**UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA**

**Informatică aplicată şi programare**

*Program postuniversitar de formare și dezvoltare profesională continuă*

Site de vânzări on-line pentru un magazin sportiv

***Lucrare de disertaţie***

***Vădan Radu Dan***

***2019***

Cuprins

[1.Introducere 3](#_Toc13657640)

[2. Prezentarea soluțiilor folosite 3](#_Toc13657641)

[2.1 HTML 3](#_Toc13657642)

[2.2 CSS 4](#_Toc13657643)

[2.3 Javascript 5](#_Toc13657644)

[2.4 My SQL 7](#_Toc13657645)

[2.5 PHP 8](#_Toc13657646)

[3. Programele folosite 9](#_Toc13657647)

[4. Prezentarea structurii site-ului 11](#_Toc13657648)

[4.1 Stilizare CSS 11](#_Toc13657649)

[4.2 Pagini HTML 13](#_Toc13657650)

[4.3 Baza de date 24](#_Toc13657651)

[4.4 Scripturi PHP 26](#_Toc13657652)

[5. Bibliografie 37](#_Toc13657653)

# 

# 1.Introducere

Acest proiect urmărește realizarea unui site de vânzări on-line. Se dorește ca administratorul site-ului să poată introduce elemente din baza de date mySQL care nor apărea in paginile site-ului.

Tot prin intermediul acestui site clienții magazinului on-line vor putea cumpăra produse pe baza unui cont de utilizator inscris în baza de date mySQL.

Existau mai multe variante pentru rezolvarea acestei probleme, obligatoriu fiind ca limbajul ales să fie unul de back-end. Se putea opta pentru Java, dezvoltând aplicația prin Netbeans, sau în C#, dezvoltând aplicația prin Visual Studio.

Pentru rezolvarea problemei am ales să fac un site folosind limbajele HTML, CSS și PHP, realizând conexiunea cu baza de date mySQL, această alpicatie fiind realizat cu ajutorul compilorului Sublime text.

Pentru a putea fi utilizată aplicația se cere ca utilizatorul să aibe un cont cu nume și parolă, iar cu ajutorul acestuia va putea comanda produsele de pe site.

# 2. Prezentarea soluțiilor folosite

## 2.1 HTML

WWW (World Wide Web) este un ansamblu de protocoale şi convenţii de reprezentare a informaţiei care facilitează afişarea în aplicaţii de navigare a conţinutului paginilor Web. În paginile Web informaţia este codificată în format hipertext (denumit şi format HTML, prescurtare de la HyperText Markup Language). O aplicaţie de navigare (denumită în engleză browser) permite afişarea conţinutului paginilor Web care poate fi simplu text, secvenţe video, miniaplicaţii interactive sau secvenţe audio. Deoarece paginile pot conţine şi referinţe spre alte pagini, se crează un efect de reţea.

Într-un document hipertext hiperlegătura (eng. *hyperlink*) este un şir de caractere sau o imagine care în urma selectării cu mouse-ul determină afişarea în fereastra aplicaţiei de navigare a unui alt document sau a unei alte porţiuni din documentul curent.

Într-o hiperlegătură fişierul care trebuie încărcat este referit prin adresa lui. Sistemul de adresare în Internet este standardizat, fiecare fişier având o adresă unică (URL, *Uniform Resource Locator*).

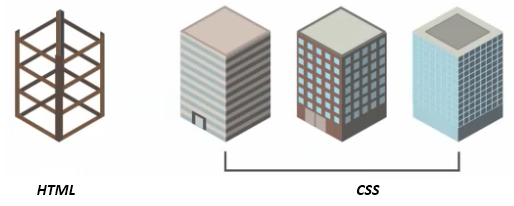
O adresă a unui fişier din WWW este formată prin alăturarea a trei componente:

* simbolul unui protocol de comunicaţie (de exemplu *http* sau *https* pentru *Hypertext Transfer Protocol* sau *ftp* pentru *File Transfer Protocol*),
* un nume de *domeniu* (sau *subdomeniu*) care deține fișierul şi
* calea spre fişier.

Numele unui domeniu (sau subdomeniu) se citește de la dreapta spre stânga și cuprinde TLD (eng. *Top Level Domain*), numele domeniului și eventuale nume de subdomenii.

## 2.2 CSS

CSS (eng. Cascading Style Sheets, trad. foi de stiluri în cascadă) este limbajul actual de descriere a aspectului şi a dispunerii în fereastra browser-ului a informaţiilor din documentele hipertext.  Relaţia dintre HTML şi CSS este prezentată în imaginea următoare:



Dacă ansamblul de elemente conţinut într-un fişier hipertext dă structura şi conţinutul unei pagini Web, dispunerea şi modul de afişare a elementelor în fereastra browser-ului este impus prin foi de stiluri (fişiere CSS).

Foile de stiluri permit realizarea şi altei funcţii esenţiale, respectiv adaptarea modului de afişare a informaţiilor la diferitele medii de afişare. Astfel, fără a interveni în fişierul hipertext, doar prin folosirea altei foi de stiluri se poate realiza afişarea corectă a unei pagini web pe un ecran normal, pe imprimantă sau pe ecranul unui telefon.

Descrierea în limbajul CSS a modul de afişare a elementelor dintr-un document HTML se realizează prin stiluri. Un document HTML poate fi formatat folosindu-se una sau mai multe foi de stiluri, ordinea acestora determinând rezultatul final. De obicei prima foaie aplicată conţine stiluri generale, valabile în toate site-urile unei firme de producţie de site-uri şi au rolul de a uniformiza modul în care diferitele browsere tratează site-urile. De multe ori numele dat acestui fişier este chiar reset.css. După acestă foaie urmează alte foi cu rol particular, legate de conţinutul fiecărui site în parte.

Un stil conţinut într-o foaie de stiluri constă la rândul lui din două părţi: un selector sau un set de selectori (elemente HTML sau selectori special definiţi) şi un bloc de descriere încadrat între acolade ({})

Ansamblul de stiluri care formează o foaie de stiluri poate fi înregistrat într-un fişier separat având extensia .css sau inserat în pagină, la început, în head. Odată definită, o foaie de stiluri poate fi ataşată tuturor paginilor unui site, dându-le acestora un aspect unitar.

Integrarea într-un fişier în format hipertext a unui fişier conţinând stiluri se realizează folosind elementul link.

n CSS pentru exprimarea dimensiunilor se folosesc diverse unităţi de măsură: px, pt, pc, %, em, ex, in, mm, cm. Importanţă practică au însă doar următoarele:

1. ***px*** – pixeli (puncte ecran). Exemplu: 14px. Pentru majoritatea aplicaţiilor de navigare, mărimea normală, implicită a fontului paragrafelor (*p*) este de 16px.
2. ***em*** – este o unitate adoptată din domeniul tipografiei şi reprezintă mărimea curentă a fontului. De exemplu 2em înseamnă de două ori mărimea curentă. Raportarea se face permanent la mărimea curentă a fontului din blocul în care dimensiunea 2em este folosită. Dacă de exemplu pentru elementul *html* mărimea este lăsată cea implicită, (16px), pentru *body* (inclus în *html*) este impusă de 0.75em (ar fi 12px) şi pentru paragrafe (*p*, evident incluse în *body*) este impusă mărimea 0.875em, mărimea reală în pixeli a fontului paragrafelor va fi 0.75\*0.875\*16px=10.5px.  
   Observație: Această unitate este scalabilă. Ea permite modificarea proporţională a mărimii tuturor fonturilor când cititorul paginii web modifică mărimea fontului de bază folosind facilităţile browser-ului (*Ctrl +* pentru zoom+ respectiv *Ctrl –* pentru zoom -);
3. ***rem*** (*root em*) – permite raportarea mărimilor fonturilor folosite la o singură mărime de font, respectiv cea impusă pentru elementul *html*. Unitatea este introdusă în CSS3.
4. ***%*** - procent din mărimea normală. Se poate folosi în loc de *em* (100% = 1em) dar se utilizează mai ales la definirea dimensiunilor blocurilor în care este divizată pagina.

## 2.3 Javascript

Paginile Web pot conţine pe lângă ansamblul de elemente care formează informaţia afişată şi secvenţe de cod care contribuie la realizarea unor efecte deosebite. Sunt deja consacrate galeriile de imagini, meniurile derulante sau efectele declanşate cu ajutorul mouse-ului care amplifică senzaţia de interactivitate. Pentru toate acestea s-a creat JavaScript, un limbaj derivat din C. Împreună cu HTML5 şi CSS3, JavaScript constituie o componentă esenţială a pregătirii dezvoltatorilor pentru componenta de front-end.

Legat de utilizarea acestui limbaj în crearea paginilor Web, în ultimii ani soluţia programării în JavaScript a fost completată cu cea a utilizării unor biblioteci de funcţii JavaScript predefinite. Succesul acestora este o consecinţă a evoluţiei din ultimii ani a limbajului JavaScript spre un limbaj de uz general, nelegat neaparat de crearea de efecte în paginile Web. Din acest motiv, pentru realizarea efectelor dorite, creatorii de pagini Web se orientează cu precădere spre utilizarea funcţiilor dedicate din biblioteci specializate, un exemplu reprezentativ fiind jQuery.

Cunoaşterea limbajului JavaScript pentru realizatorii de pagini web, cel puţin la un nivel de bază, este însă obligatorie deoarece JavaScript şi jQuery sunt inseparabile. Funcţiile din biblioteca jQuery sunt funcţii JavaScript şi sunt apelate din secvenţe de program scrise în JavaScript. De altfel, ori de câte ori soluţia oferită de JavaScript este uşor de folosit, aceasta va fi preferată soluţiei oferită de jQuery. Este bine să nu se evite utilizarea limbajului JavaScript deoarece viteza de rulare a aplicaţiilor ar avea mult de suferit.

Invăţarea limbajului JavaScript nu este posibilă fără cunoştinţe de html şi CSS deoarece, într-o pagină Web, funcţiile JavaScript sunt de regulă destinate manipulării elementelor html şi a regulilor de prezentare codificate în CSS.

Codul JavaScript se rulează în cadrul unui interpretor de comenzi integrat în aplicaţia de navigare în Internet. Se spune că JavaScript este un limbaj *interpretat* (spre deosebire de C sau C++ care sunt limbaje compilate). Limbajele interpretate se mai numesc şi limbaje de scriptare (*scripting languages*).

O aplicaţie Web este însă o aplicaţie în arhitectură client-server, componenta client executată în aplicaţia de navigare. Din această perspectivă, JavaScript este limbajul destinat scrierii componentei *client*. Pentru componenta *server*, programarea se face frecvent în PHP, un alt limbaj interpretat, care va fi studiat în partea finală a cursului.

Inserarea codului JavaScript într-un document hipertext se realizează în acelaşi mod în care s-au inserat foile de stiluri, respectiv într-un element *script* sau prin ataşarea unuia sau mai multor fişiere *.js* care conţin codul aplicației. Exemplu de utilizare a unui element *script*:

<script>

//Se inserează liniile de cod.

</script>

Deşi un script JavaScript poate fi inserat oriunde într-un fişier html, există totuşi două locuri în care acesta este inserat cu precădere, respectiv la început, în head, sau la sfârşit, înainte de </body>. Plasarea unui element script la sfârşitul paginii are avantajul că nu afectează sensibil viteza de afişare a acesteia, deoarece browser-ul tratează elementele conţinute în fişierul hipertext în ordine. Dacă elementul script este plasat la început, pagina se va afişa doar după încărcarea fişierului (fişierelor) JavaScript.

  Scrierea codului în JavaScript nu necesită o aplicaţie specializată. Orice editor de texte poate fi folosit, de exemplu Sublime Text sau Notepad++, menționate deja. Dar mediile de programare avansate, de exemplu Visual Studio, Eclipse sau NetBeans, suportă și ele scrierea în JavaScript şi asigură chiar funcţii foarte utile, ca formatarea specifică a codului sau asistenţa la scrierea acestuia (intellisense). În firmele de software sunt mult folosite și alte aplicaţii ca WebStorm (www.jetbrains.com/webstorm), sau Aptana Studio (www.aptana.com).

