**UNIVERSITATEA POLITEHNICĂ DIN BUCUREȘTI**

FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

TEHNOLOGII DE PROGRAMARE ÎN INTERNET

# CREAREA UNEI APLICAȚII WEB UTILIZÂND ASP.NET CORE ÎN AZURE

**STUDENT:**

LEFTERACHE ALEXANDRU-GABRIEL 431A

# PROFESOR COORDONATOR:

STOICA ELENA-CRISTINA

# 2022-2023

# INTRODUCERE

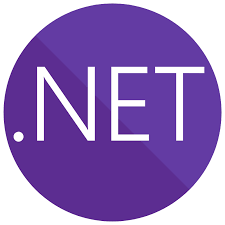
Lucrarea de față având titlul “**Crearea unei aplicații web utilizând Asp.net Core în Azure**” are în vedere atât documentarea unora dintre cele mai utilizate tehnologii de dezvoltare web, cât și exemplificarea funcționalității prin implementarea unei aplicații elementare.

Experiența personală în utilizarea mediului de dezvoltare **.Net** consta până la acest moment în dezvoltarea unor aplicații Windows cu ajutorul framework-ului **WPF** (Windows Presentation Foundation). Complexitatea uneltelor oferite și modul în care această platformă facilitează dezvoltarea aplicațiilor ”cross-platform” (destinate nu doar Windows, cât și web, Linux, Android, sau iOS) m-au determinat să aleg acest subiect, pentru a exemplifica cele menționate anterior, dar și pentru a-mi extinde arealul de cunoștințe tehnice.

# ASPECTE GENERALE

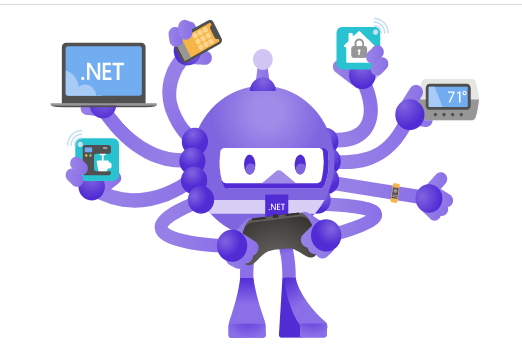
## **.NET**

.**.NET** (pronunțat în limbaj liber ”dot net”) reprezintă o platformă de dezvoltare software ”open-source”, proiectată de Microsoft începând cu sfârșitul anilor 90. Aceasta oferă posibilitatea de a implementa diverse tipuri de aplicații: mobile, desktop, sau web, ce pot rula nativ pe orice sistem de operare compatibil.



