**UNIVERSITATEA BABEŞ-BOLYAI CLUJ-NAPOCA**

**FACULTATEA DE MATEMATICǍ ŞI INFORMATICǍ**

**SPECIALIZAREA**

**MATEMATICA-INFORMATICA**

**LUCRARE DE LICENŢĂ**

**Dezvoltarea unei aplicații Web Client-Server pentru**

**gestiunea unui magazin virtual.**

**Conducător ştiinţific**

**Lector dr.Pop Andreea-Diana**

**Absolvent**

**Dorha Alex-Traian**

**2017**

1

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **CUPRINS** |  |
| [**CAPITOLUL 1**](#page3) | | | [**3**](#page3) |
| [**1.1**](#page4) | [**MOTIVAȚIA**](#page4) | | [**4**](#page4) |
| [**1.2**](#page5) | [**STRUCTURA APLICATIEI**](#page5) | | [**5**](#page5) |
| [**CAPITOLUL 2**](#page6) | | | [**6**](#page6) |
| [**2.1 INTERNETUL**](#page6) | | | [**6**](#page6) |
| [**2.2 APLICAȚII WEB**](#page6) | | | [**6**](#page6) |
| [**2.3 SITE-URILE WEB**](#page7) | | | [**7**](#page7) |
| [**CAPITOLUL 3**](#page9) | | | [**9**](#page9) |
| [**3.1 TEHNOLOGII FRONT-END**](#page9) | | | [**9**](#page9) |
| [**3.1.1 HTML**](#page9) | | | [9](#page9) |
| [**3.1.2 CSS**](#page11) | | | [11](#page11) |
| [**3.1.3 JAVASCRIPT**](#page12) | | | [12](#page12) |
| [**3.1.4 ANGULARJS**](#page13) | | | [13](#page13) |
| [**3.2 TEHNOLOGII BACK-END**](#page14) | | | [**14**](#page14) |
| [**3.2.1 MICROSOT .NET FRAMEWORK**](#page14) | | | [14](#page14) |
| [**3.2.2 ASP.NET**](#page16) | | | [16](#page16) |
| [**3.2.3 WEB API**](#page16) | | | [16](#page16) |
| [**3.2.4 JSON**](#page18) | | | [18](#page18) |
| [**3.2.5 LINQ**](#page18) | | | [18](#page18) |
| [**3.2.6 ENTITY FRAMEWORK**](#page20) | | | [20](#page20) |
| [**CAPITOLUL 4**](#page21) | | | [**21**](#page21) |
| [**4.1 DESCRIEREA APLICATIEI**](#page21) | | | [**21**](#page21) |
| [**4.2 ETAPELE PROIECTARII APLICATIEI**](#page21) | | | [**21**](#page21) |
| [**4.2.1 DIAGRAMA CAZURILOR DE UTILIZARE**](#page22) | | | [22](#page22) |
| [**4.2.2 BAZA DE DATE**](#page25) | | | [25](#page25) |
| [**4.3 DESCRIERE ARHITECTURA**](#page28) | | | [**28**](#page28) |
| [**4.4 APLICAREA INSTRUMENTELOR UTILIZATE PENTRU DEZVOLTARE**](#page47) | | | [**47**](#page47) |
| [**4.5 MANUAL DE UTILIZARE AL APLICATIEI**](#page51) | | | [**51**](#page51) |
| **4.6 C**oncluzii | | | 55 |
| **BIBLIOGRAFIE:** | | | **56** |

2

**Capitolul 1**

**INTRODUCERE**

Internetul nu este văzut doar ca un progres excepțional al tehnologiei, ci este perceput ca un mediu indispensabil de căutare a informației, un mediu de comunicare, de socializare, de divertisment și nu în ultimul rând un mediu în care iau naștere afaceri profitabile.

Cine ar fi zis că Internetul de la simpla sa funcționalitate, aceea de a transmite informații și date între persoane și instituții urma să devină o nouă modalitate de prelucrare,prezentare și comunicare a informației ce permite implementarea unor noi modele și concepte de neimaginat. Apariția Internetului a determinat transformări esențiale în publicitate.

Internetul a fost văzut de publicitari ca o piața cu reale oportunități de reușită, ca spre surprinderea lor să afle că acesta funcționează,în acest caz ca o afacere online.

Odată cu trecerea anilor, programarea Web s-a dovedit ca fiind unul dintre domeniile dezvoltării de aplicații software cu o evoluție permanentă, atât calitativă, cât și cantitativă. Un rol important în contextul acestei dezvoltări îl constituie Internetul.

Pentru această lucrare,am încercat să aprofundez cât mai multe aspecte teoretice și practice cu privire la modalitățile de proiectare și implementare a unei aplicații web.

Pentru crearea site-ului, am ales un design plăcut și modern care să atragă cât mai multe persoane.

Un alt aspect important în dezvoltarea aplicației îl reprezintă încărcarea rapidă a paginilor, conținutul site-ului să fie bine structurat , ușor de înțeles și accesibil pentru toți utilizatorii.

Aplicația poate fi rulată pe majoritatea motoarelor de căutare.

Am folosit o parte din limbajele și tehnologiile de programare utilizate în momentul de față pentru dezvoltarea și întreținerea oricărui tip de aplicații Web:HTML, CSS, AngularJS, SQL SERVER, JavaScript , JSON si LINQ.

3

**1.1 Motivația**

Stilul reprezintă o lume plina de imaginație, este un joc prin care ne putem exprima personalitatea.Ce spun alegerile vestimentare despre tine? Încă mai cred că am 16 ani! Nu imi pasă! Uită-te la mine! Vreau sa fiu invizibil! Vreau să par mai în vârstă!

Uneori trec mulți ani până ce un bărbat reușește să-și găsească stilul său vestimentar.

Acest trebuie să spună multe despre tine: mă simt confortabil în propria mea piele, sunt unic și îmi place să-mi exprim personalitatea, reușesc să găsesc cu lejeritate ceea ce mi se potrivește cel mai bine, nu am nici o reținere să încerc un anumit stil doar pentru a vedea dacă sunt și alte lucruri care s-ar putea să mi se potrivească și încă nu am știut de ele.

Principala motivație pentru această aplicație a constat în faptul că stilul vestimentar este una dintre pasiunile mele.Această pasiune am descoperit-o de la tatăl meu, el a reprezentat și încă reprezintă un model pentru mine demn de urmat pe toate planurile.

În zilele noastre, stilul s-a dezvoltat foarte mult, iar acum fiecare persoană are la dispoziție o mulțime de informații despre modă,despre lumea mondenă și cu ajutorul acestor informații își pot descoperi propriul lor stil vestimentar.

Pornind de la această afirmație, și de la faptul că, stilul vestimentar a ajuns să fie atât de avansat, am ales dezvoltarea unei aplicații web, și anume un magazin online despre haine care este special conceput pentru persoanele de sex masculin.

Aplicația este destinată tuturor persoanelor care au acces la internet, care doresc să afle informații despre anumite produse, cele mai căutate produse sau produsele care au fost adăugate cel mai recent în magazin, și multe altele.

Scopul acestei lucrări este descrierea unui mod prin care se poate dezvolta o aplicație web, mai exact un magazin în mediul online despre haine.

Astfel am încercat să respect cât mai multe aspecte și concepte care să ofere un mediu online mult mai prietenos, interactiv cu clienții, care să-i ajute să găsească informații necesare.

4

**1.2 Structura aplicatiei**

Lucrarea de licență cuprinde destul de detaliat aspecte teoretice și practice ale modalităților de concepere și implementare a unei aplicații web. Structura acestei lucrării este împărțită pe patru capitole.

În dezvoltarea aplicației web am definit câteva cerințe :

* un design, atrăgător și modern, potrivit cu profilul site-ului.
* navigare ușoară și accesibilă pentru diferite tipuri de utilizatori.
* încărcarea rapidă a paginilor.

•tehnic compatibil cu majoritatea navigatoarelor web: Chrome, Internet, Explorer,Opera.

* structura tehnică solidă, capabilă să evolueze în timp.
* bine structurat din perspectiva informațiilor oferite vizitatorilor, adică clar și

concis.

• ușor de menținut și de actualizat pentru cei responsabili de conținut.

Aplicatia este impartita in 3 nivele:

1. **Front-end**

Partea de view, cea care expune informațiile formatate utilizatorului și permite interacțiunea dintre utilizator si funcțiile aplicației. Este creată utilizând tehnologiile:

HTML, CSS, Javascript, AngularJS Framework. Folosește un serviciu web API prin requesturi HTTP la resursele acestuia.

1. **Back-End**

Această parte este inima aplicației punând la dispoziție o gamă variată de endpoint-uri care expun informația din baza de date spre afisare sau primesc informație spre manipulare, în ambele cazuri utilizându-se formatul de date JSON. Partea de back-end este creată utilizând framework-ul ASP.NET Web API configurat să accepte request-uri de tip Get, Post, Put, Delete

1. **Stocare de date**

Pentru partea de stocare de date m-am folosit de limbajul Microsoft SQL utilizând utilitarul Microsoft SQL Server Express. Baza de date a fost creată cu ajutorul migrațiilor și cuprinde 10 tabele legate prin diferite relații plus un tabel special care ține un istoric al schimbărilor aduse asupra bazei de date.

5

**Capitolul 2**

**Noțiuni introductive**

**2.1 Internetul**

Termenul de „Internet” este format din compunerea parțială a doua cuvinte provenite din limba engleză, interconnected care înseamnă interconectat și network fiind rețeaua. Acest termen desemnează o rețea mondială unitară de calculatoare care sunt interconectate conform protocoalelor sau regulilor de comunicare TCP/IP sub denumirea de Transmission Control Protocol și Internet Protocol. În Internet există două categorii importante de calculatoare:cele care acumulează,distribuie și sortează informațiile numite servere sau gazde,iar cele care solicită accesul la aceste informații și le utilizează se numesc clienții.

Internetul s-a dezvoltat foarte mult și în același timp era într-o continuă ascendență, ajungându-se la sfârșitul anului 1993 când erau înregistrate l30 servere web,iar în anul 1994 erau peste 11000.În prezent numărul calculatoarelor conectate a ajuns la sute de miliarde în întreaga lume.

Internet Protocol este un protocol de comunicații care transmite datele. Datele sunt trimise de la un calculator la altul prin Internet,și sunt identificate printr-o adresa IP. Atunci când o persoană primește sau trimite informații,mesajul este împărțit în blocuri mici numite pachete.

**2.2 Aplicații web**

O aplicație web este un program care rulează într-o arhitectură client-server.

**World Wide Web** sau WWW este baza unui protocol numit Hypertext TransferProtocol,pe scurt HTTP. Protocolul de transfer de tip hypertext(HTTP) este cea mai folosită metoda prin care se accesează informațiile în Internet și care sunt stocate pe servere World Wide Web (www).

Fișierul trimis spre destinație poate fi un document HTML(HyperText Markup Language),fișiere grafice,video ,editor de text și altele,care funcționează împreună cu HTTP.

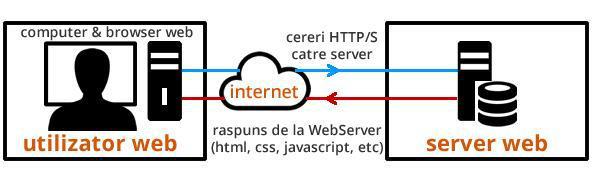
Aplicațiile înlocuiesc modelele în care serverul și clientul rulează tehnologii proprietar, mentenanța acestora este complexă pe partea de client.

6

Prin prezența browserelor,folosite ca și client s-au rezolvat toate problemele. În acest mod,aplicațiile au avansat și pe partea de server,fiind vorba de serverele de aplicații,iar pe lângă limbajele de programare au apărut framework-uri și tehnologii dedicate programării acestora.

În figura de mai jos,este prezentată schema unei aplicații web,în care browserul web face o cerere de tip http sau https către serverul web. Serverul web răspunde acestei cereri și trimite clientului un răspuns prin cod html,css,etc.

Figura de mai jos reprezintă schema unei aplicații web în care browserul web face o cerere de tip http către serverul web.

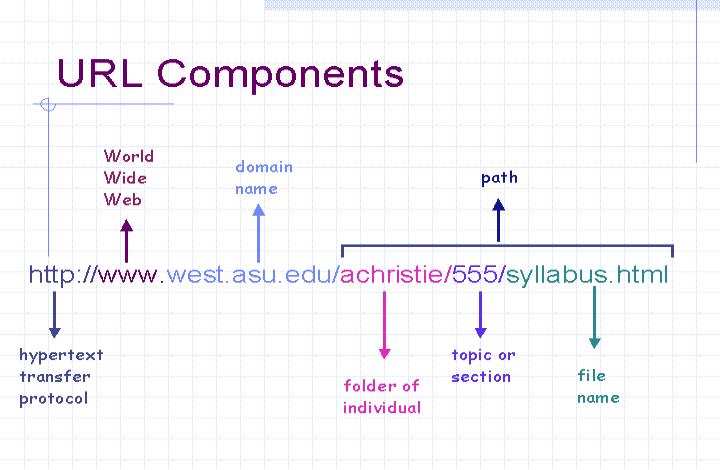


**2.3 Site-urile web**

Un site web este format din mai multe pagini web unite,acesta poate să aibă conținut multimedia identificat în mod normal printr-un nume de domeniu și publicat pe un server web. Site-ul web poate fi accesat printr-o rețea publică de Internet Protocol sau printr-o rețea locala privată (LAN).

