

# Monitorizare automată a locurilor de parcare

Autori: **ȚOC Marius-Nicolae**, clasa a X-a C, profil matematică - informatică intensiv informatică, [toc.marius@gmail.com](mailto:toc.marius@gmail.com), Colegiul Național „Horea, Cloșca și Crișan” Alba Iulia, jud. Alba  
**GRUIAN David**, clasa a X-a C, profil matematică - informatică intensiv informatică, [david.gruian@gmail.com](mailto:david.gruian@gmail.com), Colegiul Național „Horea, Cloșca și Crișan” Alba Iulia, jud. Alba

Profesori coordonatori: **conferențiar universitar doctor, RÎȘTEIU Mircea**,  
Universitatea ”1 Decembrie 1918” Alba Iulia, jud. Alba  
**profesor de informatică grad I, HUMENIUC Ramona**,  
Colegiul Național „Horea, Cloșca și Crișan” Alba Iulia, jud. Alba

## I. Introducere

În jur de 75% din populația Uniunii Europene a ales mediul urban drept loc în care să viețuiască. Conceptul de "*oraș inteligent*" este următoarea etapă în procesul de urbanizare, iar acesta câștigă teren cu factorii de decizie politică. Orașele inteligente pot fi considerate niște ecosisteme cu o componentă tehnică ridicată. Acest tip de metabolism urban este un sistem deschis și dinamic care consumă, transformă și eliberează materiale și energie, se dezvoltă și se adaptează la schimbări și interacționează cu oameni și alte ecosisteme.

Poluarea aerului dăunează sănătății oamenilor și mediului înconjurător. În ciuda faptului că emisiile industriale și cele ale automobilelor au scăzut în ultimii ani, concentrațiile de poluanți rămân ridicate, iar problemele cu calitatea aerului persistă. Acest pericol este local, regional, și de asemenea, internațional pentru că aerul poluat emis într-o țară poate străbate distanțe lungi în atmosferă spre alte locații, în cele din urmă diminuând calitatea aerului și din acele zone.

Poluarea fonică afectează, de asemenea, un mare număr de europeni, iar publicul o percepe ca fiind una dintre problemele majore legate de mediu. Ea poate afecta persoanele și din punct de vedere fiziologic și din punct de vedere psihologic, interferând cu activități de bază, precum somnul, odihna, studiul și comunicarea.

Ca răspuns la solicitare, un nou concept în ceea ce privește orașele așteaptă să îmbunătățească calitatea vieții locuitorilor, cu un impact minim vizavi de mediul înconjurător și cu costuri reduse. Problema parcarilor este una dintre cele mai importante prezente într-un oraș. În întreaga lume, poluarea atmosferică și drumurile aglomerate duc la scăderea calității vieții, rezultând timp pierdut pentru șoferi și combustibil irosit. Comisia Europeană estimează pierderile economice datorate întârzierilor din trafic undeva la aproximativ 150 de miliarde de euro pe an în Europa. Nevoia de a căuta locuri libere de parcare este un factor care contribuie semnificativ la aglomerație și o cauză majoră de stres pentru automobiliști. Pe baza calculelor făcute în Barcelona,

Spania, milioane de șoferi petrec în medie 20 de minute în fiecare zi căutând un loc liber de parcare, în acest timp ei producând peste 2400 de tone de emisii de CO<sub>2</sub>.

## II. Monitorizarea locurilor de parcare

Cea mai eficientă variantă de monitorizare a unei parcări este cea în care se găsesc senzori pe fiecare loc de parcare, deoarece acest sistem nu ține cont de regularitatea parcarii (cu intrare sau iesire standard).

Pentru a detecta prezența unei mașini am folosit un *senzor de proximitate cu infraroșu Sharp 2Y0A02*, al cărui interval de măsurare este între 20 și 150 cm. Datorită faptului că acesta poate produce tensiune analogică, poate interfața foarte ușor cu microcontrolerele, conectându-se la unul dinre porturile analogice ale acestora.

