



IT Mafia – A2

**Virna Stefan-Alexandru
Gavrila Maria-Denisa**

1. Abstract

Deoarece din ce in ce mai multi romani prezinta interes pentru domeniul auto, site-ul AutoStat vine cu o serie de informatii care redau intr-o maniera bine-structurata preferintele din tara noastra si genereaza statistici utile pentru cei care doresc sa-si achizitioneze un autovehicul, sau doar sa studieze piata in scopuri personale/profesionale.

Pentru ca numarul masinilor creste de la an la an, este interesant si totodata substantial sa cunoastem directia in care se indreapta parcul auto din Romania, urmarind treptat evolutia prin intermediul rapoartelor oferite de site-ul nostru.

2. Introducere

AutoStat este o aplicatie web care permite vizualizarea de statistici din domeniul auto, pe baza informatiilor furnizate de Guvernul Romaniei, din ultimii ani. Utilizatorii pot accesa situatia parcului auto din tara noastra, statisticile fiind structurate pe judete, iar informatiile fiind centralizate pe baza criteriilor din documentele oficiale (judet, categorie, an, marca, combustibil).

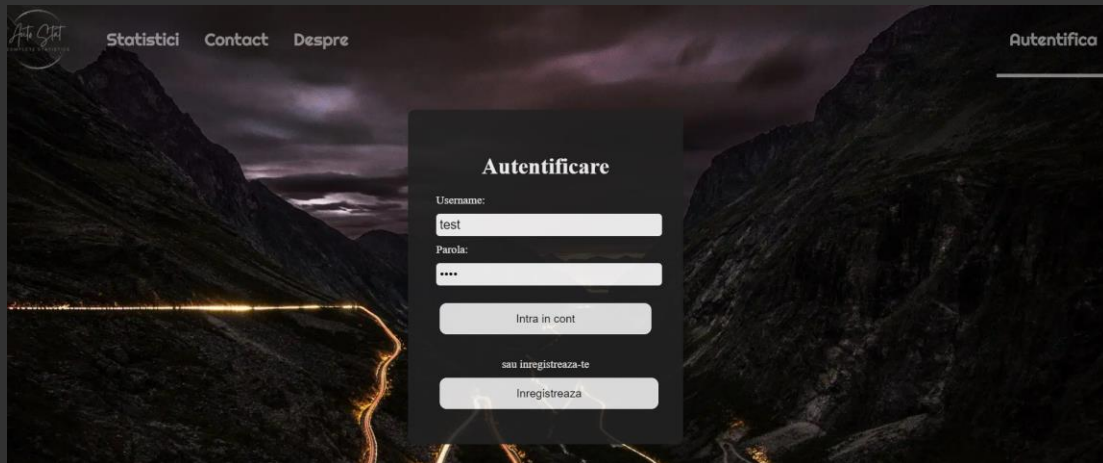
Aplicatia este dezvoltata utilizand Node.JS pentru serviciile API, si tehnologii web clasice (HTML, CSS si JavaScript) pentru interfata utilizatorului. Pentru imbunatatirea procesului de dezvoltare si exploatare a acesteia, am implementat principii moderne, cum ar fi CI/CD, aplicatia fiind automat actualizata si pe serverele Azure cu fiecare modificare in codebase-ul de pe GitHub.

3. Implementare

3.1 Inregistrare si Login

Aplicatia permite crearea unui cont de utilizator, urmata de logarea propriu-zisa, care ofera acces catre pagina principala, de unde userul isi poate selecta urmatoarea actiune.

Daca utilizatorul nu are inca un cont, acesta se poate inregistra, iar datele sale vor fi trimise catre Backend si inserate, apoi, in baza de date. La autentificare, daca datele introduse sunt corecte si se gasesc in baza de date, se genereaza automat un token gwt (are 12 ore timp pana la expirare), care este mai apoi salvat de Frontend in LocalStorage. Ulterior, acest token este trimis, in cadrul fiecarui request, intr-un header "Authorization", folosit de Backend pentru validarea identitatii utilizatorului, dupa decodificare. Din pagina contului de utilizator, acesta se poate deconecta, dand click pe butonul "logout", care il redirectioneaza catre pagina principala.



```
src > api > JS login.js > ...
1  import sqlite3 from 'sqlite3';
2  import jwt from 'jsonwebtoken';
3  import { getDate } from './statistics.js';
4  const dbFilePath = "../database.db";
5
6  > export const loginUser = (username, password, callback) => { ...
30 }
31
32 > export const registerUser = (username, password, callback) =>{ ...
57 }
58
59 > export const decodeToken = (token) => { ...
61 }
62
```