Identificarea site-ului se face cu ajutorul identificatorului de resurse uniform (URL). O adresă URL completă este în general formată din trei componente: protocolul, numele site-ului și calea absolută către document sau resursă.

Figura de mai jos reprezintă componentele unei adrese URL.



7

Paginile web sunt accesate și transportate cu Hypertext Transfer Protocol(HTTP),care pot folosi opțional criptarea pentru a asigura securitatea și confidențialitatea utilizatorului. Hyperlink-urile dintre paginile web transmit clienților structura și modul de funcționare al site-ului.

De cele mai multe ori pornește de la pagina principală care conține un director al conținutului web al site-ului. Site-urile web se pot clasifica după o mulțime de factori, dar principalul factor rămâne subiectul de activitate al site-ului. Din punct de vedere tehnologic,un site web poate fi static sau dinamic.

Site web static este un site alcătuit din pagini web stocate pe server în formatul care este trimis către un browser web client. Este programat în principal în Hypertext Markup Language (HTML),fonturile de stil cascadă (CSS) sunt folosite pentru aspect,imaginile sunt utilizate în mod normal pentru a efectua aspectul dorit.

Acest tip site web static va furniza, în general, informații standard pentru o perioadă îndelungată de timp,deși proprietarul site-ului poate face periodic actualizări.

Un site dinamic este unul care se modifică sau se personalizează frecvent și automat. Paginile dinamice din partea serverului sunt generate automat de codul computerului care produce codul HTML.

Un site poate monitoriza o situație în schimbare sau poate furniza informații într-un fel personalizat în funcție de cerințele utilizatorului individual. Codul HTML dinamic utilizează codul JavaScript pentru a instrui browserul să modifice interactiv conținutul paginii.

Figura 1 reprezintă o prescurtare la „World Wide Web”.



8

**Capitolul 3**

**Tehnologii folosite în dezvoltarea aplicației**

**3.1 Tehnologii front-end**

Termenul se referă la acea parte a unei aplicații sau a unui site, care este vizibilă și cu care interacționează utilizatorii.

Front-end-ul este alcătuit din două părți: designul fiind partea creativă și dezvoltarea interfeței reprezentând partea de cod sau implementare HTML,CSS.

Cele mai folosite limbaje pentru front-end sunt: HTML, CSS și chiar și Javascript , toate acestea fiind controlate de browser-ul pe care il folosim.

Această tehnologie include elemente ca de exemplu font-uri, imagini, butoane,formulare,etc.

Pentru proiectarea unui site obiectivul principal este să se asigure că, atunci când utilizatorii deschid site-ul, aceștia să vadă informațiile într-un format ușor de citit și relevant.

Acest lucru este în continuare complicat de faptul că, utilizatorii folosesc acum o mare varietate de dispozitive pe diferite dimensiuni și rezoluții ale ecranului, iar designerii trebuie să aibă în considerare și aceste aspecte referitoare la proiectarea site-ului.

Acest lucru se poate face prin crearea unui design web receptiv utilizând stiluri în CSS.

**3.1.1 HTML**

Hypertext Mark-up Language (HTML) este un limbaj standard de markup pentru crearea de pagini web și aplicații web ce se pot atașa într-un browser,dar HTML nu este un limbaj case-sensitive,deoarece acesta nu face diferența între litere mari și litere mici.

Scopul HTML-ului este mai degrabă prezentarea informațiilor și paragrafelor, fonturilor,tabelelor, etc decât descrierea semanticii documentului. [19]

9

Browserele web primesc documente HTML de la un server web sau stocate local și le fac în pagini web multimedia.

HTML5 a apărut cu scopul de a îmbunătăți limbajul, suportând astfel tehnologii actuale, spre exemplu multimedia.

HTML este o formă de markup orientată către prezentarea documentelor text pe o singură pagină,care duce la folosirea unui software de redare specific,aici fiind vorba de browserul web ,care poartă numele de agent utilizator HTML.

HTML procură mijloacele prin care conținutul unui document poate să completeze un text cu tipuri de metadate și indicații de redare.

Metadatele pot cuprinde detalii cu referire la titlu și autorul documentului,informații structurale despre modul în care este împărțit documentul în diferite segmente,paragrafe,etc. și folosirea hyperlink-urilor care fac legătura unui document cu alte documente.

Indicațiile de redare pot fi de la decorațiuni minore ale textului,de exemplu cuvântul trebuie subliniat și până la scripturi complicate,crearea unor formulare.

În special, HTML5 aduce multe noi caracteristici sintactice. Acestea cuprind elemente ca <video>, <audio>, <header> și <canvas> elemente HTML.

Aceste noutăți sunt proiectate pentru a facilita includerea și manipularea în web a conținuturilor multimedia și grafice fără a fi nevoie să se recurgă la proprietățile de plugin.

Imaginea de mai jos reprezintă imaginea de prezentare pentru HTML.



10

**3.1.2 CSS**

Cascading Style Sheets (CSS) este un limbaj de stilizare care este folosit pentru structurarea și modelarea paginilor web,documentelor scrise în HTML,XHTML și alte limbaje de marcare.

În primul rând acest concept de CSS este utilizat pentru a desprinde conținutul textului de partea ce conține elemente precum aspectul,culorile și fonturile. Separarea acestor informații poate îmbunătății accesul la conținutul real al paginii web .[20]

Spre deosebire de HTML, CSS deține un control mai deosebit asupra formatării unei pagini web,reducând complexitatea și repetiția semnificativă în structura conținutului.

Legătura între HTML și CSS se realizează prin intermediul tag-ului **<style>…</style>,**care trebuie așezat între **<head>…</head>.**Stilurile pentru o pagină,se potimplementa atât în partea de head a documentului HTML ,cât și într-un fișier css extern.

În exemplul de mai jos,am definit două stiluri pentru paragraful din pagina respectivă,adăugând un stil pentru tag-ul **<p>.**

Foile de stil pot fi înserate în trei moduri:foaie externă de stil,foaie internă de stil și stil în linie. Foaia externă de stil este folositoare atunci când avem mai multe pagini ,deoarece putem schimba aspectul site-ului,modificând un singur fișier,paginile trebuie să aibă tag-ul **<link>** în secțiunea **<head>.**



Foaia internă de stil se aplică în momentul în care un singur document are un anumit stil. Stilurile în linie nu sunt prea avantajoase pentru că se amestecă conținutul cu prezentarea,ca de exemplu **<p style=”color:red; text-align:center”>**Acesta e un paragraf**</p>.**

Imaginea de mai jos reprezintă un logo al tehnologiei CSS.

11

**3.1.3 JavaScript**

JavaScript (JS) este cel mai cunoscut limbaj de scripting pe partea de client.

Acesta funcționează în toate browserele importante,spre exemplu Internet Explorer, Chrome și Opera.Limbajul este recunoscut ca fiind un limbaj folosit pentru construirea site-urilor web,dar este utilizat și pentru accesul la obiecte fixate(embedded objects) în alte aplicații.

A fost dezvoltat inițial de către Brendan Elich,având prima denumire Mocha,apoi LiveScript și în final JavaScript.

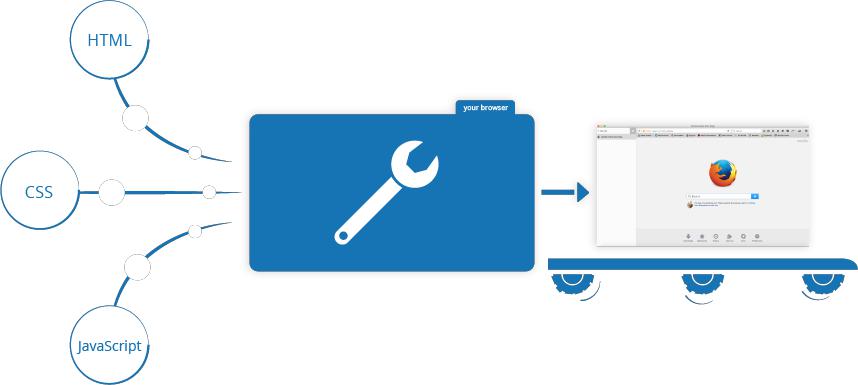
Acesta este un limbaj care oferă posibilitatea de a face ca paginile să fie mai interactive, care să comunice cu vizitatori,site-uri dinamice. Scripturile JavaScript sunt înglobate în pagina HTML ,pentru diferite activități ca de exemplu validarea datelor introduse de utilizatori într-un formular sunt interpretate și sunt executate de browser.[21]

Dacă scriptul aparține de secțiunea **head** va fi executat numai atunci când este apelat,iar dacă este inclus în secțiunea **body**,scriptul va fi executat atâta timp cât pagina se încarcă.

JavaScript reprezintă o secvență de intrucțiuni,declarații și comenzi care vor fi executate de către browser. Scriptul se inserează în pagina încadrat între etichetele **<script>…</script>.**

Browserele rețin în memorie o reprezentare a unei pagini web sub forma unui arbore de obiecte și pun la dispoziție aceste obiecte scripturilor JavaScript. Arborele de obiecte se denumesc ca fiind Document Object Model prescurtat DOM.

Figura de mai jos reprezintă tehnologiile de care am avea nevoie pe partea de front-end ca să realizăm interfața unui site web.



12

**3.1.4 AngularJS**



AngularJS (denumit în mod obișnuit "Angular.js" sau "AngularJS 1.X") este un framework de aplicații web bazate pe open-source bazate pe JavaScript, menținut în principal de Google și de o comunitate de persoane și corporații pentru a aborda multe dintre Provocările întâlnite în dezvoltarea aplicațiilor single-page.[6]

Imaginea de mai sus reprezintă un logo al acestei tehnologii AngularJS iar imaginile de mai jos reprezinta câteva informații despre această tehnologie.



Componentele JavaScript completează Apache Cordova, cadrul utilizat pentru dezvoltarea aplicațiilor mobile pe mai multe platforme.

Obiectivul său este de a simplifica atât dezvoltarea, cât și testarea unor astfel de aplicații, oferind un cadru pentru arhitecturile model-view-controller (MVC) și model-view-viewmodel (MVVM), împreună cu componentele utilizate frecvent în aplicațiile Internet bogate.[7]

13

Potrivit serviciului de analiză JavaScript Libscore, AngularJS este utilizat pe site-urile Wolfram Alpha, NBC, Walgreens, Intel, Sprint, ABC News și aproximativ 12 000 de alte site-uri din 1 milion testate în octombrie 2016.

**3.2 Tehnologii back-end**

Back-end-ul sau „server-side”,reprezintă un modul în care funcționează site-ul,actualizări și modificări. Acest lucru se raportează la tot ceea ce utilizatorul nu poate sa vada în browser,cum ar fi bazele de date și serverele.

Cei care lucrează pe partea de back-end se numesc programatori sau dezvoltatori,care acordă o importanță și asupra securității,structurii și gestionării conținutului. Back-end-ul este format dintr-un server,o aplicație și o bază de date.

Dezvoltarea back-end este necesară pentru a crea un site dinamic. Un site dinamic este un site care se schimbă în mod frecvent,actualizat în timp real și necesită o bază de date.

Niste exemple de astfel de site-uri sunt Facebook sau Google Maps care sunt considerate site-uri dinamice,deoarece conținutul lor se schimbă și se actualizează în mod constant.

Toate informațiile,cum ar fi profilurile utilizatorilor sunt stocate în baza de date a acestor site-uri,iar legătura se face cu tehnologii precum PHP.

Codul care este scris de programatori comunică cu serverul și apoi îi transmite browserului ce trebuie utilizat din baza de date.

**3.2.1 Microsot .NET Framework**

NET Framework (pronunțat dot net) este un schelet software dezvoltat de Microsoft, care rulează pe mai multe sisteme de operare. [2]

Acesta include o bibliotecă de clasă mare denumită Framework Class Library (FCL) și oferă interoperabilitate lingvistică în mai multe limbi de programare.

NET Framework (pronunțat dot net) este un schelet software dezvoltat de Microsoft, care rulează pe mai multe sisteme de operare.

14

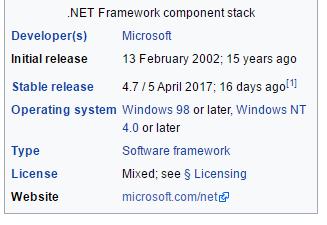


Imaginea de mai sus reprezintă un logo al acestei tehnologii.

NET Framework (pronunțat dot net) este un schelet software dezvoltat de Microsoft, care rulează pe mai multe sisteme de operare.

Acesta include o bibliotecă de clasă mare denumită Framework Class Library (FCL) și oferă interoperabilitate lingvistică în mai multe limbaje de programare.[3]

Imaginea de mai jos reprezintă niste informatii generale despre această tehnologie.



Microsoft a început să dezvolte .NET Framework la sfârșitul anilor 1990, inițial sub numele de Next Generation Windows Services (NGWS). Până la sfârșitul anului 2000, au fost lansate primele versiuni beta ale .NET 1.0.

În august 2000, Microsoft, Hewlett-Packard și Intel au lucrat pentru standardizarea infrastructurii lingvistice comune (CLI) și C #.

Organizația Internațională pentru Standardizare (ISO) a urmat în aprilie 2003.

La 3 octombrie 2007, Microsoft a anunțat că codul sursă pentru bibliotecile .NET

Framework 3.5 urma să devină disponibil sub licența Microsoft Source Source License (Ms-

RSL).

