**Wantsome 2021**

**E-commerce website**

**Front-end course**

**Student: Rusu Andrei Sebastian**

**Mentor:Popușoi Marius-Cătălin**

**Iasi 2021**

Cuprins

[Capitolul 1. Scop 3](#_Toc81743378)

[Capitolul 2. Tehnologii folosite 4](#_Toc81743379)

[2.1 Html 4](#_Toc81743382)

[2.2 Css/Sass 5](#_Toc81743383)

[2.3 Javascript 7](#_Toc81743384)

[Capitolul 3. Structura aplicatiei 8](#_Toc81743385)

[Capitolul 4. Descrierea aplicatiei 9](#_Toc81743389)

[4.1 Login/Signup 9](#_Toc81743390)

[4.2 Pagina de prezentare](#_Toc81743391) 10

[4.2 Pagina de shop 12](#_Toc81743391)

[4.2 Pagina de blog 14](#_Toc81743391)

[4.2 Pagina de features 15](#_Toc81743391)

[4.2 Pagina de about 16](#_Toc81743391)

[4.2 Pagina de cart 17](#_Toc81743391)

[Bibliografie 18](#_Toc81743393)

**Capitolul 1. Scop**

Scopul principal al proiectului este implementarea unei aplicatii web prin intermediul careia utilizatorii pot achizitiona diferite produse avand legatura cu industria cafelei.

Piaţa cafelei în România este estimată la circa 435 milioane euro în 2016, o valoare în creştere fată de anii anteriori, determinată de o putere de cumpărare mai mare a populaţiei, de eliminarea accizei în 2016 şi de dezvoltarea lanţurilor de cafenele în ultimii ani. Piaţa de cafea a crescut constant în ultimii ani, atât în volum, cât şi în valoare.

Peste 90% dintre români beau cafea acasă sau în oraş în mod regulat. Consumul de cafea continuă să aibă şi va avea încă mult timp de acum înainte foarte mult loc de creştere. Cafeaua este considerată cea mai consumată băutură din țările europene. Mai mult, aceasta este a doua cea mai tranzacționată marfă  de pe piată, după industria petrolului.

Europa are cel mai mare consum de cafea pe cap de locuitor din lume. Italia, Germania, Franța și Marea Britanie au cel mai mare consum total de cafea, în timp ce Finlanda, Austria, Olanda și Germania au cel mai mare consum pe cap de locuitor.

Valoarea pieței de cafea din Europa este de așteptat să înregistreze un CAGR(Rata anuală de creștere compusă) de 5,4% și să atingă o valoare de piață de milioane de dolari, în perioada de prognoză, până în 2022. Aproximativ 155 de milioane de pungi de cafea au fost consumate la nivel global în 2016. În America, aproape 70% dintre consumatori beau cel puțin două cești de cafea pe zi. Potrivit unui sondaj național realizat de Nielsen Scarborough în primăvara anului 2016, peste 36 de milioane de americani au vizitat o cafenea în ultima lună.

**Capitolul 2. Tehnologii folosite**

**2.1 Html**

**Html** sau **HyperText Markup Language** este unul dintre cele mai vechi **limbaje de marcare web**. Acesta sta la baza crearii unui site web. Cu alte cuvinte, inainte de a invata alte limbaje de programare trebuie sa cunosti cel putin **HTML**.

Un fisier HTML este format din *elemente*. Elementele sunt definite folosind o serie de ***opening*** *(<>)* si ***closing*** *(</>)* tags.

Fiecare element poate avea multiple **atribute** prin care putem controla comportamentul elementului.

De obicei, relatiile intre elemente sunt specificate prin relatii de rudenie. Astfel putem avea elemente de tip *parinte*, *copil* sau *frate*.

Elementul special *head* a unui document HTML este acel element care nu este vizibil si contine informatii precum *titlul* paginii, legaturi catre fisiere de stilizare de tip CSS, link-uri catre *favicon*-uri custom si alte tipuri de metadata despre pagina (tipul de caractere folosit, autori, etc).

Elementul HTML *<body>* reprezinta contentul afisat de pagina noastra web.

Toate elementele care vrem sa fie vizibile si cu care utilizatorii pot interactiona trebuie sa se afle in aceasta sectune a paginii noastre.

Principalele elemente in **HTML** sunt**<html>, <head>, <title>** si <***body>***. Iata si un exemplu simplu de construire a unei pagini **HTML**.

Pe langa atributul align putem folosi pentru a alinia elementele HTML urmatoarele valori: left, right center si justify.

Alte tipuri de titluri in **HTML**: h2, h3, h4, h5, h6. Unde h1 este cel mai mare si h6 cel mai mic.

<br /> este un salt in linie in ***HTML*** - spatiul intre br si slesh face diferenta intre cod ***valid XHTML*** si **HTML 4.1** de exemplu.

<hr /> se foloseste pentru a desena o linie orizontala in ***HTML*** - spatiul intre hr si slesh face diferenta intre ***cod valid XHTML*** si **HTML 4.1** de exemplu..

**2.2 Css/Sass**

**Cascading Style Sheets** pe num ele lui mic **CSS** se foloseste pentru a personaliza tagurile HTML. In principiu HTML a fost conceput pentru a marca elementele unei pagini:

Cu ajutorul **CSS**-ului putem crea pagini web simple dar si complexe folosind efecte diverse. Cu **CSS** putem stabili culoarea, marimea si fontul textului, deasemenea putem crea un **layout** (un suoport pentru elementele HTML) personalizat adaugand margine culoare sau imagine de fond si multe altele. In concluzie **CSS** te poate scapa de multe batai de cap, ajutandute in a mentine codul html cat mai simplu si mai ordonat.

Odata cu introducerea HTML 3.2 au fost introduse si atributele de personalizare a tag-urilor precum "font", "color" etc. A fost atunci cand limbajul de programare HTML a devenit greoi. Fiecare pagina a websitului trebuia luata separat si modificate proprietatile elementelor principale.

Aceasta problema a fost rezolvata in versiunea 4.0 a HTML-ului. Toate atributele de personalizare au fost scoase si salvate intr-un fisier extern cu extensia ".css".

In felul acesta modificand un singur fisier putem schimba forma in care sunt afisate toate paginile unui website. Putem schimba culoarea textului, fontul, marimea, putem personaliza div-uri, formulare si multe altele.

Foile de stiluri contin mai multe elemente de urmatoarele tipuri:

* **selectori:**  folositi pentru a selecta un anumit element dintr-un document HTML pe care vor fi aplicate anumite:
* **proprietati:** folosite pentru a defini modul in care elementul(ele) selectat(e) sunt afisate pe ecran
* **valoare**: modificarea de proprietat pe care o dorim pentru elementul(ele) selectat(e)

Exemplu: Vreau ca pe **toate paragrafele din documentul HTML** pe care l-am creat sa aiba **marimea font-ului 16px** si **culoarea textului rosu**.

**CSSOM** sau **Cascading Style Sheets Object Model** reprezinta o structura arborescenta alcatuita din obiecte care definesc obiectele elementele ce structureaza un anumit document HTML, alaturi de proprietatile CSS corespunzatoare definite in cadrul uneia sau a mai multe foi de stiluri.

Aceasta structura este creata de catre browser in momentul in care o pagina este incarcata, asociind fiecarui element HTML specificat cate un obiect si serveste ca o **interfata** pentru accesarea, modificarea si stergerea proprietatilor CSS asociate unui element. ( vom vedea mai tarziu cine foloseste aceasta interfata )

**Sass** (prescurtare pentru foi de stil minunat din punct de vedere sintactic) este un limbaj de script preprocesor care este interpretat sau compilat în foi de stil în cascadă (CSS). SassScript este limbajul de scriptare în sine.

