_produs.php?intIdProdus=$intIdProdus\"

```

```

us\"
class=\"ProdNameLink\">$strName

\n";
echo "<img alt=\"$strName\"
src=\"./images/produse/$strImage\"
style=\"width: 140px; height: 110px; border: 1px #9EB79E solid;\" />
\n";
echo "<p style=\"text-align: left; padding: 4px;\">\n";
//echo nl2br($strSpecs);
echo "</p>\n";
echo "$strPrice RON

\n";
echo "<a
href=\"detalii_produs.php?intIdProdus=$intIdProdus\"
class=\"DetailsLink\">detalii\n";
echo "</td></tr>\n";
echo "<tr><td
class=\"prodsumbottom\"></td></tr>\n";
echo "</table>\n";
echo "</td>\n";
}
//-----
function fInsertCartLink() {
 echo "<table id=\"cart\"><tr><td>\n";
 echo "<img alt=\"Cosul de cumparaturi\" src=\"./images/cart.gif\"
/>\n";
 echo "</td></tr></table>\n";
}
//-----
function fClearCart($intIdUtilizator) { //
goleste cosul
 $strSQL = "DELETE FROM tcos WHERE
fIdUtilizator=$intIdUtilizator";
 mysql_query($strSQL);
}
?>

```

## adminfunctions.inc.php

```

<?php
require("config.inc.php");
function fCheckAdminLogin() {
 $intIdAdmin=
isset($_SESSION['intIdAdmin'])?
$_SESSION['intIdAdmin'] : '';
 //if (($SESSION['intIdAdmin'] == "") ||
(!isset($_SESSION['intIdAdmin']))) {
if (($intIdAdmin == "") || (!isset($intIdAdmin))) {
 return false;
 } else {
 return true;
 }
}
//-----
function fCheckUserPassAdmin($strUsername,
$strPassword) {

```

```

 $strSQL = "SELECT fIdAdmin, fUsername
FROM tadmins WHERE fUsername='$strUsername' AND
fPassword='$strPassword'";
 $result = mysql_query($strSQL);
 if (mysql_num_rows($result) < 1) {
 return false;
 } else {
 $row =
mysql_fetch_array($result);
 $_SESSION['intIdAdmin'] =
$row['fIdAdmin'];
 $_SESSION['strUsername'] =
$row['fUsername'];
 return true;
 }
 }
//-----
function fPrintLoginAreaAdmin($bIsLoggedIn,
$strError) {
 if ($bIsLoggedIn) {
 // the admin is logged in, say HI
 and display link to cart and link to logout
 $strUsername =
$_SESSION['strUsername'];
 echo "<table
class=\"modelspecs\"><tr><td>\n";
 echo "
Bine ai venit
$strUsername

";
 echo "<a class=\"genLinkMenu\"
href=\"modificaparola.php\">Modificare
parola
";

 echo "<a class=\"genLinkMenu\"
href=\"prodicatori.php\">Administrare
prodicatori";
 echo "<a class=\"genLinkMenu\"
href=\"categorii.php\">Administrare masini";
 echo "<a class=\"genLinkMenu\"
href=\"subcategorii.php\">Administrare modele
masini+motorizari";
 echo "<a class=\"genLinkMenu\"
href=\"produse.php\">Administrare piese
auto
";

 echo "<a class=\"genLinkMenu\"
href=\"utilizatori.php\">Administrare
utilizatori";

 echo "<a class=\"genLinkMenu\"
href=\"comenzi.php\">Administrare comenzi";

 echo "<a class=\"genLinkMenu\"
href=\"rapoarte.php\">Vanzari totale";
 echo "<a class=\"genLinkMenu\"
href=\"rapoartel.php\">Vanzari pe clienti";
 echo "<a class=\"genLinkMenu\"
href=\"rapoarte2.php\">Vanzari pe o
perioada";
 echo "<a class=\"genLinkMenu\"
href=\"rapoarte3.php\">Vanzari produse";
 }
}

```