În noiembrie 2014, Microsoft a produs, de asemenea, o actualizare a granturilor sale de brevete, care extind în continuare domeniul de aplicare dincolo de angajamentele anterioare.

La 31 martie 2016, Microsoft a anunțat la Microsoft Build că vor relicva complet Mono sub o licență MIT chiar și în scenarii în care era nevoie de o licență comercială.

15

**3.2.2 ASP.Net**

ASP.NET este un tip de aplicație web server-side concepută pentru dezvoltarea aplicațiilor web ce are ca scop producerea unor pagini web dinamice. Acesta a fost dezvoltat de Microsoft pentru a permite programatorilor să construiască site-uri web dinamice, aplicații web și servicii web. A fost lansat pentru prima dată în ianuarie 2002 cu versiunea 1.0 a .NET Framework și este succesorul tehnologiei Microsoft Active Server Pages (ASP).

ASP.NET este construit pe baza Common Language Runtime (CLR), permițând programatorilor să scrie cod ASP.NET folosind orice limbaj .NET acceptat.[5]

Framework-ul acestei extensii ASP.NET SOAP permite componentelor ASP.NET să proceseze mesaje SOAP. Succesorul ASP.NET este ASP.NET Core.

Acest succesor reprezinta o re-implementare a ASP.NET ca un framework web modular, împreună cu alte framework-uri ca Entity Framework.

Noul framework utilizează noua platformă .NET Compiler (codename "Roslyn") care este platformă transversală. ASP.NET MVC, API Web ASP.NET și pagini Web ASP.NET (o platformă care utilizează numai pagini Razor) s-au îmbinat într-un MVC 6 unificat .

Imaginea de mai jos reprezintă un logo al tehnologiei ASP.NET.



**3.2.3 Web API**

Un API este o [interfață](https://es.wikipedia.org/wiki/Interfaz) de [programare](https://es.wikipedia.org/wiki/Programaci%C3%B3n) a aplicațiilor (API engleză: Application Programming Interface).

Acesta este un set de rutine care oferă acces la caracteristici ale unui anumit [software.](https://es.wikipedia.org/wiki/Software)

16

Pe web, API sunt publicate de site-uri pentru a oferi posibilitatea de a efectua o acțiune sau de a avea acces la orice caracteristică sau conținutul furnizat de site-ul. [4]

Unele dintre cele mai cunoscute sunt API-uri:

**1)Client Side**

Un web API client-side este o interfață de programare folosită pentru a extinde funcționalitatea unui browser web sau un alt client HTTP.

Inițial,acesta a fost cel mai frecvent folosite sub forma de nativ plug-in în extensii de browser. Google a creat arhitectura nativă, care este proiectată pentru a ajuta la înlocuirea plug-in-

urilor nesigure cu extensii native sigure sandboxed și aplicații.

Ei au făcut această arhitectură prin utilizarea unui compilator LLVM AOT modificat.

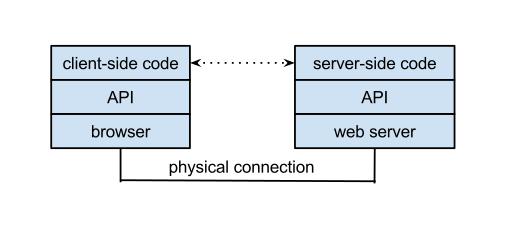
**2)Server side**

API-uri web server-side sunt interfețe pentru lumea exterioară pentru a interacționa cu logica de afaceri.

Pentru multe companii această logică de afaceri interne și de proprietate intelectuală asociate cu ea sunt ceea ce le diferențiază de alte companii, și, posibil, ceea ce le conferă un avantaj competitiv.

Ei nu doresc ca aceste informații să fie expuse. Cu toate acestea, în scopul de a oferi un API Web de înaltă calitate, trebuie să existe un nivel suficient de documentare.

Figura de mai jos reprezintă o conexiune care se stabileste între partea de client și partea de server.



17

**3.2.4 JSON**

În computere, JavaScript Object Notation sau JSON ,reprezintă un format de fișier, care utilizează un text citit de om pentru a transmite obiecte de date care constau în perechi de atribute-valoare și tipuri de date array.

Acesta este un format de date foarte comun care este folosit pentru comunicarea asincronă a browser-ului / serverului, inclusiv ca înlocuitor pentru XML în unele sisteme de tip AJAX. JSON este un limbaj independent. [22]

Acesta a fost derivat din JavaScript, dar începând din 2017 multe limbaje de programare includ codul pentru a genera și analiza date în format JSON.

Tipul media oficial media pentru JSON este aplicația / json. Numele de fișiere JSON utilizează extensia .json.

Douglas Crockford a specificat inițial formatul JSON la începutul anilor 2000; Două standarde concurente, RFC 7159 și ECMA-404, l-au definit în 2013.

Standardul ECMA descrie doar sintaxa permisă, în timp ce RFC acoperă anumite considerente de securitate și interoperabilitate.

Un profil restricționat al JSON, cunoscut sub numele de I-JSON (reprezintă o prescurtare pentru "Internet JSON"), încearcă să depășească unele probleme de interoperabilitate cu JSON. Acesta este definit în RFC 7493.

Imaginea de mai jos reprezintă un logo al acestei tehnologii JSON.



**3.2.5 Linq**

LINQ(care se pronunța "link") este o componentă Microsoft .NET Framework care adaugă capacități de interogare a datelor native a limbajelor .NET, deși porturile există pentru PHP (PHPLinq), JavaScript (linq.js), TypeScript (linq.ts ) Și ActionScript (ActionLinq) - dar nici unul dintre aceste porturi nu este strict echivalent cu LINQ în C # (de exemplu, în cazul în care

18

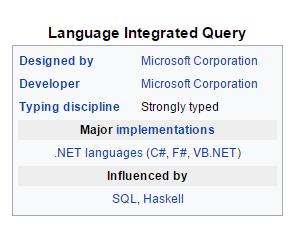
este o parte a limbajului, nu o bibliotecă externă și unde se adresează adesea unei zone mai largi de nevoi).[17]

LINQ extinde limbajul prin adăugarea de expresii de interogare, care sunt asemănătoare cu instrucțiunile SQL și pot fi folosite pentru a extrage și procesa date din arhitecturi, clase enumerabile, documente XML, baze de date relaționale și surse de date .

Imaginea de mai jos ilustrează un logo pentru această tehnologie.



Imaginea de mai jos ilustrează informații despre această tehnologie.



LINQ definește, de asemenea, un set de nume pentru metode, împreună cu regulile de traducere utilizate de compilator pentru a traduce expresii de interogare în expresii fluente folosind expresii precum lambda.

Multe dintre conceptele pe care LINQ le-a introdus au fost inițial testate în proiectul Microsoft de cercetare .

LINQ a fost lansat ca o parte importantă a .NET Framework 3.5 pe data de 19 noiembrie

2007.

19

În timp ce LINQ este implementat în primul rând ca o bibliotecă pentru .NET Framework 3.5, această tehnologie definește de asemenea extensii de limbaje opționale care fac ca interogările să fie construite ca un limbaj de primă clasă.

Aceste extensii de limbaj au fost inițial implementate în C # 3.0, VB 9.0, F # și Oxygene, cu alte limbaje, cum ar fi Nemerle.

**3.2.6 Entity framework**

Entity Framework reprezintă un set de tehnologii în ADO.NET care ajută la dezvoltarea de aplicații software orientate spre date.

Programatorii de aplicații orientate spre date s-au luptat de obicei cu necesitatea de a realiza două obiective foarte diferite.

Ele trebuie să modeleze entitățile, relațiile și logica problemelor de afaceri pe care le rezolvă și trebuie să lucreze și cu motoarele de date utilizate pentru stocarea și recuperarea datelor.[23]

Datele pot cuprinde mai multe sisteme de stocare, fiecare cu propriile protocoale;

Chiar și aplicațiile care funcționează cu un singur sistem de stocare trebuie să echilibreze cerințele sistemului de stocare cu cerințele de scriere a codului de aplicație eficient și întreținut.

Framework-ul de entități permite programatorilor să lucreze cu date sub formă de obiecte și proprietăți specifice domeniului, cum ar fi clienții și adresele clienților, fără a trebui să se preocupe de tabelele și coloanele bazei de date care stau la baza acestor date.

Imaginea de mai jos reprezintă un logo al acestei tehnologii Entity Framework.



20

**Capitolul 4**

**Dezvoltarea aplicatiei**

**4.1 Descrierea aplicatiei**

În acest capitol,am să prezint următoarele aspecte legate de etapele realizării aplicației, modul de realizare și funcționarea aplicației web.

Site-ul pe care le-am creeat reprezintă partea aplicativă a lucrării mele de licență prin intermediul căruia se configurează un magazin în mediul online, acesta începând cu puncte generale, pentru a continua mai apoi cu informatiile tehnice despre implementarea și dezvoltarea acestui proiect.

Toate aceste pagini din magazinul online funcționează dinamic , iar toate datele sunt extrase din tabelele din baza de date pe care am creeat-o inițial.

Magazinul permite promovarea produselor ,mai exact a îmbrăcămintei masculine, oferă posibilitatea clientului de a avea acces la toate informațiile referitoare la un anumit produs, posibilitatea de a adaugă acest produs de la o listă de dorințe până la coșul final de cumpărături.

De asemenea,sunt precizate informații legate de adresă,numărul de telefon și e-mailul magazinului.

Accesul la informațiile generate de această aplicație este restrictionată in funcție de postura în care se află utilizatorul.Acest site poate să fie vizualizat în postura de Vizitator,postura de Utilizator și în postura de Administrator.

Scopul proiectului prezentat în această lucrare a fost realizarea unui magazin online.

Prin tema aleasă, am considerat că descrie mult mai bine conţinutul actualizat în conformitate cu noile tendinţe şi dezvoltări din domeniul informatic abordat, lucrarea dezvăluie direct, fără echivoc, subiectul stabilit ca obiectiv principal: programarea web.

Evident că plaja de referinţă a unui asemenea obiectiv este practic imposibil de cuprins şi tratat complet, în vastitatea ei, într-o singură lucrare.

**4.2 Etapele proiectarii aplicatiei**

Pe baza cerințelor aplicației, pentru început am proiectat o Diagramă a Cazurilor de Utilizare care cuprinde enunțarea funcționalităților majore ale aplicației.

21

**4.2.1 Diagrama cazurilor de utilizare**

Diagrama cazurilor de utilizare este un tip de diagramă din care reiese modul de utilizare a sistemului informatic și anume modul în care utilizatorii interacționează cu aaplicatia.

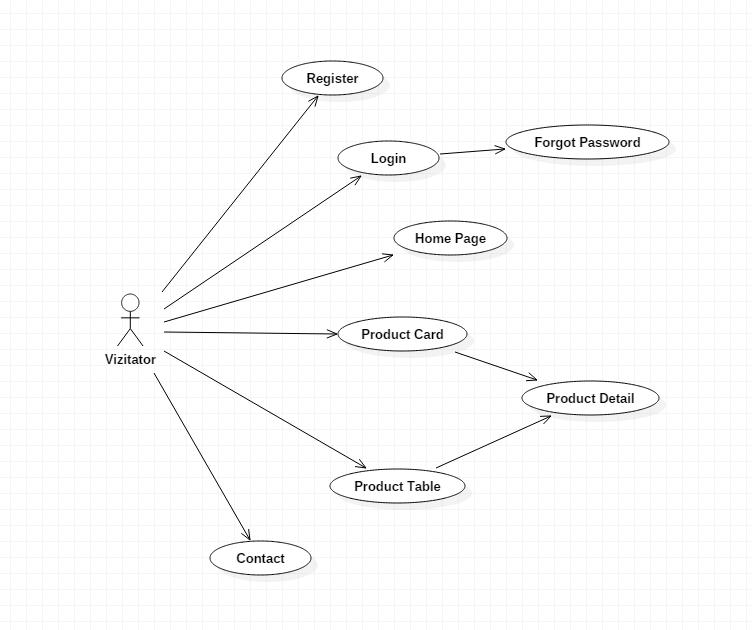
Această diagramă nu este absolut necesară pentru a scrie o specificație cu use case-uri, dar este foarte utilă pentru a ne face o imagine generală asupra sistemului.

Un actor poate să fie un utilizator sau orice sistem care poate interacționa cu sistemul modelat.

Aplicația poate să fie folosită pentru 3 tipuri de utilizatori:Vizitator,Membru și Admin.