Sass este format din două sintaxe. Sintaxa originală, numită „sintaxa indentată”, folosește o sintaxă similară cu Haml. Folosește indentarea pentru a separa blocurile de cod și caracterele de linie nouă pentru a separa regulile. Sintaxa mai nouă, „SCSS” (Sassy CSS), folosește formatarea blocurilor ca cea a CSS. Folosește paranteze pentru a indica blocuri de cod și punct și virgulă pentru a separa regulile dintr-un bloc. Sintaxa indentată și fișierele SCSS primesc în mod tradițional extensiile .sass și respectiv .scss.

CSS3 constă dintr-o serie de selectoare și pseudo-selectoare care grupează regulile care se aplică acestora. Sass (în contextul mai larg al ambelor sintaxe) extinde CSS furnizând mai multe mecanisme disponibile în limbaje de programare mai tradiționale, în special limbaje orientate obiect, dar care nu sunt disponibile pentru CSS3 în sine. Când SassScript este interpretat, creează blocuri de reguli CSS pentru diferiți selectori, așa cum sunt definiți de fișierul Sass. Interpretul Sass traduce SassScript în CSS. Alternativ, Sass poate monitoriza fișierul .sass sau .scss și îl poate traduce într-un fișier .css de ieșire ori de câte ori fișierul .sass sau .scss este salvat.

Sintaxa indentată este un limbaj metalic. SCSS este un metalimbaj imbricat, deoarece CSS valid este SCSS valid cu aceeași semantică.

SassScript oferă următoarele mecanisme: variabile, cuibărire, mixini și moștenire selector.

**2.3 Javascript**

JavaScript (JS) este un [limbaj de programare](https://ro.wikipedia.org/wiki/Limbaj_de_programare) [orientat obiect](https://ro.wikipedia.org/wiki/Programare_orientat%C4%83_pe_obiecte) bazat pe conceptul [prototipurilor](https://ro.wikipedia.org/wiki/Prototip_(POO)). Este folosit mai ales pentru introducerea unor funcționalități în paginile web, codul JavaScript din aceste pagini fiind rulat de către [browser](https://ro.wikipedia.org/wiki/Browser_web). Limbajul este binecunoscut pentru folosirea sa în construirea [siturilor web](https://ro.wikipedia.org/wiki/Website), dar este folosit și pentru accesul la obiecte încapsulate (embedded objects) în alte aplicații. A fost dezvoltat inițial de către [Brendan Eich](https://ro.wikipedia.org/wiki/Brendan_Eich) de la [Netscape Communications Corporation](https://ro.wikipedia.org/wiki/Netscape_Communications_Corporation) sub numele de Mocha, apoi LiveScript, și denumit în final JavaScript.

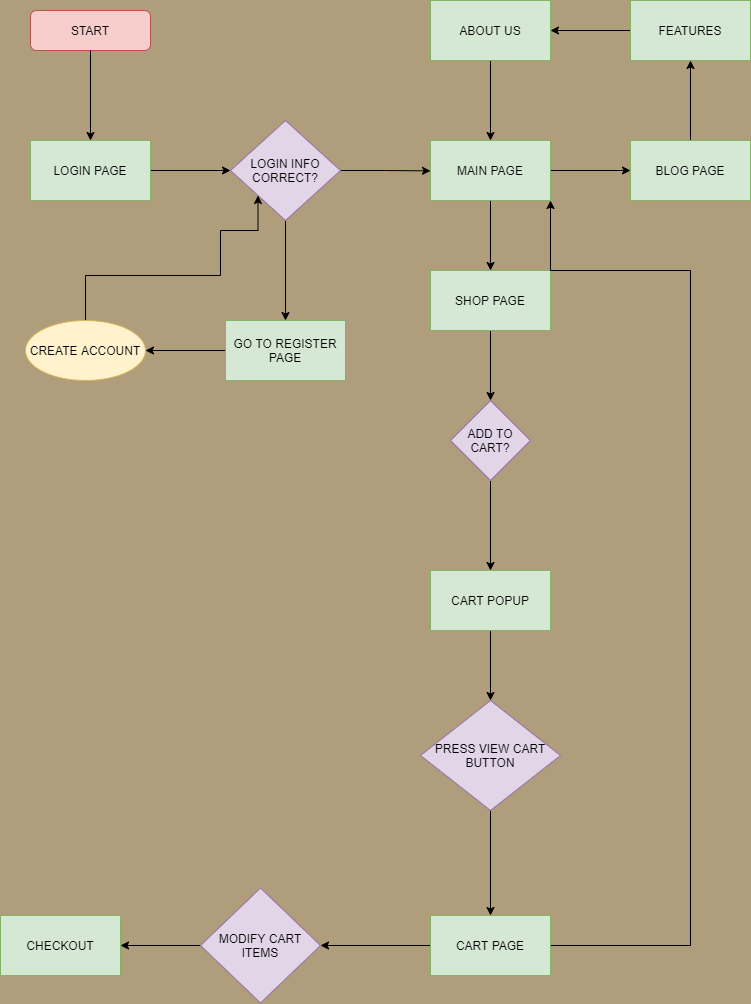
În ciuda numelui și a unor similarități în sintaxă, între JavaScript și limbajul Java nu există nicio legătură. Ca și [Java](https://ro.wikipedia.org/wiki/Java), JavaScript are o sintaxă apropiată de cea a [limbajului C](https://ro.wikipedia.org/wiki/Limbajul_de_programare_C), dar are mai multe în comun cu [limbajul Self](https://ro.wikipedia.org/w/index.php?title=Limbajul_Self&action=edit&redlink=1) decât cu [Java](https://ro.wikipedia.org/wiki/Limbajul_Java).

Cea mai des întâlnită utilizare a JavaScript este în scriptarea [paginilor web](https://ro.wikipedia.org/wiki/Pagin%C4%83_web). Programatorii web pot îngloba în paginile HTML script-uri pentru diverse activități cum ar fi verificarea datelor introduse de utilizatori sau crearea de meniuri și alte efecte animate.

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui [arbore](https://ro.wikipedia.org/wiki/Arbore) de [obiecte](https://ro.wikipedia.org/wiki/Obiect_(informatic%C4%83)) și pun la dispoziție aceste obiecte script-urilor JavaScript, care le pot citi și manipula. Arborele de obiecte poartă numele de [Document Object Model](https://ro.wikipedia.org/wiki/Document_Object_Model) sau [DOM](https://ro.wikipedia.org/wiki/DOM). Există un standard [W3C](https://ro.wikipedia.org/wiki/W3C) pentru DOM-ul pe care trebuie să îl pună la dispoziție un browser, ceea ce oferă premiza scrierii de script-uri [portabile](https://ro.wikipedia.org/w/index.php?title=Portabilitate&action=edit&redlink=1), care să funcționeze pe toate browserele. În practică, însă, standardul W3C pentru DOM este incomplet implementat. Deși tendința browserelor este de a se alinia standardului W3C, unele din acestea încă prezintă incompatibilități majore, cum este cazul [Internet Explorer](https://ro.wikipedia.org/wiki/Internet_Explorer).

O tehnică de construire a paginilor web tot mai întâlnită în ultimul timp este [AJAX](https://ro.wikipedia.org/wiki/AJAX), abreviere de la „Asynchronous JavaScript and XML”. Această tehnică constă în executarea de cereri [HTTP](https://ro.wikipedia.org/wiki/HTTP) în fundal, fără a reîncărca toată pagina web, și actualizarea numai anumitor porțiuni ale paginii prin manipularea DOM-ului paginii. Tehnica AJAX permite construirea unor interfețe web cu timp de răspuns mic, întrucît operația (costisitoare ca timp) de încărcare a unei pagini HTML complete este în mare parte eliminată. JavaScript este cu adevărat un limbaj dinamic. Rar este necesar să utilizați operatorul de evaluare pentru lucruri de genul acesta, pentru că, practic, puteți scrie orice doriți, dacă sintaxa este corectă și dacă ceea ce ați scris nu există, veți primi o eroare de execuție.