În aplicația pe care noi o propunem, pe fiecare loc de parcare se află un ansamblu local de dispozitive cu rol de detecție, format din: *microcontroler* (propunerea noastră pentru standul experimental este *Arduino Uno Rev3* cu *Atmega328P*), *senzor de proximitate cu infraroșu*, *senzor de temperatură* (Dallas DS18B20 pentru monitorizarea temperaturii ambientului în care se afla senzorul) și *modul XBee* configurat tip router (acesta este însoțit de un *adaptor* care convertește semnalul serial de 5V al microcontrolerului la 3.3 V, tensiune necesară comunicării cu modulele Xbee); acest ansamblu de dispozitive are rol de rutare (*fig. 1* și *fig. 2*).

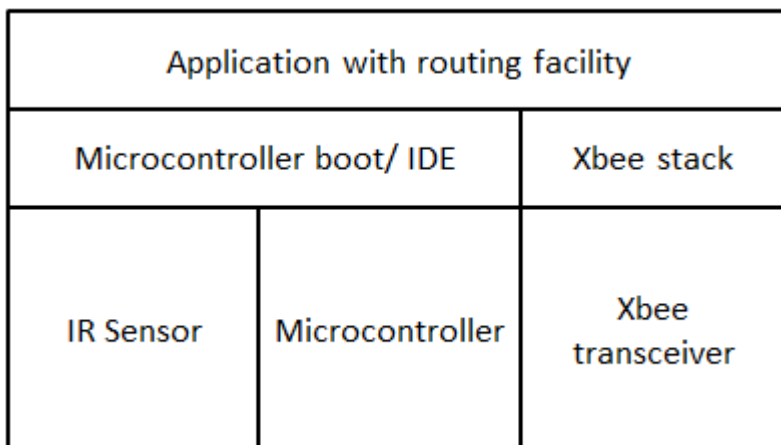


Fig.1 Schema bloc a unui sistem local de detecție

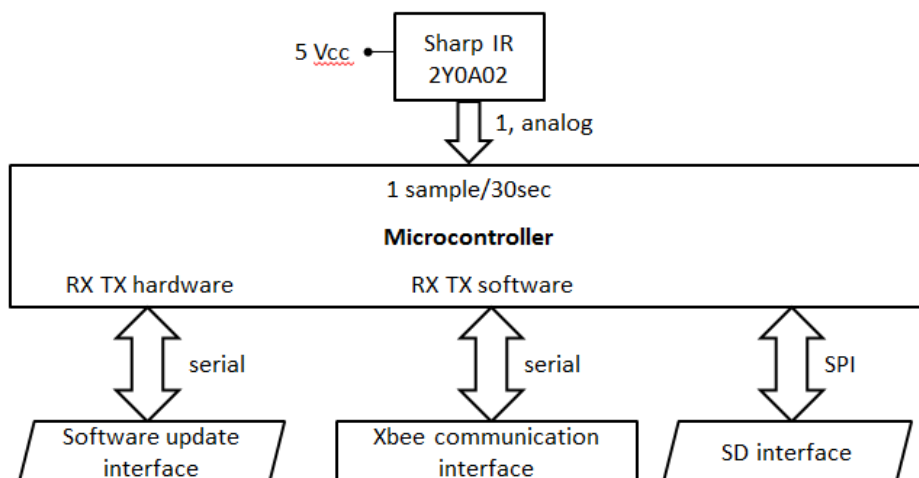


Fig.2 Ansamblu local de dispozitive cu rol de detecție și rutare

La un interval de 30 de secunde, microcontrolerul citește starea senzorului și, implicit, dacă o mașină se află în raza sa de acțiune citește și starea bateriei atașate modulului, iar aceste date le înaintează către modulul XBee, configurat tip *router*, atașat.

Pentru fiecare parcare există un ansamblu de dispozitive cu rol de coordonare, format din: microcontroler (propunerea noastră pentru standul experimental este *Arduino Uno + Ethernet Shield*) și modul XBee configurat tip coordonator (acesta este, de asemenea, însoțit de un adaptor asemănător dispozitivului configurat tip router) (fig. 3 și fig. 4).