În funcție de tipul de utilizator acesta are acces la diferite secțiuni,după cum urmează:

**a)Vizitator**

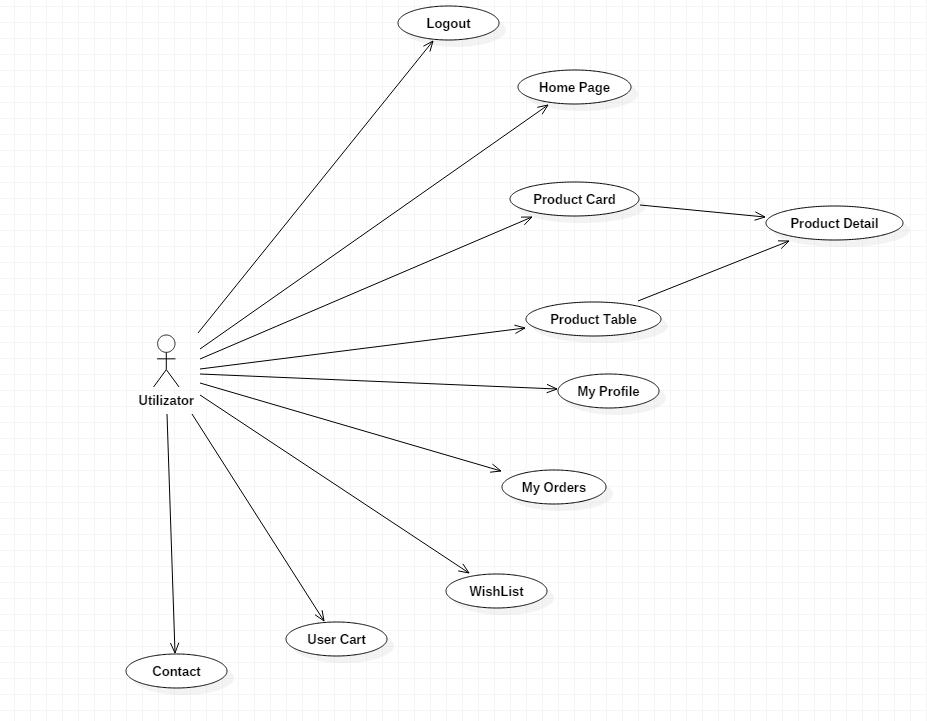


* Vede pagina unde are posibilitatea să iși creeze un cont
* Vede pagina unde se poate înregistra cu contul pe care il are
* Vede pagina unde iși poate recupera parola în cazul în care nu și-o mai amintește

22

* Vede pagina de start a acestui site
* Vede pagina unde sunt afișate produsele în diferite forme
* Vede pagina unde sunt afișate informații despre un anumit produs
* Vede pagina de contact unde are la dispoziție informații despre site și date de contact

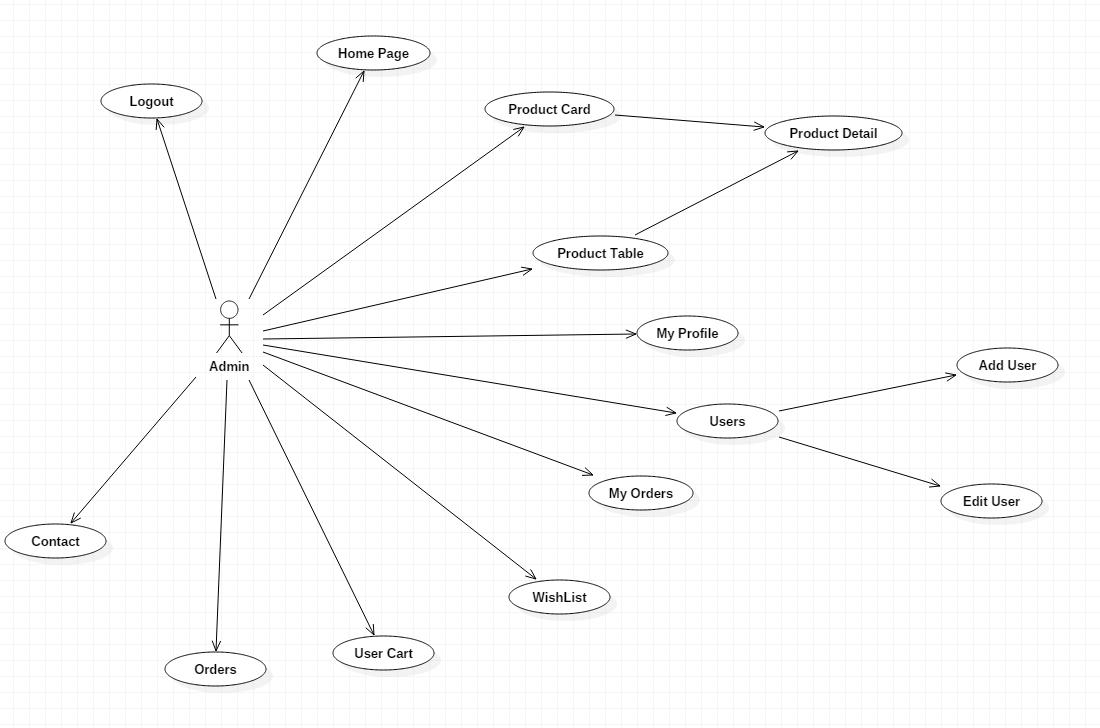
**b)Utilizator**



* Vede pagina de start a acestui site
* Vede pagina unde poate să iasă de pe cont
* Vede pagina unde sunt afișate produsele în diferite forme
* Vede pagina unde sunt afișate informații despre un anumit produs
* Vede pagina unde poate să vizualizeze și să modifice informații despre profil
* Vede pagina unde poate să vizualizeze comenzile pe care acesta le-a facut
* Vede coșul de cumpărături pe care acesta îl are din momentul în care și-a creeat contul
* Vede lista de dorințe unde poate să iși adauge produse pe care ar dori să le achiziționeze
* Vede pagina de contact unde are la dispozitie informații despre site și date de contact

23

**c)Admin**

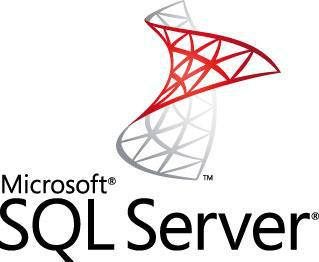


* Vede pagina de start a acestui site
* Vede pagina unde poate să iasă de pe cont
* Vede pagina unde sunt afișate produsele în diferite forme
* Vede pagina unde sunt afișate informații despre un anumit produs
* Vede pagina unde poate să iși editeze informațiile legate de profilul sau
* Vede pagina unde poate să vadă toți utilizatorii care se află pe site
* Vede pagina unde poate să adauge un utilizator sau să editeze informații legate de un utilizator
* Vede pagina unde sunt toate comenzile pe care le fac utilizatorii,el având posibilitatea să le accepte sau refuze
* Vede pagina unde poate să vizualizeze comenzile pe care acesta le-a facut
* Vede coșul de cumpărături pe care acesta îl are din momentul în care și-a creeat contul
* Vede lista de dorințe unde poate să își adauge produse pe care ar dori să le achiziționeze
* Vede pagina de contact unde are la dispoziție informații despre site și date de contact

24

**4.2.2 Baza de date**

O bază de date este o colecţie de date, creată şi menţinută computerizat, în scopul prelucrării datelor în contextul unui set de aplicaţii. Prelucrarea datelor se referă la operaţiile de introducere, ştergere, actualizare şi interogare a datelor.[1]



Microsoft SQL Server este un sistem de management al bazelor de date relaționale dezvoltat de Microsoft.

Ca server de baze de date, este un produs software cu funcția principală de stocare și recuperare a datelor, așa cum este solicitat de alte aplicații software, care pot fi difuzate pe același computer sau pe alt computer dintr-o rețea (inclusiv pe Internet).

Imaginea de mai sus reprezinta un logo al acestei tehnologii Microsoft SQL Server. Microsoft comercializează cel puțin o duzină de ediții diferite ale Microsoft SQL Server,

destinate publicului diferit și pentru sarcini de lucru, de la aplicații mici, cu o singură mașină, până la aplicații mari, orientate către Internet, cu mulți utilizatori simultani.

O bază de date este alcătuită dintr-unul sau mai multe tabelele. Comanda CREATE TABLE este utilizată în momentul în care se dorește crearea unui tabel,în timp ce comanda ALTER este utilizată pentru adăugarea, ștergerea sau modificarea coloanelor într-un tabel existent. Fiecare tabel este alcătuit din coloane sau câmpuri având atribute, o lista de elemente. Fiecare coloană conține o anumită informație despre fiecare element din tabel. Cheia primară este un atribut alcătuit din valori distincte. Atunci când creăm o tabela și dorim să specificăm o coloană ca fiind cheie primară folosim specificatorul “primary key”.Cheia unică este constituită dintr-un singur câmp,iar pentru aceasta utilizăm specificatorul “unique key”.

Comenzile standard SQL sunt următorarele: ADD,DROP,INSERT și UPDATE.

25

Pentru inserarea datelor într-o bază de date apelăm la comanda INSERT,comanda UPDATE actualizează date ,DELETE șterge datele,iar comanda SELECT se folosește pentru a extrage date din baza respectivă. Comanda DISTINCT utilizată în cazul în care sunt coloane cu valori duplicate și dorim să facem distincție între acestea.SQL folosește **‘ ’** pentru delimitarea valorilor de tip text sau string.Valorile numerice nu se delimitează prin **‘ ’** sau **“ ”**.

Limbajul SQL acceptă cinci tipuri de funcții: SUM,AVE , MIN,MAX,COUNT,sunt aplicate pentru o coloană întreagă și totodată returnează o singură valoare.

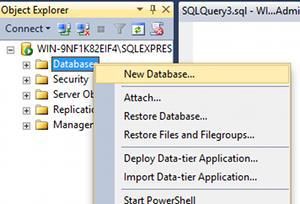
Pentru realizarea bazei de date eu am folosit Microsoft SQL Server 2014.

Baza de date pentru aceasta aplicatie a fost realizata pe baza urmatorilor pasi de mai jos:

**Pasul 1:**

-vom creea o bază de data care va avea numele **ForMEN**

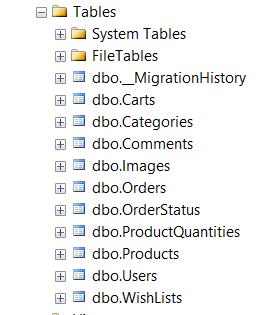
-în această bază de date vom stoca informații legate de produsele pe care le vom prezenta -imaginea respectivă reprezintă începutul proiectării unei baze de date



**Pasul 2:**

-vom creea tabelele cu datele necesare pentru proiectul nostru

-imaginea respectivă reprezintă tabelele pe care le are această aplicație

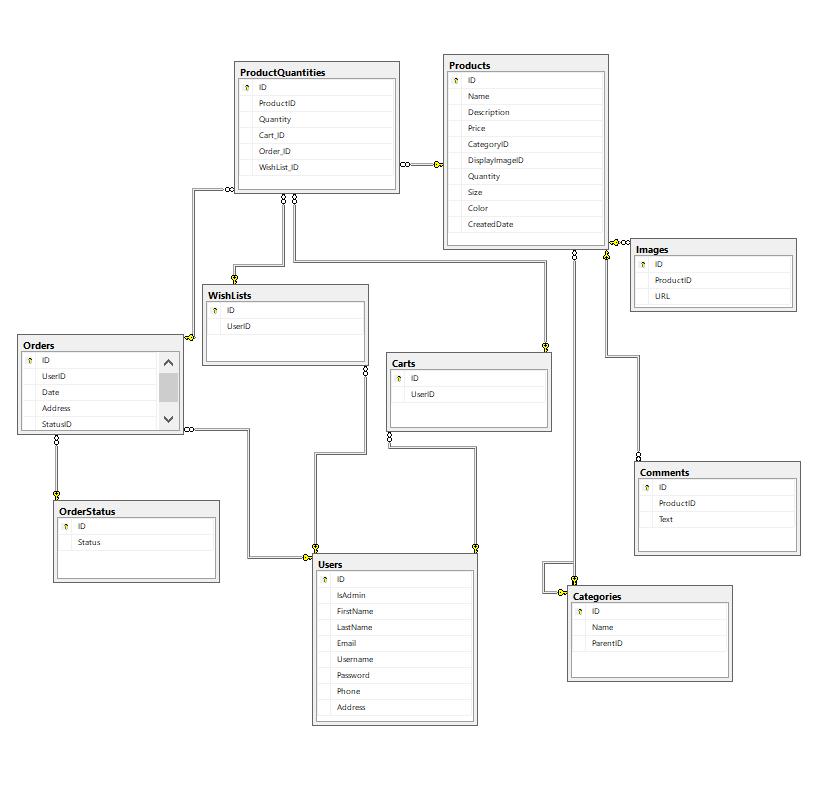


26

**Pasul 3:**

-vom reprezenta relațiile dintre aceste tabele printr-o diagramă și vom explica logica din spatele acestei diagrame

-imaginea respectivă reprezintă diagrama tabelelor din baza noastră de date

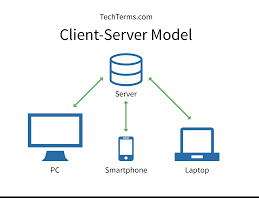


27

**4.3 Descriere arhitectura**

În această parte a capitolului 4 urmează să descriem arhitectura aplicației:

**1.Models**



Imaginea de mai sus reprezintă un server care trimite și primeste informații de la diferite dispozitive.

Fiecare Model conține o clasă care are mai multe atribute specifice clasei.

Clasele sunt urmatoarele:

-User-conține atribute și metode specifice pentru un utilizator

-Cart- conține atribute și metode specifice pentru coșul de cumpărături al unui utilizator -Category- conține atribute și metode specifice pentru o categorie -Product- conține atribute și metode pentru despre un produs

-ProductQuantity- conține atribute și metode specifice pentru cantitatea unui produs -Image- conține atribute și metode specifice pentru imaginea unui produs

-Comment- conține atribute și metode specifice pentru diferite comentarii despre un produs

-Order- conține atribute și metode specifice pentru a face o comandă la un produs

-OrderStatus- conține atribute și metode specifice pentru statusul unei comenzi

-Wishlist- conține atribute și metode specifice pentru creearea unei liste de dorințe

Pentru fiecare atribut avem 2 metode specifice care ne vor ajuta mai departe în realizarea proiectului.Aceste metode sunt {get;set}.