Sintaxa limbajului JavaScript a fost preluată din limbajul C. Regulile de scriere a codului sunt deci similare:

* se face distincţie între literele mari şi mici (limbaj *case sensitive*);
* fiecare instrucţiune se va încheia cu ';'. Ca şi în C, se pot scrie mai multe instrucţiuni pe aceeaşi linie, dar este nerecomandabil.
* spaţiile şi taburile nu contează, deci se poate evidenţia structura codului prin indentarea diferitelor blocuri de cod;
* blocurile formate din mai multe linii sunt plasate între acolade '{' '}' ;
* comentariile în linie încep cu // şi ţin până la sfârşitul linei, iar comentariile pe mai multe linii sunt încadrate între /\* şi \*/.

## 2.4 My SQL

MySQL este un sistem de gestiune a bazelor de date relaţionale produs de compania suedeză MySQL AB (preluată în 2008 de Sun Microsystems, cumpărată la rândul ei în 2009 de Oracle Co.) şi distribuit sub Licenţa Publică Generală, GNU. Este cel mai popular SGBD open-source şi o componentă principală a pachetului de aplicaţii WAMP.

Deşi este folosit foarte des împreună cu limbajul PHP, cu MySQL se pot construi aplicaţii în orice limbaj major: C, C++, C#, Java, Perl, sau Python.

Licenţa GNU GPL nu permite încorporarea MySQL în aplicaţii comerciale. Cei care doresc să facă acest lucru pot achiziţiona, contra cost, o licenţă comercială de la compania Oracle (aprox. 900$/server).

Pentru a administra bazele de date MySQL se poate folosi modul linie de comandă sau aplicaţia gratuită, scrisă în PHP, phpMyAdmin.

MySQL poate fi rulat pe multe dintre platformele software majore: Windows 7/8, Linux, Mac OS, etc.

În cele ce urmează MySQL va fi componenta aplicaţiilor web destinată păstrării datelor aplicaţiei. Deşi limbajul PHP în care vor fi scrise secvenţele de cod care prelucrează informaţiile aplicaţiei web are funcţii destinate scrierii de fişiere pe discul serverului, de regulă se apelează la varianta mult mai sigură a folosirii serverului de baze de date MySQL.

O aplicaţie web prelucrează date de diferite naturi: şiruri de caractere, valori numerice, date calendaristice etc. Exemple : denumiri de produse, nume de utilizatori sau de clienţi, parole, adrese de e-mail, preţuri, cantităţi, date de livrare sau de încheiere a unei comenzi, căi pentru accesarea fişierelor cu imagini, etc. În baze de date pot fi însă păstrate şi informaţii şi de alte naturi : imagini, fişiere în format .pdf, filme, fişiere audio .mp3 etc.

Informaţiile păstrate într-o bază de date relaţională sunt dispuse într-un ansamblu de fişiere. Din considerente legate de creşterea vitezei de accesare a informaţiilor, fişierele sunt formate din articole având aceeaşi structură.

Structura articolelor unui fişier de date este definită la crearea sa, prin precizarea câmpurilor pe care le va conţine fiecare articol.

O aplicaţie de tip server este un program care aşteaptă comenzi din partea unor aplicaţii client. În cazul serverelor de baze de date relaţionale (Oracle, MySQL, IBM DB2, Microsoft Access, OpenOffice Base etc.) se foloseşte limbajul SQL (Structured Query Language). Deoarece prezentarea acestui limbaj depăşeşte cadrul acestui curs, în cele ce urmează comenzile SQL vor fi construite folosind interfaţa grafică oferită de PHPMyAdmin, o aplicaţie web integrată în WAMP.

## 2.5 PHP

În cadrul ansamblului de instrumente software folosit pentru crearea unei aplicaţii Web, PHP este limbajul de programare destinat scrierii componentelor care se rulează pe server. Există desigur şi alte limbaje care pot fi folosite, dar PHP este în acest moment soluţia cea mai răspândită pentru scrierea scripturilor care prelucrează datele din formulare, care returnează date memorate în baze de date sau care generează pagini dinamice bazate pe date păstrate în baze de date.

Limbajul PHP şi interpretorul de comenzi aferent sunt gratuite. Interpretorul de comenzi este conţinut în pachetul de aplicaţii WAMP şi este instalat automat împreună cu celelalte componente ale pachetului.

Un fişier PHP poate conţine numai cod PHP sau secvenţele de cod inserate printre marcaje HTML. Lansarea în execuţie a interpretorului PHP se realizează de către serverul de Web (Apache, IIS etc.) dacă fişierul cerut de browser are extensia .php. Interpretorul va trimite browserului atât codul HTML găsit în fişier cât şi rezultatele executării secvenţelor programate în PHP.

Aplicaţiile din categoria serverelor Web nu permit transferul unor informaţii de la o pagină vizitată la alta. Un astfel de server se limitează la a trimite spre client (browser) fişierul cerut sau lansează în execuţie o aplicaţie, ca în cazul scripturilor asociate formularelor de exemplu.

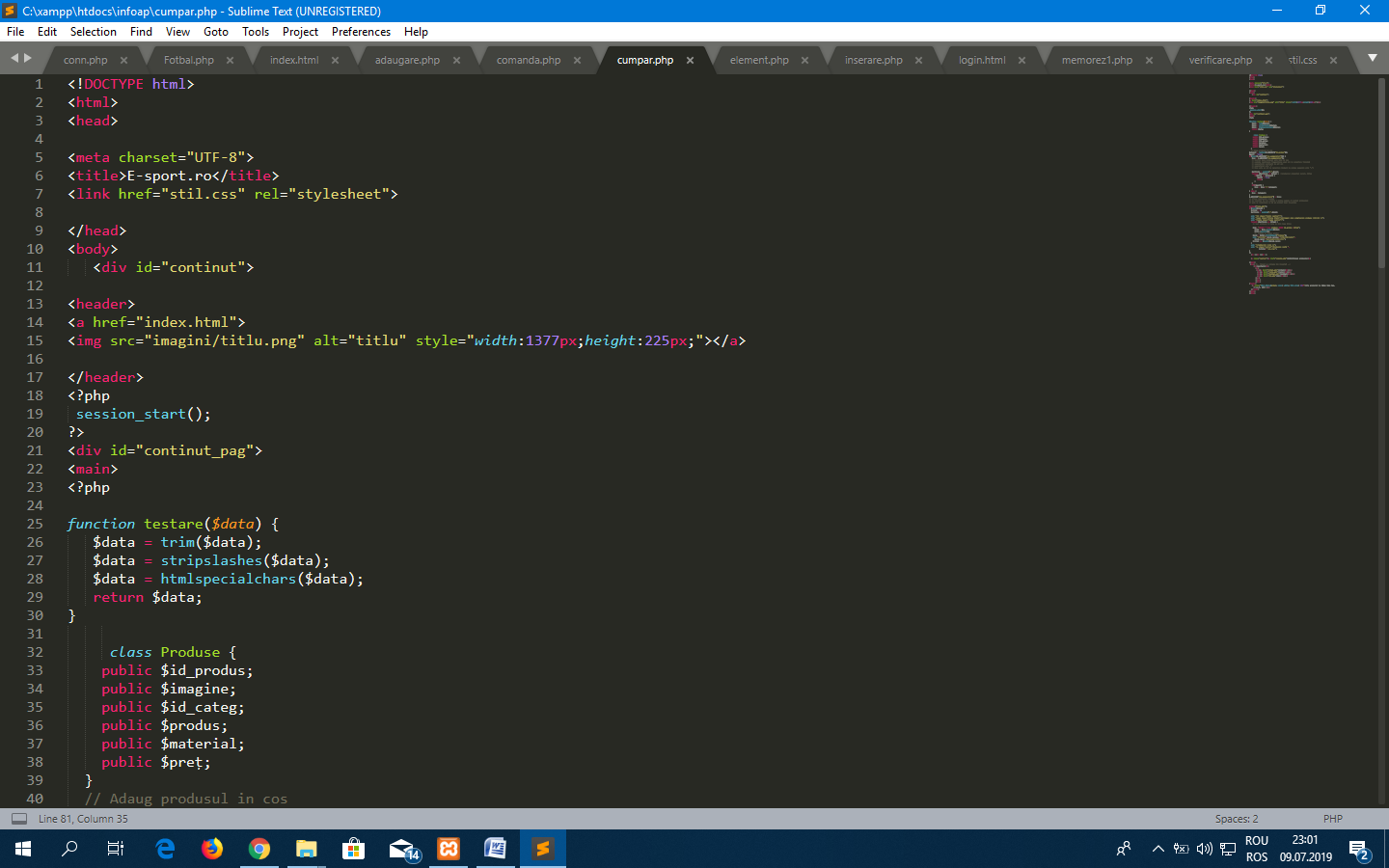
Există cazuri în care este necesar ca în timpul navigării prin site anumite informaţii să fie "transportate" de la o pagină la alta. Un exemplu clasic este procesul de cumpărare dintr-un magazin virtual. După cumpărarea unui articol dintr-o pagină se trece la o altă pagină de unde se doreşte un alt articol şi aşa mai departe. Articolele deja selectate în vederea cumpărării trebuie memorate până la definitivarea comenzii.

Limbajul PHP permite diverse soluţii pentru această problemă, în cele ce urmează fiind prezentat lucrul cu sesiuni (eng. sessions). O sesiune iniţiată într-o pagină va fi menţinută deschisă în timpul navigării şi va fi închisă doar atunci când nu mai este necesară. În cazul unui magazin virtual închiderea sesiunii se va realiza după definitivarea comenzii de cumpărare sau dacă se abandonează acţiunea şi se părăseşte site-ul.

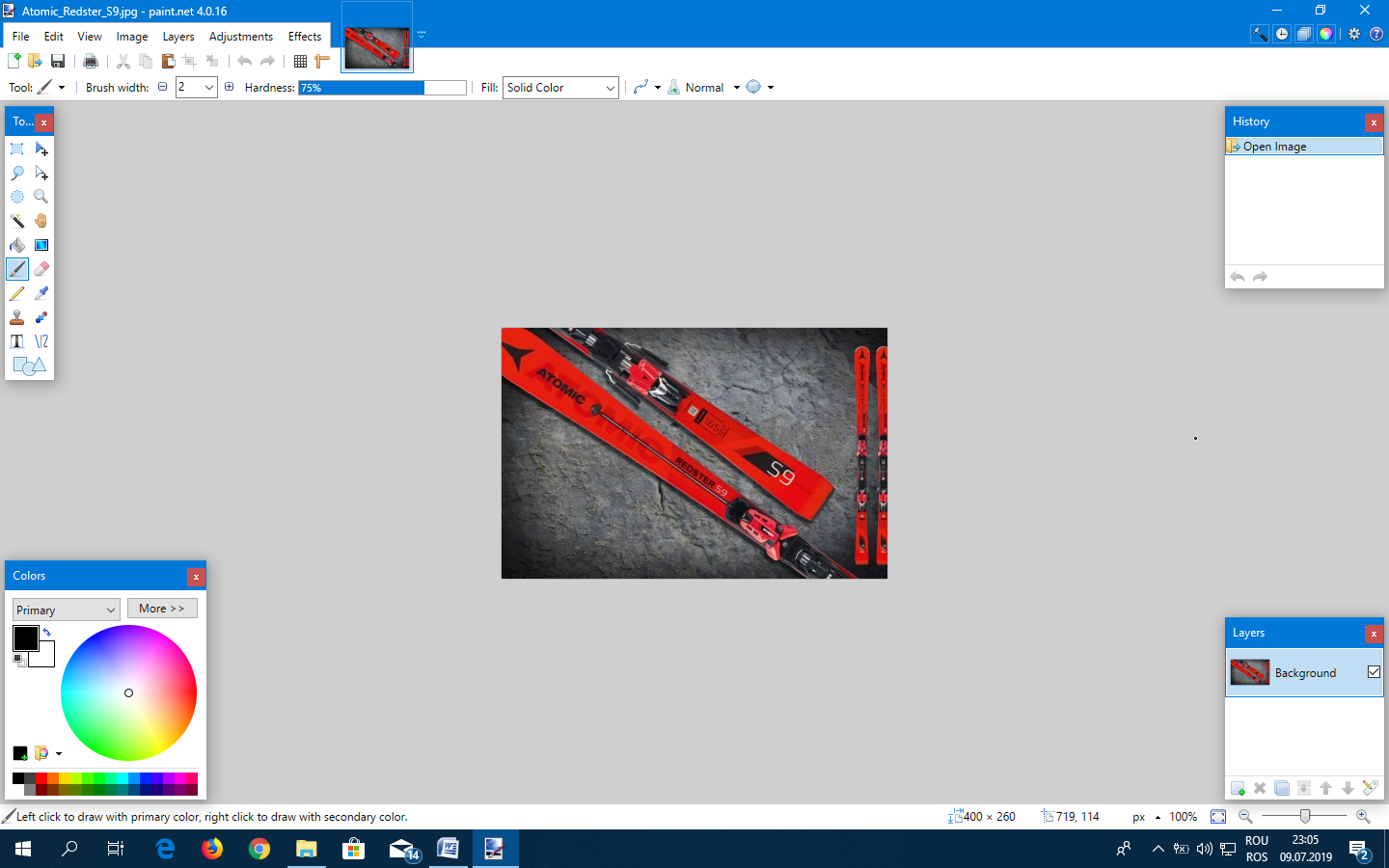
Pe toată durata existenţei unei sesiuni, în cadrul acesteia poate fi definit un set de variabile care vor păstra informaţiile dorite. De exemplu, în cazul cumpărării dintr-un magazin virtual, în cadrul sesiunii deschise se poate utiliza o variabilă de tip şir (array) sau un şir de caractere care va conţine codurile articolelor selectate la un moment dat.

