

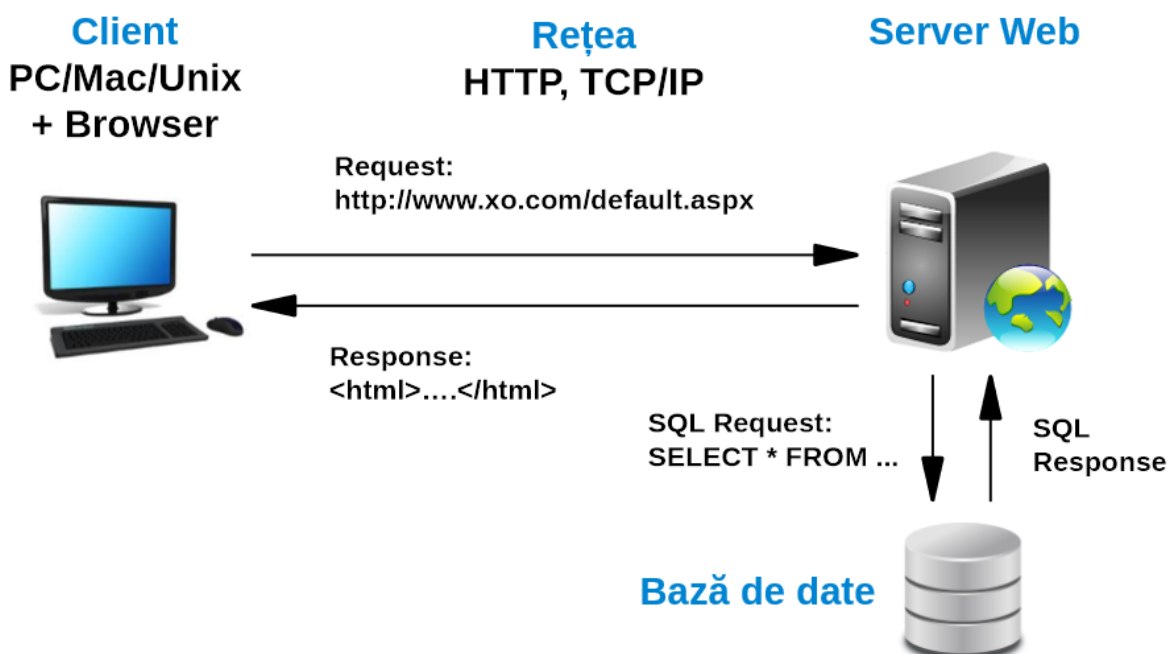
Ce este o aplicatie Web? Arhitectura Web. Avantaje/Dezavantaje ale aplicatiilor Web. Introducere in ASP.NET. Framework .NET. Introducere in C#. Conventii de notare. Ciclul de viata al unei pagini Web. ASP.NET MVC5. Instalare Visual Studio.

Ce este o aplicatie Web?

O **aplicatie web** este o aplicatie care ruleaza intr-o arhitectura Client-Server bazata pe: **protocolul HTTP, TCP/IP; browser Web; Server Web.**

Aplicatiile web sunt executate intr-un browser web si implementate folosind tehnologii precum: PHP, ASP, PYTHON, HTML, CSS, JAVASCRIPT, etc.

Arhitectura Web



Avantajele oferite de aplicatiile web

- Sunt independente de sistemul de operare
- Nu necesita instalare
- Actualizari foarte usor de facut deoarece modificarile se fac intr-un singur loc – pe server, ele propagandu-se pentru toti utilizatorii (in cazul aplicatiilor client-server clasice, interfata cu utilizatorul este asigurata prin intermediul unui program client instalat pe calculatorul fiecarui utilizator, orice modificare necesitand reinstalarea aplicatiei pentru fiecare utilizator in parte)

Dezavantaje

- Depind de conexiunea la Internet
- Riscuri de securitate

Introducere in ASP.NET

- **ASP.NET** este un framework Web open source conceput si dezvoltat de Microsoft
- Este utilizat pentru a dezvolta site-uri, aplicatii si servicii web
- Oferă o integrare foarte buna a codului HTML, CSS, JAVASCRIPT
- Este construit pe baza **CLR (Common Language Runtime)** – ruleaza **cod compilat** si permite utilizatorilor sa scrie cod folosind **orice limbaj .NET**

Ce este CLR – Common Language Runtime?