3.2 Vizualizare Statistici

Dupa logare, utilizatorul are acces pagina principala, care contine o sectiune de text informativ, iar in partea de sus se afla butoanele corespunzatoare pentru navigarea catre sectiunile importante. Butonul “Vezi harta”, aflat in ultimul paragraf, trimite utilizatorul direct catre pagina destinata statisticilor pe judete, in timp ce “Vezi statistici pe tara” va deschide pagina cu statistici corespunzatoare pentru intreg teritoriul Romaniei. Accesand harta Romaniei si apasand pe bulina verde corespunzatoare unui judet, userul este redirectat catre aceeasi pagina de statistici, cu filtrul aplicat pentru judetul selectat.



Pagina cu cel mai mare impact este cea destinata vizualizarii de statistici unde, pentru a accesa datele dorite, utilizatorul poate aplica o serie de filtre si poate opta pentru numarul de rezultate pe care doreste sa le vada pe o pagina. Daca rezultatele cautarii prezinta interes sporit, pot fi descarcate in format CSV. Pentru a oferi utilizatorului o imagine cat mai clara asupra informatiilor solicitate, pagina pune la dispozitie, in seciunea de statistici, grafice care arata distributia procentuala a rezultatelor si care pot fi cu usurinta salvate in format Webp. Constructia lor a fost realizata cu ajutorul unei librarii interne, impartite in trei clase, cate una pentru fiecare tip de grafic (Piechart, Linechart, Barchart), plus un fisier "utils" ce contine metode ajutatoare folosite in afisarea pe ecran, utilizand tehnologia canvas.

Judet

TOATE

Categorie

TOATE

An

TOATE

Marca

TOATE

Combustibil

TOATE

Salveaza cautarea

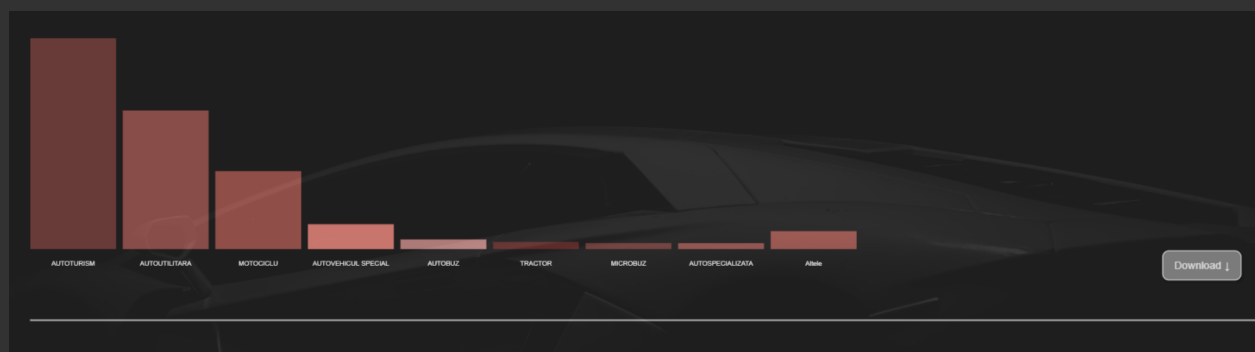
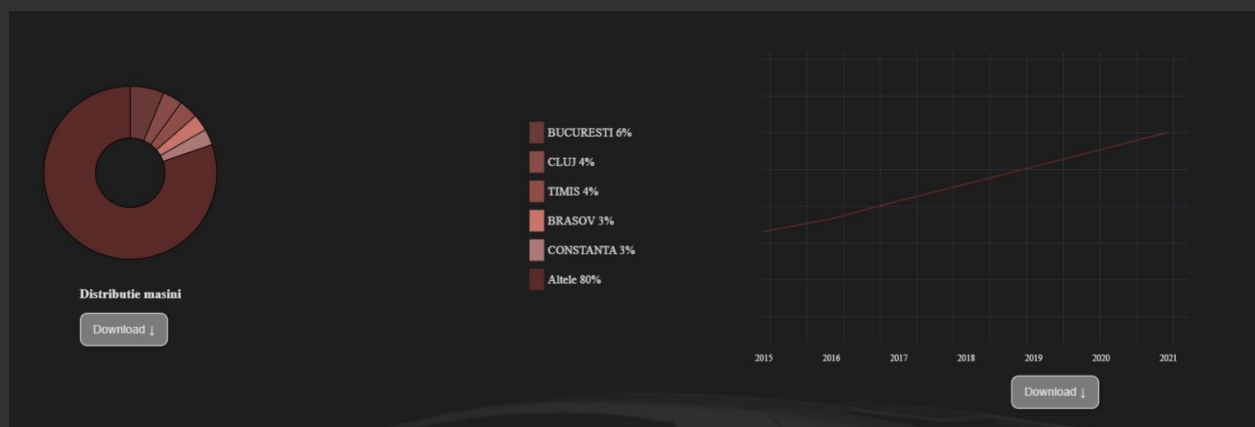
Detalii