***Figura 1.1 – Pictograma .NET***

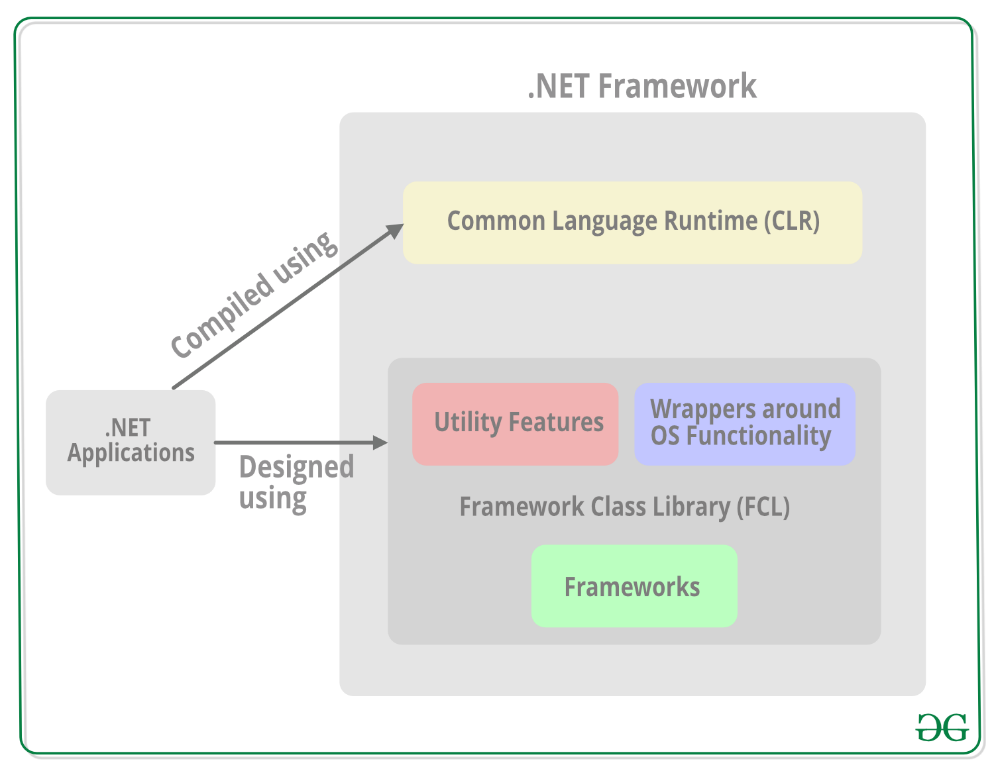
Dat fiind acest lucru, se spune despre .NET că este un mediu de dezvoltare ”**cross-platform**”: odată cu ascensiunea numărului de utilizatori ce folosesc sisteme de operare moderne, a apărut necesitatea dezvoltării unor interfețe de utilizator care să poată fi folosite pe cât mai multe echipamente hardware, cu diverse sisteme de operare. Deși o parte dintre aplicațiile ce pot fi dezvoltate utilizând .NET vizează un anumit subset de sisteme de operare și dispozitive, majoritatea sunt cross-platform. Punctul forte al platformei .NET: este independentă de hardware, oferă compatibilitate cu echipamentele noi, dar în același timp asigură menținerea compatibilității cu echipamentele mai vechi. Acest lucru se bazează preponderant pe componenta principală, anume **Common Language Runtime (CLR)**, o mașină virtuală ce oferă abstractizarea dintre codul sursă și sistemul de operare subiacent. Aceasta este datoare să asigure securitatea și stabilitatea aplicațiilor prin gestionarea corectă a memoriei și administrarea excepțiilor [[[1]](#endnote-1)].



***Figura 1.2 – .NET ca mediu de dezvoltare ”cross-platform” [4]***

O altă componentă importantă în cadrul .NET este reprezentată de FCL (Framework Class Library): o bibliotecă considerabilă de clase, spații de nume. interfețe și tipuri predefinite, ce oferă un set extins de servicii și funcționalități esenșiale dezvoltării de aplicații. Printre acestea se pot număra:

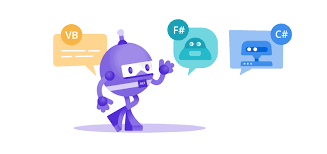
* Gestionarea fișierelor și directoarelor
* Accesul la baze de date
* Componente de interfață grafică
* Operarea pe rețea
* Gestionarea firelor de execuție [[[2]](#endnote-2)].



## ***Figura 1.3 – .NET Framework [[[3]](#endnote-3)]***

Familia de limbaje de programare suportate de .NET este alcătuită din:

* **C#** - (pronunțat ”C sharp”) este un limbaj simplu și modern, orientat către programarea pe obiecte și ”type-safe” (adică un limbaj ce impune verificări stricte privind compatibilitatea și coerența tipurilor de date utilizate în program). Este familiar cu C++ și Java, fapt ce facilitează învățarea unui astfel de limbaj. Se numără printre limbajele pe care le utilizez aproape zilnic și pe care le prefer.
* **F#** - limbaj de programare ușor de scris fiind succint și robust.
* **Visual Basic** – de asemenea limbaj de programare orientat pe obiecte.



***Figura 1.4 – Limbaje de programare***

***.NET [******[[4]](#endnote-4)]***

Dintre cele menţionate mai sus, cele popular este desigur C# (utilizat de altfel şi în cadrul acestui proiect), situat la momentul actual pe poziția 5, conform indexării **TIOBE** pe luna mai 2023. Acesta este precedat de **Python**, **C**, **Java** respectiv **C++,** fiind cel mai nou limbaj de programare (dezvoltat în 2000) din topul primelor 10 cele mai populare.



