

Proiect de practica

Student: Kotai Melinda-Antonela,
FSEGA, specializarea Informatica Economica
anul 2, grupa 1

Cuprins

Prezentarea generala a firmei, a proiectului si a modului de comunicare si organizare	3
Prezentarea activitatilor desfasurate si a contributiei personale in cadrul proiectului realizat	5
prima saptamana.....	5
a doua saptamana	5
a treia saptamana.....	8
a patra saptamana.....	12
a cincea saptamana	16
a sasea saptamana	18
a saptea saptamana	19
Concluzii personale	19

Prezentarea generala a firmei, a proiectului si a modului de comunicare si organizare

Mi-am desfasurat stagiul de practica in perioada 13 iulie – 28 august la S.C. Softex Company S.R.L. care este o companie de dezvoltare software si furnizare de servicii IT pentru companii de asigurare, banci si alte institutii financiare, avand sediu in Cluj-Napoca pe Str. Salcamului nr. 30.

Activitatea desfasurata a avut loc exclusiv online, fara prezenta fizica la sediul firmei. Comunicarea cu firma si ceilalti interni s-a realizat prin e-mail si prin intermediul aplicatiilor Discord si Google meets.

Am facut parte dintr-o echipa formata din 6 interni. Pentru a ne familiariza cu modul de lucru, de realizare de proiecte si atmosfera din cadrul unei organizatii, ni s-a oferit un proiect la care a trebuit sa lucram in echipa.

S-a cerut realizarea unei aplicatii web (site e-commerce de vanzare carti) cu urmatoarele cerinte:

- sistem logare și recuperare parolă cu doua nivele de utilizatori (cietii si administrator) + multi-factor authentication;

- **sistem vanzare carti – interfete: front-end (meniu, sigla, footer), afisare carti cu paginare, functionalitati de cautare dupa titlu/autor, sortare carti dupa titlu/autor crescator/descrescator si cautare dupa interval pret; paginile individuale ale produselor cu detalii despre acestea; cos de cumparaturi; gestiune conturi (gestiune myaccount); aprobare conturi de catre administrator cu trimitere de e-mail; **CRUD carti**, useri pentru administrator; (Eu pe parcursul stagiului de practica am lucrat pe o parte din aceste cerinte, cele boldate)**

- un sistem de gestiune versiuni preturi care permite utilizand WS-REST transmiterea pretului cartilor, pentru clientii logati, obtinut prin aplicarea unor reduceri asupra pretului de baza in functie de județul clientului, vârsta clientului, categoria cartii (ro/straina);

- sistem de rapoarte pentru administrator (top carti vandute, top cumparatori, top autori in functie de judet).

Initial a trebuit ca unul dintre noi sa isi asume functia de project manager care urma sa aiba rolul de a imparti sarcinile intre membrii echipei, de a stabili deadline-uri, de a urmări respectarea acestora si desfasurarea corecta a proiectului, de a inainta intrebarile membrilor echipei catre superiori (angajatii firmei care au fost pusi la dispozitia noastra sa ne ajute cand aveam nelamuriri). De asemenea project manager-ul a avut un rol important in coordonarea si

supravegherea realizarii version control-ului si deployment-ului proiectului asigurandu-se de asemenea ca toate configurarile sunt realizate corect pentru ca proiectul sa functioneze pe server.

Ceilalti 5 interni urmau sa ia rolul de programatori si sa realizeze sarcinile date de project manager in vederea realizarii site-ului pana la termenul limita.

Pentru dezvoltarea aplicatiei:

- pentru back-end s-a lucrat in php, in framework-ul Laravel;
- pentru partea de front-end s-a folosit: HTML, CSS, JQuery, JavaScript, Ajax;
- ca baza de date s-a folosit MySQL;

Pentru a functiona aplicatia fiecare si-a instalat pe calculatorul personal XAMPP.

Ca si IDE s-a folosit PhpStorm iar baza de date s-a realizat in Heidi.

Cu ajutorul VPN ne-am conectat la server. Pentru deployment ni s-au oferit niste credentiale prin intermediul carora ne-am conectat prin PhpStorm la server (protocol SFTP) pentru a avea posibilitatea sa uploadam fisiere si sa schimbam structura proiectului de pe server. Cu aceasta ocazie am invatat cum se realizeaza procedura de deployment in cadrul unui proiect dintr-o organizatie si comenzile de baza: compare (pentru a vedea diferentele dintre versiunea de pe calculatorul personal si cea de pe server), upload (pentru a schimba versiunea de pe server cu cea de pe calculatorul personal sau a incarca un fisier nou), download (pentru a descarca versiunea de pe server care va inlocui versiunea de pe calculatorul personal sau a descarca un fisier nou). Pentru a nu ne suprascrie codul unul altuia ni s-a explicat modul in care trebuie sa procedam cu aceste comenzi.

Pentru version control s-a folosit GIT. Cu aceasta ocazie am vazut cum se poate lucra cu mai multi oameni pe un singur proiect. Cu ajutorul comenzii merge s-a putut realiza unirea mai multor functionalitati/branch-uri realizate de coechipieri diferiti in cadrul proiectului. De asemenea am invatat cum se lucreaza cu comenzile de baza: PULL, PUSH, COMMIT.

Prezentarea activitatilor desfasurate si a contributiei personale in cadrul proiectului realizat

Stagiul de practica a durat 7 saptamani din care prima saptamana a fost organizatorica iar ultima a fost de prezentare a proiectului realizat, feedback si completare de documente.

In prima saptamana am participat in fiecare zi la sedinte pe Google meets impreuna cu tutorele de practica Aurel Filip care este directorul general al companiei si inca 2 angajati care ne-au ajutat pe parcursul stagiului de practica. In cadrul acestor intalniri din prima saptamana:

- am facut cunostinta cu membrii echipei,
- am discutat despre modul in care se va realiza comunicarea intre membrii echipei (Discord),
- ni s-au prezentat toate aplicatiile pe care urma sa le instalam necesare pentru comunicare si realizarea proiectului,
- ni s-a prezentat modul de conectare la server,
- ni s-a explicat modul de utilizare a git-ului,
- ni s-au prezentat cerintele proiectului iar in urma analizei acestora de catre fiecare membru individual ne-am comunicat nelamuririle care au fost apoi rezolvate prin raspunsurile date de tutorele de practica,
- ni s-au dat sfaturi cu privinta la modul in care trebuie sa ne organizam echipa si sa ne impartim sarcinile,
- ni s-au dat sfaturi cu privinta la modul cel mai eficient de a scrie cod (evitarea redundantei, scrierea cat mai modularizata, incorporarea a cat mai multe comentarii).

Pana in ziua a treia a saptamanii am avut project managerul ales (un coleg intern, Horatiu Spanion), sarcinile de baza impartite intre membrii echipei si ne-am propus ca prim obiectiv realizarea bazei de date si popularea acesteia pana la sfarsitul saptamanii a doua.