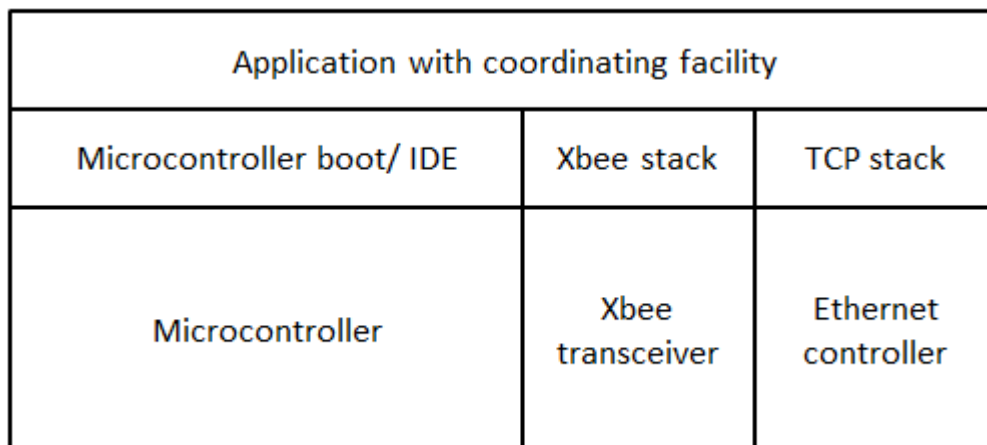


Fig. 3 Schema bloc a unui ansamblu local de coordonare

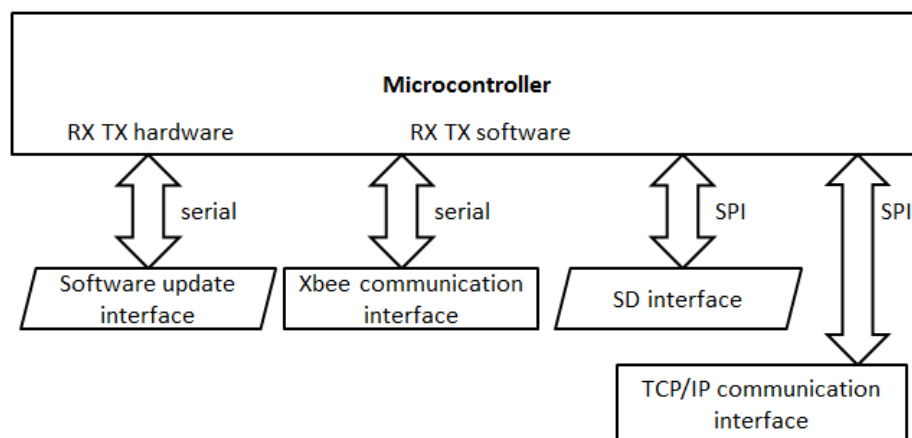


Fig. 4 Ansamblu local de dispozitive cu rol de coordonare

Modulul XBee coordonator primește datele privind fiecare loc de parcare de la modulele XBee atașate fiecărui ansamblu local cu rol de detecție. Acesta le transmite microcontrolerului atașat, care la rândul său, printr-o conexiune la internet, le încarcă într-o bază de date MySQL. O parte din codul sursă Arduino pentru ansamblul coordonator se poate vedea în fig. 5.

Transmiterea datelor colectate de acești senzori se face wireless printr-un protocol de comunicație radio numit *ZigBee*, iar dispozitivele care folosesc acest protocol se numesc *XBee*.

- ZigBee este un protocol care utilizează standardul 802.15.4 ca o bază și adaugă rutare (gateway) și funcționalitate de rețea.
- ZigBee este proiectat să se poată adăuga dispozitive în rețea tip plasă (mesh) care au la bază comunicația radio 802.15.4.

```
xbec_to_mysql_database | Arduino 1.6.1
File Edit Sketch Tools Help
[Icons]
xbec_to_mysql_database
#include <SoftwareSerial.h>
#include <Ethernet.h>
#include <SPI.h>

SoftwareSerial xbeeSerial(4, 5);

byte mac[] = { 0x90, 0xA2, 0xDA, 0x00, 0x0D };
IPAddress server(192,168,0,105);

EthernetClient client;

char cArray[10];int ic=0;
double realPower = 0,apparentPower = 0,powerFactor = 0,Vrms = 0,Irms = 0, Irmsl = 0, Freq = 0;
int currentValue, lastValue, secondLastValue; int lastValueSent; int differenceThreshold = 20;

void setup()
{Ethernet.begin(mac); Serial.begin(9600); xbeeSerial.begin(9600); delay(1000);}
int readXbeeData()
{int done = 0;int inByte;
while (xbeeSerial.available() > 0)
{inByte = xbeeSerial.read();
if (inByte=='A') realPower = atof(cArray);
if (inByte=='B') apparentPower = atof(cArray);
if (inByte=='C') powerFactor = atof(cArray);
if (inByte=='D') Vrms = atof(cArray);
if (inByte=='E') Irms = atof(cArray);
if (inByte=='F') Irmsl = atof(cArray);
if (inByte=='G') {Freq = atof(cArray); done = 1;}
```