# 3. Programele folosite

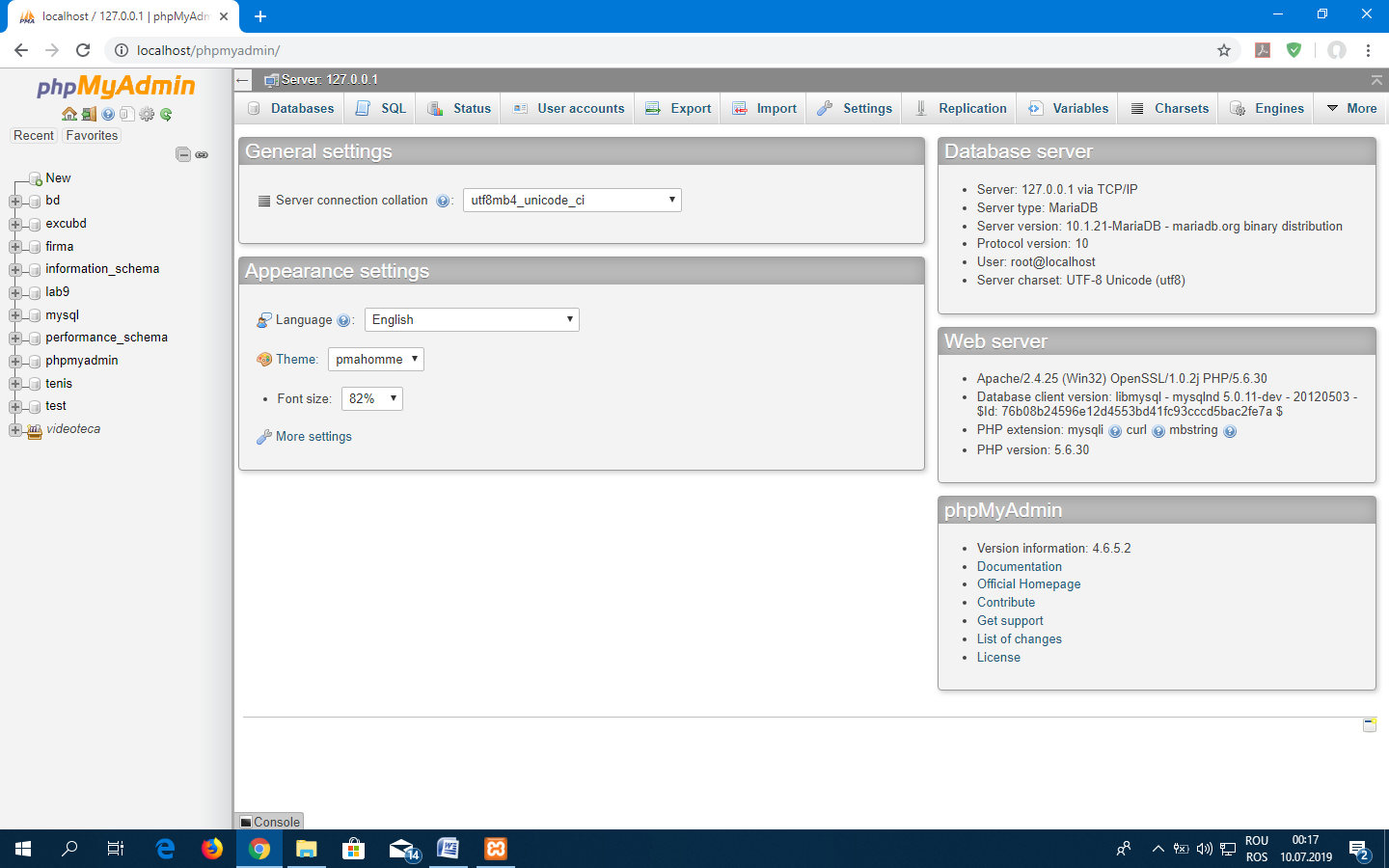
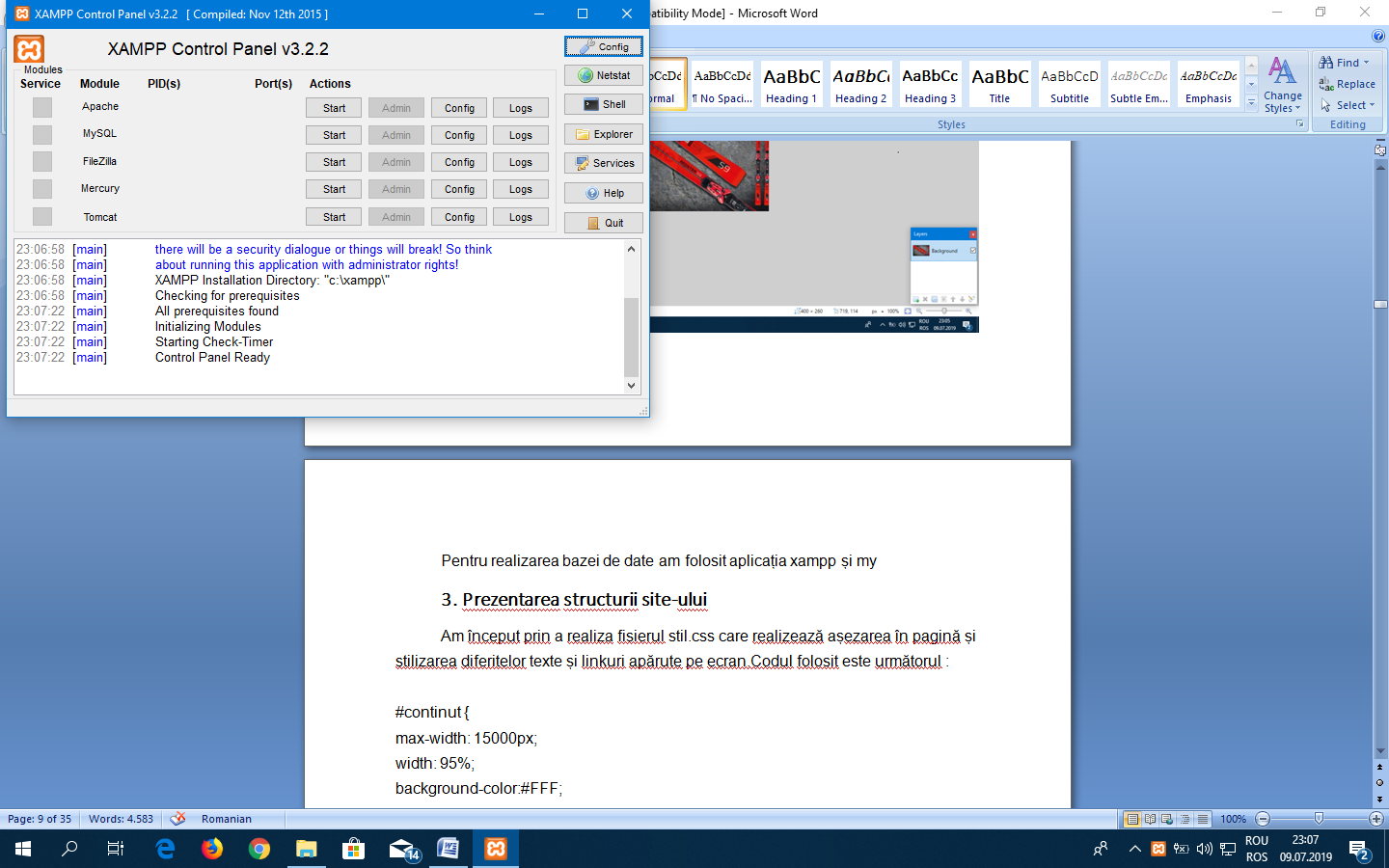
Pentru a compila codurile scrise am folosit sublime text.



Pentru prelucrarea fotografiilor astfel încât formatul lor să fie asemănător și să se potrivească în pagină am folosit aplicația paint.net.



Pentru realizarea bazei de date am folosit aplicația xampp și phpmyadmin.



# 4. Prezentarea structurii site-ului

## 4.1 Stilizare CSS

Am început prin a realiza fisierul stil.css care realizează așezarea în pagină și

stilizarea diferitelor texte și linkuri apărute pe ecran.Codul folosit este următorul :

#continut {

max-width: 15000px;

width: 95%;

background-color:#FFF;

margin: 0 auto;

position: relative;

}

#continut\_pag {

width: 100%;

background-color:#FFF;

margin: 0 auto;

overflow: hidden;

}

header {

width: 100%;

position: relative;

overflow: hidden;

}

main {

float: left;

width: 80%;

background-color:#0000c8;

}

nav {

float: right;

width: 17%;

background: #0000c8;

}

article{

float: left;

width: 80%;

background-color:#0000c8;

margin-left: 15px;

}

footer {

max-width: 1500px;

height: 1.8rem;

background-color: #0000c8;

letter-spacing: normal;

text-align: center;

font-size: 0.9em;

font-weight: bold;

margin: 0 auto;

padding: 0;

}

h1 {

background-color: #0000c8;

color: #ff7848;

font-family: ravie;

}

h2 {

background-color: #0000c8;

color: #ff7848;

font-family: arial;

font-size: 2em;

font-weight: bold;

}

ul{

background-color: #ff7848;

color: #0000c8;

font-family: ariaL;

font-size: 30px;

a{

font-family: arial;

color: white;

}

## 4.2 Pagini HTML

Am realizat apoi prima pagină a site-ului index.html, care conține toate link-urile de pe site precum și o ferestră slide-show cu promoțiile lunii.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>E-sport.ro</title>

<link href="stil.css" rel="stylesheet">

</head>

<body>

<div id="continut">

<header>

<a href="index.html">

<img src="imagini/titlu.png" alt="titlu" style="width:1377px;height:225px;"></a>

</header>

<div id="continut\_pag">

<main>

<h1> Ofertele lunii</h1>

<div class="slideshow-container">

<div class="mySlides fade">

<div class="text">Minge Nike Strike 400 lei</div>

<img src="imagini/balon-nike-strike.jpg" style="width:100%">

</div>

<div class="mySlides fade">

<div class="text">Racheta de tenis Wilson Blade Team 500 lei</div>

<img src="imagini/wilson-blade-team.jpg" style="width:100%">

</div>

<div class="mySlides fade">

<div class="text">Minge de handbal Hummel Concept 330 lei</div>

<img src="imagini/hummel concept.jpg" style="width:100%">

</div>

<div class="mySlides fade">

<div class="text">Skiuri atomic redster G9 2700 lei</div>

<img src="imagini/atomic redster.jpg" style="width:100%">

</div>

</div>

<br>

<div style="text-align:center">

<span class="dot"></span>

<span class="dot"></span>

<span class="dot"></span>

<span class="dot"></span>

</div>

<script>

var slideIndex = 0;

showSlides();

function showSlides() {

var i;

var slides = document.getElementsByClassName("mySlides");

var dots = document.getElementsByClassName("dot");

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none";

}

slideIndex++;

if (slideIndex> slides.length) {slideIndex = 1}

for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "");

}

slides[slideIndex-1].style.display = "block";

dots[slideIndex-1].className += " active";

setTimeout(showSlides, 5000);

}

</script>

</main>

<nav> <!-- Plasat pe coloana din dreapta! -->

<h1>Destinatii</h1>

<ul>

<li><a href="Fotbal.html">Fotbal</li></a>

<li><a href="Tenis.html">Tenis</li></a>

<li><a href="Handbal.html">Handbal</li></a>

<li><a href="Ski.html">Ski</li></a>

</ul>

<h1>Administrare</h1>

<ul>

<li><a href="login.html">Baza de date</li></a>

<ul>

</nav>

</div>

<footer>

<p style="font-family:arial; color: white; font-size: 150%">Site proiectat de Vădan Radu Dan.