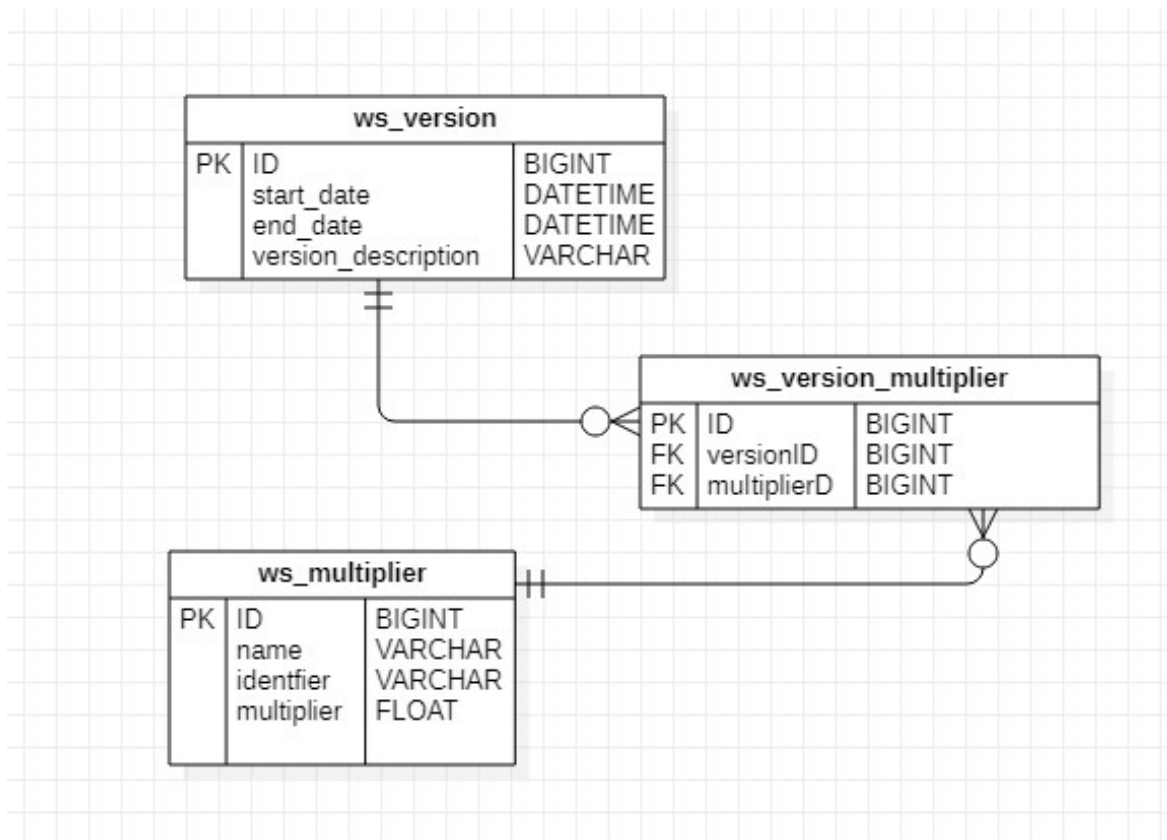
Individual am inceput sa ma uit la tutoriale pentru a invata php si laravel.

In a doua saptamana am participat in fiecare zi la sedinte mai scurte pe Google meets cu tutorele de practica in care i-am prezentat acestuia modul in care se desfasura proiectul, cum ne-am organizat, cum ne-am impartit sarcinile intre noi si deadline-urile orientative. Aceste sedinte erau urmate de o sedinta pe discord doar intre membrii echipei.

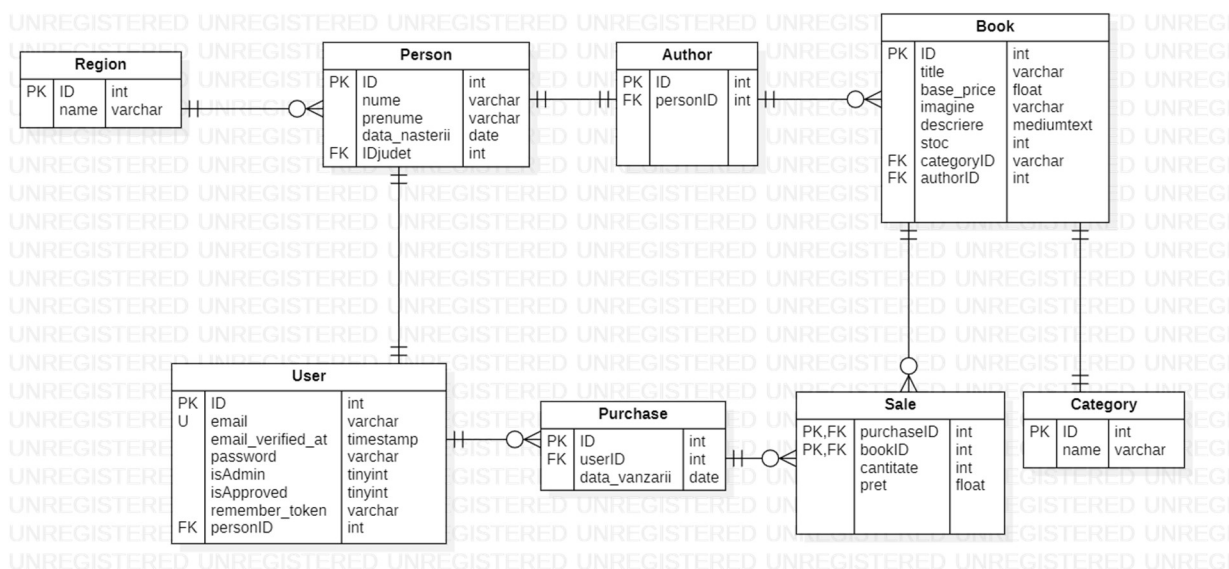
Baza de date trebuia sa fie formata din doua parti: o parte cu tabele ce tin de aplicatia web (useri, carti , vanzari) si o parte cu tabele ce tin de web service-ul care urma sa calculeze

pretul produselor pentru fiecare user. Pentru realizarea design-ului bazei de date fiecare a realizat cate un design propriu pe care l-a prezentat ulterior membrilor echipei: 3 membrii au lucrat la realizarea unui design pentru prima parte a bazei de date iar 2 membrii pentru a doua parte. In urma prezentarii design-ului creat de fiecare membru si discutarii modificarilor care se pot face s-a ajuns la un design final pentru fiecare parte a bazei de date.

Design final pentru partea de Web service:



Design final al bazei de date pentru aplicatia principala la care am lucrat eu:



Acest design final l-am realizat pana in ziua 3 a saptamanii. Pana la sfarsitul saptamanii eu am realizat codul pentru crearea si popularea primei parti a bazei de date prin migration si seeding. Migrarile sunt un tip de version control pentru baza de date oferit de laravel. Acestea permit unei echipe sa modifice schema bazei de date si sa ramana la curent cu starea actuala a schemei. Prin rularea comenzii "php artisan migrate" se ruleaza codul scris la migrari, disponibil tuturor prin git, care creaza in baza de date locala a fiecaruia tabelele.

Exemplu de clasa de migrare a unei tabele scrisa de mine:

```

class CreateBooksTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create( table: 'books', function (Blueprint $table) {
            $table->id( column: 'id');
            $table->unsignedBigInteger( column: 'authorID');
            $table->foreign( columns: 'authorID')->references( columns: 'id')->on( table: 'authors')->onDelete( action: 'cascade')->onUpdate( action: 'cascade');
            $table->string( column: 'title');
            $table->float( column: 'base_price');
            $table->string( column: 'image');
            $table->integer( column: 'stoc');
            $table->mediumText( column: 'descriere');
            $table->unsignedBigInteger( column: 'categoryID');
            $table->foreign( columns: 'categoryID')->references( columns: 'id')->on( table: 'categories')->onDelete( action: 'cascade')->onUpdate( action: 'cascade');
            $table->timestamps();
        });
    }

    public function down()
    {
        Schema::dropIfExists( table: 'books');
    }
}
    
```

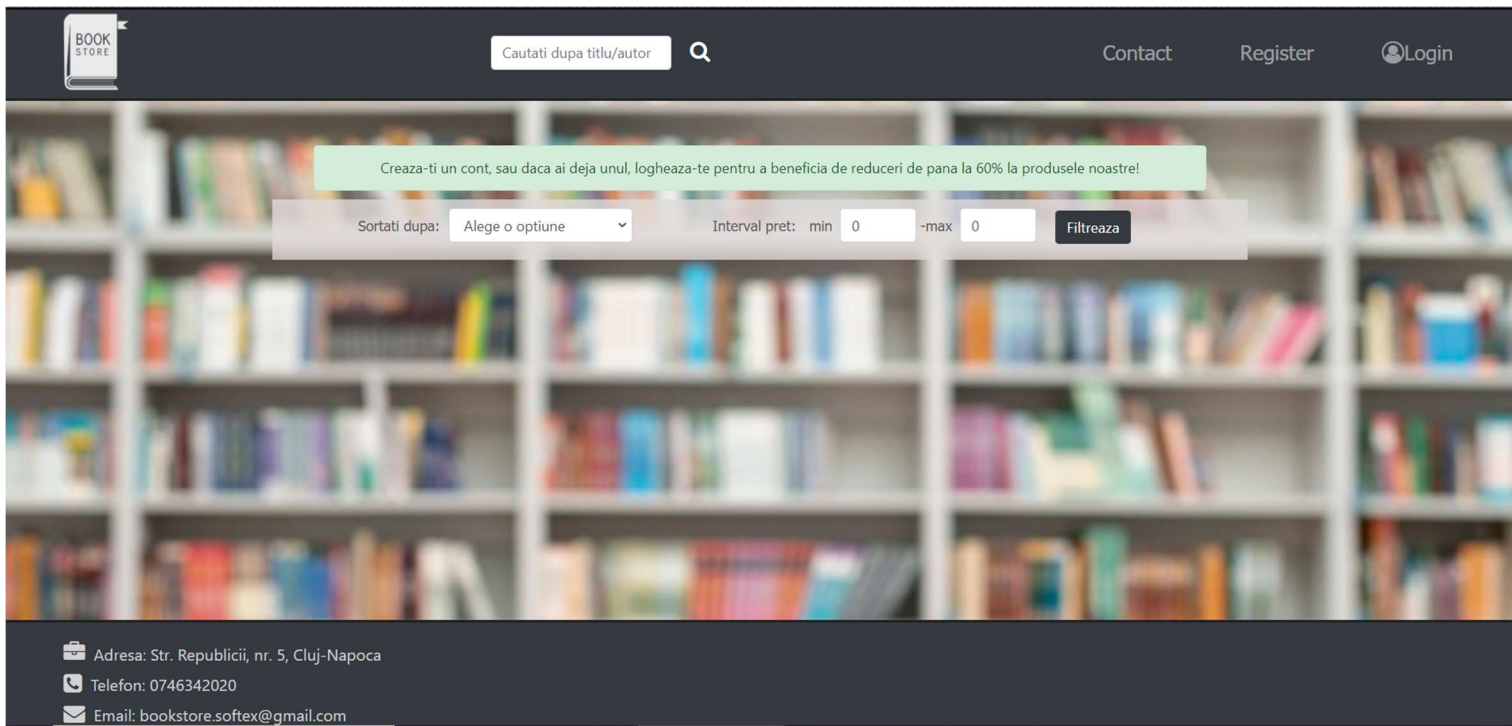
(functia up este apelata la rularea comenzii php artisan migrate si creaza tabelele in baza de date, functia down este apelata la rularea comenzii php artisan reset, aceasta sterge tabelele din baza de date)

Pentru popularea bazei de date laravel ofera un tip de clase: seed classes. Acestea sunt folosite pentru popularea rapida a bazei de date si pentru ca toti membrii echipei sa aiba aceleasi date de test.

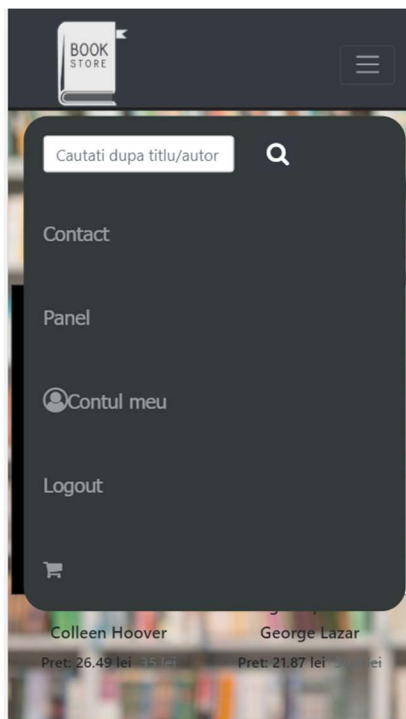
Exemplu de clasa seed a unei tabele scrisa de mine:

```
class PeopleSeeder extends Seeder
{
    public function run()
    {
        $items = [
            ['id' => 1, 'nume' => 'Manson', 'prenume'=>'Mark', 'data_nasterii'=>'1984-03-09', 'judetID'=> 1],
            ['id' => 2, 'nume' => 'Owens', 'prenume'=>'Delia', 'data_nasterii'=>'1949-04-04', 'judetID'=> 13],
            ['id' => 3, 'nume' => 'Kiyosaki', 'prenume'=>'Kiyosaki', 'data_nasterii'=>'1947-04-08', 'judetID'=> 1],
            ['id' => 4, 'nume' => 'Boyne', 'prenume'=>'John', 'data_nasterii'=>'1971-04-30', 'judetID'=> 17],
            ['id' => 5, 'nume' => 'Voss', 'prenume'=>'Chris', 'data_nasterii'=>'1988-03-31', 'judetID'=> 13],
            ['id' => 6, 'nume' => 'Zusak', 'prenume'=>'Markus', 'data_nasterii'=>'1975-06-23', 'judetID'=> 14],
            ['id' => 7, 'nume' => 'Michaelides', 'prenume'=>'Alex', 'data_nasterii'=>'1977-09-04', 'judetID'=> 1],
            ['id' => 8, 'nume' => 'Goleman', 'prenume'=>'Daniel', 'data_nasterii'=>'1946-03-07', 'judetID'=> 3],
            ['id' => 9, 'nume' => 'Obama', 'prenume'=>'Michelle', 'data_nasterii'=>'1964-01-17', 'judetID'=> 16],
            ['id' => 10, 'nume' => 'Chapman', 'prenume'=>'Gary', 'data_nasterii'=>'1938-01-10', 'judetID'=> 14],
            ['id' => 11, 'nume' => 'Brown', 'prenume'=>'Brene', 'data_nasterii'=>'1965-11-18', 'judetID'=> 15],
            ['id' => 12, 'nume' => 'Lugand', 'prenume'=>'Martin', 'data_nasterii'=>'1979-07-23', 'judetID'=> 41],
            ['id' => 13, 'nume' => 'Kishimi', 'prenume'=>'Ichiro', 'data_nasterii'=>'1956-07-23', 'judetID'=> 31],
            ['id' => 14, 'nume' => 'Carnegie', 'prenume'=>'Dale', 'data_nasterii'=>'1888-11-24', 'judetID'=> 4],
            ['id' => 15, 'nume' => 'Maxwell', 'prenume'=>'John', 'data_nasterii'=>'1947-02-20', 'judetID'=> 13],
            ['id' => 16, 'nume' => 'Lippincot', 'prenume'=>'Rachael', 'data_nasterii'=>'1980-07-23', 'judetID'=> 1],
            ['id' => 17, 'nume' => 'Moyes', 'prenume'=>'Jojo', 'data_nasterii'=>'1969-08-04', 'judetID'=> 13],
            ['id' => 18, 'nume' => 'Alcott', 'prenume'=>'Louisa', 'data_nasterii'=>'1832-11-29', 'judetID'=> 10],
            ['id' => 19, 'nume' => 'Creanga', 'prenume'=>'Ion', 'data_nasterii'=>'1837-03-01', 'judetID'=> 20],
            ['id' => 20, 'nume' => 'Setepys', 'prenume'=>'Ruta', 'data_nasterii'=>'1967-11-19', 'judetID'=> 30],
            ['id' => 21, 'nume' => 'Tolle', 'prenume'=>'Eckhart', 'data_nasterii'=>'1948-02-16', 'judetID'=> 3],
            ['id' => 22, 'nume' => 'Frankl', 'prenume'=>'Viktor', 'data_nasterii'=>'1905-03-26', 'judetID'=> 1],
            ['id' => 23, 'nume' => 'Black', 'prenume'=>'Holly', 'data_nasterii'=>'1971-11-10', 'judetID'=> 8],
            ['id' => 24, 'nume' => 'Atwood', 'prenume'=>'Margarete', 'data_nasterii'=>'1939-11-18', 'judetID'=> 4],
        ]
    }
}
```