Se ocupa de executia programelor C#. Atunci cand este compilat un program C# rezultatul compilarii **nu este un cod executabil**. In locul acestuia se produce un fisier care contine un tip de cod apropiat de codul masinii, numit limbaj intermediar sau pe scurt **IL – Intermediate Language**.

Prin intermediul unui compilator denumit **JIT – Just in Time**, CLR transforma codul intermediar in cod executabil.

Framework-ul .NET

- Este compatibil cu peste 20 de limbaje diferite, cele mai populare fiind C#, C++, Visual Basic, F#
- Pune la dispozitie o colectie impresionanta de clase, organizate in biblioteci
- Este construit din doua entitati importante:

1. Common Language Runtime (CLR)

- mediul de executie al programelor fiind cel care se ocupa cu managementul si executia codului scris in limbaje specifice .NET

2. Base Class Library

- Este biblioteca de clase .NET
- Acopera o arie larga a necesitatilor de programare, incluzand **interfata cu utilizatorul, conectarea cu baza de date, accesarea datelor**

Introducere in C#

- Este un limbaj compilat
- Este un limbaj orientat pe obiecte
- Permite dezvoltarea de aplicatii industriale, durabile
- A fost conceput ca un concurent pentru limbajul Java
- Este derivat al limbajului C++
- Numele limbajului a fost inspirat de notatia # din muzica (nota muzicala urmata de # este mai inalta cu un semiton)

Limbaj compilat vs limbaj interpretat

Limbaj compilat – codul scris, numit cod sursa, este translatat de catre compilator intr-un cod apropiat de nivelul masinii, numit **cod executabil**. Atunci cand aplicatia trece de compilare fara erori de sintaxa se va produce codul executabil, iar aplicatia va putea fi rulata. (exemplu limbaje compilate: C, C++, Java, C#).

Limbaj interpretat (la rulare) – cu ajutorul unui interpretor specific limbajului, fiecare linie de cod este interpretata chiar in momentul rularii, fiind preschimbata imediat in cod masina si executata (exemplu limbaje interpretate: PHP, Ruby, Python)

Ce tip de limbaj (compilat sau interpretat) este mai rapid?

- Cel compilat deoarece nu mai are nevoie de un interpretor

C# si Java

Hello World in Java:

```
public class Hello {  
    public static void main(String args []) {  
        System.out.println("Hello world! This is Java Code!");  
    }  
}
```

Correspondentul in C#:

```
using System; -- in Java se numeste pachet, in c# namespace  
public class Hello {  
    public static void Main(string [] args) {  
        System.Console.WriteLine("Hello world! This is C# code!");  
    }  
}
```

Conventii de notare

- Pentru variabile se utilizeaza **camelCase** (age, firstName, placeOfBirth)
- Pentru clase si metode se utilizeaza **PascalCase** (Age, FirstName, PlaceOfBirth)

Ciclul de viata al unei pagini Web

Paginile ASP.NET ruleaza pe server-ul Web Microsoft IIS (Internet Information Server). In urma prelucrarii pe server rezulta o pagina web HTML, care este trimisa catre browser.

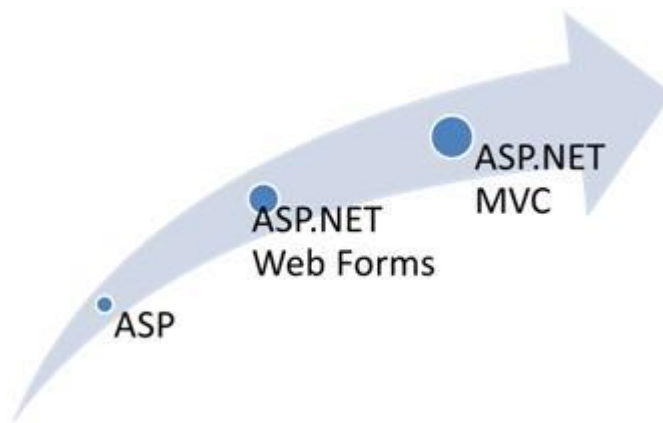
Ciclul de viata al unei pagini Web ASP.NET are urmatoorii pasi:

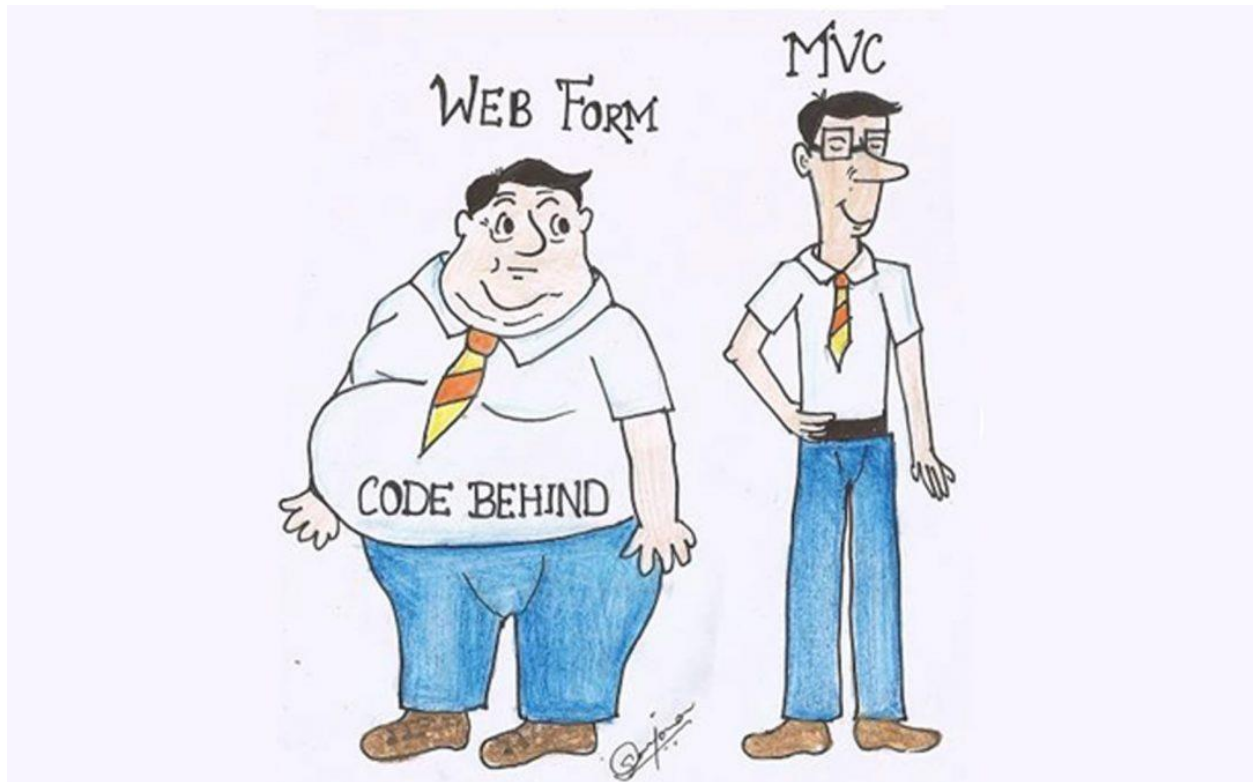
- **Page request (accesarea paginii)** – acest pas se intampla inaintea ciclului de viata, atunci cand o pagina este ceruta serverului
- **Start** – in acest stadiu se incarca proprietatile paginii, cum ar fi request-ul si raspunsul si se identifica tipul acestora (**GET – cerere resurse, POST – trimiterea de informatii catre server**)
- **Initialization (initializare)** – in acest pas se initializeaza directivele si controalele si se aplica codul din Master Page
- **Load (incarcarea)** – in aceasta faza daca cererea este de tip **postback**, controalele sunt incarcate cu informatii
- **Evenimentele Postback** – daca cererea este de tip postback se executa codul aferent. Dupa executia codului se aplica sistemele de validare
- **Rendering (ex: afisarea paginii)** – in acest pas se construiesc pagina finala pe server, care va fi afisata in browser
- **Unload (eliberarea memoriei)** – dupa ce pagina a fost trimisa utilizatorului, resursele alocate pentru aceasta sunt eliberate

ASP.NET MVC5

Are o arhitectura bazata pe modelul **Model – View – Controller**, ceea ce il face mult mai **rapid, robust** si **sigur**. ASP.NET MVC5 este urmatoarea generatie dupa ASP.NET Web Forms si ultima fata de versiunile sale anterioare, MVC1, MVC2, MVC3, MVC4.

De ce ASP.NET MVC?