***Figura 1.5 – TIOBE Index pe luna Mai 2023 [[[5]](#endnote-5)]***

De-a lungul timpului .NET a avut parte de multiple evoluții, fiecare contribuind cu noi îmbunătățiri. Principalele versiuni ale platformei sunt:

* **.NET Framework:** versiunea inițială a platformei .NET. Aceasta a fost lansată în 2002 și a fost concepută inițial pentru Windows. Ultima versiune: 4.8.
* **.NET Core:** introdus în 2016 ca o implementare cross-platform, aceasta suportând atât Windows, cât și macOS sau Linux. Ultima versiune: 3.1.
* **.NET:** successor al lui .NET Core 3.1, lansat în noiembrie 2020. Reprezintă o unificare a celor două menționate anterior [[[6]](#endnote-6)].

De menționat este faptul că începând cu versiunea .NET, Microsoft s-a concentrat pe dezvoltarea unei singure platforme .NET unificate, renunțând la cele două convenții (.NET Framework, respectiv .NET Core), pe care o referă acum simplu, ”.NET”.

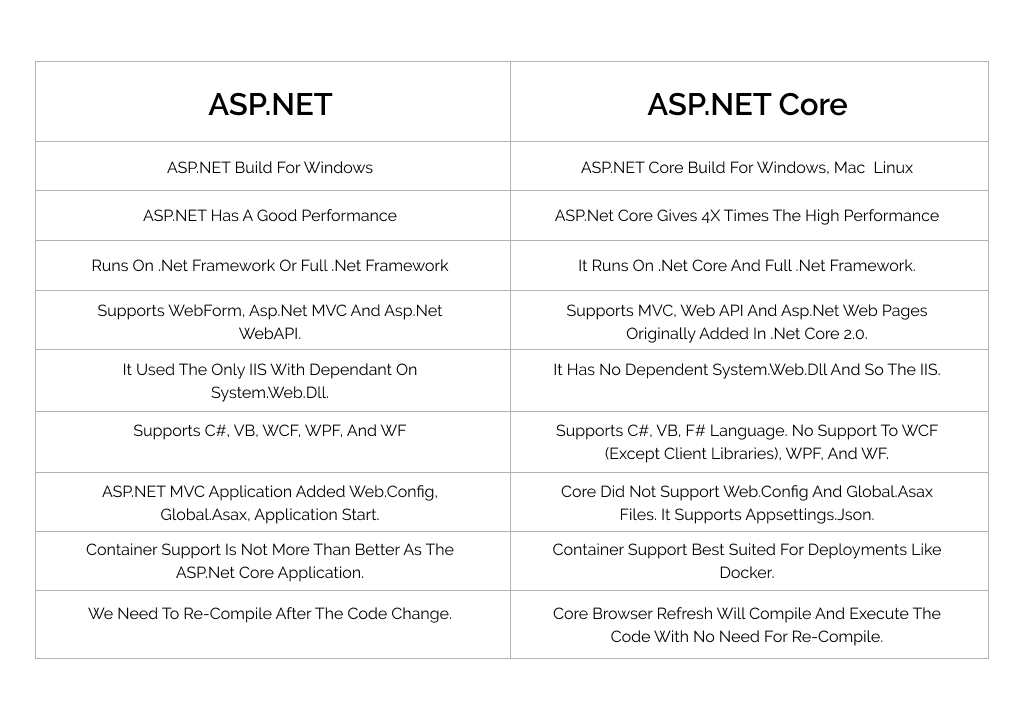
Pentru această lucrare am utilizat versiunea **.NET 6**, lansată în noiembrie 2020. Aceasta ne pune la dispoziție o multitudine de framework-uri și biblioteci cu funcționalități și suport pentru diverse domenii și tipuri de aplicații:

1. **ASP.NET Core** - framework pentru dezvoltarea aplicațiilor web și API-uri. Oferă suport pentru model-view-controller (MVC), routing, middleware pipeline, autentificare, autorizare și multe alte funcționalități pentru dezvoltarea aplicațiilor web moderne.
2. **Entity Framework Core** - ORM (Object-Relational Mapping) ce permite interacția cu baze de date relaționale.
3. **ML.NET** - Framework pentru dezvoltarea aplicațiilor de machine learning în .NET.
4. **Windows Forms** – probabil cel mai cunoscut, este un framework pentru construirea aplicațiilor desktop cu interfață grafică tradițională în stilul Windows.
5. **WPF (Windows Presentation Foundation)** - Este un framework pentru construirea aplicațiilor desktop cu interfață grafică mai avansată.