Fig. 5 Codul sursă Arduino pentru ansamblul coordonator

Rețeaua plasă este folosită în aplicații în care distanța între două puncte poate fi mai mare decât intervalul în care cele două dispozitive radio, situate în acele puncte, pot comunica, dar dispozitivele radio intermediare sunt în zonă și ar putea transmite orice mesaje spre sau de la dispozitivele radio solicitate (fig. 6).

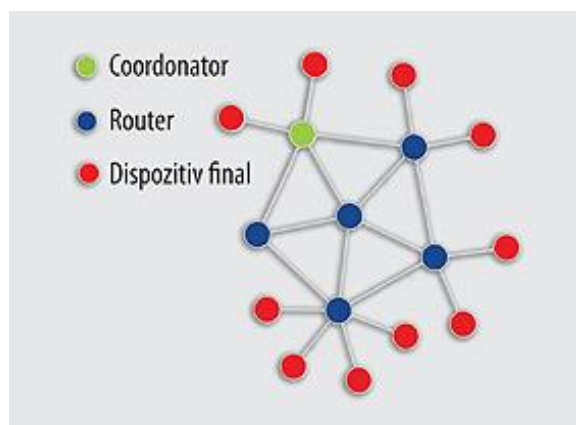


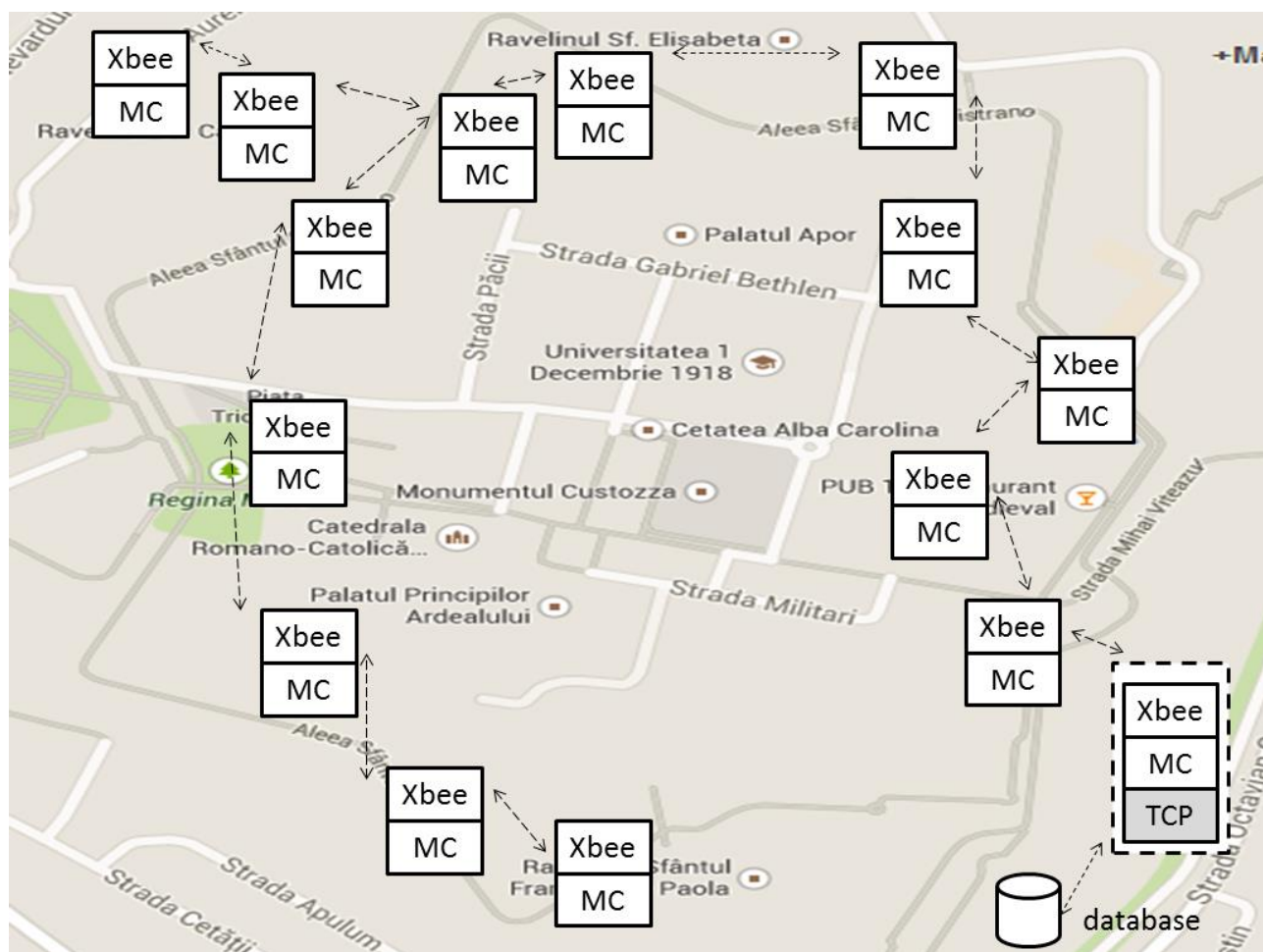
Fig. 6 Structura unei rețele XBee

Modulele de comunicație Xbee au fost concepute pentru comunicații de viteză mică pe distanțe scurte (100 m), medii (3 km) sau mari (40 km) în rețele de senzori, utilizarea lor nelimitându-se doar la aceștia. Viteza maximă de comunicație variază între 10 kbps și 65 Mbps, în funcție de distanță și banda utilizată. Ele funcționează pe benzile de frecvență pentru uz industrial, științific și medical (ISM), cu licență liberă: 2.4 GHz, 868 Mhz. Modelul folosit este *Xbee S2*,

acesta intrând în categoria modulelor pentru comunicații la distanțe mici (120 m) cu viteze de până la 250 kbps.

Rețeaua XBee este numită rețea personală sau PAN (Personal Area Network). Fiecare rețea este definită cu un identificator unic PAN (PAN ID). Acest identificator este comun între toate dispozitivele din aceeași rețea. Dispozitivele XBee sunt fie preconfigurate cu un identificator PAN ID ca să se alăture, fie ele pot descoperi rețele din