Instalare Visual Studio 2017

Link-ul pentru descarcare:

<https://developerinsider.co/download-visual-studio-2017-web-installer-iso-community-professional-enterprise/>

Instalare:

Pasul 1:

Download Visual Studio 2017

You can download any of the below editions of Visual Studio 2017 directly from Microsoft servers:

VERSION	WEB INSTALLER	ISO	LAST UPDATE
<u>Visual Studio Community 2017</u>	<u>Download</u>	<u>Download</u>	<u>January 6, 2019</u>
Visual Studio Enterprise 2017	<u>Download</u>	<u>Download</u>	January 6, 2019
Visual Studio Professional 2017	<u>Download</u>	<u>Download</u>	January 6, 2019
Visual Studio Test Professional 2017	<u>Download</u>	<u>Download</u>	August 23, 2018
Visual Studio Community 2017 for macOS	<u>Download</u>	--	January 6, 2019
Visual Studio Professional 2017 for macOS	<u>Download</u>	--	January 6, 2019
Visual Studio Enterprise 2017 for macOS	<u>Download</u>	--	January 6, 2019
Team Foundation Server 2017	<u>Download</u>	<u>Download</u>	August 23, 2018

Pasul 2:

Visual Studio Installer

Before you get started, we need to set up a few things so that you can configure your installation.

To learn more about privacy, see the [Microsoft Privacy Statement](#).

By continuing, you agree to the [Microsoft Software License Terms](#).



Continue


Pasul 3:


Installing — Visual Studio Community 2017 — 15.9.16

Workloads Individual components Language packs Installation locations


Windows (3)


 **.NET desktop development**
Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.


 **Universal Windows Platform development**
Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.

 **Desktop development with C++**
Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.

Web & Cloud (7)

 **ASP.NET and web development**
Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.

 **Python development**
Editing, debugging, interactive development and source control for Python.

 **Azure development**
Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...

 **Node.js development**
Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.

Installation details

✓ **Visual Studio core editor**
The Visual Studio core shell experience, including syntax-aware code editing, source code control and work item management.

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Community [Change...](#)

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.






Total space required 609 MB

Install while downloading









Install

Workloads Individual components Language packs Installation locations




Windows (3)

 .NET desktop development Build WPF, Windows Forms, and console applications using C#, Visual Basic, and F#.	<input checked="" type="checkbox"/> 	 Desktop development with C++ Build Windows desktop applications using the Microsoft C++ toolset, ATL, or MFC.	<input type="checkbox"/>
 Universal Windows Platform development Create applications for the Universal Windows Platform with C#, VB, JavaScript, or optionally C++.	<input checked="" type="checkbox"/> 		




Web & Cloud (7)

 ASP.NET and web development Build web applications using ASP.NET, ASP.NET Core, HTML/JavaScript, and Containers including Docker support.	<input checked="" type="checkbox"/> 	 Azure development Azure SDKs, tools, and projects for developing cloud apps, creating resources, and building Containers including...	<input type="checkbox"/>
 Python development Editing, debugging, interactive development and source control for Python.	<input type="checkbox"/>	 Node.js development Build scalable network applications using Node.js, an asynchronous event-driven JavaScript runtime.	<input type="checkbox"/>
 Data storage and processing Connect, develop, and test data solutions with SQL Server, Azure Data Lake, or Hadoop.	<input checked="" type="checkbox"/> 	 Data science and analytical applications Languages and tooling for creating data science applications, including Python, R and F#.	<input type="checkbox"/>

Mobile & Gaming (5)

 Mobile development with .NET Build cross-platform applications for iOS, Android or Windows using Xamarin.	<input type="checkbox"/>
 Mobile development with JavaScript Build Android, iOS and UWP apps using Tools for Apache Cordova.	<input type="checkbox"/>
 Game development with C++ Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.	<input type="checkbox"/>

Other Toolsets (3)


 Visual Studio extension development Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.	<input type="checkbox"/>
 .NET Core cross-platform development Build cross-platform applications using .NET Core, ASP.NET Core, HTML/JavaScript, and Containers including Docker...	<input checked="" type="checkbox"/> 


Pasul 4:


Visual Studio Installer

Products

Installed

 Visual Studio Community 2017

Downloading: 27 MB of 4.11 GB (3 MB/sec)
0% 

Installing: package 0 of 0
0% 

Verifying...

☒ Start after installation

Visual Studio Start Page:

