

ASP .NET CASE STUDY



Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Cognitive Convergence is Subject Matter Expert in Office 365, Dynamics 365, SharePoint, Project Server, Artificial Intelligence Practices, Machine Learning practices, Power Platform: Power Apps-Power BI-Power Automate-Power Virtual Agents.

Our ASP .NET Development, Consulting, Add-in Development, Customization, Integration services and solutions are dedicated to help companies maximize their business performance, overcoming market challenges, achieving profitability, and providing best customer service.

Contents

OBJECTIVE	1
INTRODUCTION	1
WHAT IS ASP .NET?	1
<i>History of ASP .NET.....</i>	<i>1</i>
WEB FRAMEWORKS	2
<i>ASP.NET.....</i>	<i>2</i>
<i>ASP.NET Core.....</i>	<i>2</i>
<i>Blazor</i>	<i>2</i>
Numerical libraries.....	2
<i>Open-source numerical libraries</i>	<i>2</i>
<i>Proprietary numerical libraries.....</i>	<i>3</i>
3D graphics	3
<i>Open-source 3D graphics</i>	<i>3</i>
<i>Proprietary 3D graphics</i>	<i>3</i>
Image processing	3
Graphical user interface	4
Quality assurance	4
Object–relational mapping.....	4
PROS OF DOT NET FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT	5
<i>Code Reusing</i>	<i>5</i>
<i>Flexible Deployment.....</i>	<i>5</i>
<i>Use for Service-Oriented Architecture (SOA).....</i>	<i>5</i>
<i>Security and Safety.....</i>	<i>5</i>
<i>Use across platform and language.....</i>	<i>5</i>
<i>Vast and Viable</i>	<i>5</i>
<i>MVC.....</i>	<i>6</i>
<i>Advanced UI Control.....</i>	<i>6</i>
<i>Cross-platform Support</i>	<i>6</i>
<i>Cloud-Based Web Application Development Support</i>	<i>6</i>
<i>Integration with Legal Systems</i>	<i>6</i>
<i>Quick development.....</i>	<i>6</i>
CONS OF DOT NET FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT	7
<i>Legacy</i>	<i>7</i>
<i>Learning Curve</i>	<i>7</i>
<i>Windows-based Development Tools</i>	<i>7</i>
.NET PIPELINE	7
BUILDING CUSTOM MIDDLEWARE	8
.NET FRAMEWORKS:	9
DATABASE MODELING, AND CONFIGURATION.....	9
<i>Database configuration.....</i>	<i>9</i>
<i>Database modeling</i>	<i>9</i>
.NET DIFFERENT LIBRARIES AND THEIR WORKING	10
<i>Platform-specific class libraries</i>	<i>11</i>

Portable class libraries	11
.NET Standard class libraries	11
Mono class libraries.....	12
SINGLE PAGE APPLICATION IN .NET.....	12
DEPLOYMENT OF .NET PROJECT	12
No-impact applications.	13
Private components by default.	13
Controlled code sharing.	13
Side-by-side versioning.....	13
XCOPY deployment and replication.....	13
On-the-fly updates.	13
Integration with the Windows Installer.....	13
Enterprise deployment.	13
Downloading and caching.....	13
Partially trusted code.	13
PROVIDING HIGH, MODERN WEB AUTHENTICATION AND AUTHORIZATION SECURITY	14
Authentication in ASP.NET	14
Authentication providers.....	14
Windows authentication and IIS	14
Passport authentication	15
Forms authentication	15
Configuring Authorization	15
BEST FRONT-END FRAMEWORKS.....	15
React	15
Pros.....	16
Cons.....	16
When to use:	16
When not to use:.....	16
Angular	16
Pros.....	17
Cons.....	17
When to use:	17
When not to use:.....	17
Vue.js	17
Pros.....	18
Cons.....	18
When to use:	18
When not to use:.....	18
Ember.js	18
Pros.....	18
Cons.....	19
When to use:	19
When not to use:.....	19
jQuery	19
Pros.....	19
Cons.....	20
When to use:	20
When not to use:.....	20
Semantic-UI	20

Pros.....	20
Cons.....	20
When to use:	20
When not to use:.....	21
Backbone.js.....	21
Pros.....	21
Cons.....	21
When to use:	21
When not to use:.....	21
Preact.....	21
Pros.....	22
Cons.....	22
When to use:	22
When not to use:.....	22
Svelte.....	22
Pros.....	22
Cons.....	22
When to use:	23
When not to use:.....	23
Foundation	23
Pros.....	23
Cons.....	23
When to use:	23
When not to use:.....	23
DESIGN PRINCIPLE	24
Interoperability.....	24
Language independence	24
Type safety	24
Portability.....	24
Security.....	24
Memory management	25
Performance.....	25
ASP .NET STARTER KIT	25
The Commerce and Portal Starter Kits	25
The Time Tracker Starter Kit.....	25
The Reports Starter Kit	26
The Community Starter Kit	26
DATABASE WITH ASP .NET	27
Microsoft SQL Server	27
Oracle	27
MySQL	27
PostgreSQL	28
SQLite	28
Firebird	28
USE CASES OF ASP .NET	29
Business Use Cases ASP.NET Would Suit	29
Any Internal Software Required to Deal with Data.....	29
Any Web Applications Which Must Generate Dynamic Content.....	29

INDUSTRIES USE .NET IN THEIR BUSINESS	29
<i>Samsung</i>	29
<i>Microsoft</i>	30
<i>JP Morgan Chase</i>	30
<i>Intel</i>	30
<i>Cisco</i>	30
<i>Dell</i>	31
<i>Capgemini</i>	31
<i>Intuit</i>	31
<i>GoDaddy</i>	32
<i>Xero</i>	32
WHY CHOOSE .NET?	32
CONCLUSION	33

OBJECTIVE

The objective of this document is to describe ASP.NET and its different modules. We will elaborate on the use of ASP .NET in various industries and how we will provide you more advance, secure web applications based on your requirement.