**User.cs**--User-ID

-IsAdmin

28

-FirstName

-LastName

-Email

-Username

-Password

-Phone

-Adress

--ResponseUser-ID

-IsAdmin

-FirstName

-LastName

-Email

-Username

-Phone

-Adress

**Cart.cs**--Cart-ID

-UserID {ForeignKey(UserID)}

-Products(virtualList)

**Category.cs**--Category-ID

-Name

-ParentID {ForeignKey(ParentID)} **Product.cs**--Product-ID

-Name

-Description

-Price

-CategoryID {ForeignKey(CategoryID)}

-DisplayImageID

-Quantity

-Size

-Color

-Date

**ProductQuantity.cs**--ProductQuantity-ID

-ProductID {ForeignKey(ProductID)}

29

-Quantity

**Image.cs**--Image-ID

-ProductID {ForeignKey(ProductID)}

-URL

**Comment.cs**--Comment-ID

-ProductID {ForeignKey(ProductID)}

-Text

**Order.cs**--Order-ID

-UserID {ForeignKey(UserID)}

-StatusID {ForeignKey(StatusID)}

-Date

-Products(virtualList)

**OrderStatus.cs**--OrderStatus-ID

-UserID {ForeignKey(UserID)}

-Status {ForeignKey(StatusID)}

-DateTime

-Adress

-Products(virtualList)

**WishList.cs**--WishList-ID

-UserID {ForeignKey(UserID)}

-Products(virtualList)

**2)Controllers**

Fiecare controller de mai jos reprezintă funcționalitățile care se pot executa în Postman.Aceste requesturi sunt de tipul CRUD.

CRUD înseamnă Create(POST),Read(Get),Update(Put),Delete(Sterge)

Fiecare controller conține cele 5 metode:

-Add

-Get all

-Get by Id

-Update

30

-Delete

Aceste controllere sunt generate de .Net pe baza modelelor pe care le-am explicat mai sus.

**User**---POST(Create)-Add

-Login

---GET(Read)-Get all

-Get by ID

---PUT(Update)-Update

---DELETE(Delete)-Delete

**Cart**---POST(Create)-Add products

---GET(Read)-Get by user ID

---PUT(Update)-Update products quantity

---DELETE(Delete)-Delete by user ID **Categories**---POST(Create)-Add

---GET(Read)-Get all

-Get by ID

---PUT(Update)-Update

---DELETE(Delete)-Delete

**Images**---POST(Create)-Add

---GET(Read)-Get all

-Get by ID

---PUT(Update)-Update

---DELETE(Delete)-Delete

**Comments**---POST(Create)-Add

---GET(Read)-Get all

-Get by ID

---PUT(Update)-Update

---DELETE(Delete)-Delete

**Orders**---POST(Create)-Add order

---GET(Read)-Get all

-Get by user ID

---PUT(Update)-Update order status

31

**WishList**---POST(Create)-Add products

---GET(Read)-Get by user ID

---PUT(Update)-Remove products

---DELETE(Delete)-Empty products

**3)Migrations**

Dupa ce am creeat un tabel în baza noastra de date poate exista posibilitatea de a mai dori să adăugăm ceva atribut în acesta.Pentru a realiza acest lucru folosim migrations care ne ajută să rezolvăm totul mult mai repede.

**Initial.cs**-creeazăcele 3 tabele {Categories,Comments,Products} cu datele specifice care ne vorajuta foarte mult pentru realizarea acestui site.

CreateTable{Categories}

-ID(int) PrimaryKey

-Name(String)

-ParentID(Int)

CreateTable{Comments}

-ID(Int) PrimaryKey

-ProductID(Int)

-Text(String)

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

-Price(Decimal)

-CategoryID(Int)

32

**Image.cs**-creeazătabelul {Images} iar apoi introduce în tabelul de mai sus {Products} o nouăcoloană care are ca proprietate oferirea unei imagini specifice produsului pe care dorim să îl achiziționăm.

CreateTable{Images}

-ID(Int) PrimaryKey

-ProductID(Int)

-URL(String)

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

-Price(Decimal)

-CategoryID(Int)

-DisplayImageID

**UserCartOrder.cs**-creeazătabelele {Carts,ProductQuantity,Orders,Users} unde vom putea găsidate despre fiecare user.Fiecare user o să aibă un coș unde poate să adauge comenzi pe care acesta urmează să le facă.Userului i se oferă informații despre produsul pe care acesta dorește să îl achiziționeze,inclusiv despre cantitatea care se află pe stoc.

CreateTable{Carts}

-ID(Int) PrimaryKey

-UserID(Int)

CreateTable{ProductQuantity}

-ID(Int) PrimaryKey

-Quantity(Int)

-CartID(Int)

-OrderID(Int)

-ProductID(Int)

33

CreateTable{Orders}

-ID(Int) PrimaryKey

-UserID(Int)

CreateTable{Users}

-ID(Int) PrimaryKey

-IsAdmin(Boolean)

-FirstName(String)

-LastName(String)

-Email(String)

-Username(String)

-Password(String)

**OrderStatus.cs**-adaugă unele detalii esențiale despre starea unei comenzi.

CreateTable{OrderStatus}

-ID(Int) PrimaryKey

-Code(Int)

-Status(String)

CreateTable{Orders}

-ID(Int) PrimaryKey

-UserID(Int)

-StatusCode(Int)

**ProductFields.cs**-adaugăunele detalii esențiale despre un produs oarecare.Scopul principal esteca pe baza acestor informații userul să poată să achiziționeze produsul perfect și care să fie la nivelul așteptărilor sale.

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

34

-Price(Decimal)

-CategoryID(Int)

-DisplayImageID

-Quantity(Int)

-Size(String)

-Color(String)

**OrderFields.cs**-adaugăinformații despre o comandăcare urmeazăsăfie procesată și apoiexpediată către cumpărător.

CreateTable{Orders}

-ID(Int) PrimaryKey

-UserID(Int)

-Date(DateTime)

-Adress(String)

**ForeignKeys.cs**-creeazăcoloane noi în tabelele existenteși seteazăcheile străine pentru toatetabelele pe care le avem deja.

CreateTable{Orders}

-ID(Int) PrimaryKey

-UserID(Int) ForeignKey

-Date(DateTime)

-Adress(String)

-StatusID(Int) ForeignKey

CreateTable{Categories}

-ID(int) PrimaryKey

-Name(String)

-ParentID(Int)

CreateTable{Carts}

-ID(Int) PrimaryKey

35

-UserID(Int) ForeignKey

CreateTable{ProductQuantity}

-ID(Int) PrimaryKey

-Quantity(Int)

-CartID(Int) ForeignKey

-OrderID(Int) ForeignKey

-ProductID(Int) ForeignKey

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

-Price(Decimal)

-CategoryID(Int) ForeignKey

-Quantity(Int)

-Size(String)

-Color(String)

CreateTable{Comments}

-ID(Int) PrimaryKey

-ProductID(Int) ForeignKey

-Text(String)

CreateTable{Images}

-ID(Int) PrimaryKey

-ProductID(Int) ForeignKey

-URL(String)

**UseFields.cs**-adaugă coloana “Phone” in tabela existenta “Users”

CreateTable{Users}

-ID(Int) PrimaryKey

36

-IsAdmin(Boolean)

-FirstName(String)

-LastName(String)

-Email(String)

-Username(String)

-Password(String)

-Phone(String)

**WishList.cs**-creează tabelul “WishList” și seteazălegaturi cu alte tabele pentru a funcționacorespunzator

CreateTable{WishList}

-ID(Int) PrimaryKey

-UserID(Int)

CreateTable{ProductQuantity}

-ID(Int) PrimaryKey

-Quantity(Int)

-CartID(Int) ForeignKey

-OrderID(Int) ForeignKey

-ProductID(Int) ForeignKey

-WishListID(Int) ForeignKey

**DisplayImageID.cs**-creeazăo nouăcoloanăîn tabelul “Products” care oferăposibilitatea de aadăuga poze la un anumit produs

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

-Price(Decimal)

-CategoryID(Int) ForeignKey

-Quantity(Int)

37

-Size(String)

-Color(String)

-DisplayImageID(Int) ForeignKey

**UserAdress.cs**-adaugăo nouăcoloanăîn tabelul “Users” care oferăposibilitatea userului săișiadauge adresa unde acesta locuiește

CreateTable{Users}

-ID(Int) PrimaryKey

-IsAdmin(Boolean)

-FirstName(String)

-LastName(String)

-Email(String)

-Username(String)

-Password(String)

-Phone(String)

-Adress(String)

**ProductCreateDate.cs**-adaugădata la care a fost creeat un anumit produs.

CreateTable{Products}

-ID(Int) PrimaryKey

-Name(String)

-Description(String)

-Price(Decimal)

-CategoryID(Int) ForeignKey

-Quantity(Int)

-Size(String)

-Color(String)

-DisplayImageID(Int) ForeignKey

-CreateDate(DateTime)

38

**Configuration.cs**-ilustreazătoate etapele prin care trebuie sătreacăo comandăpe care o face unuser.Fiecare etapă are un cod specific iar în momentul în care o comandă trece la etapa urmatoare,se modifică si codul

0={ Status = "In procesare" },

1={ Status = "Anulat" },

2={ Status = "Livrat" },

3={ Status = "Receptionat" },

4={ Status = "Returnat" }

**Î**n urmatoarea parte urmeazăsăfacem o descriere a arhitecturii a acestei aplicatii dinpostura de programator.

1. **Dezvoltarea partii de back-end**

In procesul de dezvoltare a aplicației am început cu partea de back-end.

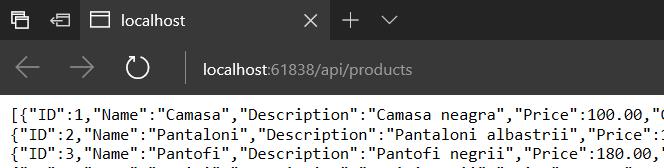
Utilizănd framework-ul ASP.NET Web API am construit un serviciu care prin endpoint-uri expune informațiile necesare sub formă de modele în format JSON și permite manipularea acestora prin operații CRUD.

Un model este o clasă care cuprinde atributele unei entități.

Primul pas a fost crearea partială a modelelor, dupa care s-au mai adaugat sau modificat, pe baza cărora au fost create ulterior controllere pentru fiecare dintre acestea.

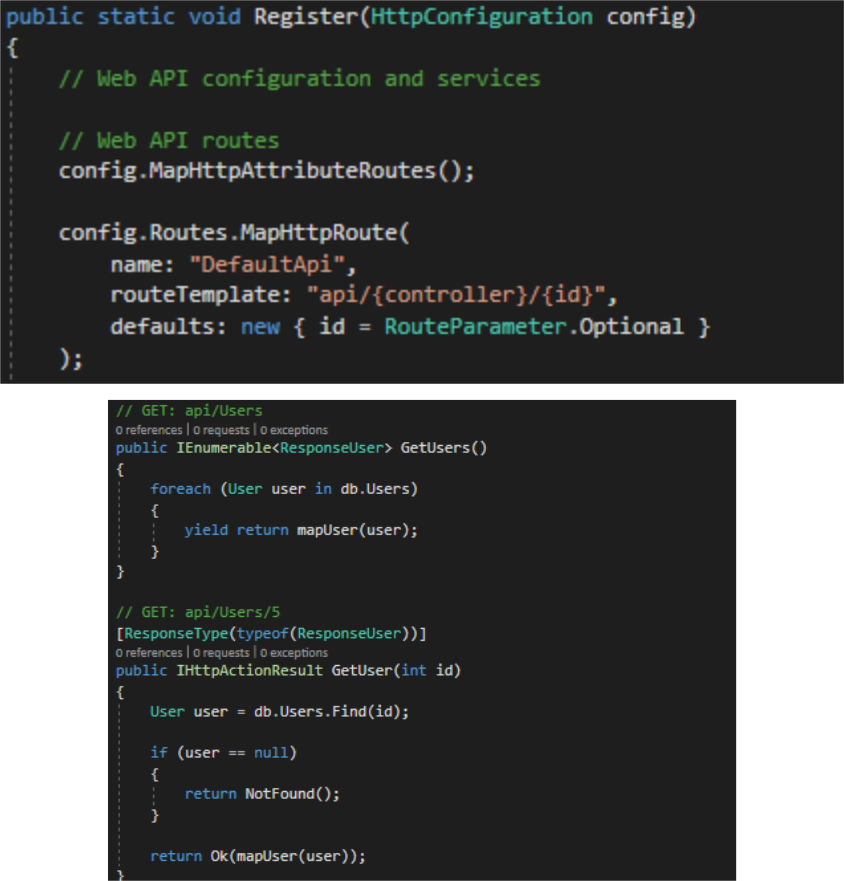
De exemplu, pentru modelul User avem UsersController. Un controller crează funcțiile care vor fi accesate de endpoint-uri în funcție de requestul dorit.

De exemplu, pe endpoint-ul „/api/products” se pot face request-uri de tip GET sau POST (dar si altele) iar controllerul va asocia aceste requesturi cu funcția corespunzatoare lor.