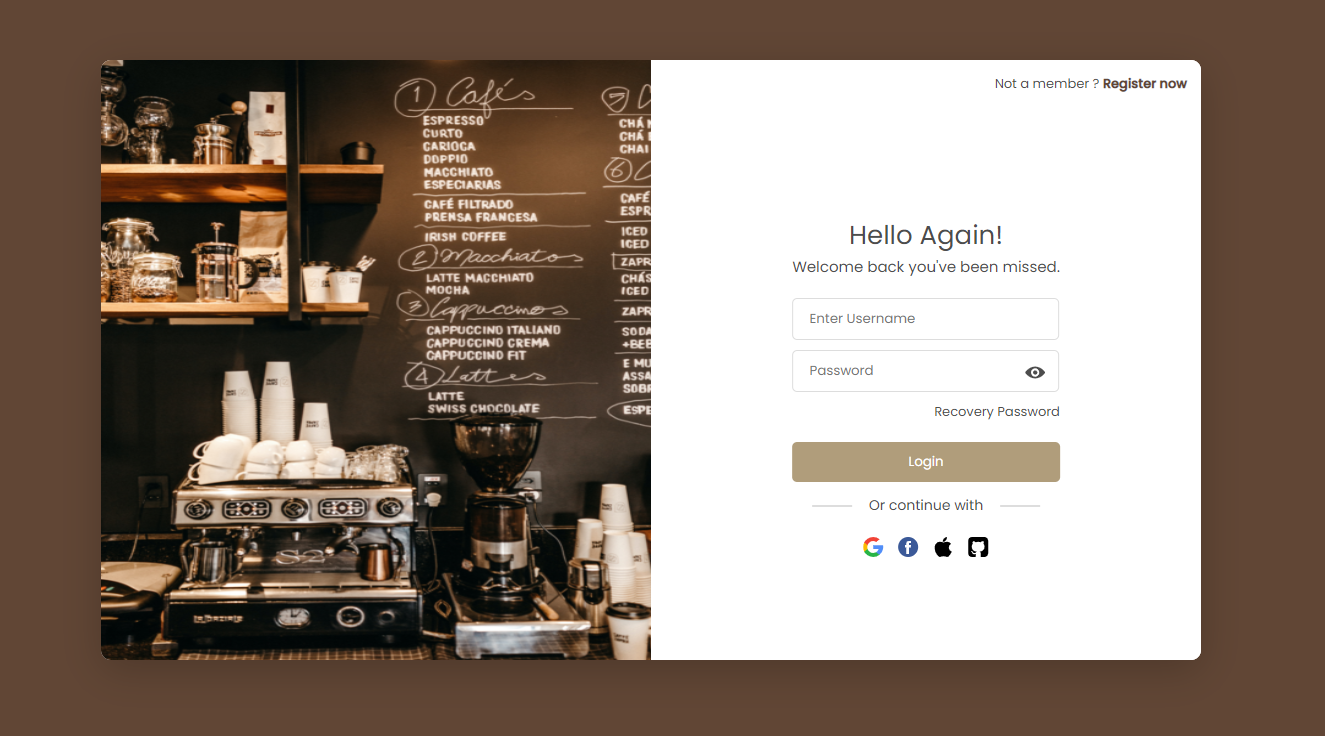
**Capitolul 3. Structura aplicatiei**

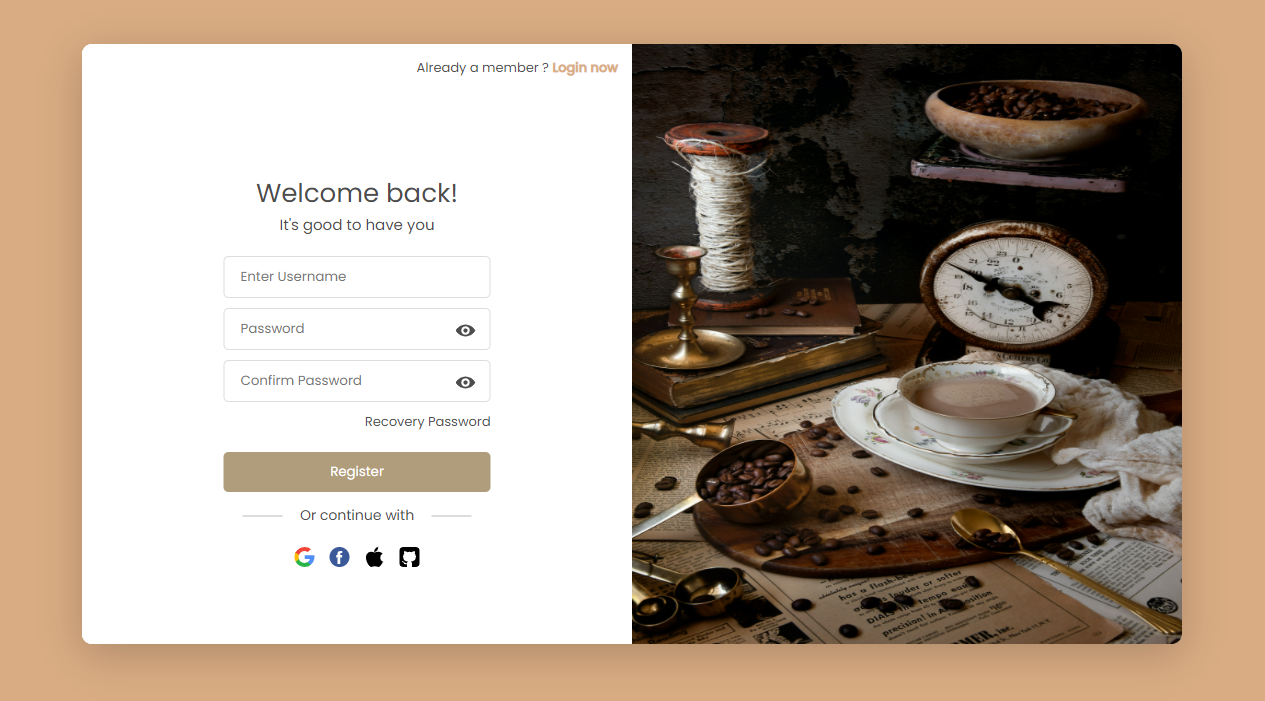
****

**Capitolul 4. Descrierea aplicatiei**

**4.1 Login/Signup**

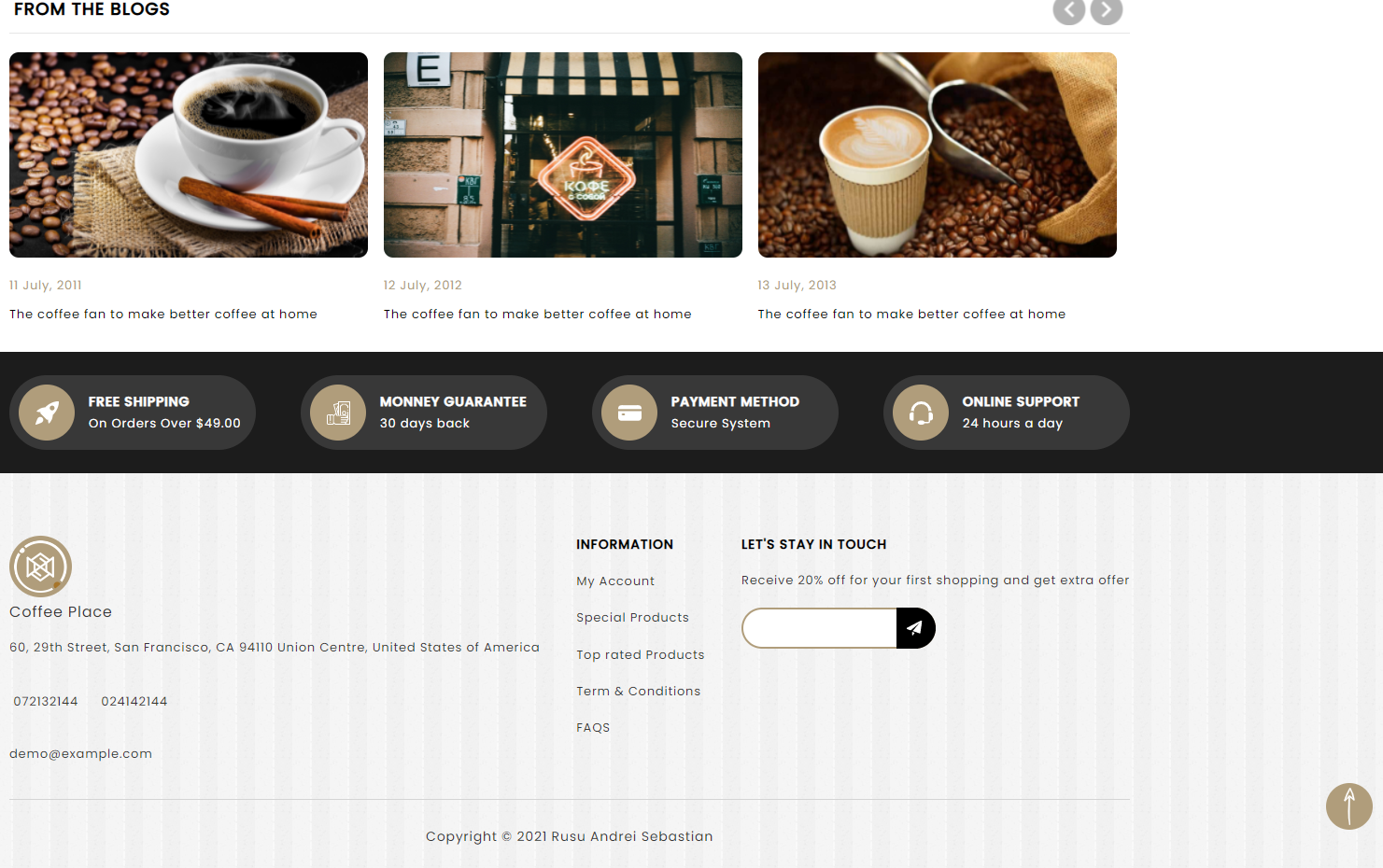
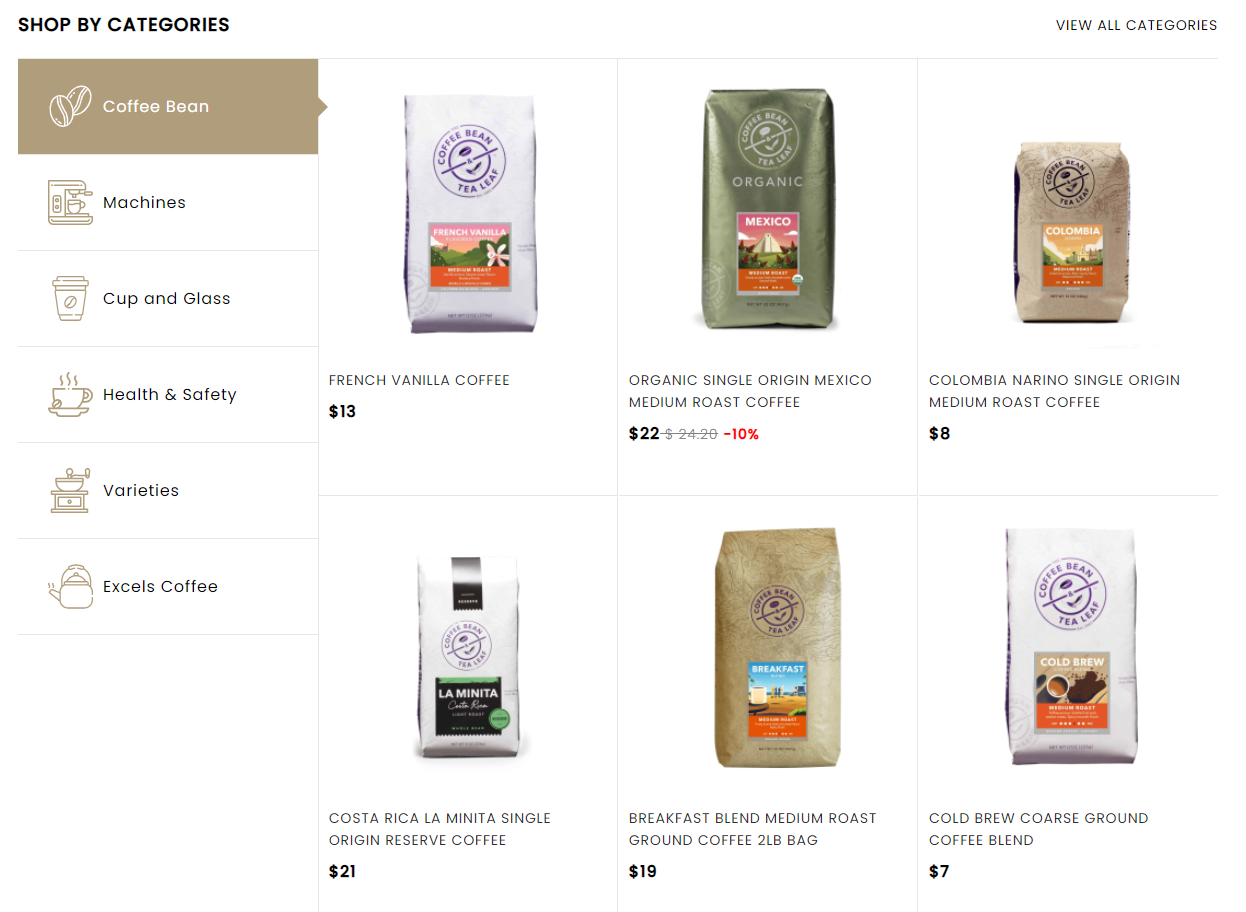
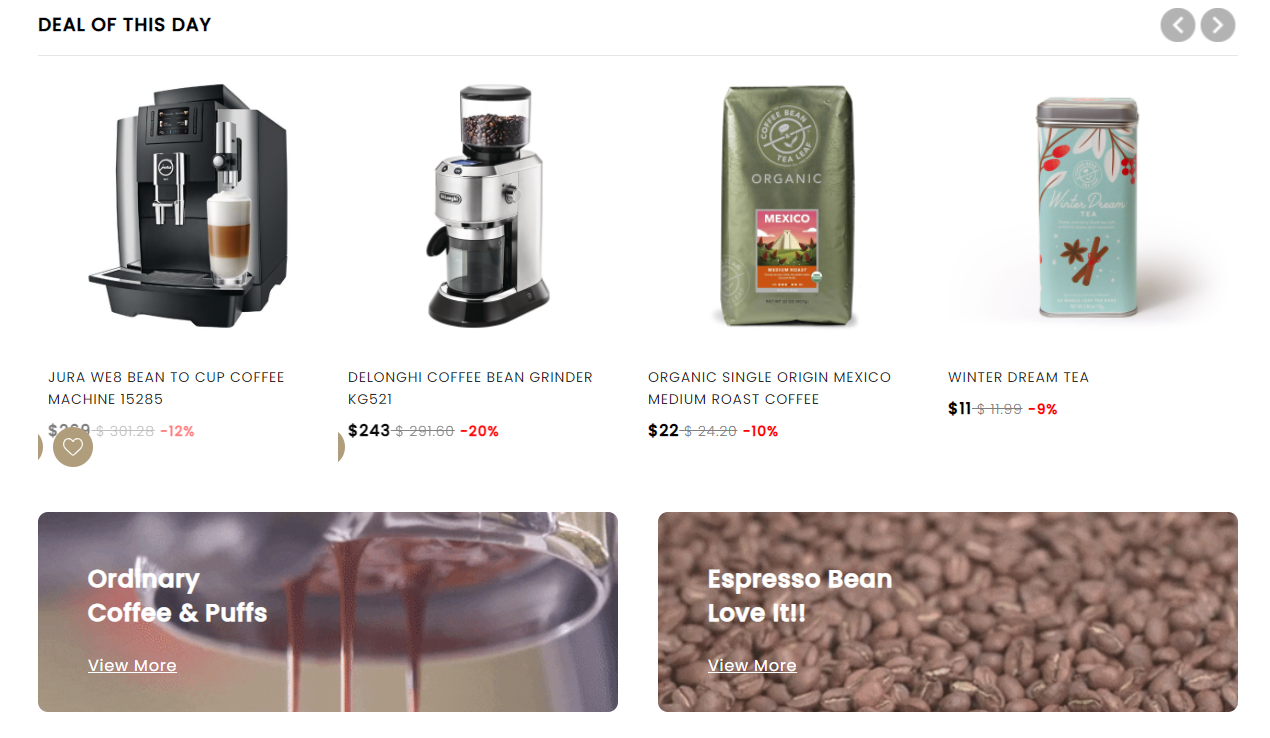