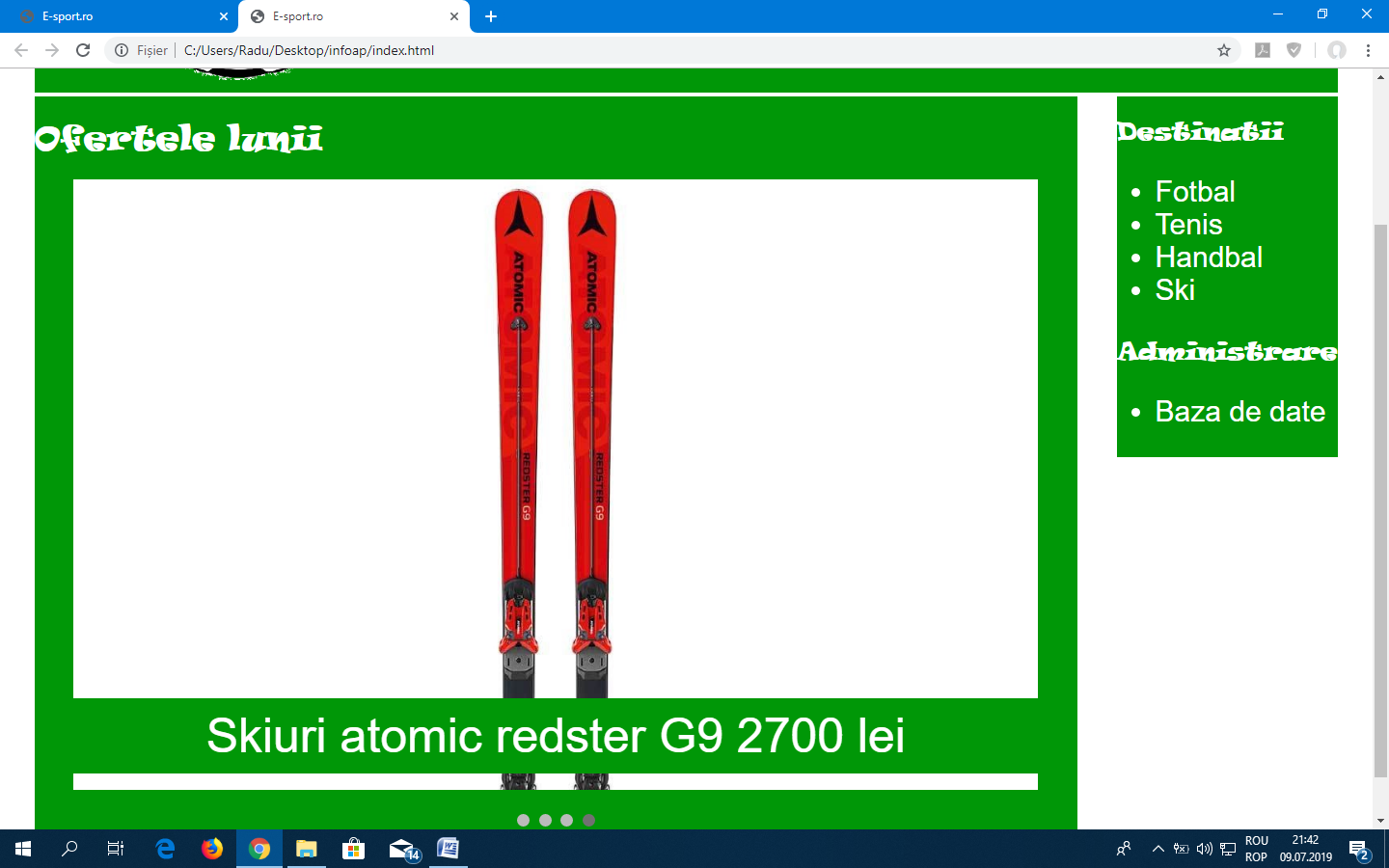
© I.A.P. 2019 </p>

</footer>

</body>

</html>

După scrierea acestor coduri și deschiderea paginii în browser, aceasta va arăta așa:



Următorul pas a fost crearea paginilor cu produsele disponibile pentru fiecare sport în parte. Prima astfel de pagină a fost fotbal.html, care ne arată produsele din categoria fotbal.

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>E-sport.ro-Fotbal</title>

<link href="stil.css" rel="stylesheet">

</head>

<body>

<div id="continut">

<header>

<a href="index.html">

<img src="imagini/titlu.png" alt="titlu" style="width:1377px;height:225px;"></a>

</header>

<div id="continut\_pag">

<main>

<h1> Fotbal</h1>

<article>

<img src="imagini/fotbal/balon-nike-strike.jpg" style="float:left" border="5">

<div class="descriere">

<a href="nikestrike.html">

<h2> Minge Nike Strike </h2></a>

</div>

</article>

<article>

<img src="imagini/fotbal/adidas-football-fifa-confederations-cup.jpg" style="float:left" border="5">

<div class="descriere">

<a href="adidas.html">

<h2> Minge Adidas Football Fifa Confederations Cup </h2></a>

</div>

</article>

<article>

<img src="imagini/fotbal/atletico.jpg" style="float:left" border="5">

<div class="descriere">

<a href="atletico.html">

<h2> Tricou oficial de joc Atletico Madrid </h2></a>

</div>

</article>

<article>

<img src="imagini/fotbal/nike mercurial.jpg" style="float:left" border="5">

<div class="descriere">

<a href="nikemercurial.html">

<h2> Ghete de fotbal Nike Mercurial </h2></a>

</div>

</article>

<article>

<img src="imagini/fotbal/forza-mundo.jpg" style="float:left" border="5">

<div class="descriere">

<a href="forza.html">

<h2> Mănuși de portar Forza Mundo </h2></a>

</div>

</article>

</main>

<nav> <!-- Plasat pe coloana din dreapta! -->

<h1>Sporturi</h1>

<ul>

<li><a href="Fotbal.html">Fotbal</li></a>

<li><a href="Tenis.html">Tenis</li></a>

<li><a href="Handbal.html">Handbal</li></a>

<li><a href="Ski.html">Ski</li></a>

</ul>

</nav>

</div>

<footer>

<p style="font-family:arial; color: white; font-size: 150%">Site proiectat de Vădan Radu Dan.

© I.A.P. 2019 </p>

</footer>

</body>

</html>

După scrierea secvențelor de cod și rularea paginii web aceasta va arăta on felul următor:



În fișierul CSS creat inițial am mai creat stiluri pentru afisarea slide-show-ului de pe prima pagină, care ne arată pe scurt promoțiile lunii.

}

.descriere {

float: right;

width: 50%;

padding: 0,5rem;

text-align: justify;

color: white;

font-size: 16px;

}

.descriere1 {

float: right;

width: 98%;

padding: 0,5rem;

text-align: justify;

color: white;

font-size: 16px;

}

\* {box-sizing:border-box}

body {font-family: Verdana,sans-serif;margin:0}

.mySlides {display:none}

.slideshow-container {

max-width: 1000px;

position: relative;

margin: auto;

}

/\* Caption text \*/

.text {

background-color: rgb(0, 151, 7);

color: #ffffff;

font-family: arial;

font-size: 50px;

padding-top: 10px;

padding-right: 10px;

padding-bottom: 10px;

padding-left: 10px;

position: absolute;

bottom: 20px;

width: 100%;

text-align: center;

}

/\* Number text (1/3 etc) \*/

.numbertext {

color: #ffffff;

font-size: 12px;

padding: 8px 12px;

position: absolute;

top: 0;

}

/\* The dots/bullets/indicators \*/

.dot {

height: 13px;

width: 13px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active {

background-color: #717171;

}

/\* Fading animation \*/

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 2.5s;

animation-name: fade;

animation-duration: 2.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

/\* On smaller screens, decrease text size \*/

@media only screen and (max-width: 300px) {

.text {font-size: 11px}

}

.login table, .login td, .login th {

border: 2px solid orange;

padding:15px;

}

.centrat {

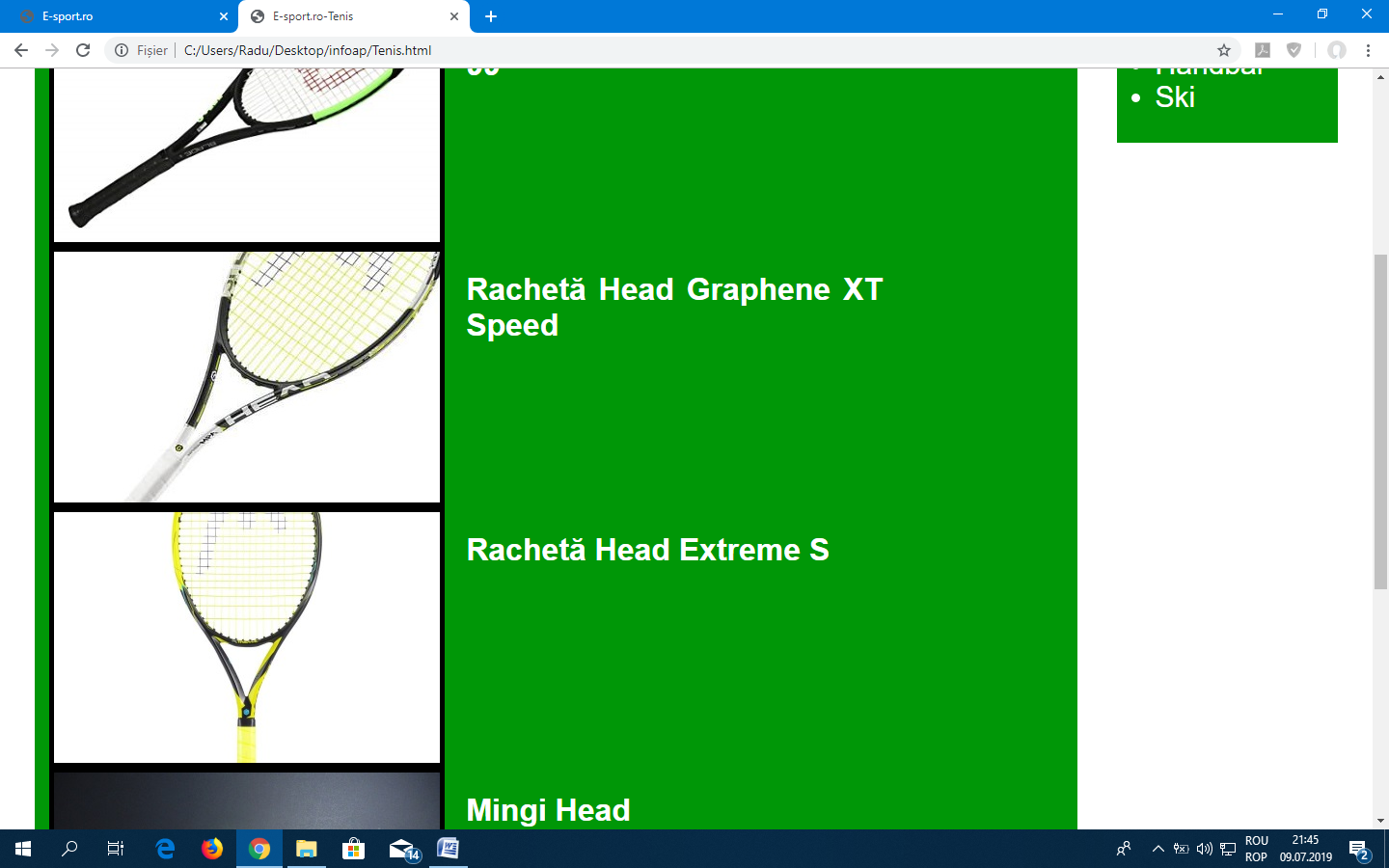
margin: 0 auto;

text-align: center;

