**UNIVERSITATEA DIN BUCUREȘTI**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**LUCRARE DE LICENȚĂ**

**COORDONATOR ȘTIINȚIFIC**

**Prof. Univ. dr. Radu Ionescu**

**ABSOLVENT**

**Gavrilă Timea-Maria**

**BUCUREȘTI, 2015**

**UNIVERSITATEA DIN BUCUREȘTI**

**FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ**

**SPECIALIZAREA INFORMATICĂ**

**LUCRARE DE LICENȚĂ**

**CATALOG ONLINE PENTRU EVIDENȚA CĂRȚILOR**

**COORDONATOR ȘTIINȚIFIC**

**Prof. Univ. dr. Radu Ionescu**

**ABSOLVENT**

**Gavrilă Timea-Maria**

**BUCUREȘTI, 2015**

**CUPRINS**

**INTRODUCERE**

**CAP. I Noțiuni generale privind tehnologiile folosite**

*1.1. Definirea aplicațiilor web*

*1.2. Aplicații existente vs. aplicația propusă*

**CAP. II Tehnologii utilizate în proiectarea și funcționalitatea aplicației**

*2.1. Programarea web utilizând ASP.NET*

*2.1.1 Introducere*

*2.2. Limbajul C#*

*2.2.1. Introducere*

*2.3. Baza de date*

**CAP. III Tehnologii utilizate în design-ul aplicației**

*3.1. HTML*

*3.2. CSS*

*3.2.1. Introducere*

*3.2.2. LESS*

*3.2.3. Bootstrap*

*3.2.4. FontAwesome*

*3.3. Javascript*

*3.3.1. Introducere*

*3.3.2. jQuery*

**CONCLUZII**

**BIBLIOGRAFIE**

**ANEXE**

**INTRODUCERE**

Prezenta lucrare de licență, își propune să aducă un nou concept de site web, românesc, pentru toate persoanele pasionate de cărți. Aceasta este o adaptare după binecunoscutul goodreads.com, adoptat într-o formulă locală, pentru România (site-ul fiind în limba română), iar limbajele folosite pentru crearea lui sunt în principal ASP.NET și C#, precum și câteva framework-uri pentru interfață (ex.: jQuery, Bootstrap, etc.).

Site-ul permite vizitatorilor (atât cei autentificați cât și cei neînregistrați) acces la un catalog online de peste 100 de cărți, la review-urile altor utilizatori precum și la un sistem de rating pentru toate cărțile existente în baza de date. De asemenea, aceștia au posibilitatea creării unui profil personalizat, ceea ce le va permite la rândul lor să posteze recenzii și să voteze cărțile preferate. Totodată, utilizatorii înregistrati pot adăuga cărți în portofoliul personal după următoarele categorii:

* Cărți favorite
* Cărți citite
* Cărți de citit

Pentru utilizatorii autentificați, deja “antrenați”, există posibilitatea recomandării de cărți, atât pe baza comportamentului acestuia cât și pe baza profilului unic deja creat. Sistemul de recomandări va fi disponibil pe pagina principală.

Interfața se dorește a fi cât mai ușor de utilizat, iar design-ul acestuia, este unul cât mai simplu și modern, permițând atât utilizatorului obișnuit cât și persoanelor cu diferite dificultăți acces cât mai rapid și simplu la informațiile căutate.

Spre deosebire de alte site-uri web similare, #LaOCarte, prezintă și stochează cărțile în baza de date ca entități unice, i.e. pentru fiecare carte se cunoaște titlul, autorul, descrierea și genul din care face parte. Editura și, respectiv codul ISBN, au fost lăsate deoparte iar editura a fost stocată separat în baza de date, deoarece se consideră că pentru utilizatori este mai important ce carte au citit sau doresc să citească decât de la ce editură este, evitând astfel duplicatele. Totuși, pentru evitarea oricăror neclarități, pe pagina de detaliu a cărților se vor putea regăsi informații despre multiplele edituri unde a fost tipărită cartea precum și un link extern către adresa editurii.

Nu în ultimul rând, site-ul beneficiază de o pagină de contact unde orice utilizator poate trimite mail, având certitudinea că va primi și un răspuns la acesta, precum și o confirmare de trimitere.

**CAPITOLUL I.**

**Noțiuni generale privind tehnologiile folosite**

*Definirea aplicațiilor web*

Din punct de vedere tehnic, web-ul este un mediu programabil care permite personalizarea unei game largi și diverse de aplicații la nivel global, pentru milioane de utilizatori. Cele 2 componenete ale unui site web modern sunt browser-ul și respectiv aplicația web, ambele disponibile pentru toată lumea, fără nici o cheltuială.