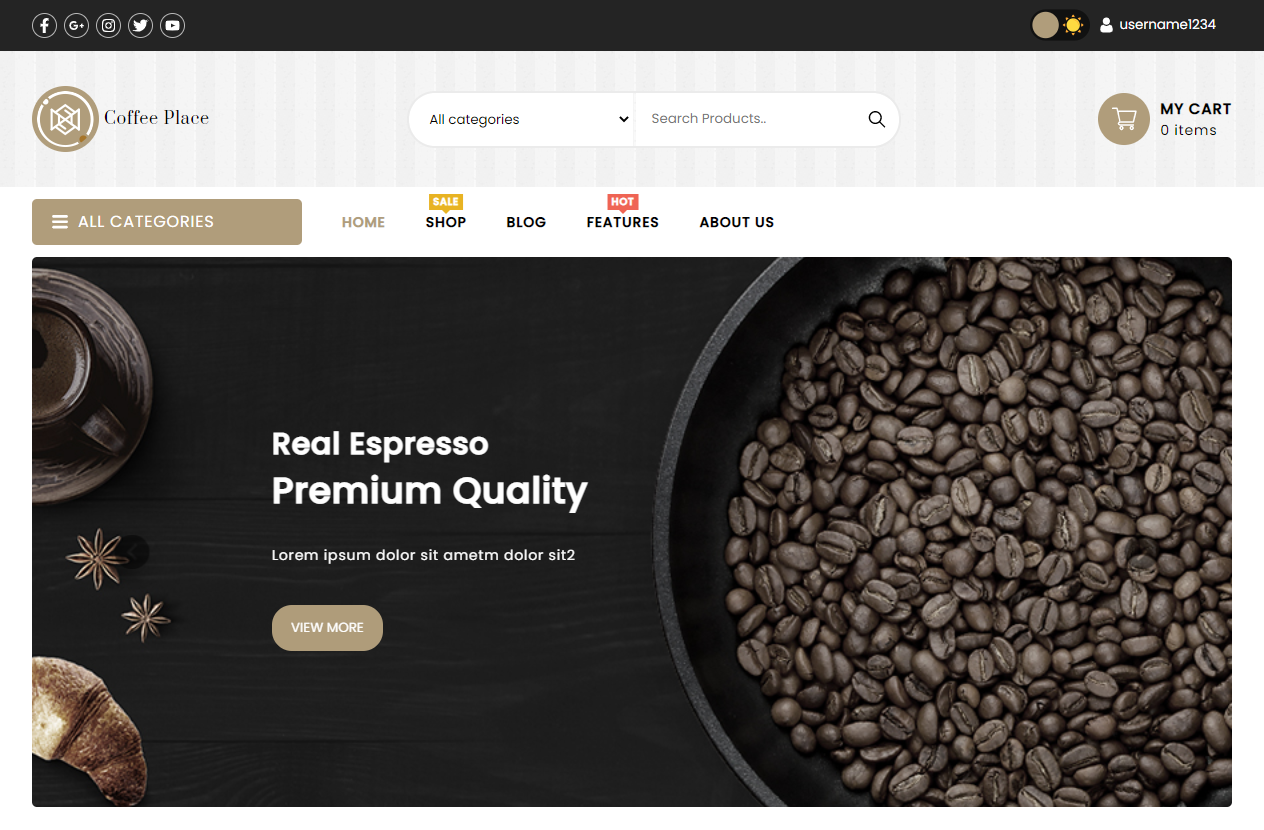
Este prima pagina a aplicatiei in care utilizatorul se poate conecta la un cont deja creat sau se poate inregistra pentru prima data. Acesta este obligat sa-si introduca corect datele de utilizator pentru a continua la pagina principala .