Din saptamana a treia nu am mai facut sedinte pe google meets cu tutorele de prctica (doar in saptamana a saptea, la prezentarea proiectului ne-am mai intalnit cu el) insa incepand din aceasta saptamana pana la sfarsitul stagiului am participat la sedinte zilnice cu echipa pe discord. In aceasta saptamana eu am realizat front-end-ul pentru pagina. Utilizand bootstrap am creat un layout, care este folosit de toti membrii echipei pentru paginilor lor individuale, care cuprinde o poza de fundal, un meniu de navigare responsive cu o sigla care duce la pagina "home" si un footer cu date de contact fictive.

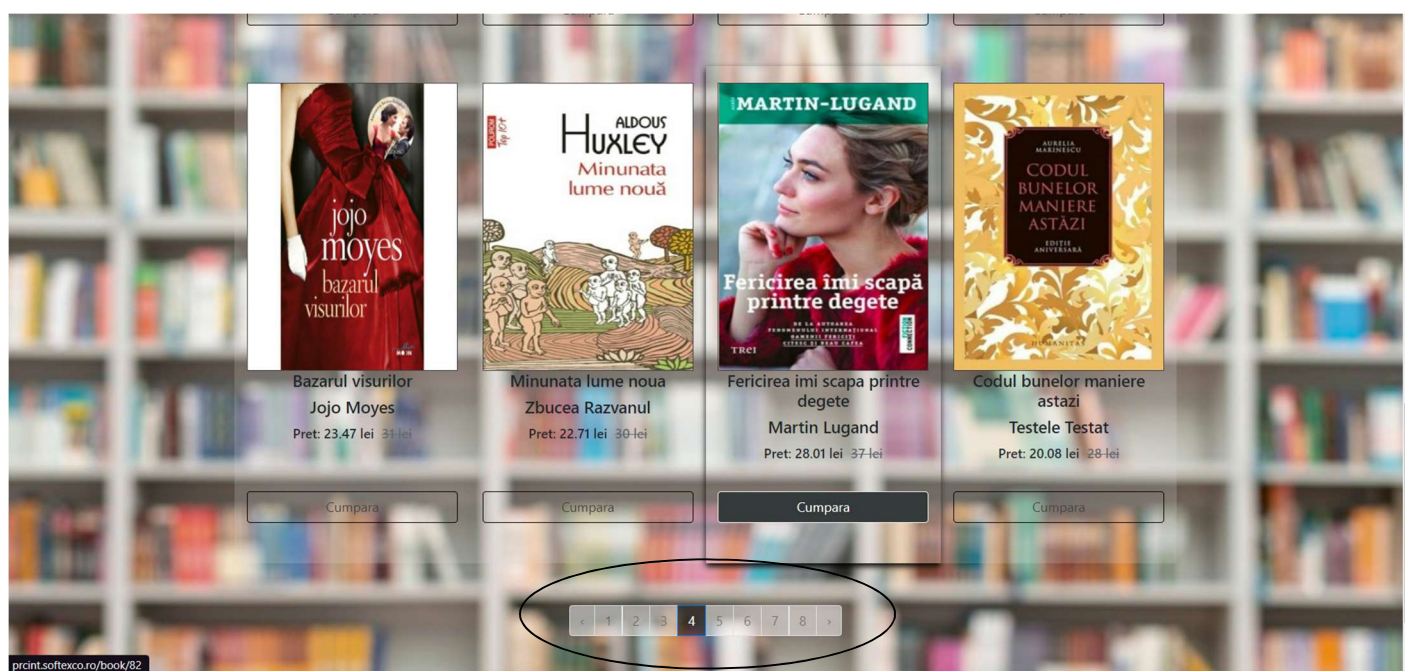


(prezentare layout front-end de pe ecranele mari: in momentul in care nu esti logat nu apar multe butoane in meniul + footer)



(in aceasta poza se observa cum se modifica meniul cand accesezi site-ul de pe un telefon mobil, acesta poate fi accesat prin intermediul unui drop-down, de asemenea se observa aparitia mai multor butoane in acest caz datorita faptului ca am fost logata pe site ca si administrator. Insa daca te loghezi de pe un cont de user nu apare butonul "Panel" care duce la pagina "Admin panel" cu mai multe functionalitati pentru admin)