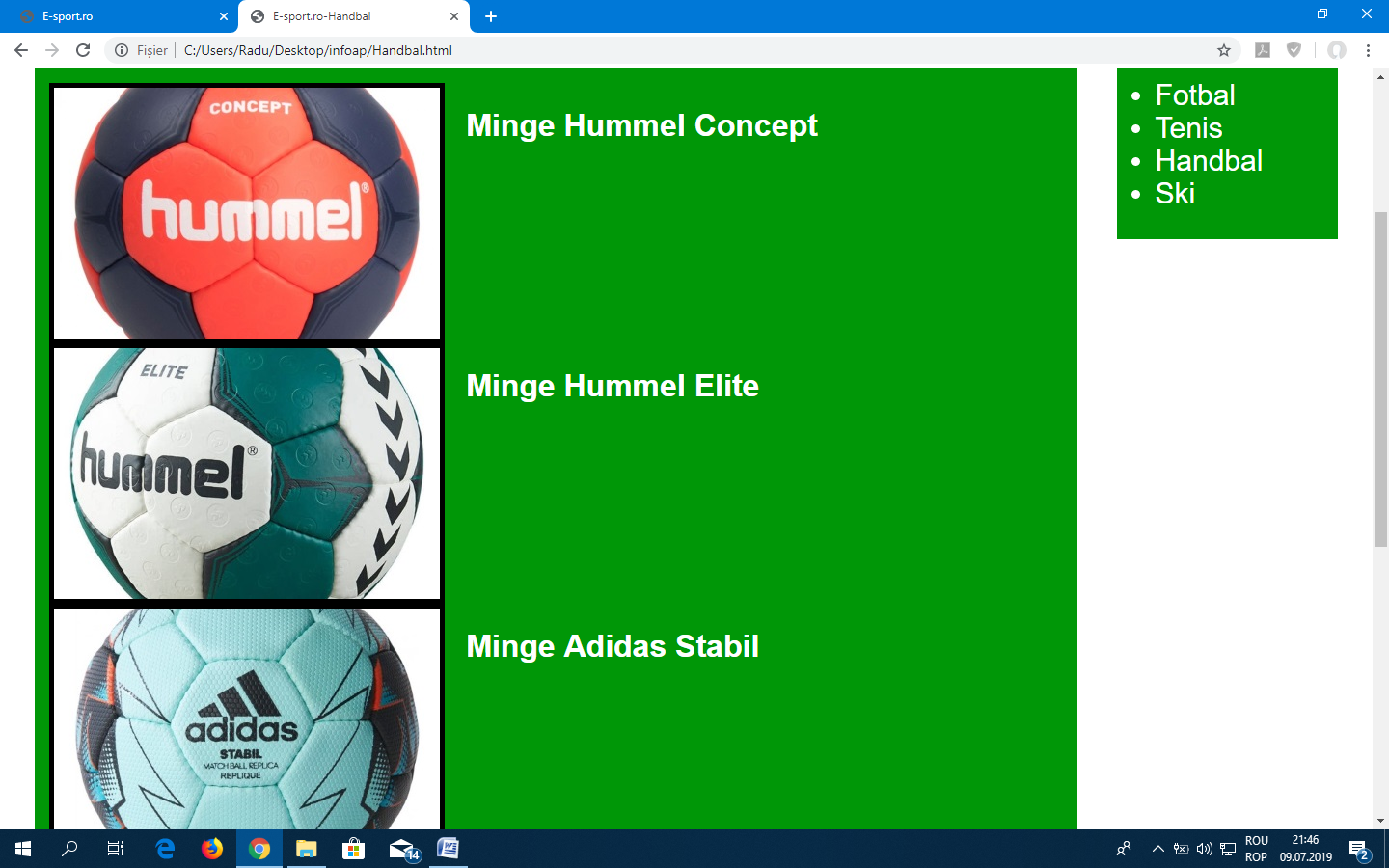
}

Pentru rerallizarea celorlalte pagini cu produsele pentru alte sporturi s-a procedat la fel ca și la pagina fotbal.html, formatul paginilor fiind același, schimbate fiind doar imaginile cu produsele disponibile pentru fiecare sport in parte:

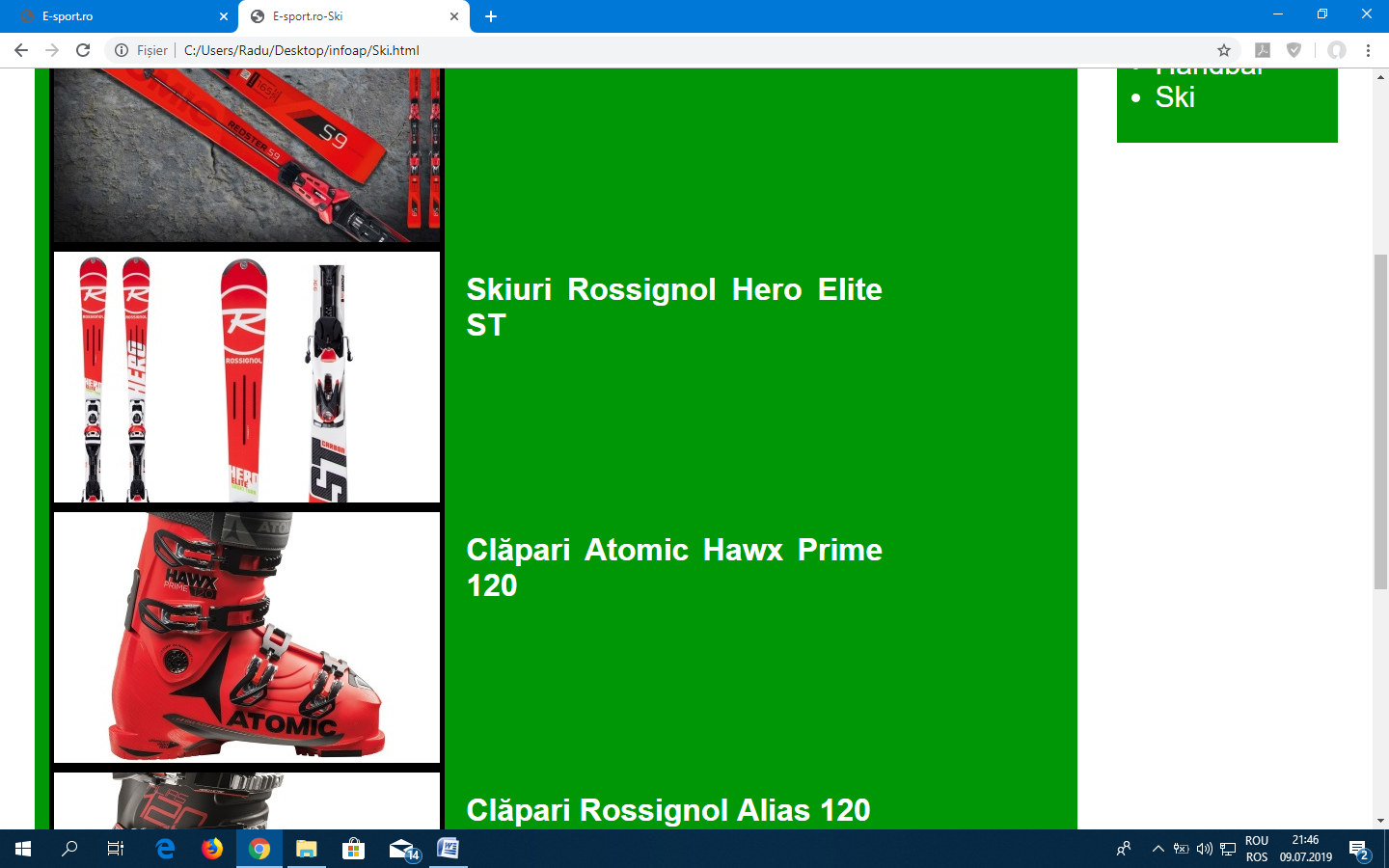
Pagina Tenis.html:



Pagina Handbal.html

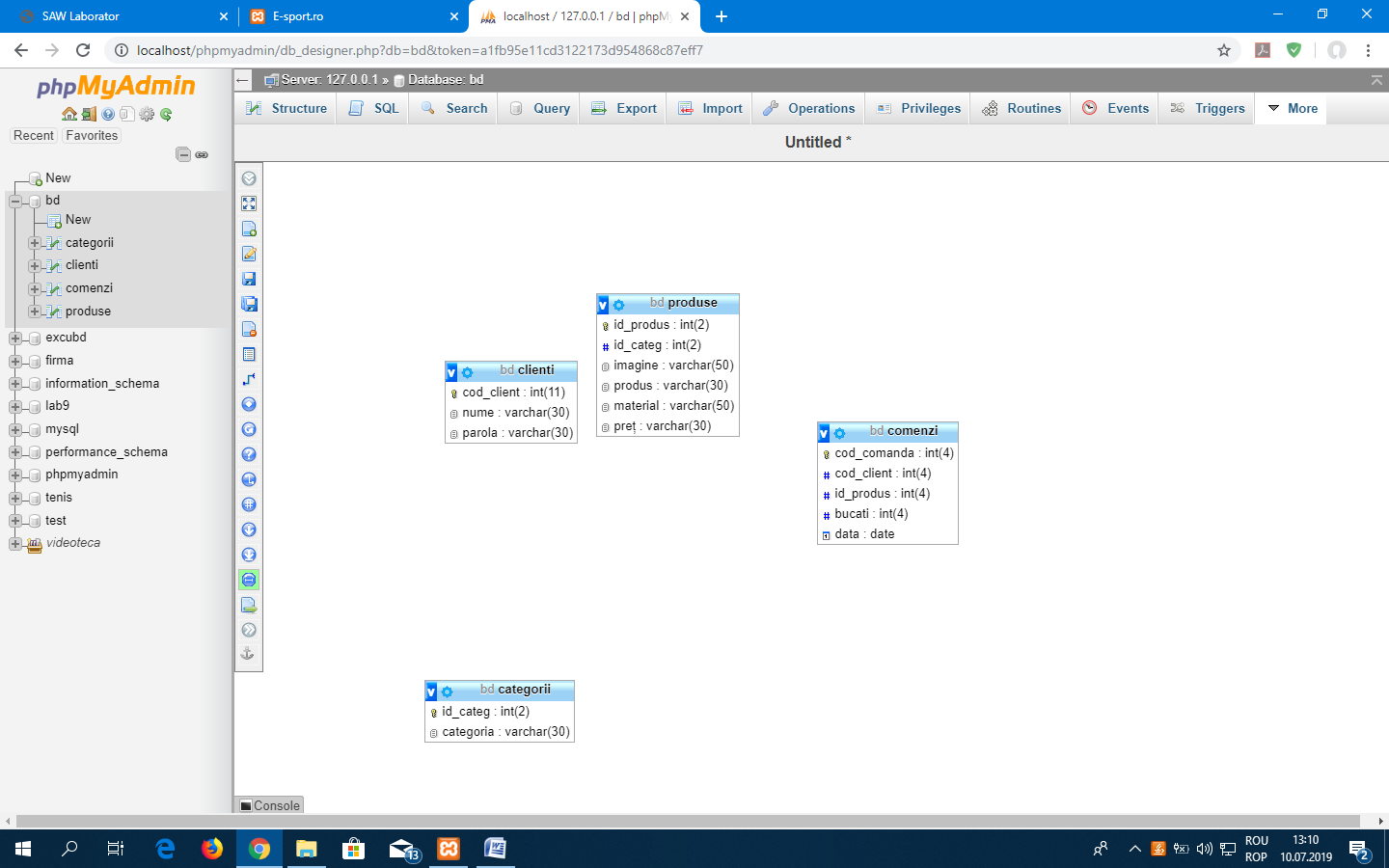


Precum și pagina ski.html:



## 4.3 Baza de date

Din fișierul xampp instalat pe partiția C a calculatorului am activat serverul localhost/phpmyadmin care rulează în browser și am inceput crearea bazei de date cu numele bd, în cae am creat tabelele admin, categorii, clienți, comenzi, produse.