Descarca CSV



| Judet | An | Marca | Categorie | Categorie Comunitara | Combustibil | Descriere | Total |
|-------|------|--------|-----------|----------------------|-------------|-----------|-------|
| ALBA | 2015 | IVECO | AUTOBUZ | M2 | MOTORINA | DAILY | 18 |
| ALBA | 2015 | IVECO | AUTOBUZ | M2 | MOTORINA | | 2 |
| ALBA | 2015 | A8 | AUTOBUZ | M3 | MOTORINA | | 1 |
| ALBA | 2015 | BMC | AUTOBUZ | M3 | MOTORINA | TBX | 3 |
| ALBA | 2015 | BMC | AUTOBUZ | M3 | MOTORINA | | 14 |
| ALBA | 2015 | BOVA | AUTOBUZ | M3 | MOTORINA | FUTURA | 3 |
| ALBA | 2015 | BOVA | AUTOBUZ | M3 | MOTORINA | | 5 |
| ALBA | 2015 | DAC | AUTOBUZ | M3 | MOTORINA | | 5 |
| ALBA | 2015 | DENNIS | AUTOBUZ | M3 | MOTORINA | | 1 |
| ALBA | 2015 | EVOBUS | AUTOBUZ | M3 | MOTORINA | CITARO | 1 |

« 1 2 3 ... 176674 »

10

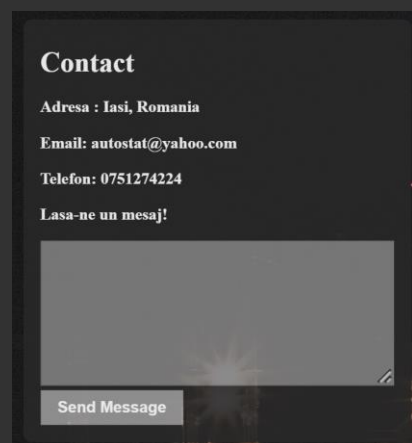


Logarea in pagina ofera posibilitatea de a salva un filtru de cautare aplicat, acesta putand fi ulterior vizualizat din pagina contului personal.

| Cautari | | |
|-------------------------|-----------|---|
| Data | Nume | |
| 02-06-2023 | cautare 1 |  |
| 02-06-2023 | cautare 2 |  |
| <button>Logout</button> | | |

3.3 Pagina de Administrare

Aceesand pagina de Contact, se deschide sectiunea in care se afla datele de contact ale companiei Auto Stat si un textbox in care clientii pot trimite mesaje directe catre administrator. Dupa tastarea textului dorit si apasarea butonul “Trimite”, pe ecran se va afisa un mesaj care sa confirme realizarea cu success a actiunii.