Browserele web sunt aplicații software pentru preluarea și prezentarea resurselor informaționale de pe internet (World Wide Web). O resursă de informații este identificată printr-un Uniform Resource Identifier (URI/URL), acesta putând fi o pagină web, o imagine, un video sau un fragment de conținut. Link-urile prezente în aceste resurse permite utilizatorilor navigarea în browser spre resursele aferente. Printre principalele și cele mai populare browsere se enumeră: Google Chrome, Mozilla Firefox, Internet Explorer, Opera și Safari.

Prin definiție, o aplicație web este un program de calculator ce rulează într-un browser web care permite vizitatorilor site-lui accesul și inserarea de informații într-o o bază de date. Aplicațiile web interoghează server-ul (acesta cuprinzând baza de date) și produce în mod dinamic documentul web. Acesta este generat într-un format standard care permite afișarea aceluiaș conținut în toate browserele (ex. HTML sau XHTML, CSS).

Câteva avantaje semnificative pentru construirea și menținerea unei aplicații web este că aceasta își îndeplinește scopul indiferent de sistemul de operare folosit, nu are aproape nici o cerință de instalare și pot fi implementate rapid fără nici un cost pentru utilizator.

De asemenea, website-urile moderne permit capturarea, procesarea, stocarea și transmiterea informațiilor private ale clienților (i.e. date personale, numerele de card, informații de securitate socială, etc.) pentru utilizare imediată și recurentă. Caracteristici precum pagini de login, mail, coșuri de cumpărături online etc. formează site-urile moderne și oferă întreprinderilor mijloacele necesare pentru a comunica cu clienții. Pe măsură ce numărul firmelor care se bucură de avantajele afacerilor pe internet crește, și alte tehnologii legate de aceasta vor continua să se dezvolte.

Figura 1.1 de mai jos detaliază modelul unei aplicații web în trei straturi. Primul este în mod normal browser-ul web cu interfața utilizatorului. Al doilea este instrumentul de tehnologie dinamic care generează conținut și al treilea strat este cel de baze de date care depozitează conținutul (ex. articole, știri) și datele clienților (ex. nume de utilizator, parole, numere de securitate socială, etc.).

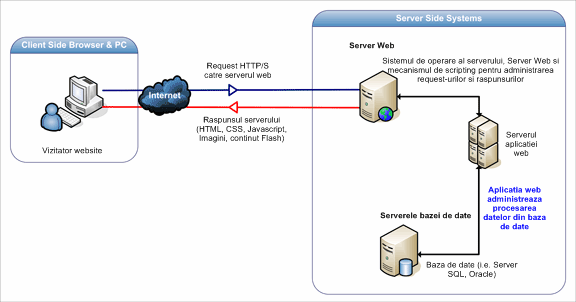


Figura 1.1 Modelul unei aplicații web

*Aplicații existente vs. Aplicația propusă*

În urma căutărilor pe internet putem împărți aplicațiile existente asemănătoare în două categorii:

1. Aplicațiile web românești
2. Aplicațiile web străine

Cele românești sunt reprezentate majoritar din bloguri personale ale iubitorilor de cărți unde se pot găsi prezentări și review-uri pentru cărțile citite de către proprietarul blogului. Pe lângă acestea, mai există site-urile editurilor și ale partenerilor (ex. elefant.ro), majoritatea lor fiind de tipul magazin online Nu există nici un site web, româneasc, de tip comunitate unde să se regăsească toate specificațiile aplicației descrise.

Printre aplicațiile străine, probabil cea mai cunoscută este Goodreads, aceasta reprezentând și oarecum ca sursă de inspirație pentru #LaOCarte. Totuși, spre deosebire de aceasta, prezenta aplicație web își propune să ofere utilizatorilor o comunitate unică, oarecum simplificată, o bază de date cu cărți doar în limba română, fără duplicate, în funcție de editura la care au apărut și codul ISBN.

**CAPITOLUL II.**

**Tehnologii utilizate în proiectarea și funcționalitatea aplicației**

*Programarea web utilizând ASP.NET*

**In****troducere**

ASP.NET este un framework open source creat de Microsoft, special pentru dezvoltarea aplicațiilor dinamice și a serviciilor web. Acesta este succesorul ASP-ului (Active Server Pages) introdus pentru prima dată în 1998 ca și prim program de tip server-side al aplicațiilor. În plus față de acesta, ASP.NET beneficiază de avantajele platformei de dezvoltare .NET și de instrumentele oferite de aplicația “Visual Studio .NET”.