Pentru a se putea face conexiunea la baza de date am realizat fisierul conn.php, a cărui rol este de a realiza această conexiune și care va fi apelat in toate fisierele php care necesită conexiune cu baza de date.Scriptul adăucat in fișier este următorul, în care specificăm numele bazei de date la care ne vom conecta:

<?php

$dsn='mysql:host=localhost;charset=utf8;dbname=bd';

$utilizator='root';

$parola='';

$cnx = null;

try {

$cnx = new PDO($dsn, $utilizator, $parola);

$cnx ->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

}

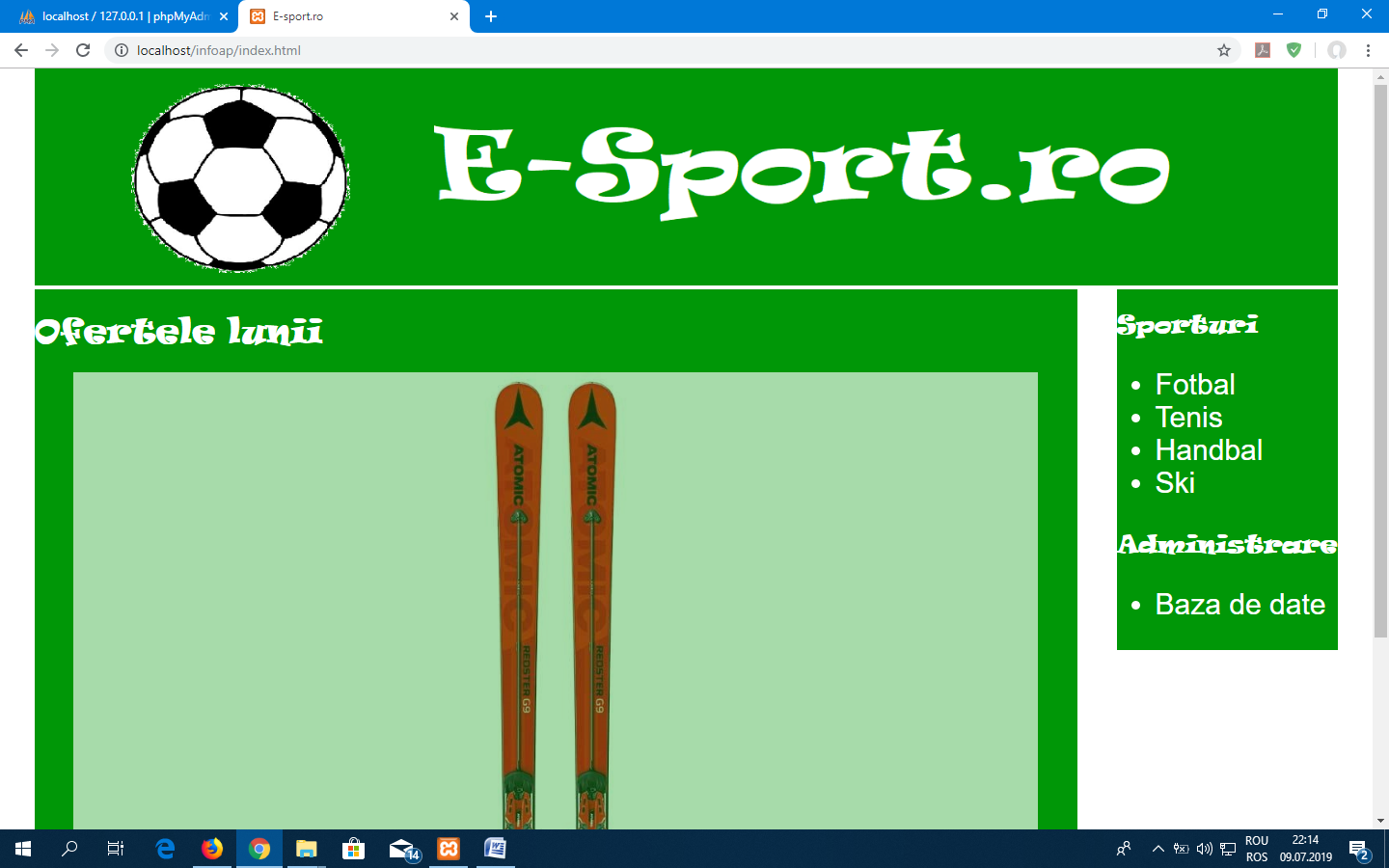
catch (PDOException $e) {

echo 'Conectare nereusita: ' . $e->getMessage();

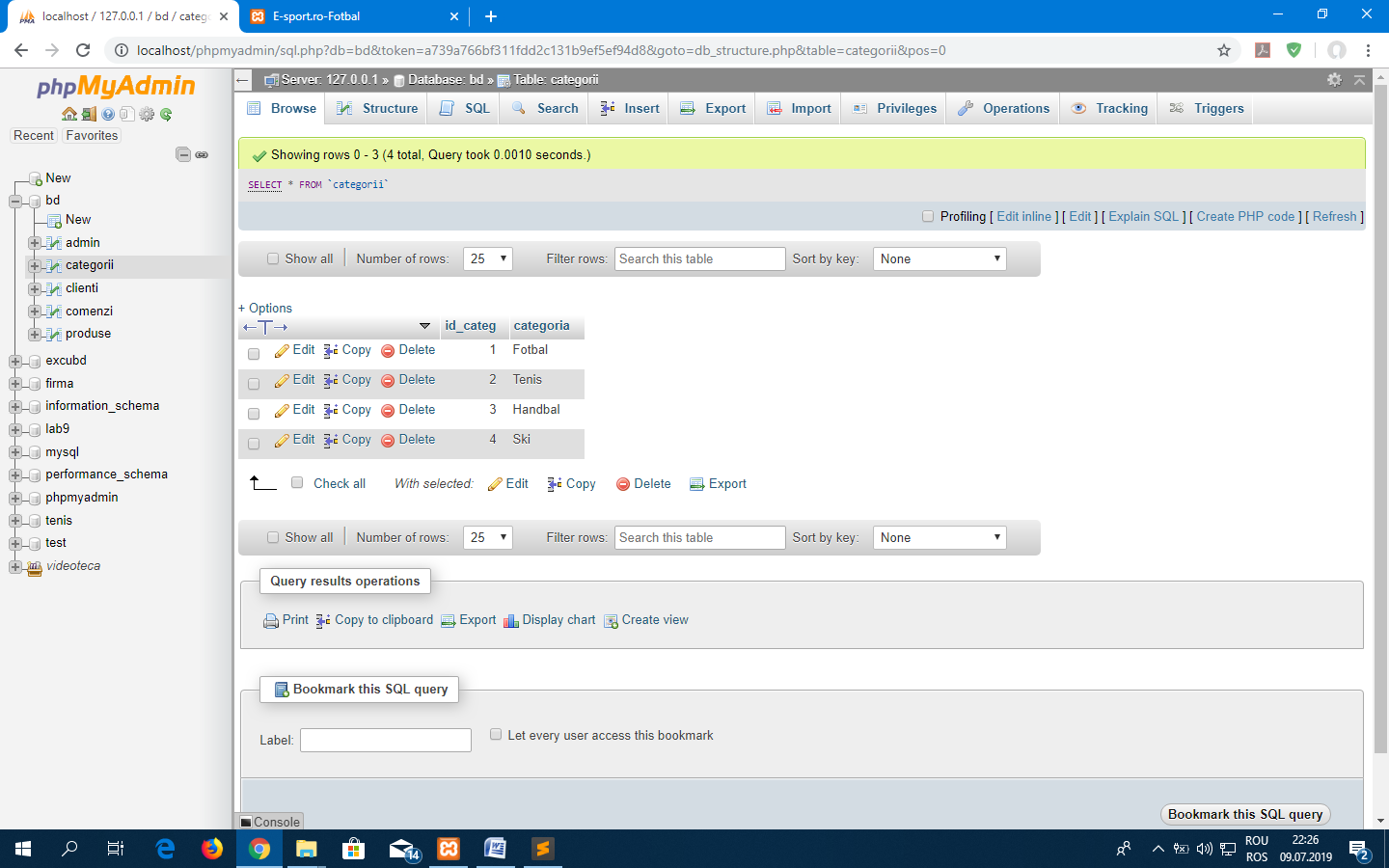
};

?>

După mutarea fișierelor la care am lucrat în fișierul aplicației xampp în fișierul htdocs, vom putea intra pe site direct din browser tastând localhost/infoap, numele fișierului aplicației.



După ce am făcut primul fișier php din proiect trecem la modificarea main-urilor din paginile html create anterior pentru a puutea face în acest fel paginile dinamice, adică ceea ce vom scrie în baza de date și fotografiile cu produsele vor putea aparea ulterior în site-ul nostru. Pentru ca aceste pagini să funcționeze se creazăo legatură între acestea și tabelul categorii din baza de date, pentru ca browserul să facă conexiunea intre produsele din baza și afișarea lor.



## 4.4 Scripturi PHP

În tabelul categorii din baza de date am scris câte un id pentru fiecare categorie în parte pentru că bazele de date fac legtura între tabele după id-uri, iar scriptul php care va inlocui main-ul fiecărei pagini de afișare produse este:

<?php

include("conn.php");

class Produse {

public $id\_produs;

public $imagine;

public $id\_categ;

public $produs;

public $material;

public $preț;

}

if(isset($cnx)) {

$cda= "SELECT \* from produse WHERE id\_categ =1";

$stmt = $cnx->prepare($cda);

$stmt->execute();

echo "<div class=\"imagini\">";

while ($prod = $stmt->fetchObject('Produse')) {

$img = $prod->imagine;

$id = $prod->id\_produs;

echo '<a href="element.php?idprod='.$id.'" class=\"balon-nike-strike\"><img src="imagini/'.$img.'" alt=""/></a>';

}

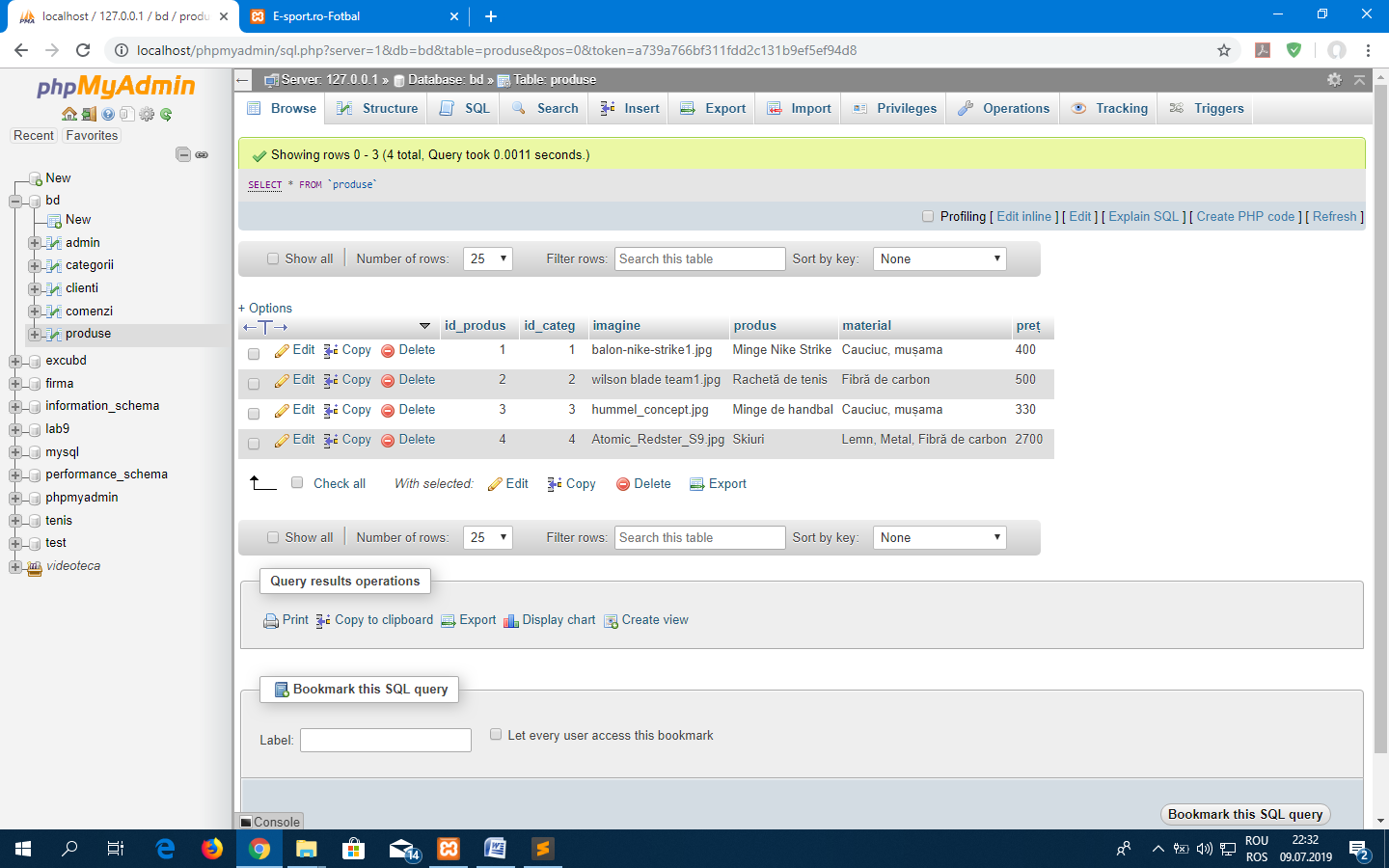
$cnx = null;

}

?>

Trebuie ca la fiecare pagină să îi scriem în SELECT-ul sql SELECT \* from produse WHERE id\_categ ="; id-ul corespunzător fiecărei categorii în parte.