```

 echo "<a class=\"genLinkMenu\"
href=\"execute.php?action=logout\">Iesire";
 echo "</td></tr></table>\n";
 } else {
 // the admin is not logged in,
 display the login/register form
 echo "<form name=\"frmLogin\"
method=\"post\"
action=\"execute.php?action=login\"
onsubmit=\"return fValidateLogin();\">\n";
 if ($strError == "loginerror") {
 echo "<center>Eroare
autentificare!</center>\n";
 }
 echo "<table
class=\"modelspecs\"><tr><td>Utilizator:
\n";
 echo "<input class=\"text\"
type=\"text\" style=\"width: 150px;\"
name=\"txtUsername\" id=\"txtUsername\"
maxlength=\"32\">
\n";
 echo "Parola:
\n";
 echo "<input class=\"text\"
type=\"password\" style=\"width: 150px;\"
name=\"txtPassword\" id=\"txtPassword\"
maxlength=\"32\">

\n";
 echo "<input class=\"buttons\"
type=\"submit\" style=\"width: 150px;\"
name=\"cmdLogin\" id=\"cmdLogin\"
value=\"Autentificare\">

\n";
 echo
"</td></tr></table></form>\n";
 }
}
??

```

## execute.php

```

<?php
 session_start();
 include_once("../include/functions.inc.ph
p");

 $myConn = fDBConnect();

 $action = $_GET['action'];

 //-----
 if ($action == "login") { //
 autentificarea utilizatorului. daca datele sunt
 corecte se asigneaza variabilele sesiune
 utilizate pentru identificare, altfel se
 afiseaza mesaj eroare autentificare
 $strNumeUtilizator =
 trim(htmlspecialchars($_POST['txtNumeUtilizator'
]));
 $strParola =
 trim(htmlspecialchars($_POST['txtParola']));
 $strSQL = "SELECT fIdUtilizator,
fNumeUtilizator FROM tutilizatori WHERE

```

```

fNumeUtilizator='$strNumeUtilizator' AND
fParola='$strParola';
 $result = mysql_query($strSQL);
 if (mysql_num_rows($result) > 0)
{ // login successfull
 $row =
mysql_fetch_array($result);

 $_SESSION['intIdUtilizator'] =
$row['fIdUtilizator'];

 $_SESSION['strNumeUtilizator'] =
$row['fNumeUtilizator'];

 fClearCart($_SESSION['intIdUtilizator'])
;

 Header("Location:index.php");
 } else { // login fail

 Header("Location:loginerror.php");
 }
}
//-----
if ($action == "logout") { // iesire din
cont, se sterg variabilele sesiune utilizate
pentru identificare
 $_SESSION['intIdUtilizator'] = -
1;
 $_SESSION['strNumeUtilizator'] =
"";
 Header("Location:index.php");
}
//-----
if ($action == "addtocart") { // adauga
produs in cosul de cumparaturi
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 if ($intIdUtilizator > 0) { //
userul este logat
 //echo "
user logat
";
 $intIdProdus =
$_POST['hddIntIdProdus'];
 $strIdSesiune =
session_id();
 $strSQL = "SELECT * FROM
tcos WHERE fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus AND
fIdSesiune='$strIdSesiune'";
 //echo "
$strSQL
";
 $result =
mysql_query($strSQL);
 if
(mysql_num_rows($result) > 0) { // produsul se
afla deja in cos, incrementam cantitatea
 //echo "
deja
exista
";
 $row =
mysql_fetch_array($result);

```

```

 $intIdItem =
$row['fIdItem'];
 $intCantitate =
$row['fCantitate'];
 $intCantitateNoua
= $intCantitate + 1;
 $strSQL = "UPDATE
tcos SET fCantitate=$intCantitateNoua WHERE
fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus AND
fIdSesiune='$strIdSesiune' AND
fIdItem=$intIdItem";

 mysql_query($strSQL);
 } else { // produsul nu
se afla in cos, il adaugam
 //echo "
nu
exista
";
 $strSQL = "INSERT
INTO tcos (fIdUtilizator, fIdProdus, fCantitate,
fIdSesiune) VALUES ($intIdUtilizator,
$intIdProdus, 1, '$strIdSesiune')";

 mysql_query($strSQL);
 }
}
Header("Location:
cos_cumparaturi.php");
}
//-----
if ($action == "emptycart") {
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 fClearCart($intIdUtilizator);
 Header("Location:
cos_cumparaturi.php");
}
//-----
if ($action == "delfromcart") {
 $intIdProdus =
$_GET['intIdProdus'];
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 $strSesiune = session_id();
 $strSQL = "DELETE FROM tcos WHERE
fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus AND
fIdSesiune='$strSesiune'";
 mysql_query($strSQL);
 Header("Location:
cos_cumparaturi.php");
}
//-----
if ($action == "updatecart") {
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 $strSesiune = session_id();
 $strSQL = "SELECT * FROM tcos
WHERE fIdSesiune='$strSesiune' AND
fIdUtilizator=$intIdUtilizator";

```