ASP.NET a introdus o mulțime de noi caracteristici pentru dezvoltatori, incluzând cod server-side compilat, tehnică folosită pentru a separa logica serverului de interfața utilizatorului, un model extensibil de control pe partea de server, o metodă ușor de utilizat de implementare și suport pentru formularele de validare, atât pe partea de interfată cât și server-side.

Ca și avantaje ale utilizării ASP.NET, se numără:

* Oferă un model de programare orientată pe obiecte, acesta conținând un set larg de componente, bazate pe XML;
* Are performanță ridicată deoarece rulează cod compilat. Codul sursă este împărțit de obicei în 2 fișiere, unul pentru codul executabil și unul pentru conținutul paginii (textul din pagina și cod HTML);
* .NET este compatibil cu mai multe limbaje de programare diferite, cele mai utilizate fiind Visual Basic și C#.

Microsoft Visual Studio este, un IDE (Integrated Development Environment) creat de Microsoft folosit pentru dezvoltarea programelor pe desktop, a website-urilor, a aplicațiilor și serviciilor web.

*Limbajul C#*

**Introducere**

C# este un limbaj de programare orientat pe obiect, imperativ, declarativ și generic, care permite printre altele atât dezvoltarea de aplicații Windows cât și aplicații de tip client-server și baze de date. Acesta a fost dezvoltat de Microsoft, în interiorul framework-ului .NET.

Sintaxa limbajului este în același timp expresivă și simplă. Pentru dezvoltatorii deja familiarizați cu C, C++ sau Java, C# este cu atât mai ușor de învățat cu cât sintaxa simplifică mult complexitatea C++-ului, oferă caracteristici precum variabile de tip null, enumerații, acces direct la memorie și multe altele care nu se regăsesc în Java. Totodată, C# suporta metode și tipuri generice, care ofera siguranță și sporește performanța.

Ca un limbaj de programare orientat pe obiect, C# permite încapsularea, moștenirea și polimorfismul. Toate variabilele și metodele (incluzând și metoda Main), sunt încapsulate în definițiile claselor. În plus, C# faciliteaza dezvoltarea componentelor software prin mai mulți constructori inovatori, cum ar fi:

1. Semnături de metode încapsulate numite, care permit notificări pentru evenimentele type-safe.

2. Proprietăți care permit accesul la variabilele private.

3. Atribute care oferă date declarative despre tipurile folosite la momentul execuției.

4. Documentație sub forma de comentarii inline pentru XML.

5. Language-Integrated Query (LINQ) – permite interogări pe o varietate de surse de date.

*Baza de date*

Hsghfg

**CAPITOLUL III.**

**Tehnologii utilizate în design-ul aplicației**

*HTML*

HyperText Markup Language (HTML) este un limbaj de markup standard folosit pentru crearea paginilor web. Acesta este format din elemente specifice HTML-ului, numite tag-uri, care sunt interpretate de către browser pentru afișarea conținutului unei pagini. Tag-urile sunt scrise între paranteze unghiulare, iar majoritatea lor vin în pereche (ex. <html> </html>). Totuși există și câteva excepții, caz în care tag-urile sunt self-closing (ex. <img>, <input>, <br>).

Principalele tag-uri HTML pe care ar trebui să le conțină orice pagină web sunt:

* <html> </html>
* <head> </head>
* <title> </title>
* <body> </body>

HTML5 reprezintă ultima versiune standardizată a HTML-ului lansat în octombrie 2014 și aprobată de World Wide Web Consortium (W3C). Scopul acestuia a fost îmbunătățirea limbajului, asigurând suport pentru multimedia și, în același timp, să îl mențină ușor de înțeles pentru oameni și ușor de interpretat pentru calculatoare și dispozitive (browsere web). Acesta a adaugat noi tag-uri, atât pentru a îmbogăți conținutul semantic al paginilor (ex. <article>, <header>, <footer>, <aside>, <nav>, <figure>) cât și pentru manipularea mai ușoară a conținutului media și graficii (ex. <video>, <audio>, <canvas>, <object>).