Astfel cu fiecare produs introdus în baza de date, acesta va apărea in pagină.



Pagina având link-ul scris element.php, înseamnă că va trebui să creăm acum pagina element.php care ne va arăta fiecare produs în parte cu informații despre el. În acest fel, în loc să facem pagini în html pentru fiecare produs în parte vom avea o singura pagină, care în funcție de id-ul fiecărui produs va afișa informații despre el. }n același timp va avea un link „Ad[ugar în coș” , care va fi premergator achiziției produsului selectat.

Scriptul php este:

<?php

include("conn.php");

class Produse {

public $id\_produs;

public $imagine;

public $id\_categ;

public $produs;

public $material;

public $preț;

}

if(isset($cnx)) {

$idp = $\_REQUEST['idprod'];

$cda = "SELECT \* from produse WHERE id\_produs=$idp";

$stmt = $cnx->prepare($cda);

$stmt->execute();

$prod = $stmt->fetchObject('produse');

echo "<article class=\"produs\"><h1>$prod->produs</h1>";

echo "<div class=\"mostra\">";

echo '<img src="imagini/'.$prod->imagine.'" alt="" />';

echo "</div>";

echo "<div class=\"descriere\">";

echo "<h2>Material</h2><p>$prod->material</p>";

echo '<h2>Preț</h2><p class="bold">'.$prod->preț.' lei</p>';

echo "</div>";

echo '<div class="pret"><a href="cumpar.php?id\_produs='.$idp.'">Adăugare în coș</a></p></div>';

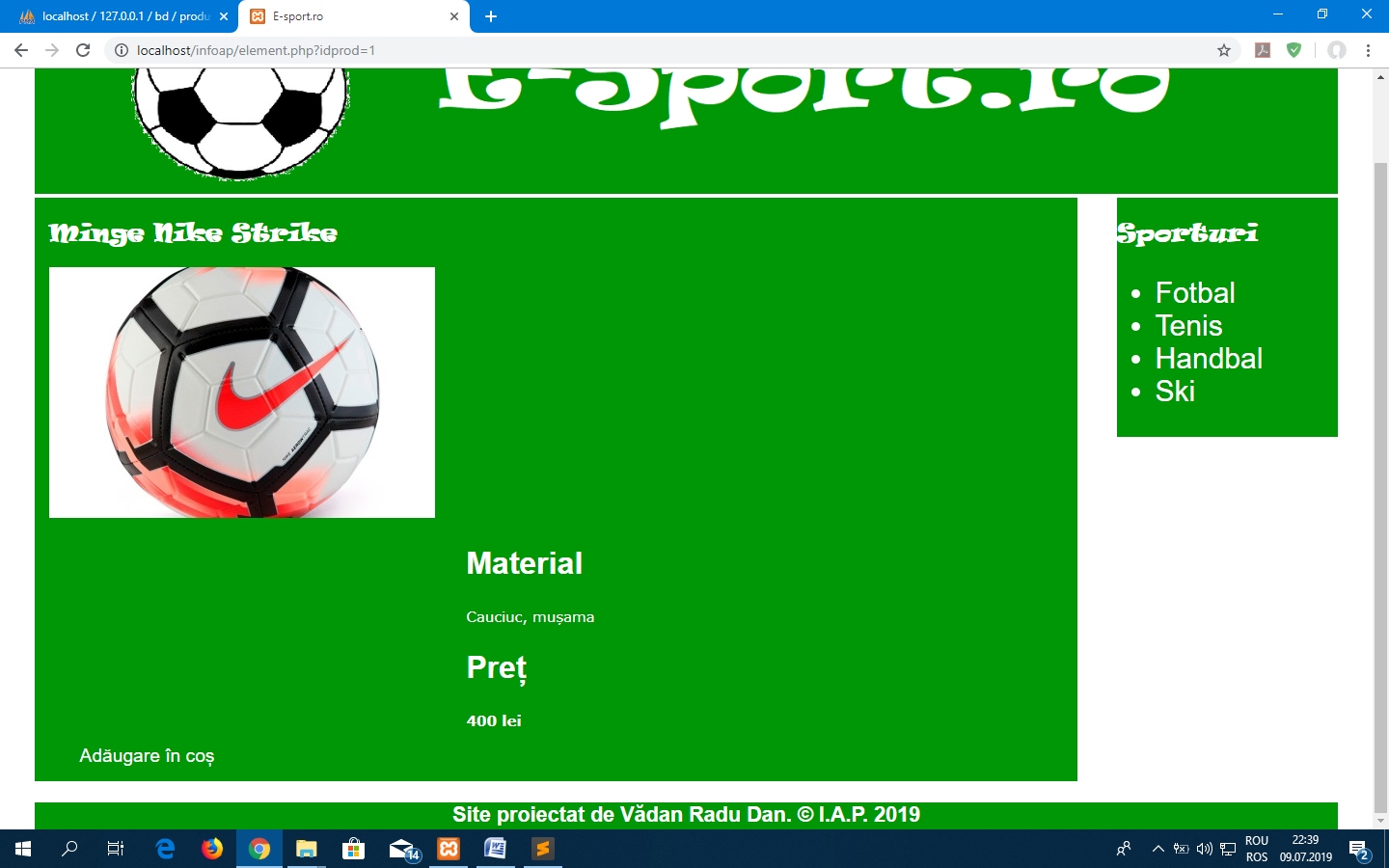
echo "</article>";

$cnx = null;

}

?>

Informațiile din baza de date despre produs vor apărea acum in browser.



Trebuie să modificăm la toate paginile făcute până acum link-urile din <nav> care duc la celelalte pagini carer afișează produse pentru ca browserul să deschidă varianta php a fiecăreia.

<nav> <!-- Plasat pe coloana din dreapta! -->

<h1>Sporturi</h1>

<ul>

<li><a href="Fotbal.php">Fotbal</li></a>

<li><a href="Tenis.php">Tenis</li></a>

<li><a href="Handbal.php">Handbal</li></a>

<li><a href="Ski.php">Ski</li></a>

</ul>

</nav>

Pagina creată anterior conține un link-ul „Adăugare în coș”, care ne duce spre pagina cumpăr.php. În main, aceasta are scriptul php care face un tabel care calculează prețul total al produselor selectate și îl afișează în pagină:

<?php

session\_start();

?>

<div id="continut\_pag">

<main>

<?php

function testare($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

class Produse {

public $id\_produs;

public $imagine;

public $id\_categ;

public $produs;

public $material;

public $preț;

}

// Adaug produsul in cos

$element = testare($\_REQUEST['id\_produs']);

$gasit = false;

if(isset($\_SESSION['cos\_cumparaturi'])) {

$cos = $\_SESSION['cos\_cumparaturi'];

// Verific daca produsul este deja in cos

// explode realizeaza fragmentarea unui sir de caractere folosind

// separatorii continuti in alt sir

// separatorul este ','

// $cos este un sir de caractere (coduri de produs separate prin ',')

$articole = explode(',',$cos);

foreach ($articole as $item) { //prelucrez elementul curent, $item

if($item == $element) {

$gasit = true;

break;

}

}

if(!$gasit) {

$cos = $cos.','.$element;

}

} else {

$cos = $element;

}

$\_SESSION['cos\_cumparaturi'] = $cos;

// afisez continutul cosului

// ma conectez la b.d. pentru a prelua numele si pretul produsului

// pana la conexiunea cu db am preluat doar id\_produs

include("conn.php");

if(isset($cnx)) {

$vtotal = 0;

$articole = explode(',',$cos);

echo '<h1 class="italic centrat">';

echo '<span class="litera italic">Ai</span> ales urmatoarele produse </h1><br />';

echo '<table class="login centrat">';

foreach ($articole as $item) {

// Caut produsul in baza de date dupa $item

$cda= "SELECT \* from produse WHERE id\_produs =$item";

$stmt = $cnx->prepare($cda);

$stmt->execute();

// Exista un singur produs

$prod = $stmt->fetchObject('Produse');

echo '<tr><td>'.$prod->produs.'</td><td><nobr>'.

$prod->preț.'lei</nobr></td></tr>';

$vtotal += (double)$prod->preț;

}

echo "</table><br /><br />";

echo '<p class="centrat">Produsele costă '.

$vtotal." lei.</p>";

}

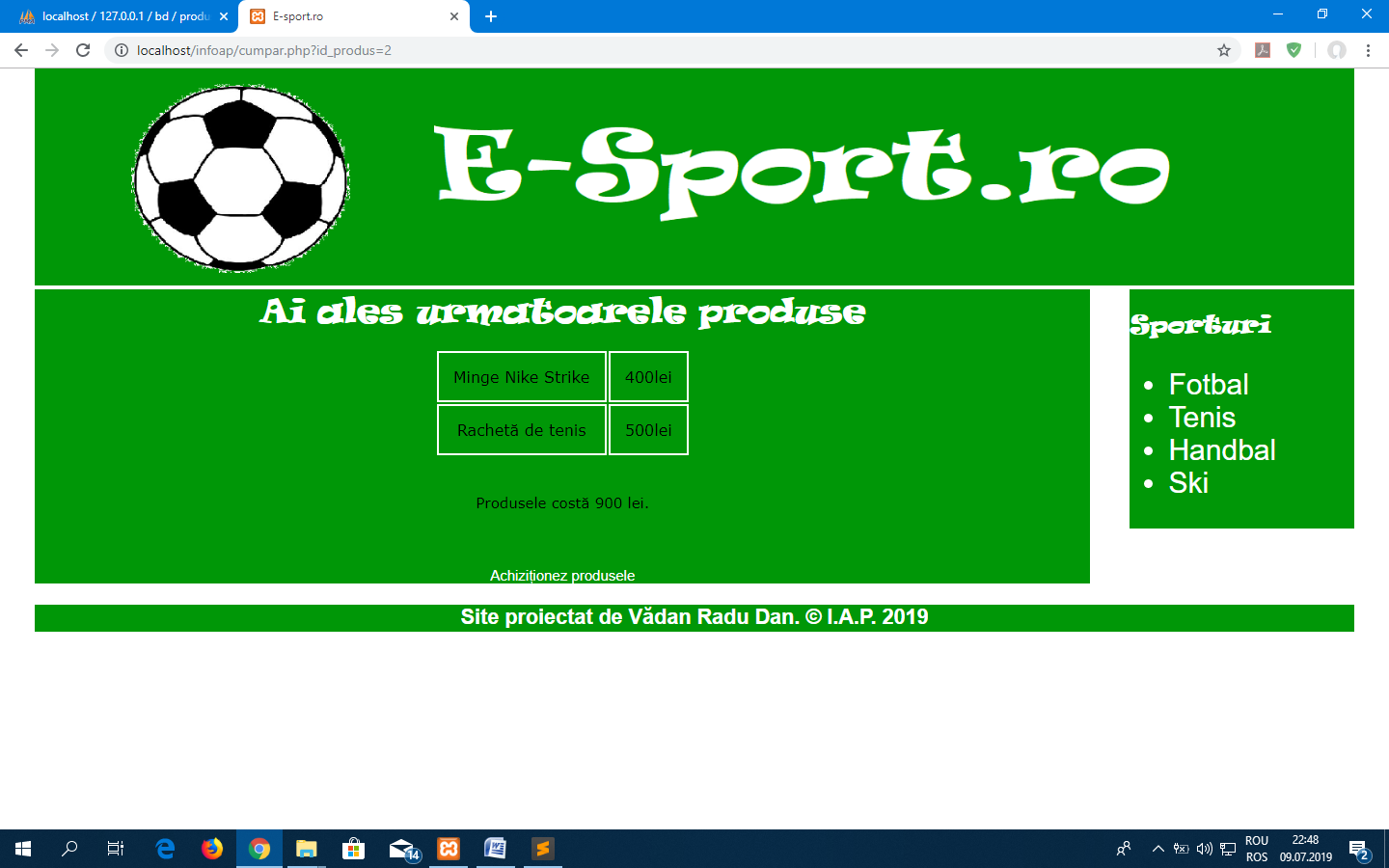
?>

<br /><br /><br />

<p class="centrat"><a href="comanda.php">Achiziționez produsele</a>

</main>

În imaginea de mai jos am selectat două produse, iar browser-ul ne arată suma dintre prețurile celor două.