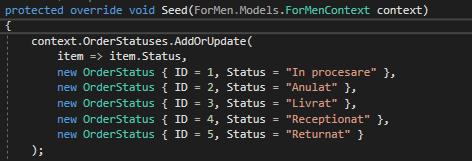


39

Funcția Register din fisierul WebApiConfig.cs și funcția RegisterRoutes din fișierul RouteConfig.cs permite manipularea rutelor. Aceste funcții ajută la construirea de URL-uri clare și specifice cum ar fi „/api/users” sau „/api/users/1”.



Functia Seed din fișierul Configuration.cs permite inserarea de date implicite la crearea bazei de date. O folosim pentru a adăuga statusurile comenzilor imediat dupa crearea tabelelor.



40

Migrațiile sunt un sistem de versionare a schemei bazei de date care permit controlarea acesteia mult mai usor decât daca am crea-o manual.

Baza de date a fost generată și modificată prin migratii astfel:

În consola .Net Package Manager s-au rulat comenzile:

* Enable-Migrations

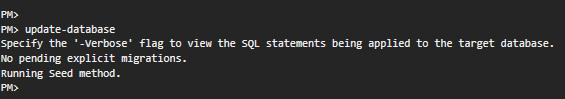
Pentru activarea sistemului de migrații

* Add-Migration NumeMigratie Pentru a adăuga o nouă migrație
* Update-Database

Pentru a aduce baza de date la ultima versiune a migrațiilor

* Update-Database –TargetMigration NumeMigrație

Pentru a aduce baza de date la o anumită versiune a migrațiilor și a putea șterge migrațiile ulterioare acesteia



Clasa speciala ForMenContext este un model de acces la informațiile bazei de date. Utilizând tehnologia Linq a .Net-ului în combinația cu acest model putem manipula cu ușurintă informațiile tuturor tabelelor.

De exemplu, pentru a selecta un utilizator cu toate proprietățile acestuia avem nevoie de o singură linie de cod:



Pentru accesul la baza de date, backend-ul (serviciul Web API) a trebuit configurat cu datele de autentificare la respectiva baza de date. Acest lucru se întâmplă în fișierul Web. Config în secțiunea <connectionStrings>



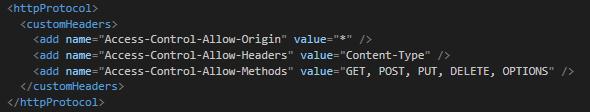
Conexiunea se realizează prin credențialele sitemului de operare, prin securitatea integrată, nefiind nevoie de specificare unei adrese a bazei de date, a unui utilizator și a unei parole întrucât atât baza de date cât și serviciul Web API se află (momentan) pe același server sau cu alte cuvinte pe acelasi laptop.

Tot in fisierul Web.config am configurat serviciul cu tipurile de request-uri acceptate.

41

Am avut nevoie de requesturi de tipuri GET, POST, PUT, DELETE așa că le-am adugat în secțiunea <httpProtocol>

<customHeaders>

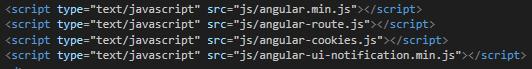


Suplimentar am specificat headerul OPTIONS utilizat pentru asigurarea faptului că un request de un anumit tip poate fi executat în cadrul serviciului.

1. **Dezvoltarea partii de front-end**

Deoarece am folosit framework-ul AngularJS, primul lucru în dezvoltarea părtii de front-end a fost includerea resurselor necesare în proiect.

Javascript:



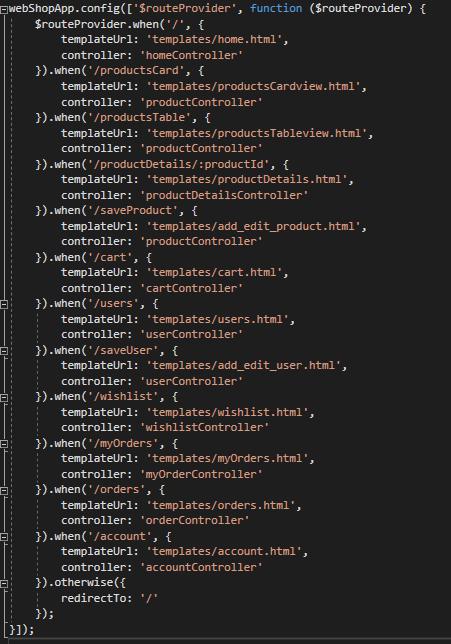
CSS:



Definirea aplicației incluzând dependențele acesteia și a rutelor a fost lucrul imediat următor ce a trebuit făcut:



42



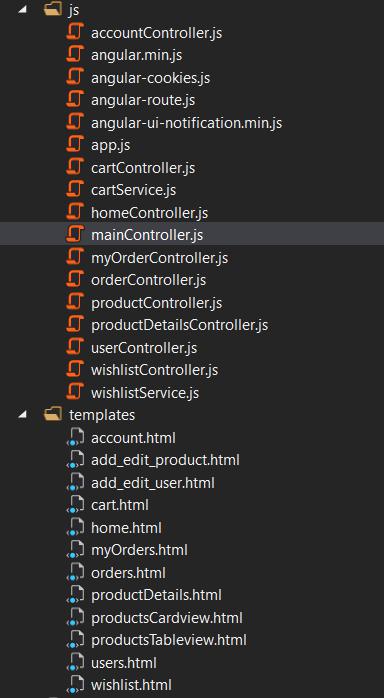
AngularJS utilizează modelul MVVM (Model-View View-Model) care este foarte asemănător modelului MVC.

După cum se poate observa și în rute, pentru fiecare fisier HTML avem un controller.

Fișierele HTML ale proiectului reprezintă view-urile care se încarcă la accesarea URL-urilor asociate lor odata cu controllerul specific fiecaruia salvat într-un fișier JS.

Am creat de asemenea și doua servicii Angular pentru scalabilitatea proiectului și usurarea dezvoltării acestuia: CartService si WishListService.

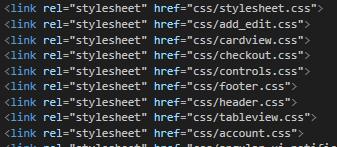
43



De exemplu, pe ruta „#!/cart” se incarca:

* View-ul „cart.html”
* Controller-ul „cartController”

Partea de design a fost organizata in fisiere CSS dupa cum urmeaza:



Proprietatea Angular de „dual-binding” a fost un ajutor esențial în construirea aplicației. Angular asociază proprietățile de pe variabile $scope a fiecărui controller cu variabile din

model astfel că dacă o variabilă este modificată într-unul dintre capete această modificare este vizibilă instant la celălalt capăt.

De exemplu:

In controllerul UsersController avem: $scope.users = valoare Această „valoare” este vizibilă în model, in variabila „users”

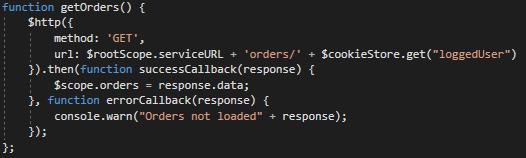
44

* Daca $scope.users se modifică în controller (de exemplu în momentul primirii unui raspuns de la server), se asociază automat variabilei users din model noua valoare
* Daca users din model se schimbă prin interacțiunea utilizatorului, $scope.users din controller se actualizează automat cu noua valoare.

Pentru management-ul sesiunii unui utilizator am folosit dependinta $cookieStore care permite adăugarea, editarea și ștergerea fișierelor cookie.

In schimbul mesajelor de alertă am ales utilizarea modulului Notification pentru a notifica utilizatorul cu privire la rezultatul acțiunilor sale.

Un request la serviciul Web API al aplicatiei arată de forma:



Imbunătățiri ulterioare ale aplicatie se pot face pe partea de securitate. Fiind o aplicație de tipul „Single Page Application”, mult mai multe informații sunt expuse și pot fi manipulate prin utilitare de editare a documentelor cookie sau prin schimbare de variabile prinse în breackpoint-uri ale codul javascript.

1. **Dezvoltarea partii de stocare de date**

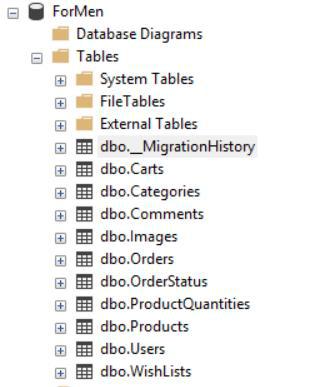
Cu această parte am avut cel mai putin de lucru considerând faptul că migrațiile au făcut munca efectivă de creare a bazei de date, de tabele și de relații între tabele.

Am utilizat framework-ul Entity în serviciul Web API iar baza de date a fost creată dupa idea „code-first”.

Munca a constat în crearea cu grijă a modelelor din serviciu astfel încât Entity să poată mapa modelele intr-o baza de date.

Baza de date rezultată arată in felul urmator:

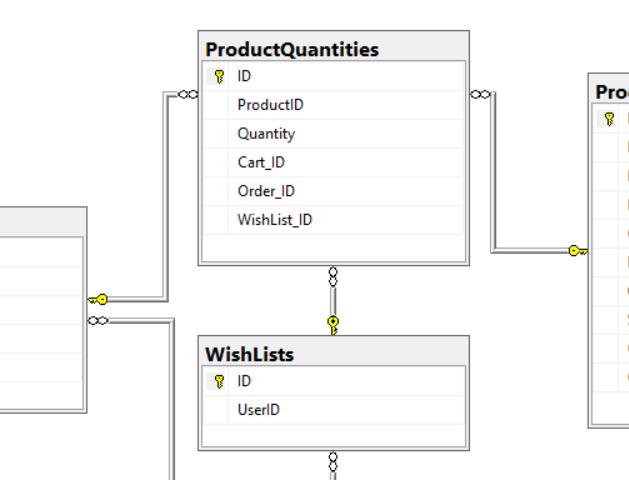
45



Toate tabelele au ID ca cheie primară iar după necesitate, unele tabele au chei străine.

Am utilizat relații de tipurile:

* One to many
* Many to many



46

**4.4 Aplicarea instrumentelor utilizate pentru dezvoltare**

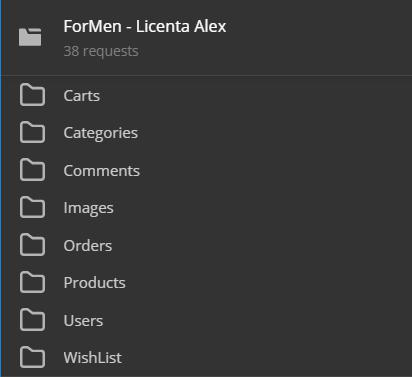
**POSTMAN**



Postman este o aplicație Google Chrome care interacționează cu API-urile HTTP. Această aplicație ne oferă o interfață grafică prietenoasă pentru construirea de cereri și răspunsuri la lectură. Persoanele din spatele Postman oferă de asemenea un pachet add-on numit Jetpacks, care include câteva instrumente de automatizare și, cel mai important, o bibliotecă de testare Javascript. Postman reprezintă o modalitate mult mai simplă și mai rapidă de a interacționa cu codul pe care noi l-am scris,face să fie totul mult mai ușor și mult mai atractiv.[10]

Pentru lucrarea mea de licența,această aplicație Postman înseamnă o reprezentare a models-urilor despre care am vorbit mai sus,a controller-elor cu toate metodele pe care le-am implementat și a migrations-urilor care m-au ajutat să adaug informații esențiale pentru fiecare model în parte.In această imagine vom ilustra modul în care sunt prezentate folderele in Postman.

Fiecare folder conține metodele specifice pe care le poate face,fiind un total de 38 de metode care pot să fie executate.



47

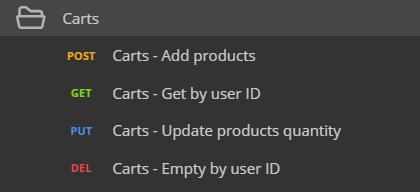
Mai departe urmează să ilustram o imagine cu fiecare folder în parte și să descriem funcționalitățile care se află în fiecare folder.