jur și selectează un PAN ID să se alăture. În cazul în care mai multe rețele XBee funcționează într-o arie comună în care se acoperă una pe cealaltă, fiecare ar trebui să aibă câte un PAN ID. În rețele XBee, coordonatorul trebuie să selecteze un PAN ID și un canal pentru a da start unei rețele. După aceea, se comportă în esență, ca un router. Coordonatorul și routerele pot permite altor dispozitive să se alăture rețelei și pot ruta date pe traseu (dispozitivele pot fi senzori, afișaje, echipamente de uz casnic, etc.). După ce un dispozitiv final se alătură la un router sau coordonator, acesta trebuie să fie capabil de a transmite sau primi date RF prin acest router sau coordonator. Router-ul sau coordonatorul, care a permis unui dispozitiv final să se alăture, devine “părinte” al dispozitivului final. Deoarece dispozitivul final poate “dormi”, părintele trebuie să fie capabil să preia și să rețină pachetele de date primite, dar destinate dispozitivului final, până când dispozitivul final este capabil să se „trezească” și să primească date.

În *fig.7* propunem schema pentru monitorizarea locurilor de parcare din Cetatea Alba Carolina.



*Fig. 7 Monitorizarea locurilor de parcare din Cetatea Alba Carolina*

### III. Interpretarea și afișarea datelor culese de la senzori

Pentru a interpreta și afișa datele culese de la senzori am realizat o aplicație web. Schema bloc a acestei aplicații se află în fig. 8.