*CSS*

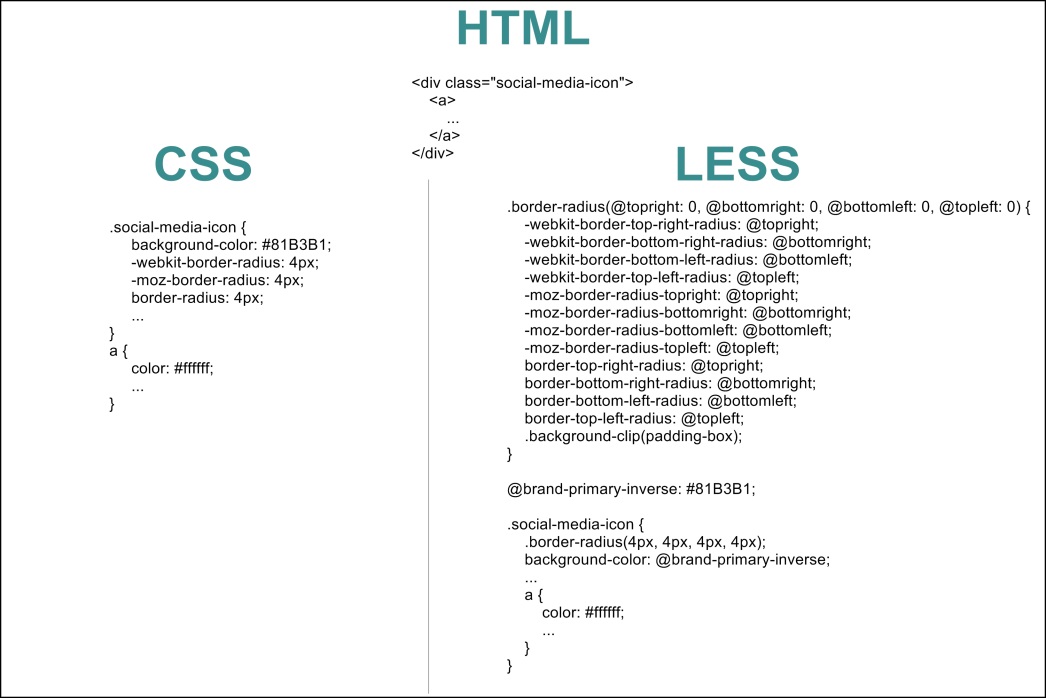
**Introducere**

Cascading Style Sheets (CSS) este un limbaj de stiluri folosit pentru descrierea aspectului și formatării unui document web. Acesta este folosit pentru crearea unor pagini web captivante, pentru interfața utilizatorului atât pentru aplicații web cât și pentru aplicații mobile. CSS-ul este conceput în primul rând pentru a permite separarea conținutul documentului de prezentarea acestuia, incluzând elemente care permit aranjarea aspectului, schimbarea culorilor și a font-urilor. Această separare poate îmbunătăți accesul la conținut, oferă mai multă flexibilitate și control și permite accesul mai multor pagini HTML la aceeași formatare prin același fișier .css reducând astfel complexitatea și repetiția în conținutul structural.

**LESS**

Less este un pre-procesor de CSS care extinde limbajul adăugând noi caracteristici precum variabile, mixing-uri, calcule matematice și funcții. Totodată, spre deosebire de CSS acesta permite scrierea codului sub formă arborescentă. Un exemplu despre cum funcționează mai exact Less-ul se poate vedea în Figura 3.1. unde se regăsește o parte din codul scris pentru icoanele de social media din header-ul paginii aplicației.

Figura 3.1 Exemplificarea codului LESS



**Bootstrap**

Inițial creat de către doi angajați la Twitter (un designer și un dezvoltator), Bootstrap a devenit rapid unul dintre cele mai populare framework-uri de front-end, open source. Numit inițial Twitter Blueprint, acesta este o combinație de HTML, CSS și Javascript care ajută la crearea componentelor interfeței utilizatorului.

Printre avantajele folosirii Bootstrap-ului se enumără:

1. Se poate personaliza - După preferințe, se poate adapta în conformitate cu specificațiile proiectului. Dezvoltatorii au capacitatea de a alege caracteristicile necesare, iar restul poate fi ignorat.