În continuare voi prezenta particularitățile framework-ului **ASP.NET Core**, cu ajutorul căruia voi dezvolta o aplicație web elementară, pe care o vom lansa în cele din urmă prin intermediul Azure.

## **ASP.NET CORE**

**ASP.NET Core** reprezintă o platformă **open-source** pentru dezvoltarea de aplicații web. Aceasta a fost creată și dezvoltată de compania americană **Microsoft Corp.**, care o evoluție a framework-ului **ASP.NET**.



***Figura 1.6 – Comparație între cele două framework-uri;***

***ASP.NET Core drept evoluție pentru APS.NET [[[7]](#endnote-7)]***

Acest framework cuprinde o gamă largă de funcționalități și instrumente necesare dezvoltării de aplicații web sigure și scalabile, utilizând orice limbaj de programare **.NET**: **C#**, **VB.NET** sau **F#**.

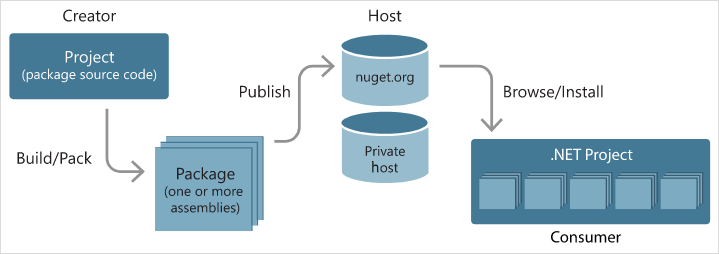
Una dintre caracteristicile importante ale **ASP.NET Core** este faptul că este construită folosind o arhitectură modulară: platforma este împărțită în module, fiecare având propriul set de funcții și instrumente. Aceste module sunt cunoscute sub numele de pachete **NuGet**, iar dezvoltatorii au libertatea de a instala doar pachetele necesare pentru proiectul lor, în loc să instaleze întreaga platformă. Modularizarea vine cu o multitudine de **beneficii**, printre care: platforma este ușor de întreținut și de actualizat, deoarece actualizările pentru fiecare pachet pot fi gestionate independent, fără a afecta alte pachete sau funcții ale platformei, iar dezvoltatorii au flexibilitatea de a crea aplicații web personalizate și optimizate pentru nevoile lor.

### **Ce este și cum funcționează un NuGet?**

Reprezintă mecanismul prin care codul este partajat în cadrul platformei .NET și definește modul în care pachetele sunt create, găzduite și utilizate [[[8]](#endnote-8)].

Mai exact, un pachet NuGet este un simplu fișier ZIP având extensia specifică **”.nupkg”,** ce conține cod compilat (DLL), precum și alte fișiere, inclusiv fișiere informative cu referire la versiunea pachetului. Prin urmare, dezvoltatorii software pot crea astfel de pachete și le pot ulterior publica. Un simplu consumator descarcă respectivul pachet și îl adaugă în proiectul propriu pentru a se putea folosi de funcționalitățile implementate [8].