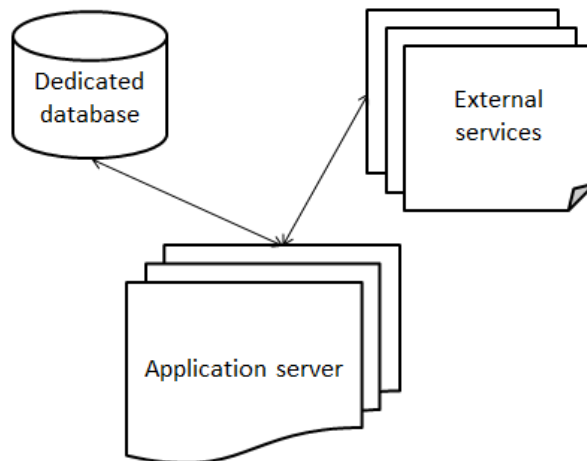


Fig. 8 Schema bloc pentru aplicația web

**HTML (Hyper Text Mark-up Language)** este limbajul de marcare pe care l-am utilizat în crearea de pagini web pentru aplicație fiind unul dintre primele elemente fundamentale ale WWW (World Wide Web) iar multe din trăsăturile lui cum ar fi independența față de platformă, structurarea formătărilor și legăturile hypertext, fac din el un foarte bun format pentru documentele Internet și Web. Acest limbaj este folosit în general pentru a prezenta informațiile dintr-o pagină web: paragrafe, fonturi, tabele, iar aspectul siteului este determinat de alte limbaje precum CSS (Cascade Style Sheets).