```

 $result = mysql_query($strSQL);
 while ($row =
mysql_fetch_array($result)) {
 $intIdProdus =
$row['fIdProdus'];
 $newval =
intval($_POST['txt' . $intIdProdus]);
 if ($newval == 0) { //
sterge produs din cos
 $strSQLDelete =
"DELETE FROM tcos WHERE fIdSesiune='$strSesiune'
AND fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus";

 mysql_query($strSQLDelete);

 } else { // actualizeaza
cantitatea
 $strSQLUpdate =
"UPDATE tcos SET fCantitate=$newval WHERE
fIdSesiune='$strSesiune' AND
fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus";

 mysql_query($strSQLUpdate);
 }
 Header("Location:
cos_cumparaturi.php");
 }
 //-----
 if ($action == "voteproduct") {
 $intIdProdus =
intval($_POST['hddIntIdProdus']);
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 $intVoteValue =
intval($_POST['selVoteValue']);
 $strSQL = "SELECT * FROM tvoturi
WHERE fIdUtilizator=$intIdUtilizator AND
fIdProdus=$intIdProdus";
 $result = mysql_query($strSQL);
 if (mysql_num_rows($result) > 0)
 {
 Header("Location:
detalii_produs.php?intIdProdus=$intIdProdus");
 } else {
 $strSQL = "INSERT INTO
tvoturi (fIdProdus, fIdUtilizator, fValoareVot)
VALUES ($intIdProdus, $intIdUtilizator,
$intVoteValue)";

 mysql_query($strSQL);
 Header("Location:
detalii_produs.php?intIdProdus=$intIdProdus");
 }
 }
 //-----
 if ($action == "inregistrare") {

```

```

 $strNum =
htmlspecialchars(trim($_POST['txtNumInregistrare']));
 $strMasina =
htmlspecialchars(trim($_POST['txtNumMasina']));
 $strLocalitate =
htmlspecialchars(trim($_POST['txtNumLocalitate']));
 $strData =
htmlspecialchars(trim($_POST['txtData']));
 $strMesaj =
htmlspecialchars(trim($_POST['txtMesaj']));
 $strToday = date("Ymd");

 $strSQL = "INSERT INTO trezervari
(fNumUtilizator, fMasina, fLocalitate, fData,
fMesaj, fDataInregistrare) VALUES ('$strNum',
'$strMasina', '$strLocalitate', '$strData',
'$strMesaj', '$strToday')";
 mysql_query($strSQL);
 Header("Location:
felicitari.php");
 }
 //-----
 if ($action == "inregistrare") {
 $strNum =
htmlspecialchars(trim($_POST['txtNumInregistrare']));
 $strParola =
htmlspecialchars(trim($_POST['txtParolaInregistrare']));
 $strEmail =
htmlspecialchars(trim($_POST['txtEmail']));
 $strAdresa =
htmlspecialchars(trim($_POST['txtAdresa']));
 $strToday = date("Ymd");

 $strSQL = "INSERT INTO
tutilizatori (fNumUtilizator, fParola, fEmail,
fAdresa, fDataInregistrare) VALUES ('$strNum',
'$strParola', '$strEmail', '$strAdresa',
'$strToday')";
 mysql_query($strSQL);
 Header("Location:
felicitari.php");
 }
 //-----
 if ($action == "actualizarecont") {
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];
 $strParola =
htmlspecialchars(trim($_POST['txtParolaInregistrare']));
 $strEmail =
htmlspecialchars(trim($_POST['txtEmail']));
 $strAdresa =
htmlspecialchars(trim($_POST['txtAdresa']));

```

```

 $strSQL = "UPDATE tutilizatori
SET fParola='$strParola', fEmail='$strEmail',
fAdresa='$strAdresa' WHERE
fIdUtilizator=$intIdUtilizator";
 mysql_query($strSQL);
 Header("Location:
contul_meu.php");
 }
 //-----
 if ($action == "comanda") {
 $intIdUtilizator =
$_SESSION['intIdUtilizator'];