****

****

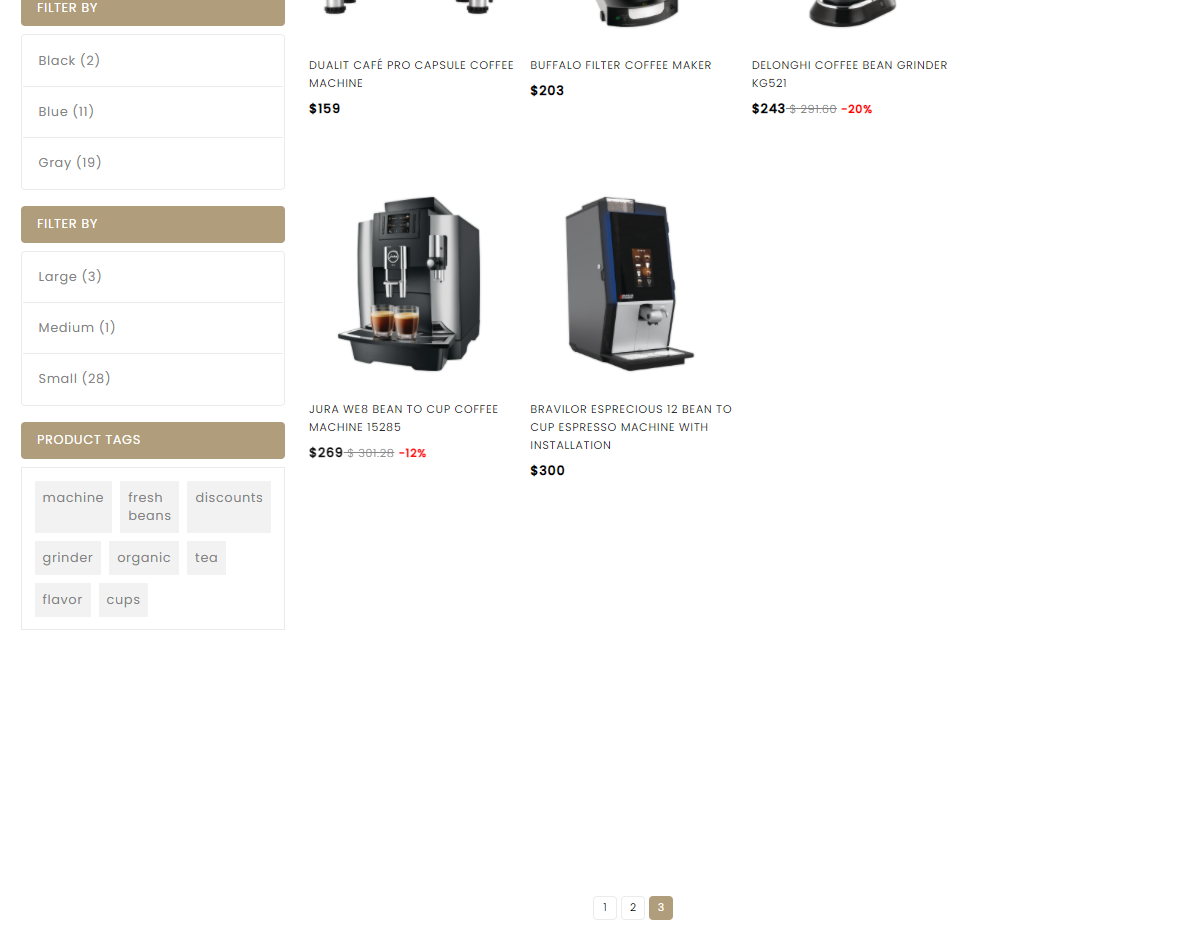
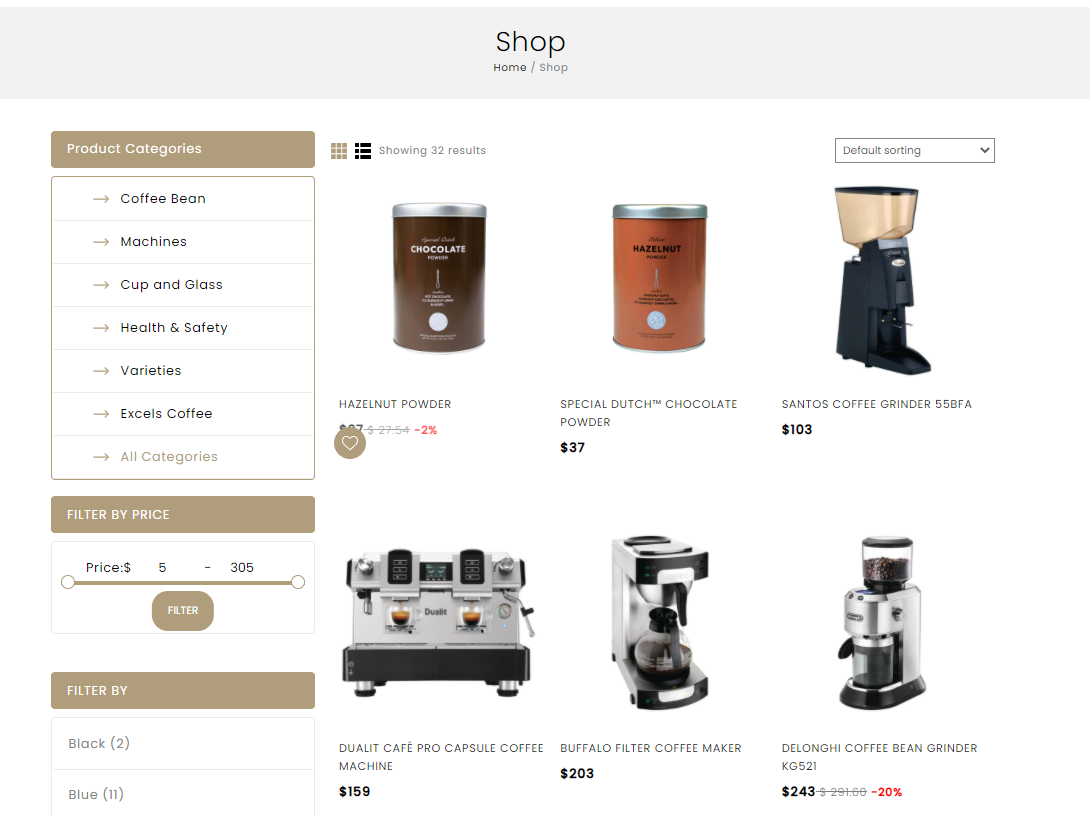
**4.2 Pagina de prezentare**