In contextul acestui proces, backendul se ocupa de adaugarea mesajului in baza de date, impreuna cu data adaugarii, prin intermediul functiei “AddNewMessage”, obtinerea ulterioara a mesajelor in scopul afisarii fiind asigurata de functia “GetMesaje”.

```
1  import sqlite3 from 'sqlite3';
2  import { getDate } from './statistics.js';
3  const dbFilePath = "../database.db";
4
5  > export const AddNewMessage = (mesaj, callback) => { ...
20 }
21
22 > export const GetMesaje = (callback) => { ...
37 }
```

La logarea in aplicatie cu date corespunzatoare administratorului, in sectiunea “Contul meu” vor putea fi vizualizare mesajele primite, dar si un grafic care arata numarul de utilizatori noi inregistrati in ultimele zile. Pentru obtinerea acestora si afisarea lor se face cate un apel AJAX de tip GET catre ruta corespunzatoare. Tot in pagina de administrare se gaseste o sectiune care permite incarcarea de noi CSV-uri, ale caror date urmeaza a fi trimise si inserate in baza de date, pentru a pune intotdeauna la dispozitia utilizatorilor o suita de informatii complet actualizate.

Mesaje

| Data | Nume |
|------------|------------|
| 02-06-2023 | Good job! |
| 02-06-2023 | :) :) :) |
| 02-06-2023 | Best team! |

Utilizatori noi



Incercare fisiere

Alege fisier: No file chosen

```
3 > $(function() {
4 >   $.ajax({
5 >     url: "/api/mesaje",
6 >     type: "GET",
7 >     headers: { ...
9 >   },
10 >   success: function(data) { ...
19 >   }
20 >
21 > });
22 >
23 > $.ajax({
24 >   url: "/api/get_admin_statistics_register_users",
25 >   type: "GET",
26 >   success: function(data) { ...
35 >   }
36 > });
37 > });
38 >
39 > const loadSearch = (id) => { ...
46 > }
```

3.4 Prelucrarea datelor

În urma descărcării CSV-urilor cu informații de pe site-ul guvernului, datele sunt introduse într-un folder specific, apoi sunt preluate cu ajutorul unui microserviciu și inserate într-o bază de date SQLite, care conține un tabel având coloane corespunzătoare tuturor specificațiilor pentru mașini. În aceeași bază de date, este creat un tabel care conține mesajele trimise de utilizator către administrator, cel din urmă având posibilitatea de a le vizualiza din pagina de admin, dar și un tabel care permite stocarea căutărilor salvate, în scopul afișării ulterioare. Tot în baza de date, există un tabel specific utilizatorilor, care conține username-ul și parola.

```
const db = connectToDatabase();
db.exec(`
  CREATE TABLE masini
  (
    an          NUMERIC,
    judet       VARCHAR(50),
    categorie    VARCHAR(50),
    categorie_com VARCHAR(50),
    marca        VARCHAR(50),
    desc         VARCHAR(50),
    combustibil  VARCHAR(50),
    total        NUMERIC
  );

  CREATE TABLE users
  (
    username  VARCHAR(50),
    password  VARCHAR(50),
    added_time TEXT
  );

  CREATE TABLE saved_searches
  (
    id          INTEGER PRIMARY KEY AUTOINCREMENT,
    username    VARCHAR(50),
    nume        VARCHAR(50),
    added_time  TEXT,
    filters     TEXT
  );

  CREATE TABLE mesaje_prime
  (
    mesaj      TEXT,
    added_time TEXT
  );
`);
```

Prelucrarea datelor are loc într-un serviciu separat, pentru a nu afecta performanța aplicației. O primă execuție are loc la pornirea aplicației pe server, ulterior serviciul runde automat în fiecare zi la miezul nopții prin intermediul unui cronjob, realizat cu ajutorul bibliotecii “node-cron”.

```
data > JS auto_import.js > ...
1  import cron from 'node-cron';
2  import { runAll } from './import.js';
3
4  console.log("Starting up auto-import service. Import will occur every day at midnight");
5  cron.schedule('0 0 0 * * *', () => {
6    runAll();
7    console.log("Auto import executed...");
8  });
```