 /*$strCupon =
$_POST['txtCuponCumparator'];
 $strSQLCupon = "SELECT * FROM
tcupoane WHERE fCodCupon='$strCupon'";
 $resultCupon =
mysql_query($strSQLCupon);
 if (mysql_num_rows($resultCupon)
> 0) { //cupon valid
 $rowCupon =
mysql_fetch_array($resultCupon);
 $intIdCupon =
$rowCupon['fIdCupon'];
 } else {
 $intIdCupon = 0;
 }*/

 $strNume =
$_POST['txtNumeCumparator'];
 $strEmail =
$_POST['txtEmailCumparator'];
 $strAdresa =
$_POST['txtAdresaCumparator'];
 $strToday = date("Ymd");

 $strSQL = "INSERT INTO tcomenzi
(fIdUtilizator, fNumeCumparator,
fEmailCumparator, fAdresaCumparator,
fDataComanda) VALUES ($intIdUtilizator,
'$strNume', '$strEmail', '$strAdresa',
'$strToday')";
 mysql_query($strSQL);
 $intIdComanda =
mysql_insert_id();

 $strIdSesiune = session_id();
 $strSQL = "SELECT * FROM tcos
WHERE fIdUtilizator=$intIdUtilizator AND
fIdSesiune='$strIdSesiune'";
 $result = mysql_query($strSQL);
 while ($row =
mysql_fetch_array($result)) {
 $intIdProdus =
$row['fIdProdus'];
 $intCantitate =
$row['fCantitate'];
 $strSQLArticole = "INSERT
INTO tprodusecomenzi (fIdComanda, fIdProdus,

```

```

fCantitate) VALUES ($intIdComanda, $intIdProdus,
$intCantitate)";
 mysql_query($strSQLArticole);
 }
 $strSQLStergere = "DELETE FROM
tcos WHERE fIdUtilizator=$intIdUtilizator";
 mysql_query($strSQLStergere);

 Header("Location:
final_comanda.php");
}
//-----

fDBClose($myConn);
?>

```

## index.php

```

<?php
 session_start();
 include_once("../include/functions.inc.ph
p");

 $myConn = fDBConnect();
 fSetProductOfTheDayID();
 $intIdCategorieCautare
=isset($_GET['intIdCategorieCautare']) ?
$_GET['intIdCategorieCautare'] : '';

?>
<head>
<title> Cele mai performante produse</title>
<?php fInsertCSS(); ?>
<?php fInsertJS(); ?>
</head>
<body>
 <table class="maintable">
 <table class="maintable">
<tr><td><?php fInsertTopMenu();?>

<?php fInsertLoginArea(); ?> </td></tr>
 <tr style="height: 220px;">
 <td id="header">
 <table id="cart">
 <tr>
 <td>

 </td>
 </tr>
 </table>
 </td>
 </tr>
 </table>
 <tr>
 <td>
 <table
class="contenttable">
 <tr>

```

```

class="leftcolcontainer">
 <!--start left column-->
 <table class="leftcoltable">
 <tr>
 style="height: 5px;"><td></td></tr>
 <!--START MENU CATEGORII-->
 <?php fInsertCategLeftMenu(); ?>
 <!--END MENU CATEGORII-->
 <tr style="height: 5px;"><td></td></tr>
 <tr style="height: 5px;"><td></td></tr>
 <!--START MENU ALTELE-->
 <?php fInsertOthersLeftMenu(); ?>
 <!--END MENU ALTELE-->
 </table>
 <!--end left column-->
</td>
id="maincontent">
-START MAIN CONTENT-->
 <table style="width: 100%;">
 <tr>
 <?php
 //$intTodayId =
 $_SESSION['intProdofDay'];

 //$strSQL = "SELECT tproduse.fIdProdus,
 RAND() as rnd_id FROM tproduse LEFT OUTER JOIN
 tpromotii ON
 (tpromotii.fIdProdus=tproduse.fIdProdus) WHERE
 (tpromotii.fIdProdus is null) AND
 tproduse.fIdProdus<>$intTodayId ORDER BY rnd_id
 LIMIT 6";

 $strSQL = "SELECT tproduse.fIdProdus,
 RAND() as rnd_id FROM tproduse ORDER BY rnd_id
 LIMIT 6";