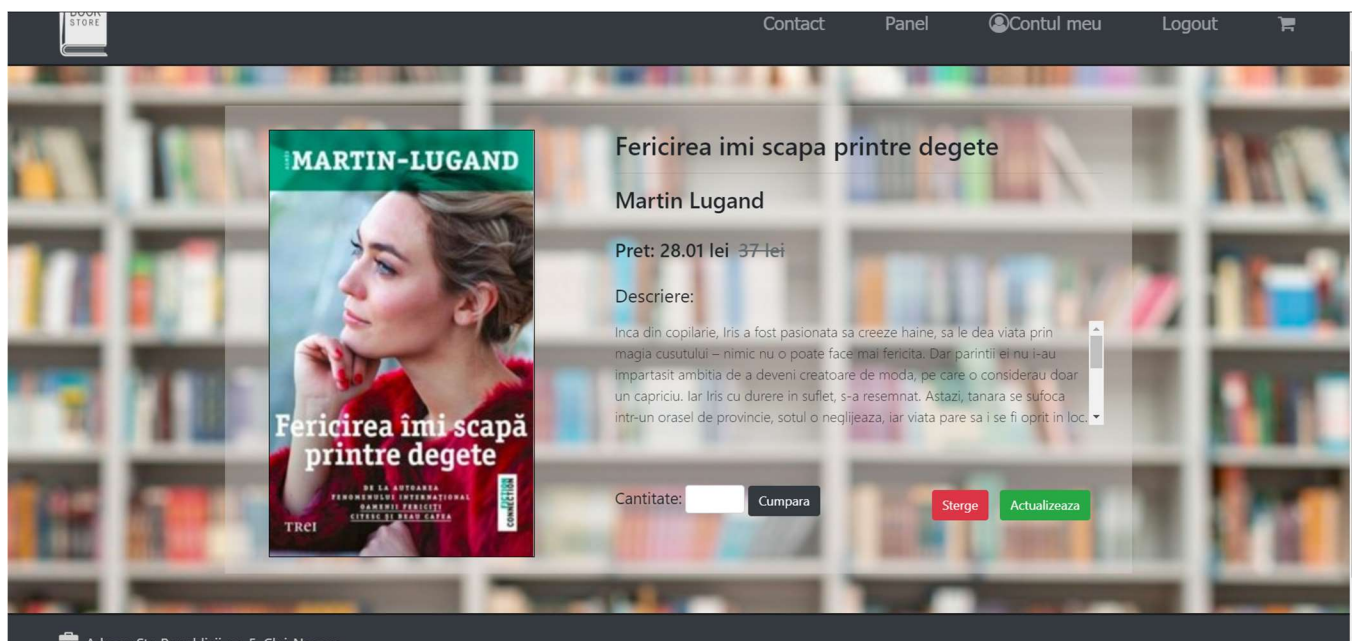
Tot in aceasta saptamana am realizat afisarea pe pagina principala a cartilor din baza de date cu prezentarea detaliilor acestora (imagine, titlu, autor, pret de baza) si cu paginare. In momentul in care un user se logheaza, in loc de pretul de baza, i se afiseaza pretul generat de web service, acesta urma sa il apelez in saptamana a cina prin transmiterea parametrilor necesari (judet,varsta user + categorie carte) functiei create de un alt membru al echipei care inmultea pretul de baza al cartii respective cu niste multiplicatori luati din baza de date. Am realizat de asemenea si pagina individuala pentru fiecare carte care apare cand apesi pe imaginea unei carti sau pe butonul “Cumpara” din pagina principala a site-ului.



(paginare + afisare carti)



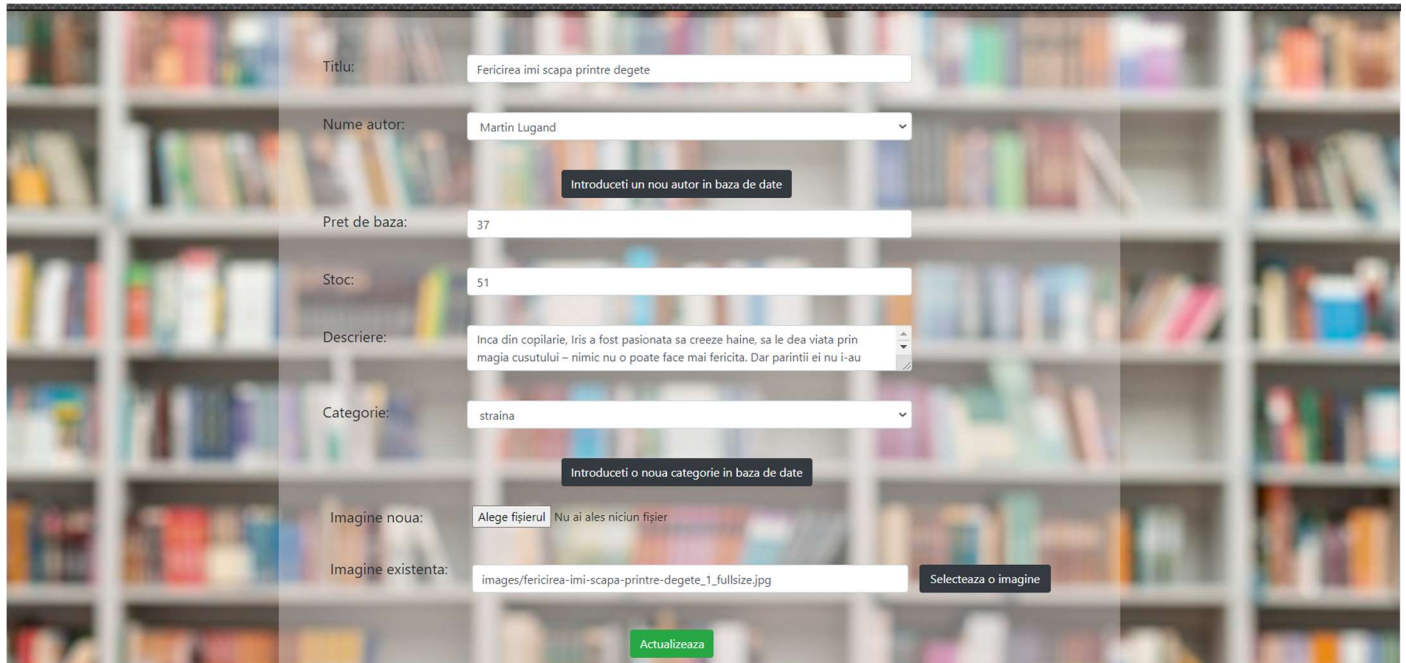
(afisare carti + paginare pe telefoane mobile)



(Pagina produsului)

Butoanele “Sterge” si “Actualizeaza” sunt vizibile doar pentru admin pentru a permite acestuia sa stearga o carte din baza de date sau sa actualizeze datele ei (functionalitate creata in saptamana 4). In cazul in care un client nelogat incearca sa apese butonul “Cumpara” acesta va fi trimis catre pagina de Login, in caz contrar, produsul va fi adaugat in cosul de cumparaturi cu cantitatea selectata.

In saptamana a patra am realizat functionalitatea de CRUD carti pentru administrator. Aceasta consta in butoanele “Sterge”, care sterge cartea din baza de date, si “Actualizeaza”, care trimite administratorul catre un formular de actualizare a cartii, care sunt puse in pagina fiecarui produs individual si un formular de creare a unei carti noi care poate fi accesat din “Admin panel”.

The image shows a web form for updating a book, overlaid on a background of bookshelves. The form contains the following fields and controls: 'Titlu:' with a text input containing 'Fericirea imi scapa printre degete'; 'Nume autor:' with a dropdown menu showing 'Martin Lugand' and a button 'Introduceti un nou autor in baza de date'; 'Pret de baza:' with a text input containing '37'; 'Stoc:' with a text input containing '51'; 'Descriere:' with a text area containing 'Inca din copilarie, Iris a fost pasionata sa creeze haine, sa le dea viata prin magia cusutului – nimic nu o poate face mai fericita. Dar parintii ei nu i-au'; 'Categorie:' with a dropdown menu showing 'straina' and a button 'Introduceti o noua categorie in baza de date'; 'Imagine noua:' with a button 'Alege fisierul' and a text input 'Nu ai ales niciun fisier'; 'Imagine existenta:' with a text input showing a file path 'images/fericirea-imi-scapa-printre-degete_1_fullsize.jpg' and a button 'Selecteaza o imagine'; and a green 'Actualizeaza' button at the bottom.

(formularul de actualizare a unei carti cu date precompletate ale cartii)

Administratorul are posibilitatea de a alege autori, categorii, imagini existente deja in baza de date, in cazul autorilor si categoriilor prin intermediul unui dropdown iar in cazul imaginilor printr-un buton: “Selecteaza o imagine” care determina afisarea imaginilor existente in aplicatie iar la apasarea unei imagini se introduce in inputul “Imagine existenta” calea catre imaginea selectata, sau sa introduca un autor, categorie, imagine noua. Butonul “Introduceti un nou autor in baza de date” duce administratorul catre un formular de creare a unui nou autor, butonul “Introduceti o noua categorie in baza de date” duce administratorul catre un formular de creare a unei noi categorii, iar pentru adaugarea unei imagini noi se selecteaza “alege fisierul” si se alege imaginea de pe calculatorul personal.