**1.Carts**-reprezintăcoșul de cumpărături care are urmatoarele funcționalităti:

-Add products-adaugă produse în coș

-Get by user ID-returnează coșul unui user în funcție de ID-ul acestuia

-Update products quantity-modifică cantitatea de produse pe care un utilizator le are în coș -Empty by user ID-golește coșul unui utilizator pe baza ID-ului acestuia



**2.Categories**-reprezintăcategoriile de produse care se aflăpe site.Acestea au următoarelefuncționalităti:

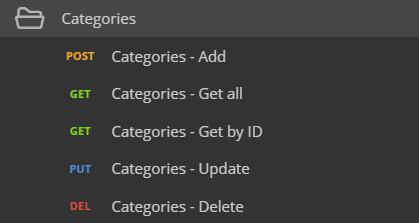
-Add-adaugă o categorie de produse

-Get All-ne arată toate categoriile de produse care se află pe site

-Get by ID-ne arată o anumită categorie în funcție de ID

-Update-modifică o categorie

-Delete-șterge o categorie



**3.Comments**-reprezintăcomentarii pe care le putem adăuga la produse.Acestea au

urmatoarele funcționalităti:

-Add-adaugă comentariu

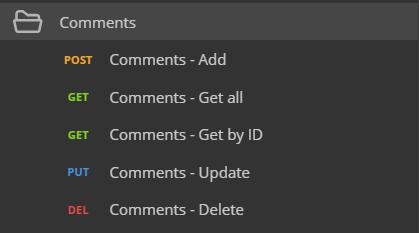
-Get all-arată toate comentariile

-Get by ID-afișează un comentariu dupa ID

-Update-modifică un comentariu

-Delete-șterge un comentariu

48



**4.Images**-reprezintăimagini pe care le putem adăuga la fiecare produs în parte.Acestea au

urmatoarele funcționalităti:

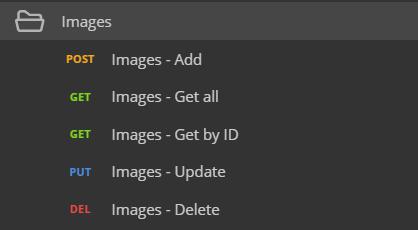
-Add-adaugă imagini

-Get all-afișează toate imaginile

-Get by ID-afișează dupa ID

-Update-modifică o imagine deja pusă

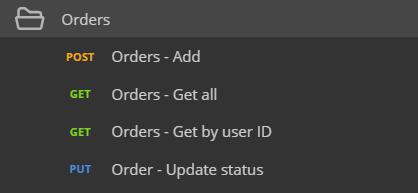
-Delete-șterge o imagine



**5.Orders**-reprezintăcomenzi pe care un user le poate face.Acestea au următoarelefuncționalităti:

-Add-adaugă o comandă

-Get all-afișează toate comenzile care sunt făcute pe site -Get by user ID-afișează o comandă dupa ID-ul userului -Update status-modifică statutul unei comenzi



49

**6.Products**-reprezintăprodusele care urmeaza săfie adăugate pe site.Acestea au

următoarele funcționalităti:

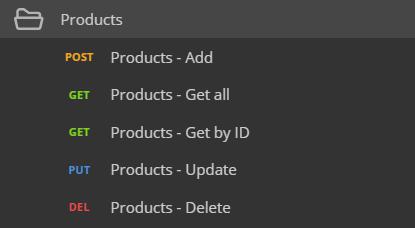
-Add-adaugă un produs

-Get all-afișează toate produsele

-Get by ID-afișează un produs dupa ID

-Update-modifică un produs

-Delete-șterge un produs



**7.Users**-reprezintăutilizatorii care sunt sau urmeazăsăfie pe site.Acest folder areurmătoarele funcționalităti:

-Add-adaugă un utilizator nou

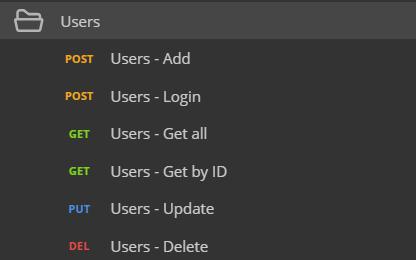
-Login-autentificarea unui utilizator pe site

-Get all-afișează toți utilizatorii de pe site

-Get by ID-afișează un utilizator dupa ID

-Update-modifică informații despre un utilizator

-Delete-șterge un utilizator



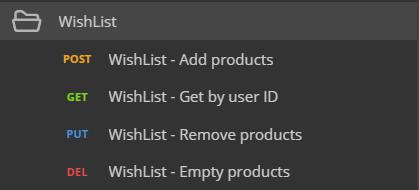
**8.WishList**-reprezintălista de dorințe a unor produse pe care un utilizator ar dori săleachiziționeze în viitor.Acest folder are urmatoarele funcționalităti: -Add products-adaugă produsele în listă

50

-Get by user ID-afișează lista de dorinte a unui utilizator dupa ID

-Remove products-șterge produsele din lista de dorinte

-Empty products-îndepărtează produsele din lista de dorinte

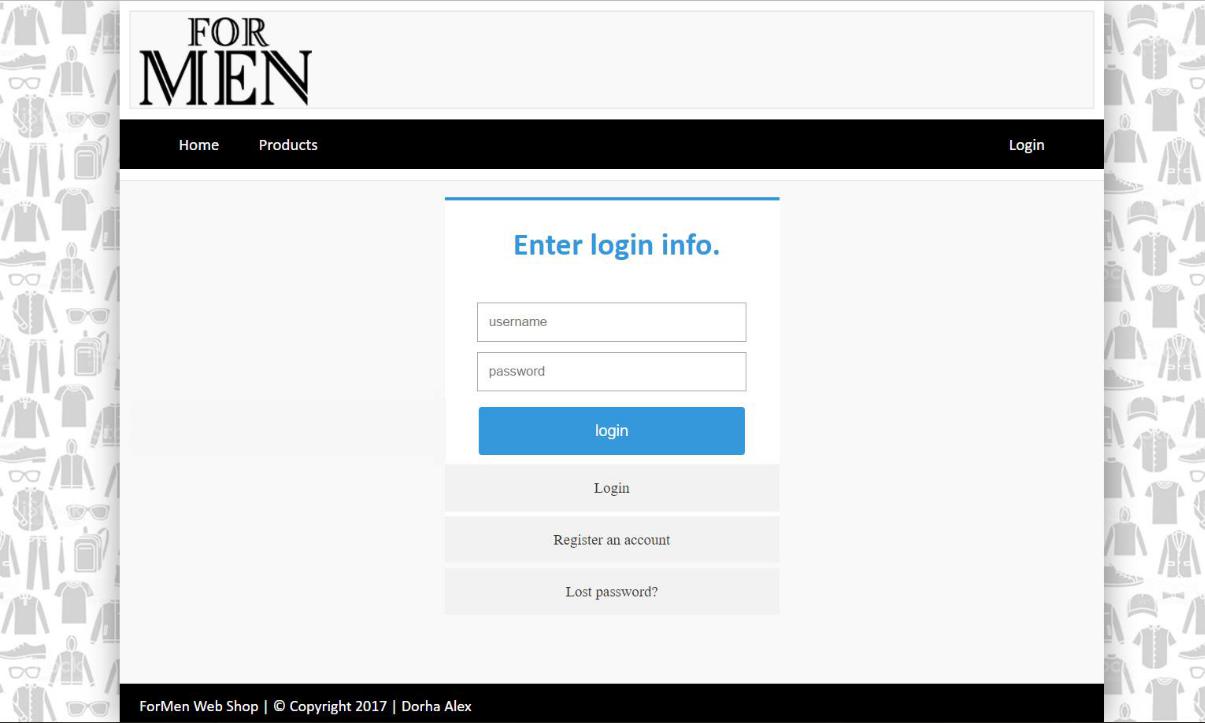


**4.5 Manual de utilizare al aplicatiei**

În acest capitol am să prezint modul efectiv de realizare și funcționare al aplicației web. Pentru a înțelege mai bine despre ce este vorba am să prezint prima dată rezultatul grafic pentru magazinul online.

Pentru început vom adăuga o imagine care reprezintă informațiile care sunt oferite de către aplicație daca este accesată în postura de Vizitator.

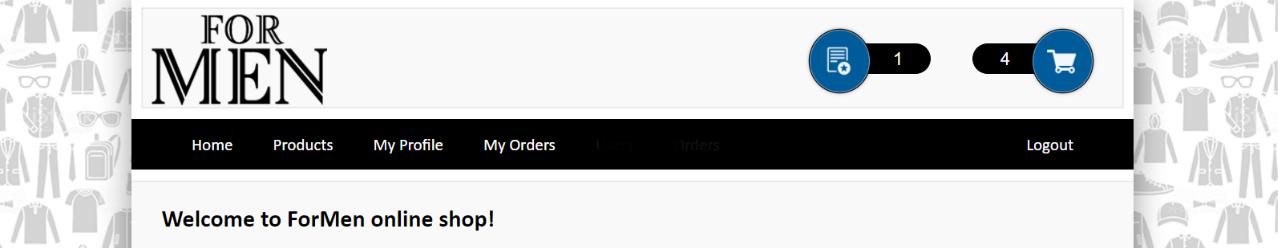
În postura de vizitator acesta trebuie să își creeze un cont pentru a avea acces la mai multe informații de pe acest site.



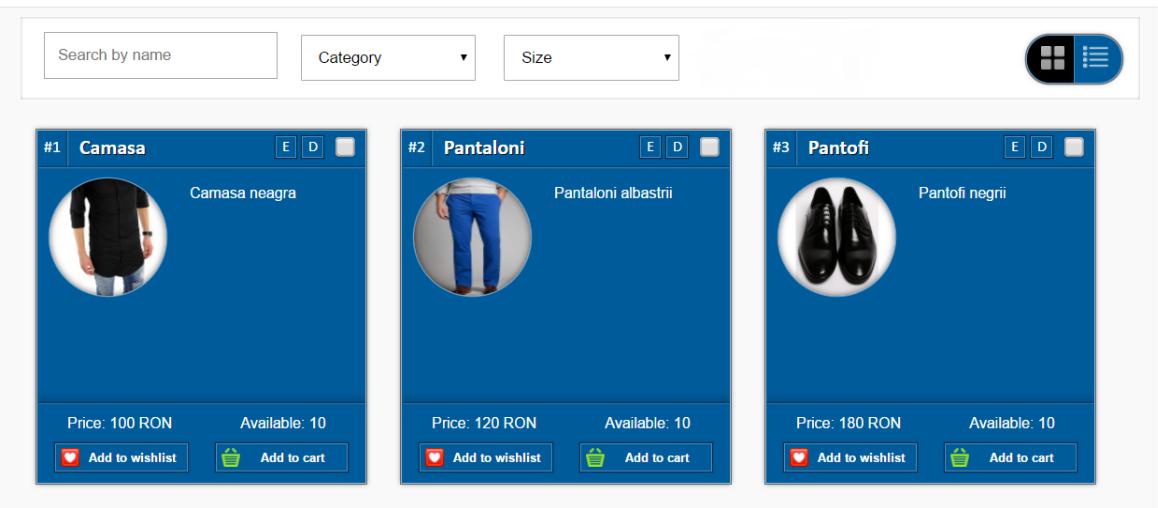
51

După ce contul a fost realizat acesta devine un utilizator care are acces la următoarele butoane:

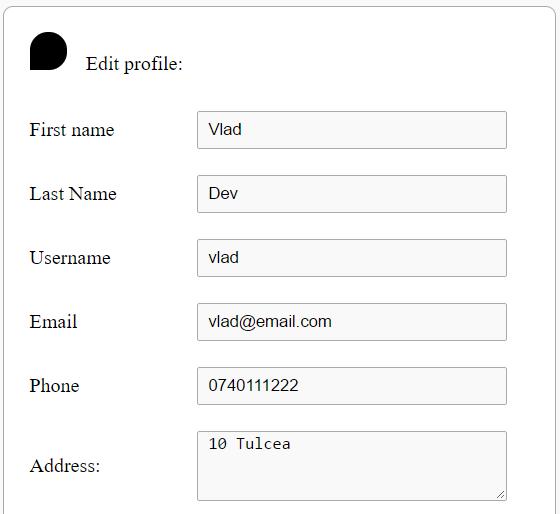
-HomePage



-Products

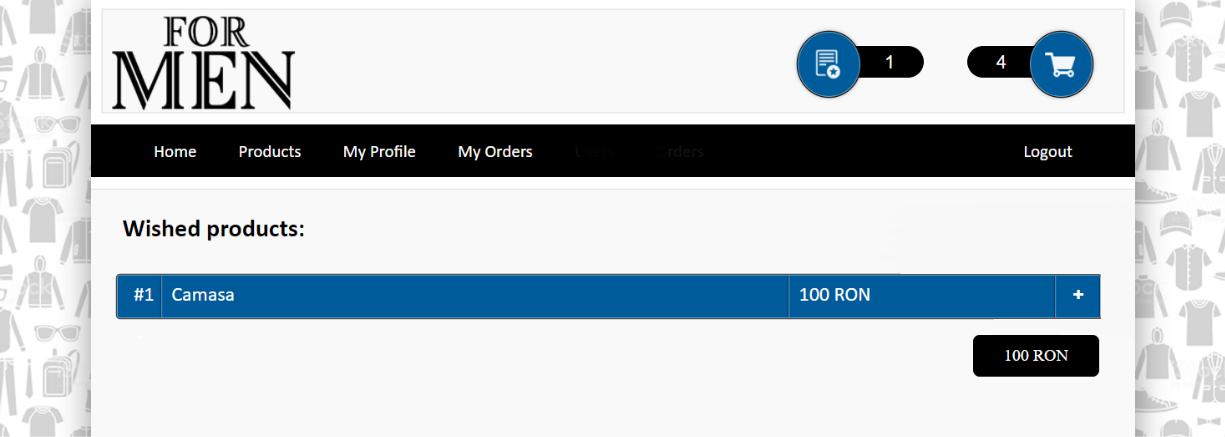


-My Profile

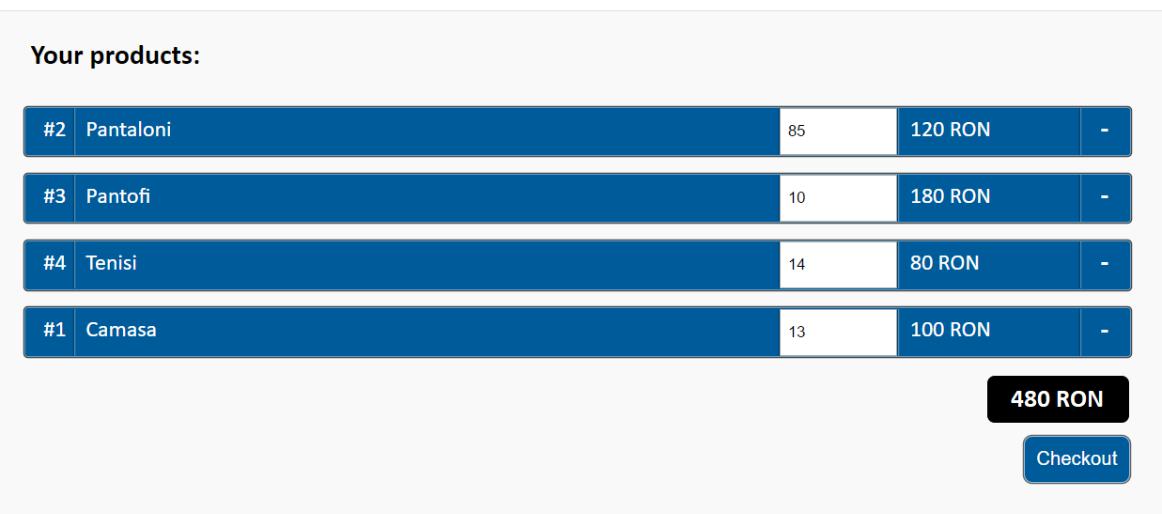


52

-WishList

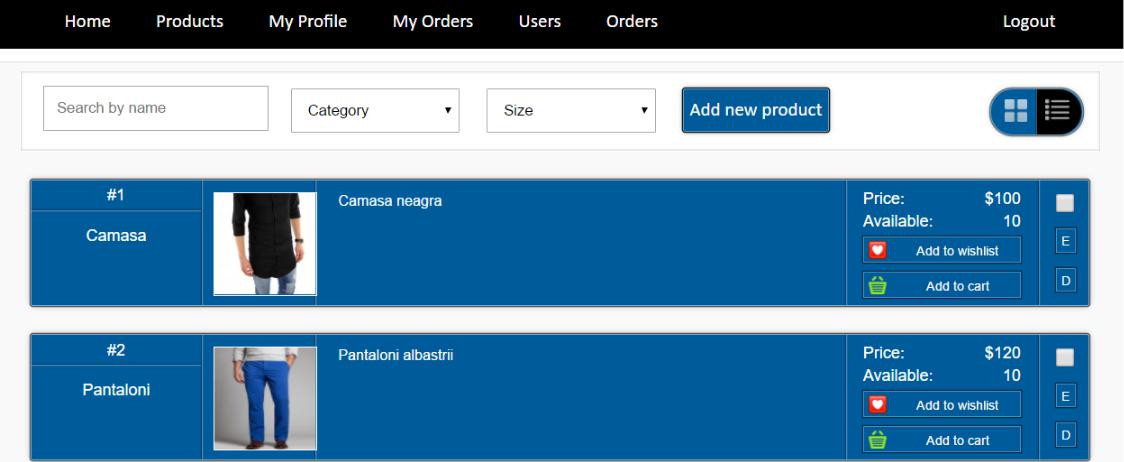


-Cart



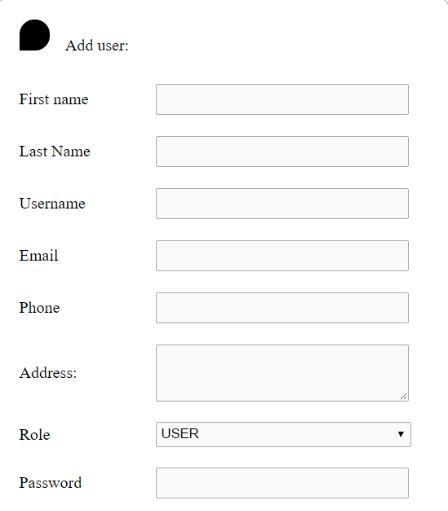
Pe lângă aceste butoane care sunt oferite utilizatorului,dacă folosești această aplicație în postura de administrator,ai la dispoziție câteva butoane în plus.

-Adăugarea unui produs

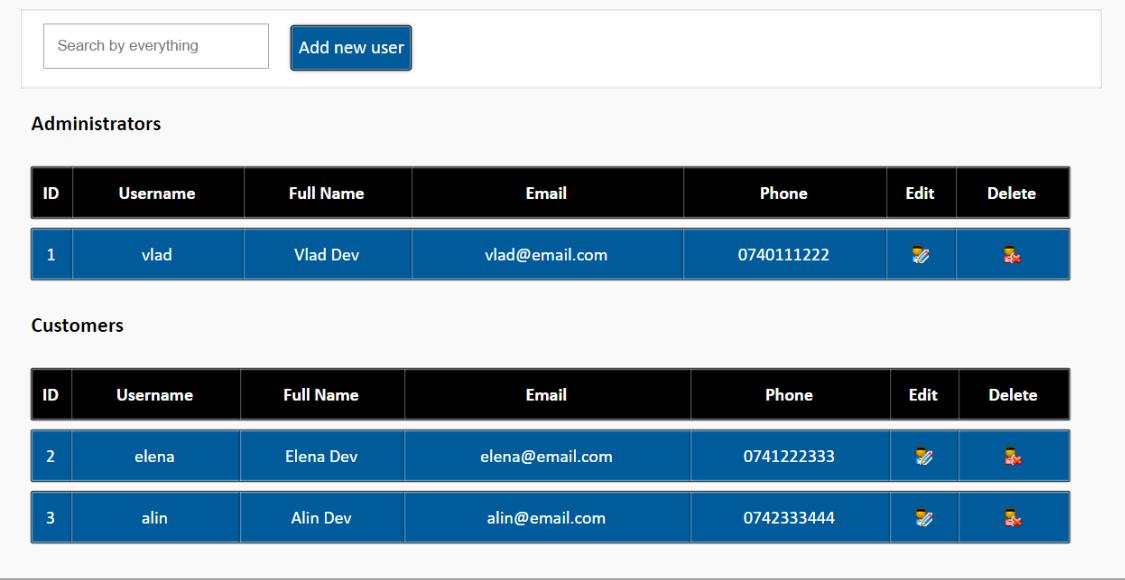


53

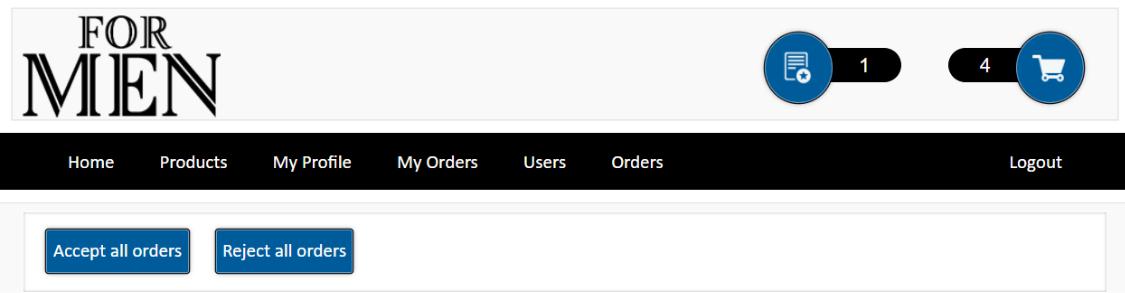
-Adăugarea unui utilizator



-Users



-Orders



54

**4.6 Concluzii**

În acest capitol urmează să vorbesc puțin despre unele concluzii pe care le-am tras pe tot parcursul realizării acestei aplicații.

Personal mă bucur că am reușit să realizez această aplicație ,mă bucur că totul a ieșit așa cum mi-am dorit și pe viitor dacă o să am posibilitatea de a extinde această aplicație la un nivel mai înalt nu voi ezita deoarece consider că poate să fie extinsă și pentru acest lucru aș avea câteva idei pe care mi-ar plăcea să le realizez.

Pe tot parcursul realizării acestei aplicații am încercat să evidențiez scopul principal al lucrări, acela de a descrie un mod prin care se poate dezvolta o aplicație web, mai exact un magazin în mediul online.

Cele 4 capitole pe care le-am descris mai sus conțin destul de detaliat aspecte teoretice și practice ale modalităților de concepere și implementare a unei aplicații web.

Aplicația creată de mine poate să fie extinsă și pentru alte tipuri de îmbrăcăminte,în special îmbrăcăminte feminină care ar reprezenta un mare plus cand vine vorba de vânzări deoarece o să fie pusă la dispoziție o gamă largă de produse specific feminine de la îmbrăcăminte până la accesorii pentru ținute.

Un alt lucru care s-ar putea face pentru extinderea acestei aplicații este permiterea utilizatorilor să comunice între ei în momentul în care se află pe site.

O altă posibilitate pe care utilizatorii o vor avea este aceea de a posta diferite articole cu produsele achiziționate de pe site pentru a oferi o mai mare încredere persoanelor care vor să achiziționeze de pe site articole vestimentare dar nu în ultimul rând și posibilitatea de a posta fotografii sau videouri cu sau despre aceste produse.

Consider că tehnologiile pe care le-am folosit până acum pentru realizarea acestei aplicații m-au ajutat foarte mult și cred că o să mă ajute dacă voi avea posibilitatea de a extinde mai mult aceasta aplicatie și de a realizea lucrurile pe care le-am enuntat mai sus.

55

**Bibliografie:**

1. [**https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server**](https://en.wikipedia.org/wiki/Microsoft_SQL_Server)
2. [**https://en.wikipedia.org/wiki/.NET\_Framework**](https://en.wikipedia.org/wiki/.NET_Framework)
3. Wigley, Andy, et al. *Microsoft. net compact framework: Core reference*. Microsoft Press, 2002.
4. [**https://en.wikipedia.org/wiki/Web\_API**](https://en.wikipedia.org/wiki/Web_API)
5. [**https://msdn.microsoft.com/en-us/library/hh833994(v=vs.108).aspx**](https://msdn.microsoft.com/en-us/library/hh833994(v=vs.108).aspx)
6. Darwin, Peter Bacon, and Pawel Kozlowski. *AngularJS web application development*. Packt Publ., 2013.
7. [**https://en.wikipedia.org/wiki/AngularJS**](https://en.wikipedia.org/wiki/AngularJS)
8. [**https://www.google.ro/search?q=web+api&rlz=1C1CHBF\_enRO738RO738&source=**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918)[**lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ\_**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918)[**AUIBigB&biw=1920&bih=918**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918)
9. [**https://www.google.ro/search?q=web+api&rlz=1C1CHBF\_enRO738RO738&source=**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918#tbm=isch&q=sql+server)[**lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ\_**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918#tbm=isch&q=sql+server)[**AUIBigB&biw=1920&bih=918#tbm=isch&q=sql+server**](https://www.google.ro/search?q=web+api&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&sqi=2&ved=0ahUKEwiHuf2Ts8zUAhUoJJoKHTAXCooQ_AUIBigB&biw=1920&bih=918#tbm=isch&q=sql+server)
10. [**https://www.getpostman.com/apps**](https://www.getpostman.com/apps)
11. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795)[**biw=1232&bih=795**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795)
12. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=css)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=css)[**biw=1232&bih=795#tbm=isch&q=css**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=css)
13. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=javascript)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=javascript)[**biw=1232&bih=795#tbm=isch&q=javascript**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=javascript)
14. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=asp.net)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=asp.net)[**biw=1232&bih=795#tbm=isch&q=asp.net**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=asp.net)
15. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=json)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=json)[**biw=1232&bih=795#tbm=isch&q=json**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=json)
16. [**https://www.google.ro/search?q=html&rlz=1C1CHBF\_enRO738RO738&source=lnm**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=entity+framework)[**s&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc\_UAhWkAJoKHeNOAdkQ\_AUIBigB&**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=entity+framework)[**biw=1232&bih=795#tbm=isch&q=entity+framework**](https://www.google.ro/search?q=html&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiC5I6flc_UAhWkAJoKHeNOAdkQ_AUIBigB&biw=1232&bih=795#tbm=isch&q=entity+framework)
17. [**https://en.wikipedia.org/wiki/Language\_Integrated\_Query**](https://en.wikipedia.org/wiki/Language_Integrated_Query)
18. [**https://www.google.ro/search?q=Language+Integrated+Query&rlz=1C1CHBF\_enR**](https://www.google.ro/search?q=Language+Integrated+Query&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiw9uPk0c_UAhUmP5oKHWMPBc8Q_AUICigB&biw=922&bih=795)[**O738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiw9uPk0c\_UAhUmP5o**](https://www.google.ro/search?q=Language+Integrated+Query&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiw9uPk0c_UAhUmP5oKHWMPBc8Q_AUICigB&biw=922&bih=795)[**KHWMPBc8Q\_AUICigB&biw=922&bih=795**](https://www.google.ro/search?q=Language+Integrated+Query&rlz=1C1CHBF_enRO738RO738&source=lnms&tbm=isch&sa=X&ved=0ahUKEwiw9uPk0c_UAhUmP5oKHWMPBc8Q_AUICigB&biw=922&bih=795)
19. [**https://en.wikipedia.org/wiki/HTML**](https://en.wikipedia.org/wiki/HTML)
20. [**https://en.wikipedia.org/wiki/Cascading\_Style\_Sheets**](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)
21. [**https://en.wikipedia.org/wiki/JavaScript**](https://en.wikipedia.org/wiki/JavaScript)
22. [**https://en.wikipedia.org/wiki/JSON**](https://en.wikipedia.org/wiki/JSON)
23. [**https://en.wikipedia.org/wiki/Entity\_Framework**](https://en.wikipedia.org/wiki/Entity_Framework)

56