 $result = mysql_query($strSQL);
 </tr>
 </table>
 </td></tr>
 <tr>
 <td id="footer">
 <!--START BOTTOM MENU-->
 $intIndex = 0;

 while ($row =
 mysql_fetch_array($result)) {
 $intIndex++;

 $aMainIDs[$intIndex] =
 $row['fIdProdus'];
 }
 ?>
 <!--
 -START MAIN - PRODUCT LEFT-->
 <?php fInsertProdLeft($aMainIDs[1]); ?>
 <!--
 -END MAIN - PRODUCT LEFT-->
 <!--
 -START MAIN - PRODUCT RIGHT-->
 <?php fInsertProdRight($aMainIDs[2]); ?>
 <!--
 -END MAIN - PRODUCT RIGHT-->
 </tr><tr>
 <!--START MAIN - PRODUCT LEFT-->
 <?php fInsertProdLeft($aMainIDs[4]); ?>
 <!--
 -END MAIN - PRODUCT LEFT-->
 <!--
 -START MAIN - PRODUCT RIGHT-->
 <?php fInsertProdRight($aMainIDs[5]); ?>
 <!--
 -END MAIN - PRODUCT RIGHT-->
 </tr>
 </table>
 </td></tr>
 <tr>
 <td id="footer">
 <!--START BOTTOM MENU-->

```



```

 <?php
fInsertBottomMenu(); ?>
 <!--END BOTTOM MENU-->
 <?php
fInsertBottomStdText(); ?>
 </td>
 </tr>
 </table>

</body>
</html>
<?php fDBClose($myConn); ?>
index.php (admin)
<?php
 session_start();
 include_once("../include/mysql.inc.php");
;
 include_once("../include/functions.inc.p
hp");
 include_once("../include/adminfunctions.
inc.php");
 $myConn = fDBConnect();
 $bIsLoggedIn = fCheckAdminLogin();
 $strError = isset($_GET['strError'])?
$_GET['strError'] : '';

?>
<html>
 <head>
 <title>AUTOMAG.ro :: Sectiunea
de administrare</title>
 <link rel="stylesheet"
href="../css/admin.css" type="text/css">
 <script language="javascript"
src="../js/functions.js"></script>
 </head>
 <body>
 <table cellpadding="0"
cellspacing="0" border="1" width="100%"
height="100%" >
 <tr style="height: 50px;
width: 100%;">
 <td colspan="2"
style="font-weight: bold; text-align: center;
height: 50px; width: 100%; font-family: Tahoma,
Verdana, Arial; font-size: 10pt;">AutoMag.ro ::
Sectiunea de administrare</td>
 </tr>
 <tr>
 <td style="width:
200px;" valign="top">
 <?php
fPrintLoginAreaAdmin($bIsLoggedIn, $strError);
?>
 </td>
 <td style="width:
700px; font-family: Verdana, Tahoma, Arial;
font-size: 10pt; font-weight: bold;"
valign="middle" align="center">Va rugam sa va

```

```

autentificati sau sa alegeti o optiune din
meniul de administrare.</td>
 </tr>
 </table>
 </body>
</html>
rapoarte3.php
<?php
 session_start();
 include_once("../include/mysql.inc.php");
;
 include_once("../include/functions.inc.p
hp");
 include_once("../include/adminfunctions.
inc.php");
 $bIsLoggedIn = fCheckAdminLogin();
 if (!$bIsLoggedIn) {
 Header("Location:index.php");
 }
 $myConn = fDBConnect();
 $strError = isset($_GET['strError'])?
$_GET['strError'] : '';

?>
<html>
 <head>
 <title>Sectiunea de
administrare</title>
 <link rel="stylesheet"
href="../css/admin.css" type="text/css">
 <script language="javascript"
src="../js/functions.js"></script>
 </head>
 <body>
 <table cellpadding="0"
cellspacing="0" border="0" width="100%"
height="100%">
 <tr style="height: 50px;
width: 100%;">
 <td colspan="2"
style="font-weight: bold; text-align: center;
height: 50px; width: 100%; font-family: Tahoma,
Verdana, Arial; font-size: 10pt;">Administrare
rapoarte</td>
 </tr>
 <tr>
 <td style="width:
200px;" valign="top">
 <?php
fPrintLoginAreaAdmin($bIsLoggedIn, $strError);
?>
 </td>
 <td style="width:
700px;" valign="top" align="center">
 <?php
 echo "<table cellpadding=\"3\"
cellspacing=\"0\" border=\"1\" width=\"700\"
class=\"modelspecs\">\n";

```