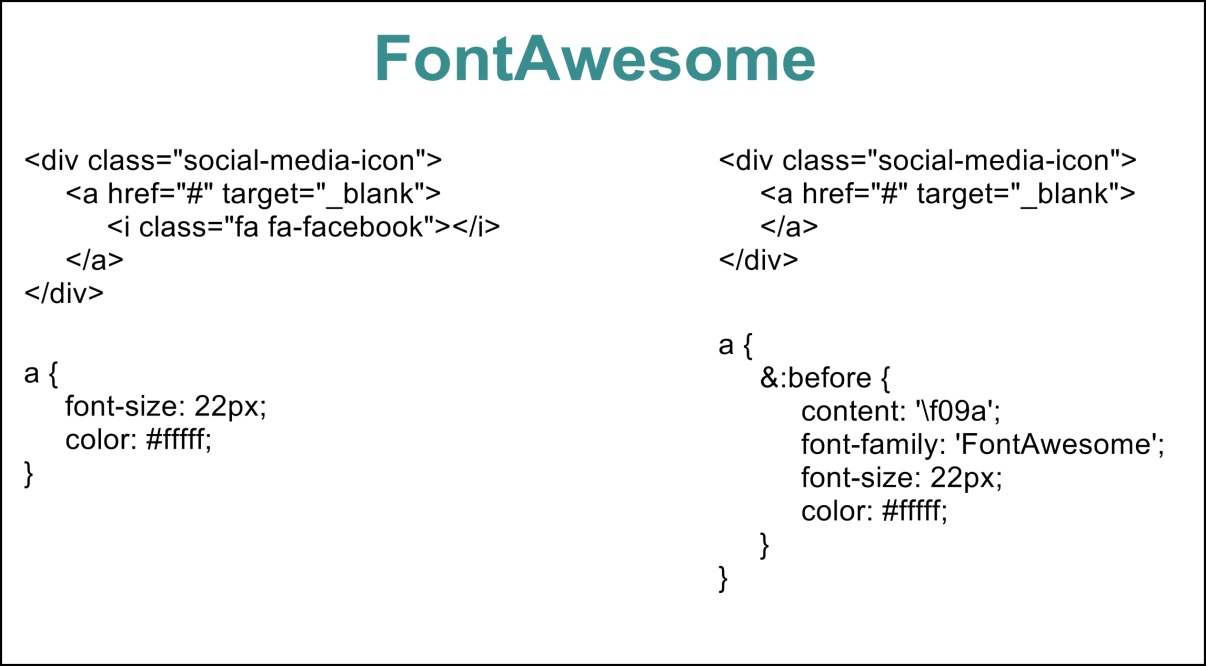
2. Este consistent – Acesta a fost fondat după acest principiu, permițând mai multor dezvoltatori care lucrează la același proiect să aibă aceleași rezultate, indiferent de platforma folosită iar interfața finală să arate la fel pe toate browserele.

3. Asigură suport – Bootstrap are o comunitate imensă în spatele lui a.î. pentru orice fel de probleme există soluții. În plus, acesta este în curs de actualizare continuă și în același timp, dezvoltat și găzduit pe GitHub.

4. Este responsive – Indiferent de dispozitivul folosit (de la desktop, la iPad și mobile), Bootstrap-ul se adapteaza automat în funcție de platformă.

**Font Awesome**

Font awesome reprezintă un set de instrumente de font-uri și imagini bazate pe CSS și LESS. Așa cum se poate observa în Figura 2.3 acestea pot fi folosite atât în codul HTML cât și în LESS în pseudo-elementele :before și :after. Exemplul de mai jos descrie construirea imaginilor de social media din header-ul paginii aplicației. În stânga este partea de HTML, cum de fapt au fost și create elementele, iar în partea dreaptă se poate observa o alternativă, realizată în LESS.



*JavaScript*

**Introducere**

JavasScript (adesea prescurtat JS), este un limbaj de programare dinamic. Acesta a fost creat în 10 zile în mai 1995 de către Brendan Eich (Netscape). Numit inițial Mocha, a fost lansat în septembrie 1995 sub numele oficial LiveScript, redenumit apoi în decembrie, același an, JavaScript. Este cel mai des folosit în aplicațiile web, ceea ce permite rularea codului de către browser. JavaScript este clasificat ca fiind un limbaj de programare imperativ, orientat pe obiect.

În ciuda similarităților de sintaxa și a numelui, între JavaScript și Java nu există absolut nici o legătură.

**JQuery**

JQuery este o librărie open-source Javascript creată pentru a simplifica și a îmbunătăți codul rulat de tip client-side, pentru ușurarea proceselor precum traversarea DOM-ului (Document Object Model) în HTML, pentru crearea de animații, tratarea evenimentelor și pentru a dezvoltarea aplicațiilor folosind cereri tip AJAX.