Întregul circuit este surpins în următoarea diagramă:



***Figura 1.7 – The flow of packages between***

***creators, hosts, and consumers [8]***

### **1.2.2 Funcționalitățile ASP.NET Core**

**ASP.NET Core** include, de asemenea, o gamă largă de funcționalități și instrumente pentru dezvoltarea de aplicații web și API-uri:

* Modelul **Model-View-Controller (MVC)**
* **Razor Pages**
* **Razor markup**
* **Tag Helpers**
* **Model binding**
* **Model validation**

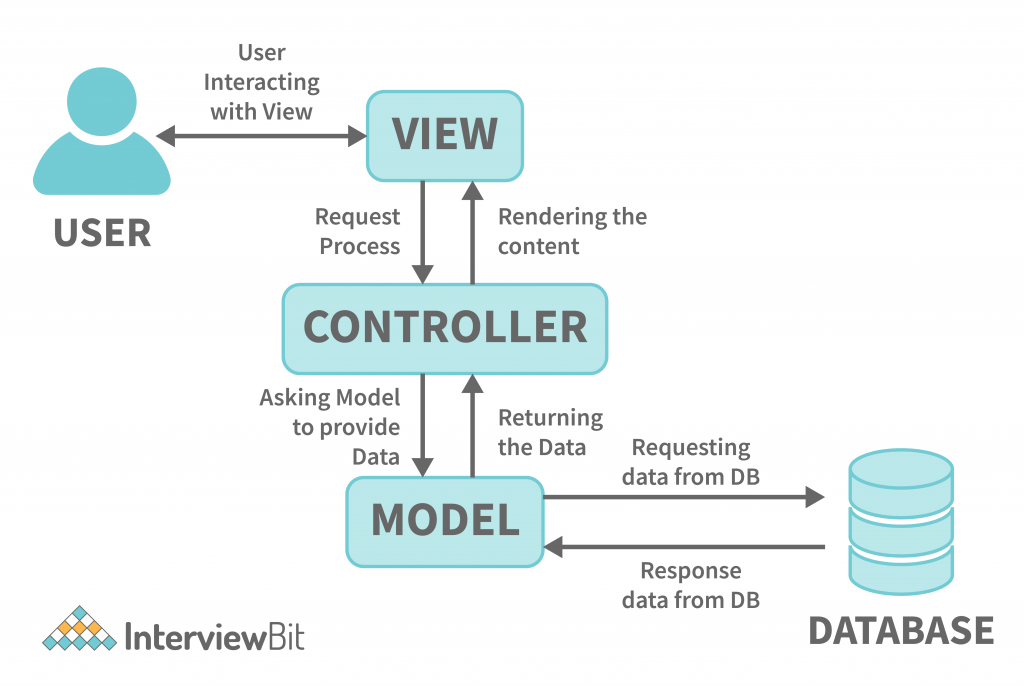
În cadrul acestei lucrări vom implementa modelul arhitectural Model-View-Controller, datorită structurii sale robuste, scalabile și a separării clare dintre logica de business, respectiv prezentarea datelor.

### **1.2.3 Model-View-Controller**

Este un model arhitectural utilizat în ingineria software, ce împarte logica programului în trei mari diviziuni interconectate și care a devenit foarte popular pentru dezvoltarea aplicațiilor web [[[9]](#endnote-9)].

Componentele caracteristice ale acestui model sunt:

* **Model** – gestionează datele sistemului software, răspunde la interogări referitoare la stare (de obicei solicitate de View) și realizează operații de modificare a datelor (de obicei invocate de controller). În cazul sistemelor bazate pe evenimente, modelul notifică observatorii (de obicei view-urile), atunci când informația se modifică pentru ca aceștia să poată reacționa la aceste modificări [[[10]](#endnote-10)].
* **View** - redă modelul într-o formă care permite interacțiunea cu utilizatorul, de obicei prin intermediul elementelor de interfață cu utilizatorul. Pentru un singur model pot exista mai multe view-uri pentru a deservi diferite scopuri [10].
* **Controller** – recepționează acțiunile utilizatorului și răspunde interogând modelul [10].

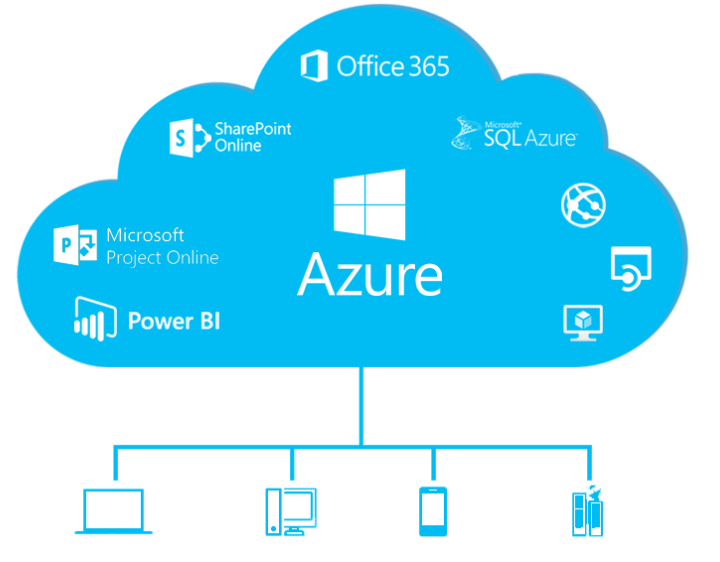


***Figura 1.8 – Arhitectura MVC (Model-View-Controller) [[[11]](#endnote-11)]***

## **2.1 Microsoft Azure**

### **2.1.1 Informații generale**

**Azure** este o platforma profesională de [servicii **cloud**](https://www.kodingtech.com/solutii-cloud/) (un ansamblu distribuit de servicii de calcul, aplicații, acces la informații și stocare de date, fără ca utilizatorul să aibă nevoie să cunoască amplasarea și configurația fizică a sistemelor care furnizează aceste servicii [[[12]](#endnote-12)]), oferită de Microsoft, care îți permite sa construiești rapid, să implementezi și să administrezi în siguranță aplicații scalabile. Proiectul a fost anunțat in luna octombrie a anului 2008 si dezvoltat sub numele de cod “Red Dog Project”. Oficial, Microsoft a lansat sistemul la data de 1 februarie 2010, sub numele de **Windows Azure**, după care, pe 25 martie 2014 și-a schimbat denumirea in **Microsoft Azure** [[[13]](#endnote-13)].



***Figura 1.9 – Are you making full use of***

***cloud technology? [[[14]](#endnote-14)]***

Aceasta le oferă clienților săi o multitudine de instrumente: de la dezvoltarea, testarea și găzduirea aplicațiilor, la stocarea datelor și furnizarea mașinilor virtuale. De aici fiecare poate selecta exact funcționalitățile dorite pentru a-și dezvolta proiectul și plătesc doar ce folosesc. Gândiți-vă ca la un abonament pe o platformă de streaming online, pentru care aveți puse la dispoziție o multitudine de filme, seriale, desene animate, etc., deși pe dvs. poate v-ar interesa să vizualizați un singur sezon al serialului Mr. Robot. Evident, nu este foarte plăcut să plătiți pentru o întreagă ofertă lună-de-lună, în timp ce utilizați această platformă pentru maxim 2h/săptămână, atunci când apare un nou episod. În schimb, folosind **Microsoft Azure** utilizatorul are posibilitatea de a consuma și plăti doar ce are nevoie, când are nevoie.

### **2.1.2 Serviciile Microsoft Azure**

Am precizat anterior, pe larg, câteva dintre serviciile oferite de platforma Microsoft Azure, cu toate că în realitate există mai mult de **600** de astfel de funcționalități. Evident, tema acestui proiect presupune utilizarea unei mici părți, dar cu toate acestea merită să enumerăm o parte dintre ele.

1. **Servicii informatice**
   1. Mașini virtuale – Windows sau Linux
   2. Servicii de aplicații – permit utilizatorilor să publice și administreze ușor website-uri
   3. Website-uri – **Azure Web Sites** permite utilizatorilor să dezvolte site-uri utilizând **ASP.NET**, PHP, Node.js, Java, sau Python
2. **Autentificare**
   1. Azure Active Directory – serviciu de gestionare a accesului bazat pe cloud.
3. **Servicii mobile**
   1. Analiza în timp real a comportamentului utilizatorilor. De asemenea oferă notificări către dispozitivele mobile.
4. **Stocare**
   1. API-uri REST și SDK pentru stocarea și accesarea datelor în cloud
   2. Servicii file
5. **Servicii de comunicații**
   1. Azure Communication Service – crearea de aplicații mobile și web ce utilizează SMS, apeluri video, VOIP (”voice over internet protocol”), sau chat web, etc.
6. **Management-ul datelor**
   1. Azure Data Explorer – analiza și explorarea datelor
   2. Azure Search – căutare de text
   3. Azure SQL Database – crearea, dezvoltarea și scalarea aplicațiilor în cloud utilizând Microsoft SQL Server.
7. **Development**
   1. Azure DevOps
8. **IA (Inteligență Artificială)**
9. **Blockchain**
10. **IoT**