```

echo "<tr align=\"center\" style=\"font-weight:
bold;\>";

 echo "<td width=\"350\">Nume
produs</td>";

 echo "<td width=\"150\">Nr produse
vandute</td>";

 echo "<td width=\"200\">Total suma
vanzare</td>";

 echo "</tr>";

$strSQL = "SELECT * FROM tproduse";
$result = mysql_query($strSQL);
$max1 = 0; $s= 0;$nume="";
while ($row = mysql_fetch_array($result)) {
 $fNumeProdus = $row['fNumeProdus'];
 $intIdProdus = $row['fIdProdus'];
 $pret= $row['fPret'];
 $intCount = 0;$intsuma = 0;
 $strSQL1 = "SELECT tcomenzi.fIdComanda,
tprodusecomenzi.fIdComanda,
tprodusecomenzi.fCantitate,
tprodusecomenzi.fIdProdus FROM tcomenzi,
tprodusecomenzi WHERE tcomenzi.fIdComanda =
tprodusecomenzi.fIdComanda AND
tprodusecomenzi.fIdProdus= $intIdProdus";
 $result1 = mysql_query($strSQL1);

 while ($row = mysql_fetch_array($result1)) {
 $fCantitate=$row['fCantitate'];

 $intCount =$intCount +$fCantitate;
 $intsuma = $intsuma + ($pret * 1.19) *
 $fCantitate;
 // $intnrimob = 0;
 // $inttotal = 0;
 }

 if($intCount>$max1) { $max1= $intCount;
 $nume=$fNumeProdus ;$s= $intsuma;}
 //else if($intCount=$max1 and
 $intsuma>$s) { $max1= $intCount;
 $nume=$fNumeProdus ;$s= $intsuma;}
 if($intCount>0){
 echo
"<tr align=\"center\" style=\"font-weight:
bold;\>";

 echo "<td width=\"50\">
$fNumeProdus</td>";

 echo "<td width=\"50\"> $intCount</td>";

 echo "<td width=\"375\">$intsuma</td>";

```

```

echo "</tr>";
 }

 echo "</table>\n"; echo "
";
 echo "
";

 echo "<table cellpadding=\"3\" cellspacing=\"0\"
border=\"1\" width=\"700\" class=\"modelspecs\"
>\n";
 echo "<tr align=\"left\" style=\"font-weight:
bold;\> CEL MAI VANDUT PRODUS</tr>\n";
 echo "<tr align=\"left\" style=\"font-weight:
bold;\>";

 echo "<td width=\"80\">NUMAR TOTAL
PRODUSE VANDUTE</td>";

 echo "<td width=\"20\">$max1</td>";
 echo "</tr>\n";

 echo "<tr align=\"left\" style=\"font-weight:
bold;\>";

 echo "<td width=\"70\">NUME PRODUS
</td>";

 echo "<td width=\"30\">$nume </td>";
 echo "</tr>\n";
 echo "<tr align=\"left\" style=\"font-weight:
bold;\>";

 echo "<td width=\"70\">VANZARE REALIZATA
</td>";

 echo "<td width=\"30\">$s </td>";
 echo "</tr>\n";
 echo "</table>\n"; echo "
"; echo
"
"; echo "
";

?>

</td>

</tr>

</table>

</body>
</html>

```

## functions.js

```
function fConfirmEmptyCart() {
 return (confirmSubmit("Doriti sa
stergeti continutul cosului ?"));
}
//-----
function numeralsOnly(evt) {
 evt = (evt) ? evt : event;
 var charCode = (evt.charCode) ?
evt.charCode : ((evt.keyCode) ? evt.keyCode :
((evt.which) ? evt.which
: 0));
 if (charCode > 31 && (charCode < 48 ||
charCode > 57)) {
 alert("Caractere permise
pentru acest camp: 0123456789");
 return false;
 }
 return true;
}
//-----
function fValidareAdaugareProducator() {
 if
(document.getElementById("txtNume").value == "")
{
 alert ("Completati campul 'Nume
Producator'.");

 document.getElementById("txtNume").focus
();
 return false;
 }
 if
(document.getElementById("txtURL").value == "")
{
 alert ("Completati campul 'Site
Web'.");

 document.getElementById("txtURL").focus(
);
 return false;
 }
 if
(document.getElementById("txtDescriere").value
== "") {
 alert ("Completati campul
'Descriere'.");

 document.getElementById("txtDescriere").
focus();
 return false;
 }
 return true;
}
//-----
function fValidareAdaugareCategorie() {
 if
(document.getElementById("txtNume").value == "")
{
```