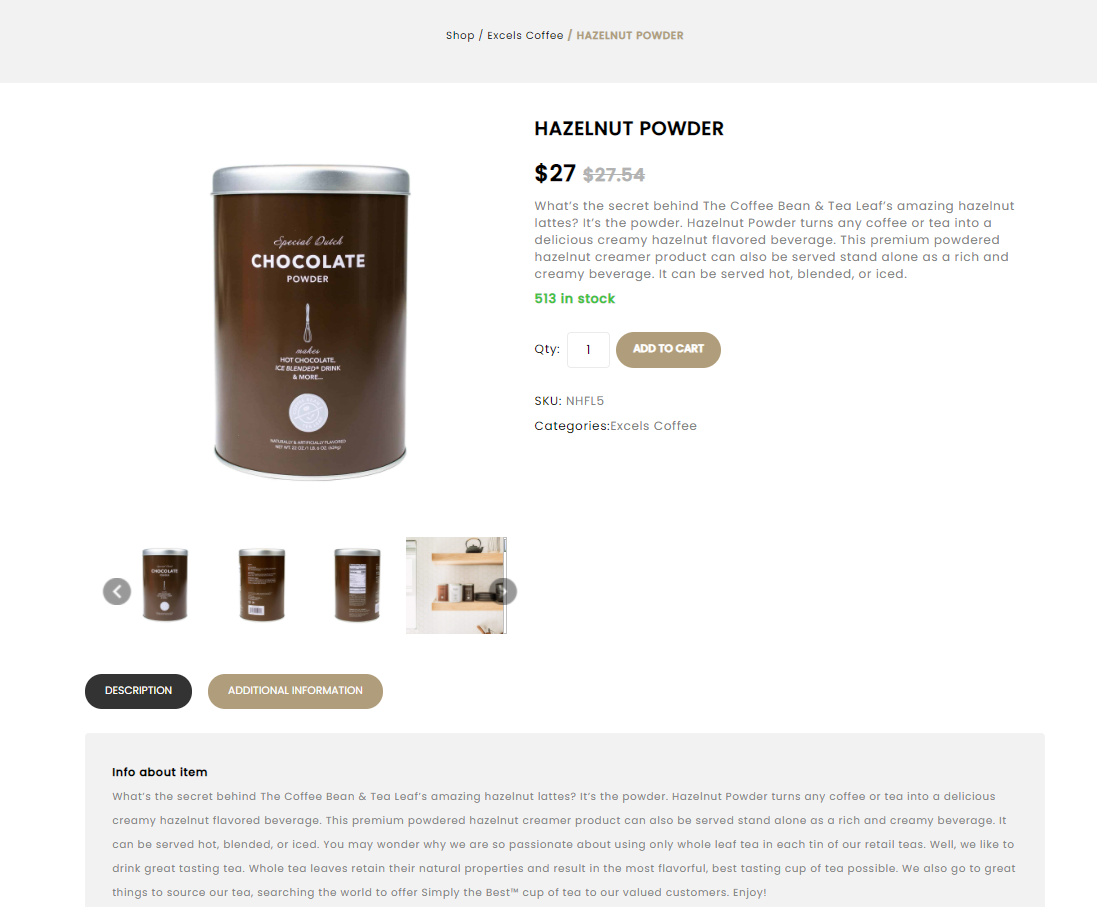
Pe pagina principala a aplicatiei web, utilizatorul are la indemana o gama larga de functionalitati, ca de exemplu: Dark-mode, Hamburger-menu, search-bar,dropdown content, slideshow, view item modal,animatii,select categories cat si responsive design.