Nume autor:

Pret de baza:

Stoc:

Descriere:

Categorie:

Introduceti o noua categorie in baza de date

Introduceti o imagine!

(dropdown pentru selectare autor existent, insiruire autori in ordine alfabetica – similar pentru categorii)

Imagine existenta:

Ascunde imaginile

Actualizeaza

(posibilitatea alegerii unei imagini existente)

Introduceti o imagine!

Imagine noua: Nu ai ales niciun fisier

Imagine existenta:

Selecteaza o imagine

Actualizeaza

(nu se permite trimiterea formularului fara selectarea unei imagini)

Pret de baza:

Introduceti o valoare numerica!

(se verifica daca datele introduse sunt de tipul potrivit)

Form for adding a new author to the database. The form is displayed over a blurred background of bookshelves. It contains the following fields:

- Nume:
- Prenume:
- Data nasterii:
- Origine autor:

Below the fields is a green button labeled "Creaza".

(formular
introducere
autor nou
in baza de
date)

Form for creating a new category. The form is displayed over a blurred background of bookshelves. It contains the following field:

- Nume categorie:

Below the field is a green button labeled "Creaza".

(formularul
pentru
introducerea
unei noi
categorii)

Formularul pentru crearea unei carti noi arata la fel ca cel de actualizare insa nu are date precompletate. Toate aceste formulare sunt de asemenea responsive. Am implementat atat partea de front-ent cat si cea de back-end pentru aceasta functionalitate.

Tot in saptamana a patra am implementat functionalitatea de search. Aceasta presupune cautarea cartilor in functie de titlu/autor/provenienta autor(strain/roman) si combinatii de nume autor-prenume autor-titlu. Initial se verifica daca inputul introdus se potriveste cu date existente in baza de date: fie cu titlul unei carti, numele unui autor, prenumele unui autor sau combinatii a acestor 3 elemente introduse in orice ordine. Daca nu se gaseste nici un rezultat, acest lucru se poate datora faptului ca clientul a gresit sintaxa unui cuvant din titlu sau a introdus numele autorului gresit. In acest caz se analizeaza fiecare cuvant format din cel putin 4 litere din input individual si se verifica daca acesta se potriveste cu numele unui autor sau cu titlurile din baza de date si se afiseaza cartile gasite.

```

public function search(Request $request)
{
    $sstr = $request->searchstr;

    //verifica daca inputul se potriveste cu titlul unei carti, nume sau prenumele unui autor,
    // numele intreg al unui autor sau combinatii de nume,prenume,titlu si se returneaza exact cartea ceruta (sau cartile autorului cerut)

    $carte = Book::select('books.id', 'authorID', 'title', 'base_price', 'image', 'stoc', 'descriere', 'categoryID')
        ->join('authors', 'books.authorID', '=', 'authors.id')
        ->join('people', 'authors.personID', '=', 'people.id')->where('title', 'LIKE', '%' . $sstr . '%')
        ->join('regions', 'regions.id', '=', 'judetID')
        ->orWhere('people.nume', 'LIKE', '%' . $sstr . '%')
        ->orWhere('people.prenume', 'LIKE', '%' . $sstr . '%')
        ->orWhere('regions.name', 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT_WS(" ", nume, prenume)'), 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT_WS(" ", prenume, nume)'), 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT(prenume, " ",nume, " ",title)'), 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT(nume, " ",prenume, " ",title)'), 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT(title, " ",prenume, " ",nume)'), 'LIKE', '%' . $sstr . '%')
        ->orWhere(DB::raw( value: 'CONCAT(title, " ",nume, " ",prenume)'), 'LIKE', '%' . $sstr . '%')
        ->paginate(12);

    //in cazul in care clientul face o greseala in introducerea titlului, numelui unei carti/autor
    // atunci se analizeaza fiecare cuvint din string si se afiseaza e autorului similar cu unele cuvinte cautate de lungime

    if ($carte->isEmpty()) {
        $sarray = preg_split( pattern: '/\s+/', $sstr, limit: -1, flags: PREG_SPLIT_NO_EMPTY);
        $query = Book::select('books.id', 'authorID', 'title', 'base_price', 'image', 'stoc', 'descriere', 'categoryID')
            ->join('authors', 'books.authorID', '=', 'authors.id')
            ->join('people', 'authors.personID', '=', 'people.id')
            ->join('regions', 'regions.id', '=', 'judetID');
        foreach ($sarray as $s) {
            if (strlen($s) > 3) { //se evita afisarea unui numar mare de carti irelevanta prin excluderea cuvintelor <4
                foreach ($sarray as $ss) {
                    if (strlen($ss) > 3) { //se evita afisarea unui numar mare de carti irelevanta prin excluderea cuvintelor <4
                        $query->orWhere('people.nume', 'LIKE', '%' . $s . '%')
                            ->orWhere('title', 'LIKE', '%' . $s . '%')
                            ->orWhere('people.prenume', 'LIKE', '%' . $s . '%')
                            ->orWhere('regions.name', 'LIKE', '%' . $s . '%');
                    }
                }
            }
        }
        $carte = $query->paginate(12);
    }

    if (Auth::user()) {
        $personID = Auth::user()->personID;
        $person = Person::where('id', '=', $personID)->get();
        $region = Region::where('id', '=', $person[0]['judetID']->get();

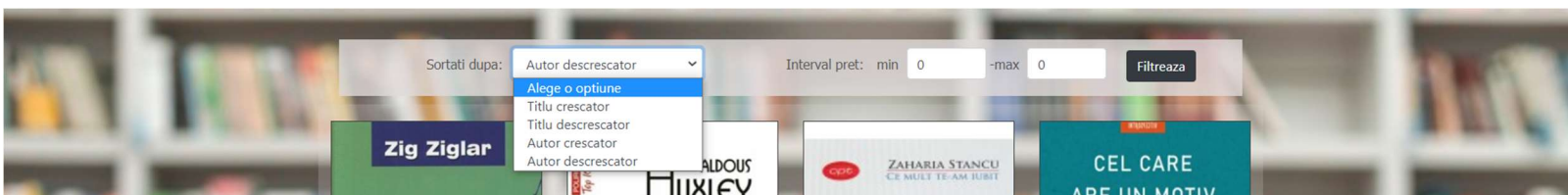
        $priceController = new PriceController();

        $b = [];
        foreach ($carte as $bk) {
            $category = ($bk->categorie['name'] == 'straina') ? 'OTHER' : 'RO';
            $b[] = array('id' => $bk['id'], 'base_price' => $bk['base_price'], 'category' => $category);
        }
        $age = Carbon::now()->diffInYears($person[0]['data_nasterii']);
        $prices = $priceController->getBundlePrices($region[0]['name'], $age, $b);
    }

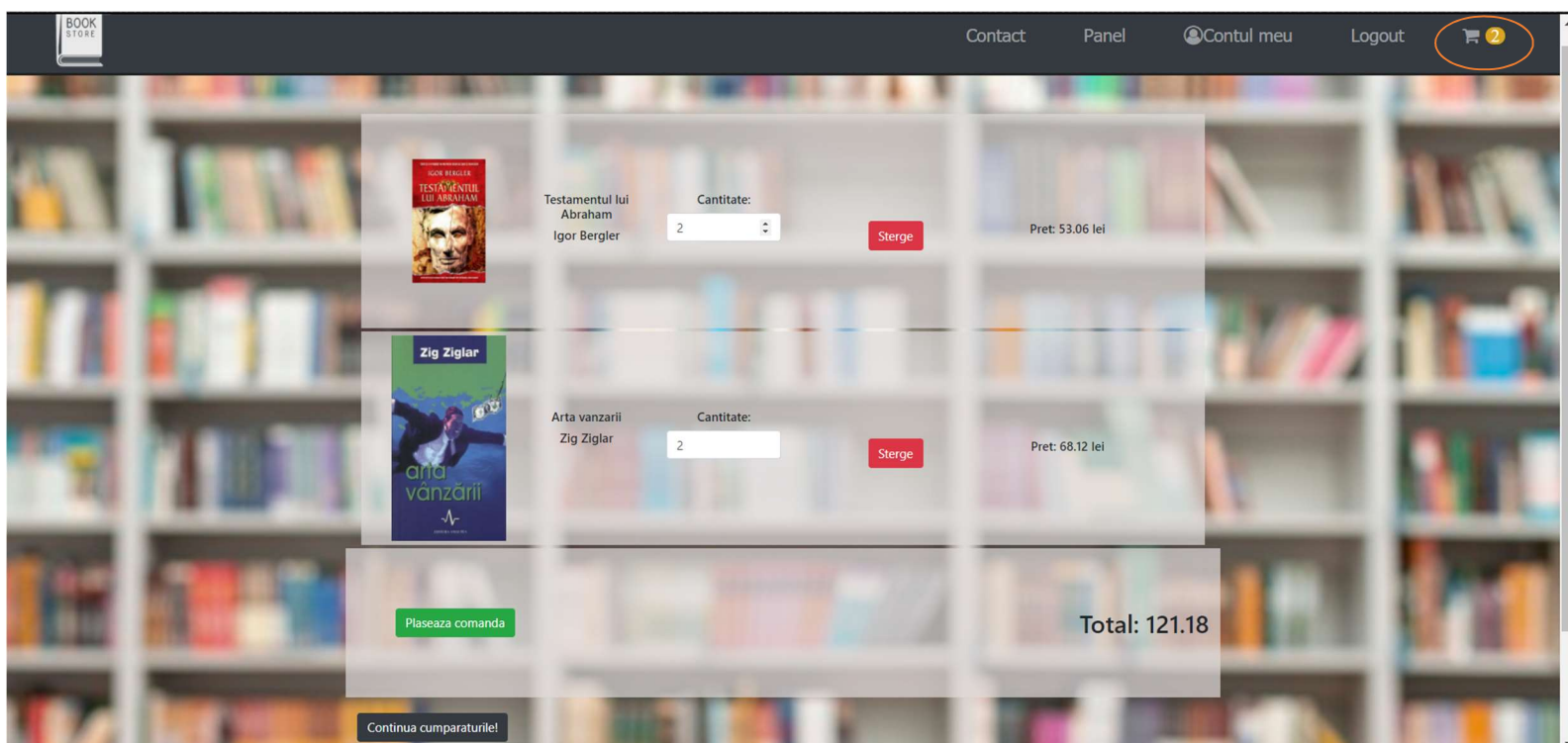
    return view( view: 'welcome', [
        'carte' => $carte,
        'person' => $person ?? "emptyPerson",
        'region' => $region ?? "emptyRegion",
        'prices' => $prices['response'] ?? "0"
    ])
}