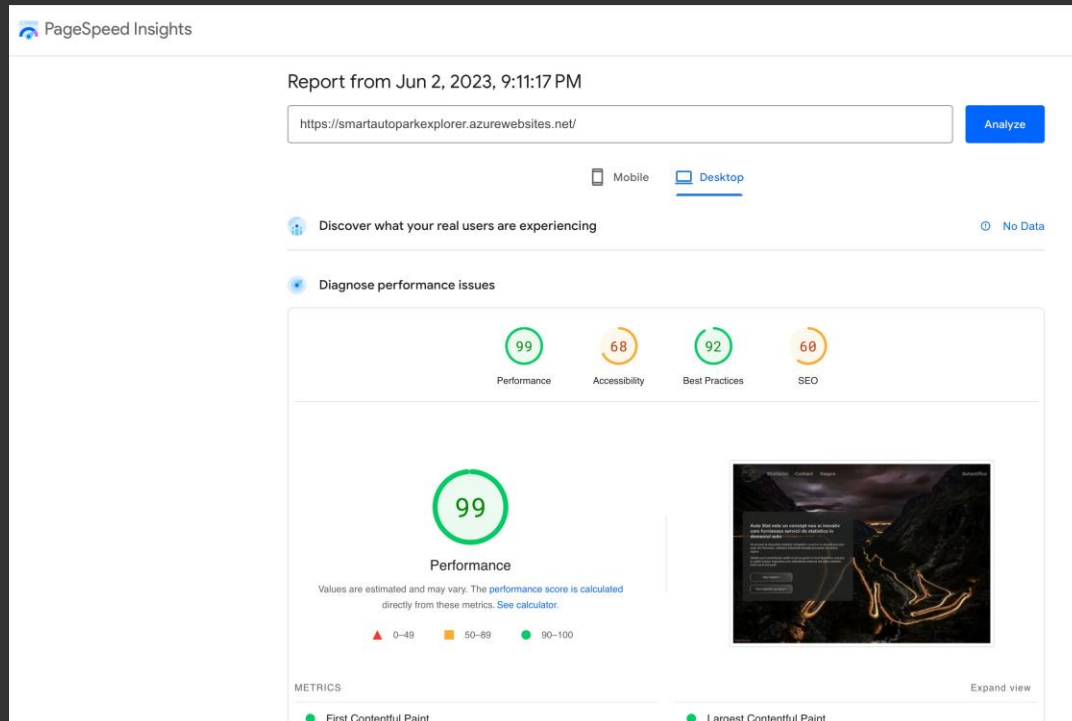
3.5 Rute

În fișierul request router sunt definite toate rutele API utilizate în construcția site-ului. Acestea sunt utilizate pentru autentificare, înregistrare, obținerea de statistici, generarea de fișiere CSV, gestionarea mesajelor și alte operațiuni relevante pentru aplicația web. La începutul fișierului, există o secțiune pentru importuri, în care sunt incluse modulele și funcțiile necesare pentru funcționalitatea rutelor, după care este definită funcția principală “routeRequest”, care gestionează toate cererile primite de la client. Aceasta verifică metoda HTTP împreună cu URL-ul fiecărei cereri și apelează funcțiile corespunzătoare pentru a trata aceste cereri. Sunt utilizate metode HTTP specific REST, pentru manipularea resurselor și comunicarea între server și client. În ceea ce privește resursele, acestea sunt reprezentate de utilizatorii din baza de date, iar informațiile despre aceștia sunt transferate între client și server sub forma unor obiecte JSON. Comunicarea între client și server este stateless, ceea ce înseamnă că fiecare cerere conține toate informațiile necesare pentru a fi procesată, iar serverul nu menține informații despre starea clientului între cereri.

```
src / api / requestrouter.js / routeRequest
1  import { decodeToken, loginUser, registerUser } from "../login.js";
2  import { getParams } from "../utilities.js";
3  import { GetJudete, GetMarca, GetCombustibil, GetStatistics, GetCategorii, GetAni, GetGraphicsTotalJudete, GetGraphicsTotalCategorii,
4  import fs from 'fs';
5  import crypto from 'crypto';
6  import { AddNewMessage, GetMesaje } from "../messages.js";
7
8  export const routeRequest = (req, response) => {
9  >   if(req.method === 'POST' && req.url === '/api/authenticate_user') { ...
29 >   } else if(req.method === 'POST' && req.url === '/api/register_user') { ...
44 >   } else if(req.method === 'GET' && req.url.includes('/api/get_statistics')) { ...
66 >   } else if(req.method === 'GET' && req.url === '/api/get_admin_statistics_register_users') { ...
76 >   } else if(req.method === 'GET' && req.url === '/api/statistics/judete') { ...
85 >   } else if(req.method === 'GET' && req.url === '/api/statistics/categorii') { ...
94 >   } else if(req.method === 'GET' && req.url === '/api/statistics/ani') { ...
103 >   } else if(req.method === 'GET' && req.url === '/api/statistics/marca') { ...
112 >   } else if(req.method === 'GET' && req.url === '/api/statistics/combustibil') { ...
121 >   } else if(req.method === 'POST' && req.url === '/api/statistics/generate_csv') { ...
155 >   } else if(req.method === 'GET' && req.url === '/api/check_auth') { ...
168 >   } else if(req.method === 'POST' && req.url === '/api/get_distribution_chart_data/judete_total'){ ...