Pe partea de web, pentru aplicație, am ales modalitatea cea mai eficientă de delimitare și de formatare a unui bloc text prin folosirea delimitatorilor `<div>...</div>` din HTML. Am utilizat și **stiluri CSS** care ne-au permis să definim formătărilor necesare (font, culori, margini, borduri, spațieri, etc.) (fig.9)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>FINAL</title>
5 </head>
6 <body background="http://pl.pichost.me/i/28/1512030.jpg">
7 <div align="center" style="margin-left: 200px; background-color: #E44424; margin-right: 200px; margin-top:100px;" >
8
9 <div align="right" style="background-color: #E44424; height:3px" > </div>
10
11 <a href="final_cetate.php" >
12 <div align="center" style="background-color: #191919;border-style: solid; border-width: 10px; border-color: #E44424;">
13
14 </a>
15
16 <div align="center" style="background-color: #191919; height:13px"> </div>
17
18 <a href="final_casa.php" >
19 <div align="center" style="background-color: #191919; border-style: solid;border-width: 10px; border-color: #E44424;">
20
21 </a>
22
23 <div align="center" style="background-color: #191919; height:13px"></div>
24
25 <a href="final_hoc.php" >
26
27 </a>
28
29 <div align="center" style="background-color: #191919; height:250px">
30 </div>
31 <div>
32 <font color="Black" face="trebuchet MS" size="6px">
33 Parcare Electronica </br>
34 Gruian David & Toș Marius
35 </font>
36 </div>
37 <div align="center" style="background-color: #191919; height:250px">
38 </div>
39 </div>
40 </body>
41 </html>
```

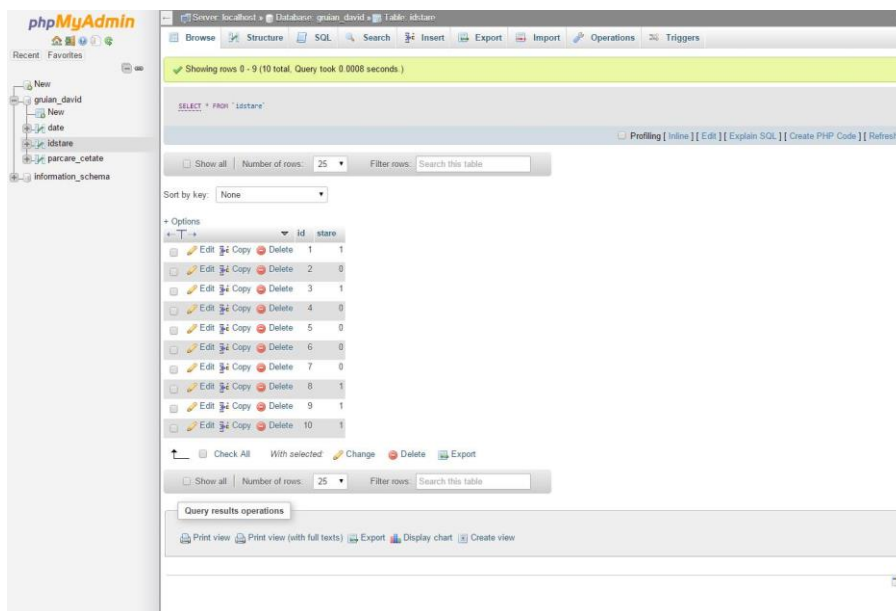
Fig. 9 Cod sursă HTML



Activitatea de programare Web presupune utilizarea unor limbaje de programare atât pe client, cât și pe server. Pentru a crea aplicația web dinamică și a realiza cuplarea la o bază de date locală am ales **limbajul PHP** (PHP: Hypertext Preprocessor)- cel mai utilizat limbaj de programare Web pentru server, un limbaj de scripting de uz general, cu cod-sursă deschis (Open Source) și pe care l-am integrat în HTML.

Pentru a putea rula scenarii php (scripturi) avem nevoie să transformăm calculatorul în server Web, server de aplicații PHP și nu în ultimul rând server de baze de date **MySQL**. După instalarea fiecăruia din aceste aplicații apare necesitatea configurării lor. Pentru a evita problemele configurării, putem instala un pachet complet ce conține Apache + PHP + MySQL + PHPMyAdmin care se numește **XAMPP (PHPTriad, EasyPhp, Base, WampServer, Denver, etc.)**. Aplicația PHPMyAdmin inclusă în pachetul **XAMPP** permite administrarea comodă a SGBD MySQL. Înainte de a rula scripturi PHP vom lansa aplicația **XAMPP**, după care se va lansa panoul de control al aplicației **XAMPP**.

Am utilizat PhpMyAdmin- utilitar PHP cu scopul de a administra baze de date MySQL pe web pentru a crea/șterge/modifica tabele, adăuga/șterge/modifica câmpuri, executa interogări SQL, administra privilegii, exporta datele în diverse forme (*fig.11*)



*Fig.11 phpMyAdmin-Crearea unei interogări SQL pentru starea locurilor din parcare*

Arhitectura site-ului a constat în asamblarea părților aplicației:

- Proiectarea unei baze de date, pentru stocarea locurilor de parcare;
- Scrierea codului SQL și PHP pentru a accede la date și la stările funcționale (locuri libere, locuri ocupate în parcare)
- Oferirea situației de a vizualiza în timp real câte locuri de parcare sunt ocupate din totalitatea locurilor disponibile printr-o interfață HTML prietenoasă (*fig. 12*)





Fig. 12 Interfața de Generare a numărului de locuri ocupate din parcare

#### IV. Concluzii

Proiectul creat răspunde unora dintre cele mai actuale nevoi cotidiene: evitarea aglomerației din trafic, reducerea poluării și menținerea scăzută a cheltuielilor. Conceptul de oraș modern nu mai e doar un vis : el începe să prindă contur și să devină din ce în ce mai eficient, iar noi suntem foarte bucuroși că putem contribui la acest lucru.

În continuare avem în lucru dezvoltarea proiectului pe partea de senzorică, pentru a fi cât mai eficient din punct de vedere al consumului de energie, dar și pe partea de afișare, pentru a avea o interfață cât mai prietenoasă și eficientă pentru conducătorul auto și o arhitectură scalabilă-odată cu creșterea numărului de locuri de parcare și a mașinilor, aplicația web să ofere funcțiile sale fără a genera erori sau a obține performanțe slabe.

#### V. Bibliografie

1. Anghel, Traian, *Dezvoltarea paginilor Web folosind XHTML, PHP și MySQL*, Editura Polirom, Iași, 2005
2. Buraga, Sabin, *Proiectarea siturilor Web- Design și funcționalitate*, Editura Polirom, Iași, 2002 (ediția I) și 2005 (ediția II)
3. <http://www.arduino.cc/>
4. <http://www.digi.com>
5. <http://www.libelium.com/>
6. <http://www.tehnorama.ro>
7. <http://www.robofun.ro>
8. <http://www.electronica-azi.ro>
9. <http://www.w3schools.com/>
10. <http://en.wikipedia.org/>
11. <http://www.codecademy.com/>