*ASP.NET și C#*

\*

us unification: a unification of languages, tools, libraries, deployment models, system design, and diagnostics. Web application developers no longer need to differentiate between components used by their pages and components used elsewhere in their architecture. They no longer have to deal with a script debugger to diagnose problems in their pages. They are no longer subject to the often mysterious subtleties of untyped scripting languages and can now use whatever .NET language they prefer in building their pages. Building Web applications is now like any other software development on the .NET platform.

[ASP.NET](http://asp.net/) introduce o mulțime de noi caracteristici pentru dezvoltator de aplicații web, inclusiv cod server-side compilat, o tehnica numita cod în spatele logicii pe partea de server separat de aspectul client-side, un model extensibil de control pe partea de server, un bine date Creată și ușor de utilizat model de implementare și suport obligatorii pentru formularul validare pe ambele clienți și servere. Mai mult decât toate că, cu toate acestea, [ASP.NET](http://asp.net/) ne oferă unificarea: o unificare de limbi, instrumente, biblioteci, modele de implementare, proiectarea sistemului, si diagnostice. Dezvoltatorii de aplicații web nu mai au nevoie să se diferențieze între componentele utilizate de pagini și a componentelor acestora utilizate în altă parte în arhitectura lor. Ei nu mai au de a face cu un program de depanare script pentru a diagnostica problemele in paginile lor. Ei nu mai sunt supuse subtilitățile adesea misterioase ale limbaje fără tip și pot folosi acum orice NET limbă pe care o preferă în construirea paginile lor. Construirea aplicații Web este acum la fel ca orice alt dezvoltare software pe platforma .NET.

\*

Prima variantă ASP.NET 1.0 a apărut în ianuarie 2002 ca parte a framework-ului .NET. Acesta

Aplicațiile web create cu ajutorul ASP.NET mai sunt cunoscute ca Web Forms, acestea fiind principalele blocuri în dezvoltarea aplicațiilor.

main building blocks for application development in ASP.NET

(

Visual Studio supports different [programming languages](http://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](http://en.wikipedia.org/wiki/C_(programming_language)),[[6]](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-6) [C++](http://en.wikipedia.org/wiki/C%2B%2B)and [C++/CLI](http://en.wikipedia.org/wiki/C%2B%2B/CLI) (via [Visual C++](http://en.wikipedia.org/wiki/Visual_C%2B%2B)), [VB.NET](http://en.wikipedia.org/wiki/VB.NET) (via [Visual Basic .NET](http://en.wikipedia.org/wiki/Visual_Basic_.NET)), [C#](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (via [Visual C#](http://en.wikipedia.org/wiki/Visual_C_Sharp)), and [F#](http://en.wikipedia.org/wiki/F_Sharp_(programming_language)) (as of Visual Studio 2010[[7]](http://en.wikipedia.org/wiki/Microsoft_Visual_Studio#cite_note-7)). Support for other languages such as [M](http://en.wikipedia.org/wiki/MUMPS), [Python](http://en.wikipedia.org/wiki/IronPython), and [Ruby](http://en.wikipedia.org/wiki/IronRuby) among others is available via language services installed separately. It also supports [XML](http://en.wikipedia.org/wiki/XML)/[XSLT](http://en.wikipedia.org/wiki/XSLT), [HTML](http://en.wikipedia.org/wiki/HTML)/[XHTML](http://en.wikipedia.org/wiki/XHTML), [JavaScript](http://en.wikipedia.org/wiki/JavaScript) and [CSS](http://en.wikipedia.org/wiki/Cascading_Style_Sheets). Java (and J#) were supported in the past.

)

**CONCLUZII**

ghddfshgfghsd

**BIBLIOGRAFIE**

1. Sitepoint 2010, The Principles of Beautiful Web Design by Jason Beaird, Canada
2. Addison-Wesley Professional 2003, Essential ASP.NET with Examples în C# by Fritz Onion
3. \*\*\* <http://www.asp.net/>
4. \*\*\*<http://www.codeproject.com/>
5. \*\*\*<http://bootsnipp.com/>
6. \*\*\* <http://www.acunetix.com/>
7. \*\*\*<https://msdn.microsoft.com/>
8. \*\*\*<http://stackoverflow.com/>
9. \*\*\*https://webthinker.wordpress.com/
10. \*\*\* <http://lesscss.org/>
11. \*\*\* <http://getbootstrap.com/>
12. \*\*\*<http://fortawesome.github.io/Font-Awesome/>