### **2.1.3 Lansarea aplicațiilor ASP.NET Core cu ajutorul Azure App Service**

Cum am menționat anterior Azure App Service este un serviciu de găzduire de aplicații web, inclusiv a aplicațiilor dezvoltate cu ASP.NET Core.

Proiectul va avea în vedere inițial crearea aplicației în mediul integrat de dezvoltare Visual Studio 2022, având o bază de date elementară, urmată de apoi de lansarea aplicației. Totți acești pași vor fi detaliați în capitolul următor.

# DEZVOLTAREA APLICAȚIEI WEB

# BIBLIOGRAFIE

1. # [] *Common Language Runtime (CLR) overview*,<https://learn.microsoft.com/en-us/dotnet/standard/clr> - accesat 01.06.2023.

   [↑](#endnote-ref-1)
2. [] *.NET Framework Class Library*, <https://www.javatpoint.com/net-framework-class-library> - accesat 01.06.2023. [↑](#endnote-ref-2)
3. # [] *.NET Framework Class Library (FCL)*, <https://www.geeksforgeeks.org/net-framework-class-library-fcl/> - accesat 01.06.2023.

   [↑](#endnote-ref-3)
4. [] *What is .NET?*, <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet> - accesat 01.06.2023. [↑](#endnote-ref-4)
5. [] *TIOBE Index for May 2023,* <https://www.tiobe.com/tiobe-index/> - accesat 01.06.2023. [↑](#endnote-ref-5)
6. [] *What is .NET?*, <https://aws.amazon.com/what-is/net/> - accesat 01.06.2023. [↑](#endnote-ref-6)
7. [] *ASP.NET vs ASP:NET Core*, [https://www.zrix.com/blog/asp-net-vs-asp-net-core - accesat 01.06.2023](https://www.zrix.com/blog/asp-net-vs-asp-net-core%20-%20accesat%2001.06.2023). [↑](#endnote-ref-7)
8. [] *An introduction to NuGet*, [https://learn.microsoft.com/en-us/nuget/what-is-nuget - accesat 01.06.2023](https://learn.microsoft.com/en-us/nuget/what-is-nuget%20-%20accesat%2001.06.2023) [↑](#endnote-ref-8)
9. [] *Model-view-controller*, <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> – accesat 01.06.2023. [↑](#endnote-ref-9)
10. [] *Modele Arhitecturale Care Asigură Separarea,* [https://www.aut.upt.ro/staff/diercan/data/PSSC/curs-11.pdf/ - accesat 01.06.2023](https://www.aut.upt.ro/staff/diercan/data/PSSC/curs-11.pdf/%20-%20accesat%2001.06.2023). [↑](#endnote-ref-10)
11. [] *MVC Architecture – Detailed Explanation*, <https://www.interviewbit.com/blog/mvc-architecture/> - accesat 01.06.2023 [↑](#endnote-ref-11)
12. [] *Cloud computing*, <https://ro.wikipedia.org/wiki/Cloud_computing> - accesat 01.06.2023 [↑](#endnote-ref-12)
13. [] *Totul despre Microsoft Azure: Ce este, beneficii, cum funcționează?,* https://www.kodingtech.com/microsoft-azure-ce-este-beneficii-functionare/ - accesat 01.06.2023 [↑](#endnote-ref-13)
14. [] Are you making full use of cloud technology?, <https://www.ntsols.com/news/microsoft-azure-the-cloud-for-modern-business> - accesat 01.06.2023 [↑](#endnote-ref-14)