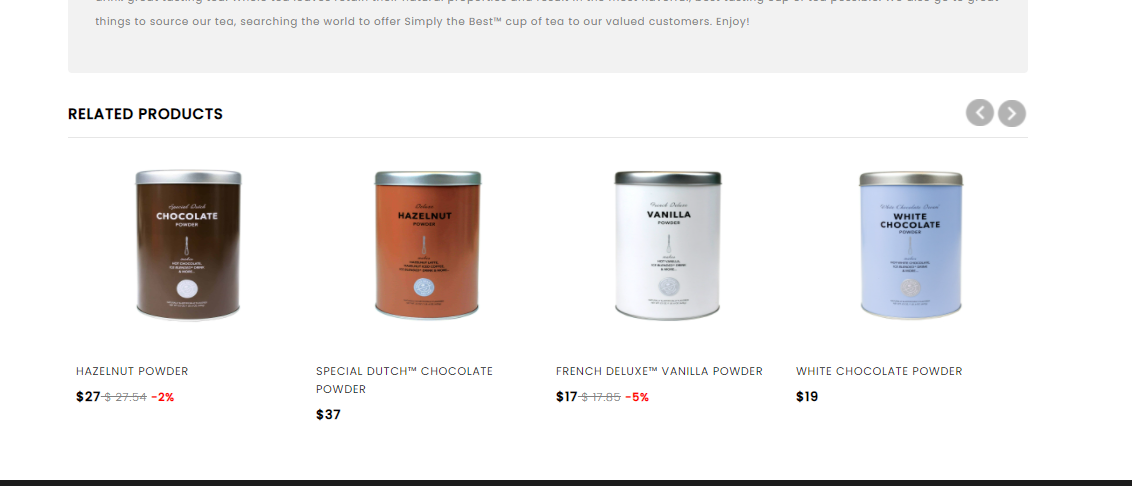
****

**4.3 Pagina de shop**

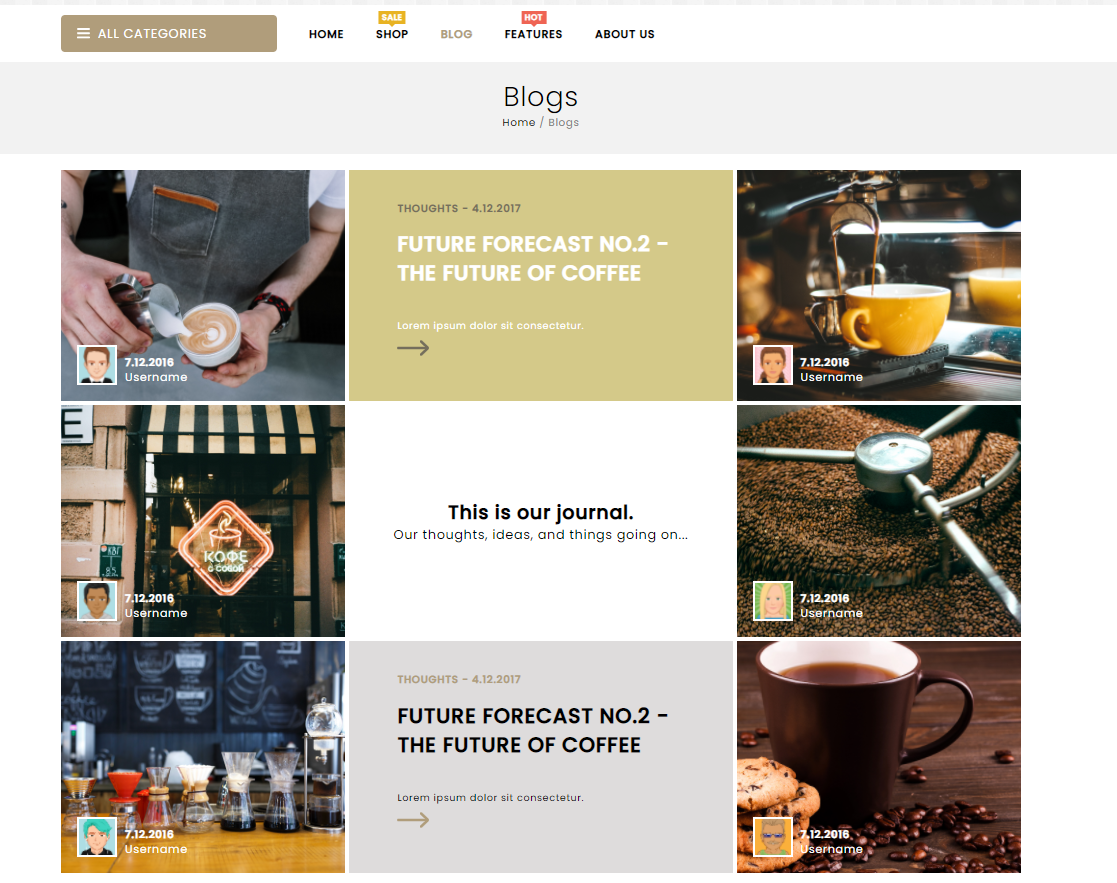
Pe aceasta pagina, utilizatorul va avea la dispozitie mai multe filtre prin care isi va putea adauga itemul ales in cart sau se poate redirectiona la pagina cu informatii despre itemul ales.