```
 alert ("Completati campul 'Nume
Categorie'.");

 document.getElementById("txtNume").focus
();
 return false;
 }
 return true;
}
//-----
function fValidareInregistrare() {
 if
(document.getElementById("txtNumeInregistrare").
value == "") {
 alert ("Completati campul 'Nume
Utilizator'.");

 document.getElementById("txtNumeInregist
rare").focus();
 return false;
 }
 if
(document.getElementById("txtParolaInregistrare"
).value == "") {
 alert ("Completati campul
'Parola'.");

 document.getElementById("txtParolaInregi
strare").focus();
 return false;
 }
 if
(document.getElementById("txtParolaInregistrare"
).value !=
document.getElementById("txtParolaInregistrare2"
).value) {
 alert ("Confirmati parola");

 document.getElementById("txtParolaInregi
strare2").focus();
 return false;
 }
 if
(document.getElementById("txtEmail").value ==
"") {
 alert ("Completati campul 'Adresa
Email'.");

 document.getElementById("txtEmail").focu
s();
 return false;
 }
 if
(document.getElementById("txtEmail").value !=
document.getElementById("txtEmail2").value) {
 alert ("Confirmati adresa
email");

 document.getElementById("txtEmail2").foc
us();
```

```

 return false;
 }
 if
(document.getElementById("txtAdresa").value ==
"") {
 alert ("Completati campul
'Adresa'.");

 document.getElementById("txtAdresa").foc
us();
 return false;
 }
 return true;
}
//-----
function fValidareActualizare() {
 if
(document.getElementById("txtParolaInregistrare"
).value == "") {
 alert ("Completati campul
'Parola'.");

 document.getElementById("txtParolaInregi
strare").focus();
 return false;
 }
 if
(document.getElementById("txtParolaInregistrare"
).value !=
document.getElementById("txtParolaInregistrare2"
).value) {
 alert ("Confirmati parola");

 document.getElementById("txtParolaInregi
strare2").focus();
 return false;
 }
 if
(document.getElementById("txtEmail").value ==
"") {
 alert ("Completati campul 'Adresa
Email'.");

 document.getElementById("txtEmail").focu
s();
 return false;
 }
 if
(document.getElementById("txtEmail").value !=
document.getElementById("txtEmail2").value) {
 alert ("Confirmati adresa
email");

 document.getElementById("txtEmail2").foc
us();
 return false;
 }
 if
(document.getElementById("txtAdresa").value ==
"") {

```

```

 alert ("Completati campul
'Adresa'.");

 document.getElementById("txtAdresa").foc
us();
 return false;
 }
 return true;
}
//-----
function fValidareComanda() {
 if
(document.getElementById("txtNumeCumparator").va
lue == "") {
 alert ("Completati campul 'Nume
Cumparator'.");

 document.getElementById("txtNumeCumparat
or").focus();
 return false;
 }
 if
(document.getElementById("txtEmailCumparator").v
alue == "") {
 alert ("Completati campul 'Adresa
Email'.");

 document.getElementById("txtEmailCumpara
tor").focus();
 return false;
 }
 if
(document.getElementById("txtEmailCumparator").v
alue !=
document.getElementById("txtEmailCumparator2").v
alue) {
 alert ("Confirmati adresa
email");

 document.getElementById("txtEmailCumpara
tor2").focus();
 return false;
 }
 if
(document.getElementById("txtAdresaCumparator").
value == "") {
 alert ("Completati campul
'Adresa'.");

 document.getElementById("txtAdresaCumpar
ator").focus();
 return false;
 }
 return true;
}

```

## pieseauto.sql

```

--
-- Table structure for table `tadmins`
--

```

```

CREATE TABLE `tadmins` (
 `fIdAdmin` int(11) UNSIGNED NOT NULL,
 `fUsername` varchar(32) DEFAULT NULL,
 `fPassword` varchar(32) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `tategorii`
--