189 >   } else if(req.method === 'POST' && req.url === '/api/get_distribution_chart_data/categorii_total'){ ...
209 >   } else if(req.method === 'POST' && req.url === '/api/get_distribution_chart_data/marci_total'){ ...
230 >   } else if(req.method === 'POST' && req.url === '/api/get_distribution_chart_data/an_total'){ ...
250 >   } else if(req.method === 'POST' && req.url === '/api/get_distribution_chart_data/combustibil_total'){ ...
270 >   } else if(req.method === 'POST' && req.url === '/api/save_search') { ...
291 >   } else if(req.method === 'POST' && req.url === '/api/mesaje'){ ...
305 >   } else if(req.method === 'GET' && req.url === '/api/mesaje'){ ...
329 >   } else if(req.method === 'GET' && req.url === '/api/get_searches'){ ...
348 >   } else{ ...
361 >   }
362 }
```


3.6 Performanta

Pentru a optimiza scorul dat de Google Page Speed, am recurs la comprimarea resurselor utilizate, prin comprimarea scripturilor css si a imaginilor. Aceste actiuni sunt parte a pipeline-ului de compilare si deployment. Codul SCSS este compilat in CSS si minimizat, iar toate fisierele din folderul /images sunt trecute printr-un script de comprimare ce are la baza biblioteca npm "compress-images", ceea ce a dus la un scor de 99.



3.7 Tehnologii Frontend

Pentru realizarea interfetei web am folosit HTML, SCSS, compilat in CSS, si implicit JavaScript. Am utilizat si AOS pentru animatii, impreuna cu JQuery pentru interactiunea cu elementele DOM si requesturile AJAX.

3.8 Tehnologii Backend

Pe partea de backend se utilizeaza NodeJs, fara framework-uri, impreuna cu librariile SQLite3, Node-cron, Compress-images, Formidable, Gifsicle, Jsonwebtoken, PNGquant-bin. Acestea sunt utilizate pentru interactiunea cu baza de date, realizarea cron-joburilor, comprimarea fisierelor si procesarea token-ului JWT utilizat la validarea identitatii utilizatorului.

3.9 Design

Pe fiecare dintre paginile aplicatiei web, domeniul auto este sugerat fie de imaginile cu drumuri de noapte, dinamice, fie de unele cu vehicule in nuante inchise, totul fiind asamblat intr-un cadru dark, rece.

Cromatica paginii are la baza nuante neutre, imbinate pe alocuri cu elemente contrastante, cum ar fi graficele din partea de jos a paginii de statistici. Zonele de text sunt construite folosind efecte de transparenta, pentru un design modern si minimalist. Textul de culoare alba, folosit pe majoritatea paginilor, vine in opozitie cu fundalurile gri, oferind utilizatorilor o experienta placuta din punct de vedere vizual.

Logo-ul foloseste atat un font elegant, pentru numele site-ului, cat si unul clasic, pentru motto, ambele zone de text fiind incadrate de forma circulara, care integreaza aceleasi nuante neutre.