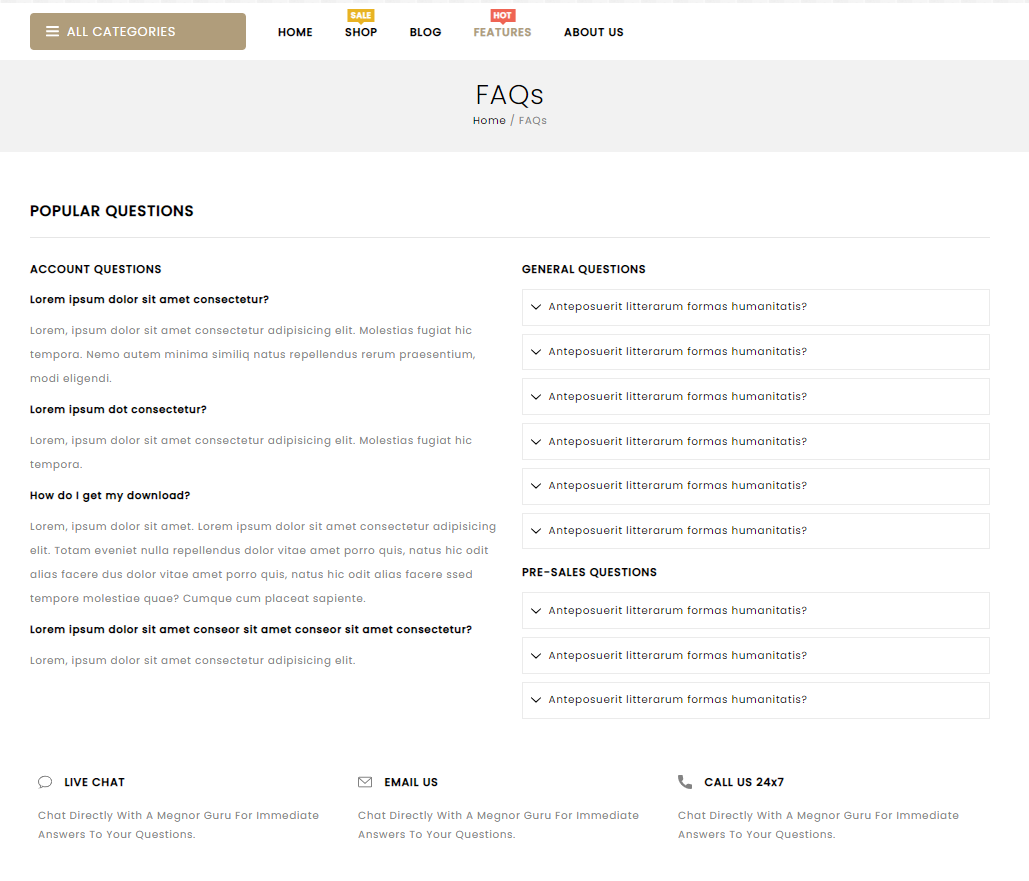




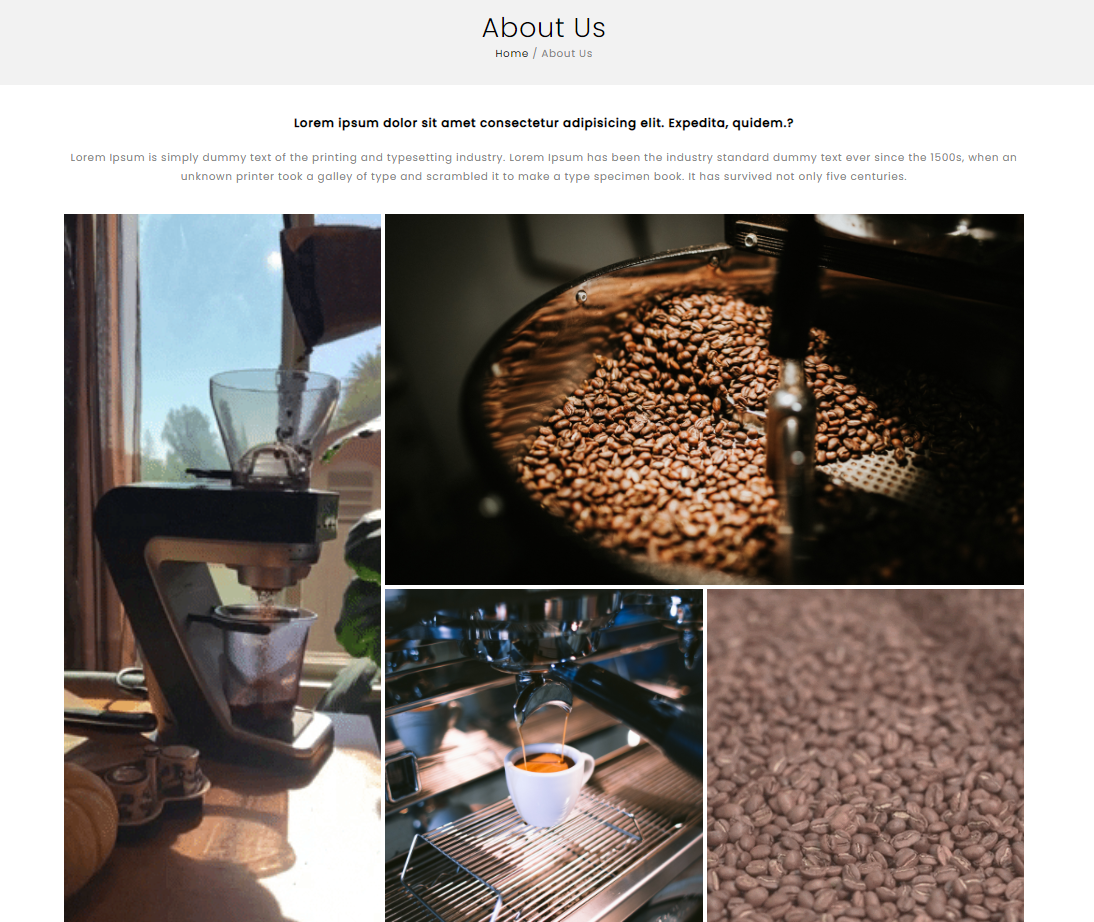
**4.4 Pagina de blog**

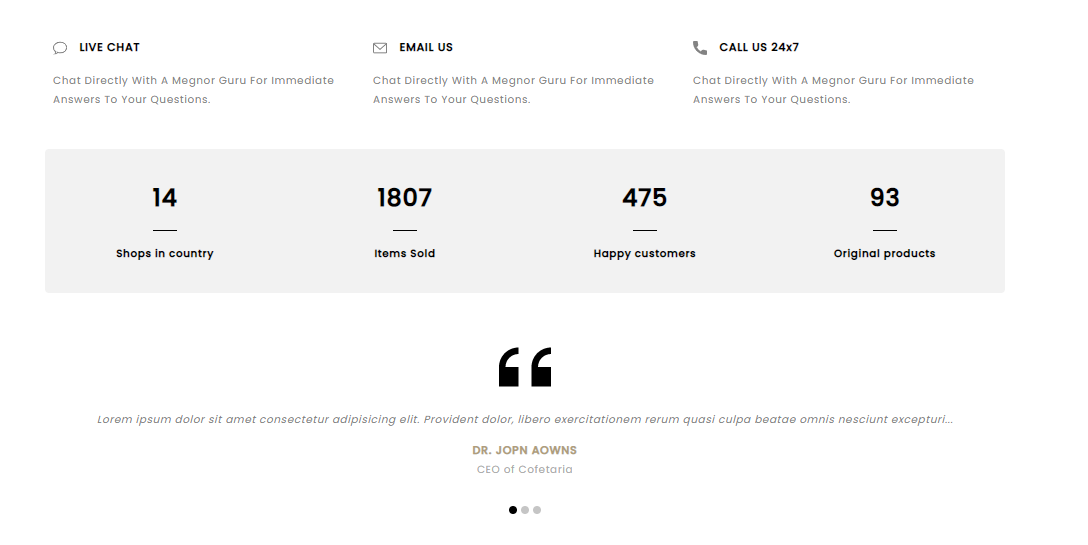


**4.5 Pagina de features**



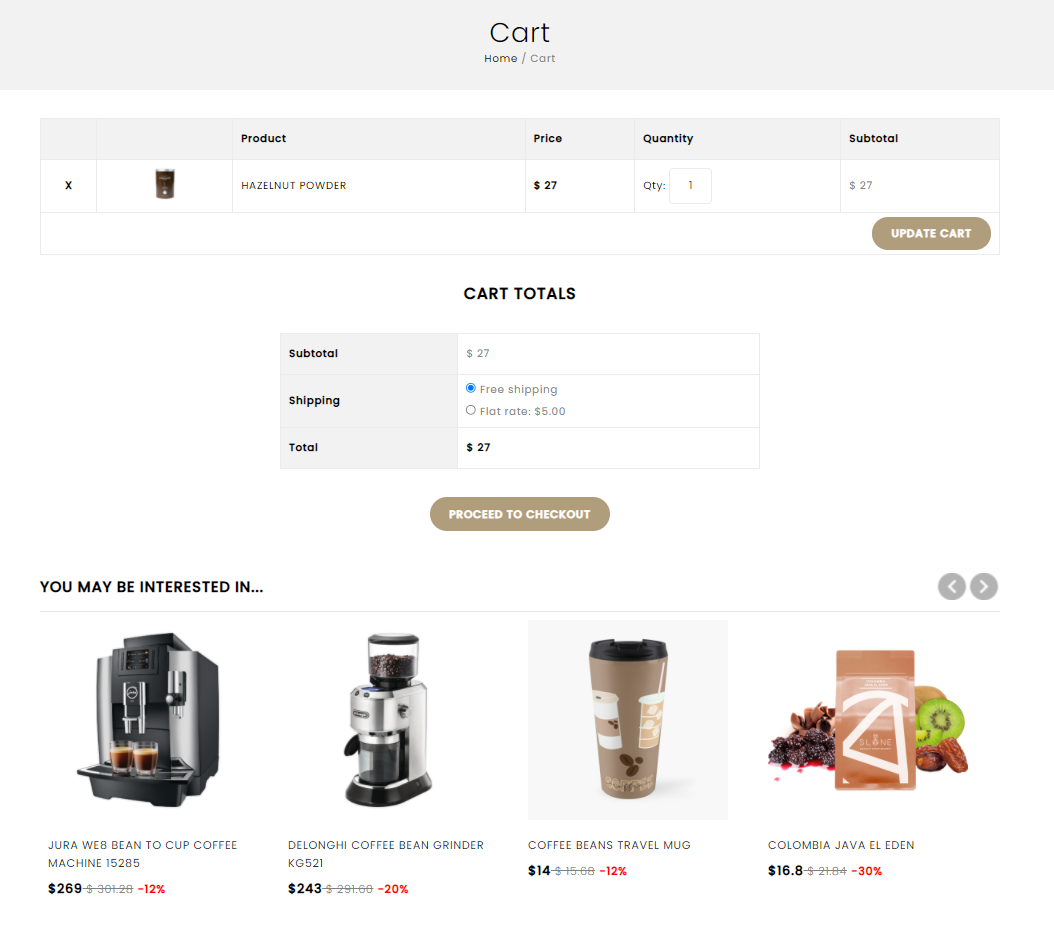
**4.6 Pagina de about us**





**4.7 Pagina de cart**

Utilizatorul poate modifica cantitatea produsului adaugat in cosul de cumparaturi si chiar sa-l elimine.



**Bibliografie**

* <https://edu.wantsome.ro>
* <https://stackoverflow.com/>
* <https://developer.mozilla.org/en-US/>
* <https://css-tricks.com/>
* <https://www.codecademy.com/learn/learn-html>
* <https://ro.wikipedia.org/wiki/Pagina_principal%C4%83>