Aceasta conține si link-ul achiziționez produsele, care ne va duca la pagina de autentificare a utilizatorului, pentru a ști care dintre clienți face comanda. Pagina comanda.php afișează un formul pentru validare date.

<?php

session\_start();

?>

<div id="continut\_pag">

<main>

<h1 class="italic centrat">

<span class="litera italic">A</span>utentificare client</h1><br />

<?php

$cos = $\_REQUEST['coscump'];

// Golesc cosul memorat in $\_SESSION['cos\_cumparaturi'], urmeaza comanda

unset($\_SESSION['cos\_cumparaturi']);

$cli = $\_REQUEST['client'];

?>

<form action="memorez1.php" method="post" class="centrat">

<table class="login centrat">

<?php

echo '<input type= "hidden" name = "coscump" value = '.$cos.'>';

?>

<tr>

<td>Nume </td>

<td><input type="text" name = num></td>

</tr>

<tr>

<td>Parola Dv. </td>

<td><input type="password" name = pw></td>

</tr>

<tr>

<td colspan = 2 class="centrat">

<input type="submit" name ="trimit1" value = "Comanda"></td>

</tr>

</table>

</form>

</main>



Dacă autentificare este reușită apare un mesaj care ne spune că s-a preluat comanda și aceasta va apărea în baza de date in tabelul comenzi, accesându-se pagina memorez1.php.

<?php

session\_start();

?>

<div id="continut\_pag">

<main>

<?php

include("conn.php");

function testare($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

class Clienti {

public $parola;

public $nume;

}

$cos = testare($\_REQUEST['coscump']);

$nume = testare($\_REQUEST["num"]);

$pw = testare($\_REQUEST["pw"]); // Parola

if(isset($cnx)) {

$cda = "SELECT \* FROM clienti";

$stmt = $cnx->prepare($cda);

$stmt->execute();

// Caut clientul in tabelul clienti

$interogare = $cnx->prepare("SELECT \* FROM clienti");

$interogare->execute();

$codcli = 0;

while ($cli = $stmt->fetchObject('Clienti')) {

if (strtoupper ($nume) == strtoupper ($cli->nume) && md5($pw) == $cli->parola) {

break;

}

}

$articole = explode(',',$cos);

foreach ($articole as $item) {

// Caut produsul in baza de date dupa $item

date\_default\_timezone\_set('Europe/Bucharest');

$data = date('Y-m-d'); // data in format aaaa-ll-dd

$interogare1 = $cnx->prepare("INSERT INTO COMENZI VALUES

(NULL, '$codcli', '$item', '1', '$data')");

$interogare1->execute();

}

echo '<h1 class="italic centrat"><span class="litera italic">C</span>';

echo 'omanda a fost preluată pe numele <span class="litera italic">'.$nume.'</span>< <br />Coletul dumneavoastră va ajunge în maxim 3 zile lucrătoare';

echo ' <br />Va multumim!</h1><br />';

// Golesc cosul memorat in $\_SESSION['cos\_cumparaturi']

unset($\_SESSION['cos\_cumparaturi']);

} else {

echo '<h1 class="italic centrat">';

echo 'Nume utilizator sau parola eronata!</h1>';

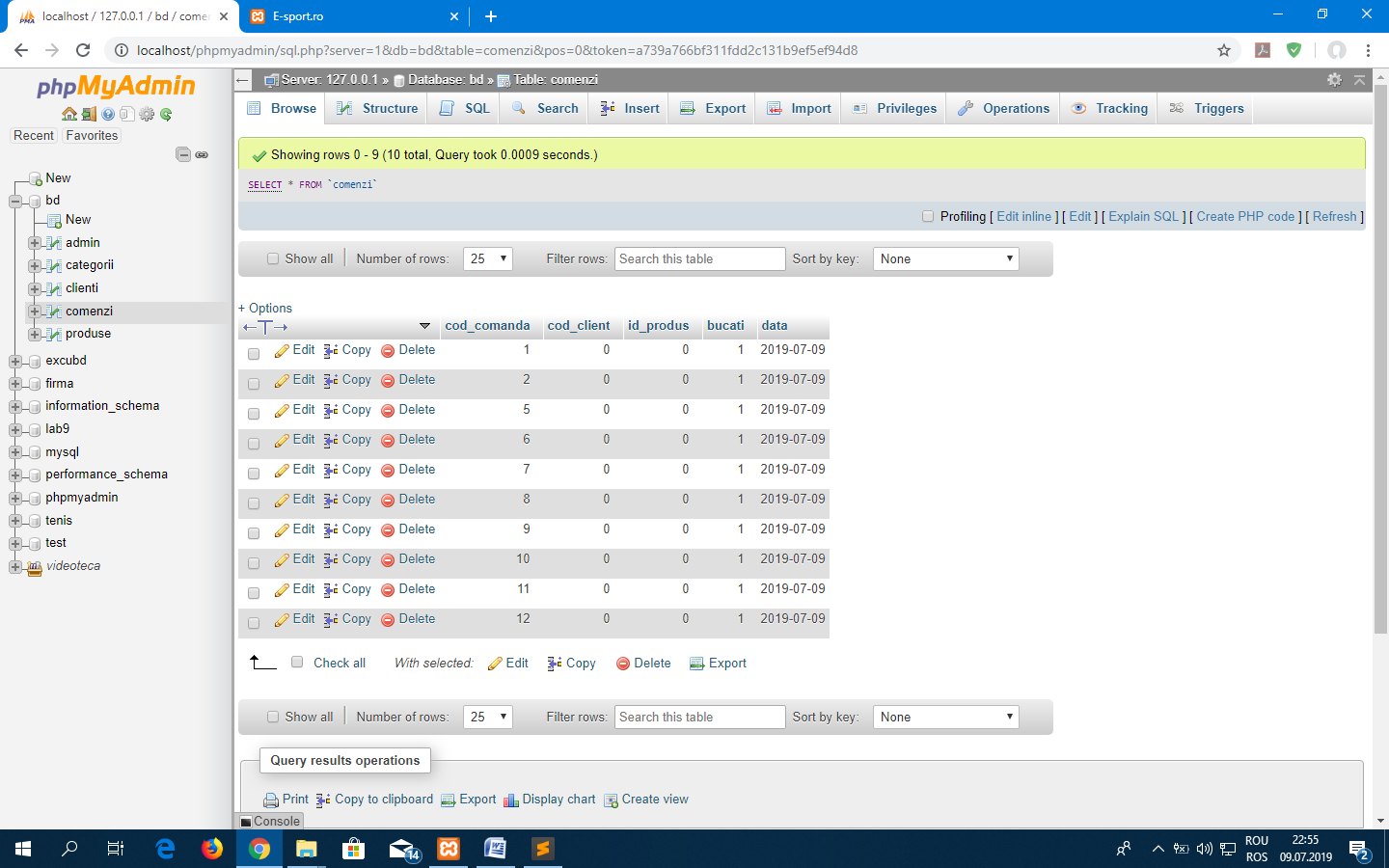
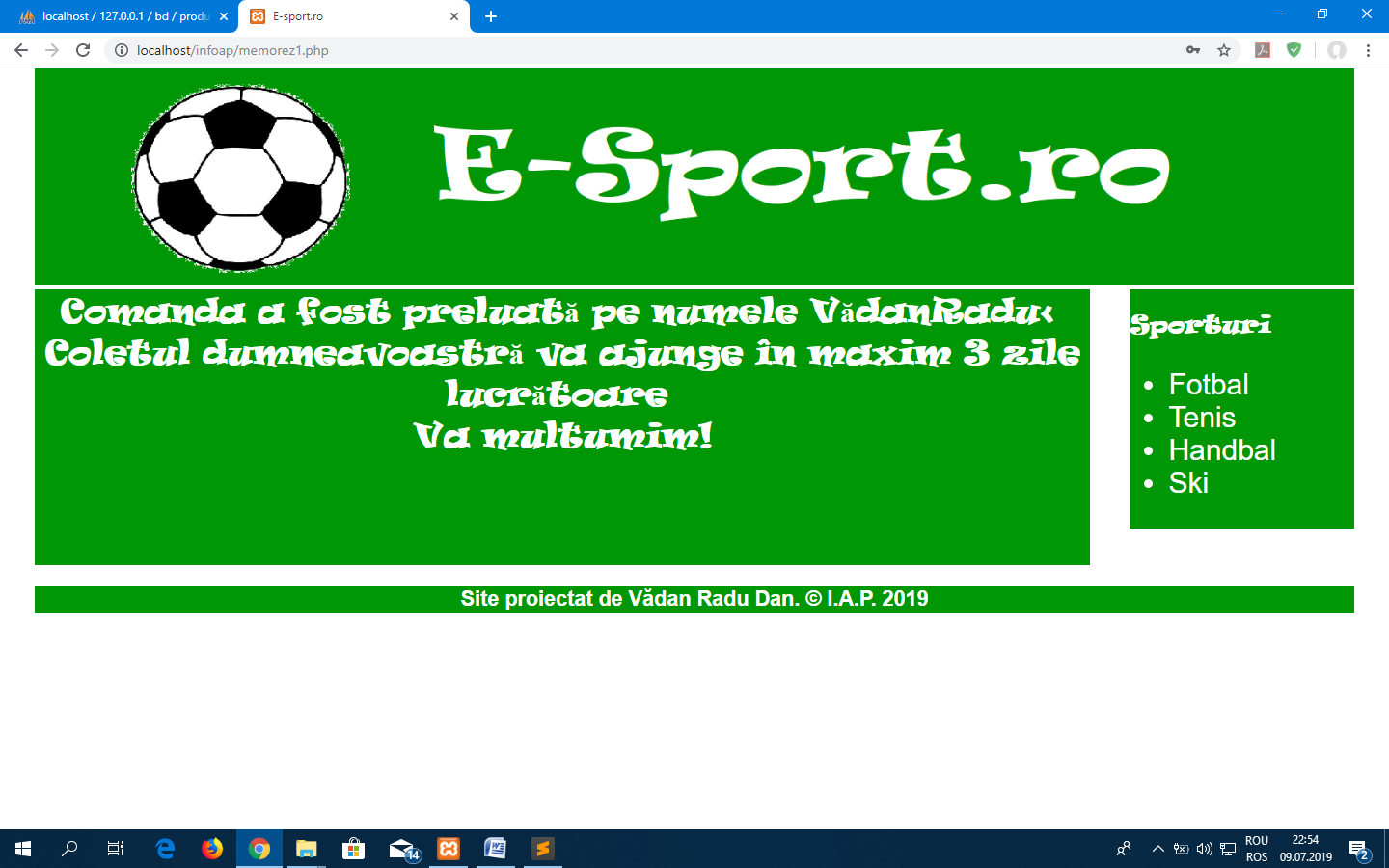
echo '<br /><input id="btn" type="button" value = "Reintroduc datele" ';

echo 'onclick="return fclick()" /<<br />';

}

$cnx = null;

?>



# Bibliografie

Infoap/cursuri – Realizarea site-urilor și a aplicațiilor pentru www