CREATE TABLE `tategorii` (
 `fIdCategorie` int(11) UNSIGNED NOT NULL,
 `fNumCategorie` varchar(50) NOT NULL DEFAULT ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tcomenzi`
--

CREATE TABLE `tcomenzi` (
 `fIdComanda` int(11) UNSIGNED NOT NULL,
 `fIdUtilizator` int(11) UNSIGNED NOT NULL
 DEFAULT '0',
 `fIdCupon` int(11) DEFAULT NULL,
 `fNumCumparator` varchar(100) DEFAULT NULL,
 `fEmailCumparator` varchar(100) DEFAULT NULL,
 `fAdresaCumparator` text,
 `fDataComanda` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tcos`
--

CREATE TABLE `tcos` (
 `fIdItem` int(11) UNSIGNED NOT NULL,
 `fIdUtilizator` int(11) UNSIGNED NOT NULL
 DEFAULT '0',
 `fIdProdus` int(11) UNSIGNED NOT NULL DEFAULT '0',
 `fCantitate` int(11) UNSIGNED NOT NULL DEFAULT '0',
 `fIdSesiune` varchar(50) NOT NULL DEFAULT ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tproduse`
--

CREATE TABLE `tproduse` (
 `fIdProdus` int(11) UNSIGNED NOT NULL,
 `fIdSubcategorie` int(11) UNSIGNED NOT NULL
 DEFAULT '0',
 `fIdProducator` int(11) UNSIGNED NOT NULL
 DEFAULT '0',
 `fNumProdus` varchar(255) NOT NULL DEFAULT '',
 `fCodProdus` varchar(50) DEFAULT NULL,
 `fImagine` varchar(100) DEFAULT NULL,
 `fDescriere` varchar(255) DEFAULT NULL,
 `fSpecificatii` blob,
 `fPret` varchar(50) NOT NULL DEFAULT '0',
 `fLuniGarantie` varchar(25) NOT NULL DEFAULT '0'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tprodusecomenzi`
--

CREATE TABLE `tprodusecomenzi` (
 `fIdArticol` int(11) UNSIGNED NOT NULL,
 `fIdComanda` int(11) DEFAULT NULL,
 `fIdProdus` int(11) DEFAULT NULL,
 `fCantitate` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Table structure for table `trezervari`
--

CREATE TABLE `trezervari` (
 `fIdUtilizator` int(11) UNSIGNED NOT NULL,
 `fNumUtilizator` varchar(32) NOT NULL DEFAULT '',
 `fMasina` varchar(32) NOT NULL DEFAULT '',
 `fLocalitate` varchar(150) NOT NULL DEFAULT '',
 `fData` varchar(32) NOT NULL DEFAULT '',
 `fMesaj` varchar(255) DEFAULT NULL,
 `fDataInregistrare` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tsubcategorii`
--

CREATE TABLE `tsubcategorii` (
 `fIdSubcategorie` int(11) UNSIGNED NOT NULL,
 `fIdCategorie` int(11) UNSIGNED NOT NULL
 DEFAULT '0',
 `fNumSubcategorie` varchar(100) NOT NULL
 DEFAULT ''
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Table structure for table `tproducatori`
--

CREATE TABLE `tproducatori` (
 `fIdProducator` int(11) UNSIGNED NOT NULL,
 `fNumProducator` varchar(255) NOT NULL
 DEFAULT '',
 `fUrl` varchar(255) DEFAULT NULL,
 `fDescriere` blob
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

```
--
-- Table structure for table `tutilizatori`
--

CREATE TABLE `tutilizatori` (
 `fIdUtilizator` int(11) UNSIGNED NOT NULL,
 `fNumeUtilizator` varchar(32) NOT NULL DEFAULT
'',
 `fParola` varchar(32) NOT NULL DEFAULT '',
 `fEmail` varchar(150) NOT NULL DEFAULT '',
 `fAdresa` text,
 `fDataInregistrare` date DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
-- Table structure for table `tvoturi`
--

CREATE TABLE `tvoturi` (
 `fIdVot` int(11) UNSIGNED NOT NULL,
 `fIdProdus` int(11) UNSIGNED NOT NULL DEFAULT
'0',
 `fIdUtilizator` int(11) UNSIGNED NOT NULL
DEFAULT '0',
 `fValoareVot` int(11) UNSIGNED NOT NULL
DEFAULT '0'
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```