```

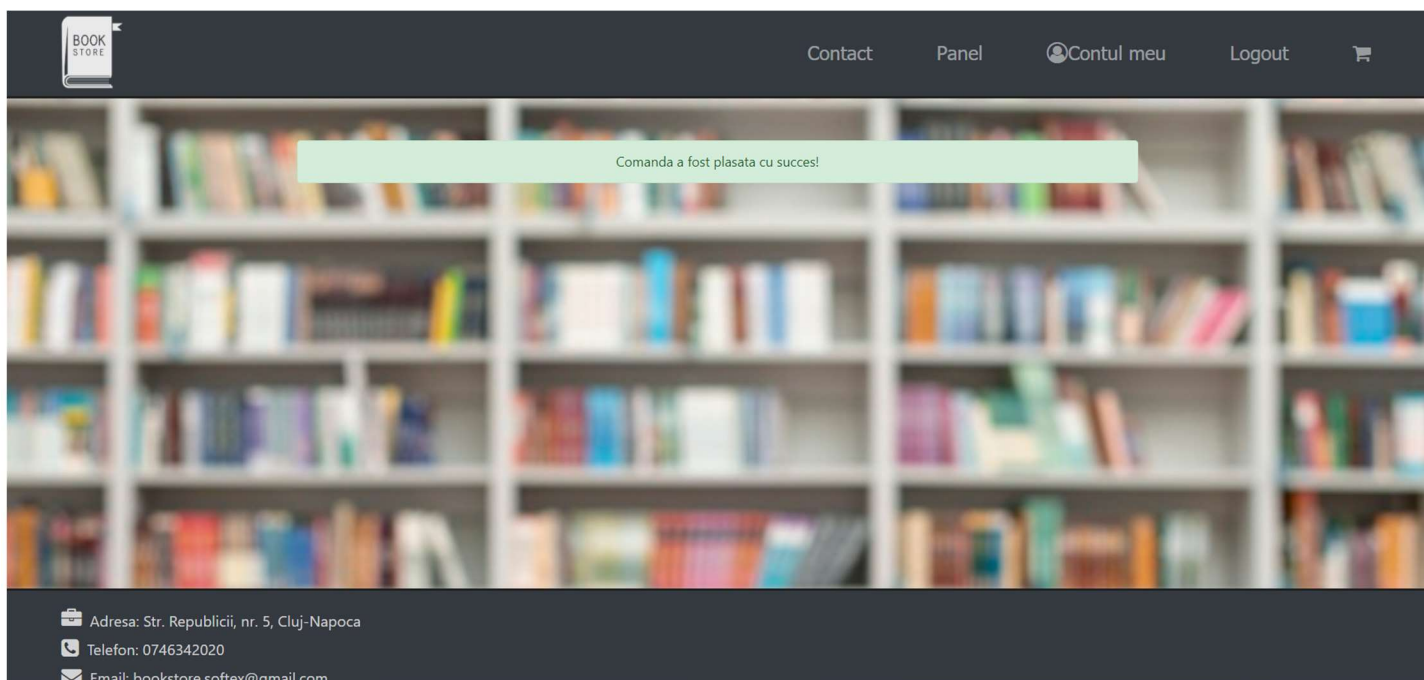

In saptamana a cincea am implementat functionalitatea de sortare a cartilor dupa autor/titlu crescator/descrescator. Clientul poate selecta modu de sortare dorit prin intermediul unui dropdown menu. Am mai implementat si cautarea cartilor dupa interval pret. In cazul in care clientul nu este logat cautarea se va face in functie de pretul de baza iar daca clientul este logat in functie de pretul obtinut prin WS.



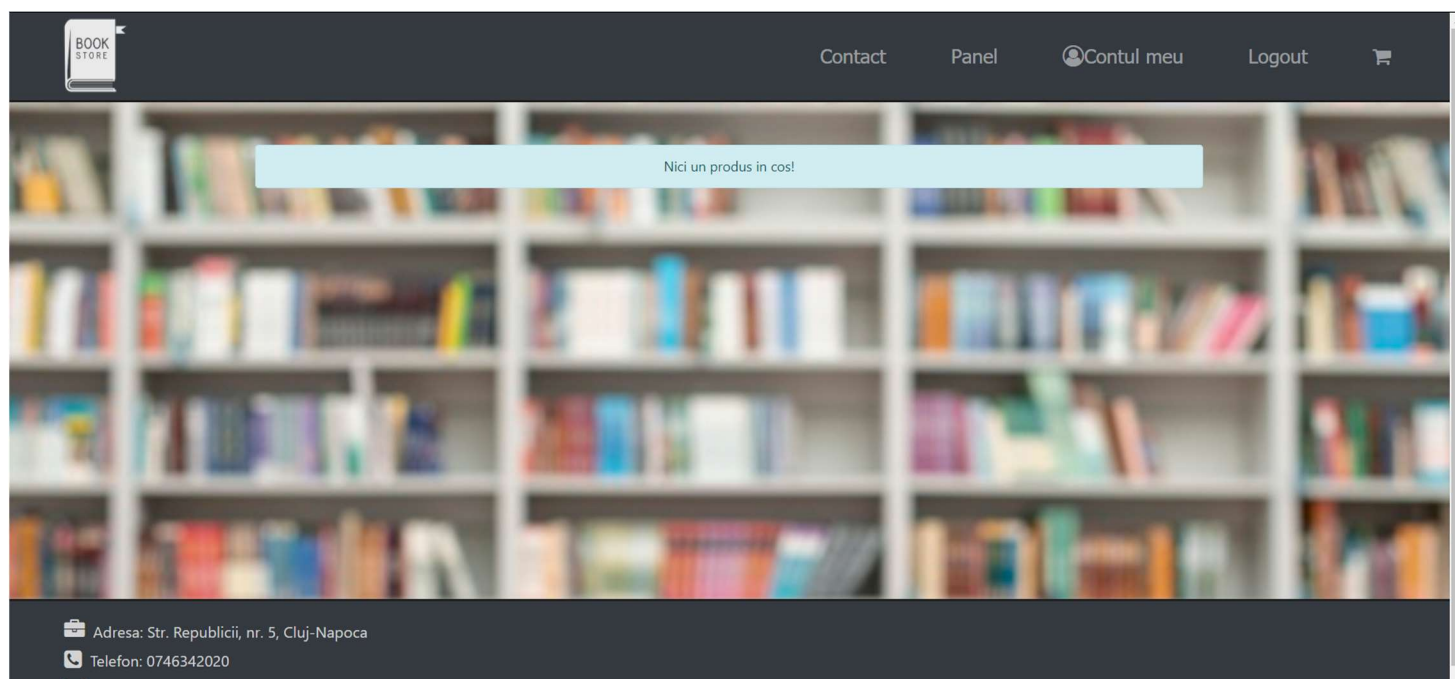
Dupa realizarea acestei functionalitati am apelat web service-ul care a fost finalizat de un coleg de echipa in toate locurile necesare in proiect dupa care am implementat cosul de cumparaturi al site-ului.



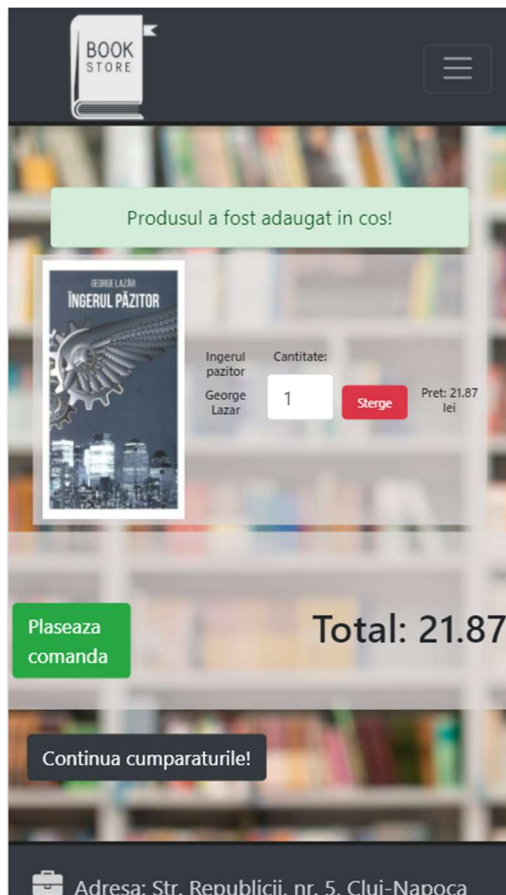
In meniul, in coltul din dreapta se poate observa numarul de produse din cos. In cosul de cumparaturi se pot sterge produse, se poate schimba cantitatea acestor produse in mod dinamic. Clientul are la dispozitie si un buton de plasare a comenzii si unul de continuare a cumparaturilor care il duce inapoi la pagina principala cu produse.



(dupa plasarea comenzii se afiseaza un mesaj de succes)



(daca se acceseaza cosul de cumparaturi gol)



(cosul de cumparaturi pe ecranele mici)

In saptamana a sasea:

- am incarcat proiectul pe server,

- am testat aplicatia,

- am finisat aplicatia, am adaugat mici imbunatatiri la fiecare functionalitate ca sa fie proiectul cat mai prezentabil (de exemplu am implementat impiedicarea retrimiterii formularelor de plasare comanda, stergere prods din cosul de cumparaturi, adaugare produs cand se da refresh la pagina dupa realizarea initiala a acestor actiuni).

Unele functionalitati care mergeau bine pe local (de exemplu actualizarea dinamica a cantitatii produselor din cosul de cumparaturi) nu functionau pe server insa am rezolvat si aceste probleme.

In saptamana a saptea luni , am anuntat ca proiectul este gata sa fie prezentat si am fost anuntati ca marti 3 dintre noi vor prezenta functionalitatile realizate si contributiile personale la proiect si miercuri vor prezenta ceilalti 3. Luni am prezentat proiectul unui angajat al firmei, care ne-a mai ajutat pe parcursul realizarii proiectului raspunzandu-ne la intrebari si nelamuriri, pentru a primi un ultim feedback inainte de prezentarea finala. Marti si miercuri am prezentat pe Google meets prin sharescreen proiectul final tutorelui de practica. Vineri am avut o ultima sedinta in care am discutat despre experienta pe care am avut-o in aceste sapte saptamani si in care am primit rapoartele de evaluare de la tutore si alte documente necesare.

Concluzii personale

Mi-a placut foarte mult experienta din cele sapte saptamani ale stagiului de practica. Am avut ocazia in aceasta perioada atat sa imi valorific, sa aprofundez si sa pun in practica cunostintele de front-end (HTML, CSS, JavaScript, responsive web design) si de baze de date (design baze de date, SQL) pe care le-am dobandit la facultate cat si sa invat si sa pun in practica cunostinte noi legate de back-end (Php- framework Laravel) si de utilizare a sistemelor de version control. Mi-am imbunatatit si abilitatile de comunicare si lucru in echipa prin participarea la sedintele zilnice si colaborarea cu membrii echipei. De asemenea, aceasta experienta mi-a dat ocazia nu doar sa ma familiarizez cu proiectele IT si atmosfera dintr-o organizatie ci si sa vad cum decurge un interviu de angajare.