INTRODUCTION

The choice of the best technology platform is most important thing to consider for the app development process. With the constant change and demand in the technology marketplace, Microsoft created Asp.Net Framework to develop feature-rich websites and web applications. Active server pages or ASP is the perfect platform for building high-end applications. Dot Net framework is the best server-side scripting technology where Windows web server is used to host ASP.NET websites and web applications. The Introduction of ASP by Microsoft corporation is one of the greatest achievements to come across the market demand of dynamic web pages which can be seen on different web browsers. With the Dot Net framework, anyone can build reliable and reusable applications.

WHAT IS ASP .NET?

Dot Net (pronounced as .NET) is a free, open-source software development framework created by Microsoft. It offers programming tools and guiding principles which can be used to develop an extensive range of applications for the web, mobile devices, desktops, Internet of Things (IoT), and game development.

Since Microsoft released the first version in 2002, web developers and businesses have used the Dot Net Framework to build form-based and web-based applications.

- The Dot Net Framework includes a large class library termed the Framework Class Library (FCL) and the Common Language Runtime (CLR), which allows software applications developed on the platform interoperability across various programming languages.
- With the Dot Net Framework, developers can build and run software applications for Windows, Windows Mobile, Microsoft Azure, Windows Server, and XML Web services on a single integrated development environment.
- The prime focus of the Dot Net framework was to remove the major challenges of application development, such as high ownership expenses, ease of deployment, difficulty in changing applications, and extended development periods. The CLR and FCL addressed lots of these issues.



History of ASP .NET

The unofficial development of .NET Framework started in 1999. The official announcement about the development of new technology took place at the Microsoft press conference on the 13th of January 2000. Finally, after a few years of active development, the world witnessed the power of this revolutionary platform. The first version of Microsoft .NET Framework was officially released on the 13th of February 2002. Since the release of the first .NET Framework version, nine more upgrades took place. The final version of .NET Framework was 4.8. Microsoft announced development of .NET Core on 12th of November 2014.

.NET Core had an official release on 27th of June 2016. Microsoft updated .NET Core and released its new versions till 3.1. However, in November 2020 the newest version of .NET Core was shipped as .NET 5. “Core” was removed

from the name and version 4.0 was skipped to avoid the confusion with .NET Framework. .NET 6 is the next major November 2021 upgrade of the .NET platform. Microsoft has already published .NET 6 and it could be downloaded from the official Microsoft website. According to the information the release of dot NET 7 is planned for November 2022. The most relevant updates as well as news about .NET 6 and .NET 7 can be found in official Microsoft's documentation.



WEB FRAMEWORKS

ASP.NET

a server-side web application framework designed for web development to produce dynamic web pages. It is the successor to Microsoft's Active Server Pages (ASP) technology built on the Common Language Runtime (CLR). It provides separate patterns for developing web applications ASP.NET MVC, ASP.NET Web API, and ASP.NET Web Pages (a platform using only Razor pages), which have merged into a unified MVC 6.

ASP.NET Core

a successor and re-implementation of ASP.NET as a modular web framework, together with other frameworks like Entity Framework. The framework uses the new open-source .NET Compiler Platform (codename "Roslyn") and is cross platform.

Blazor

a free and open-source web framework that enables developers to create Web apps using C# and HTML. Blazor Server apps are hosted on an ASP.NET Core server in ASP.NET Razor format, while Blazor WebAssembly apps are Single-page apps that are downloaded to the client's web browser before running.

Numerical libraries

Open-source numerical libraries

AForgE.NET

a computer vision and artificial intelligence library. It implements a number of genetic, fuzzy logic and machine learning algorithms with several architectures of artificial neural networks with corresponding training algorithms.

ALGLIB

a cross-platform open source numerical analysis and data processing library. It consists of algorithm collections written in different programming languages (C++, C#, FreePascal, Delphi, VBA) and has dual licensing – commercial and GPL.

Leverage our expert .NET services to harness the potential of futuristic technology

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Math.NET Numeric

aims to provide methods and algorithms for numerical computations in science, engineering and everyday use. Covered topics include special functions, linear algebra, probability models, random numbers, interpolation, integral transforms and more. MIT/X11 license.

Meta.Numerics

is a library for advanced scientific computation in the .NET Framework.

Proprietary numerical libraries

ILNumerics.Net

a commercial high performance, typesafe numerical array classes and functions for general math, FFT and linear algebra, aims .NET/mono, 32&64 bit, script-like syntax in C#, 2D & 3D plot controls, efficient memory management. Released under GPLv3 or commercial license.

Measurement Studio

a commercial integrated suite UI controls and class libraries for use in developing test and measurement applications. The analysis class libraries provide various digital signal processing, signal filtering, signal generation, peak detection, and other general mathematical functionality.

NMath

Commercial numerical component libraries for the .NET platform by CenterSpace Software, including signal processing (FFT) classes, a linear algebra (LAPACK & BLAS) framework, and a statistics package.

3D graphics

Open-source 3D graphics

Open Toolkit (OpenTK)

a low-level C# binding for OpenGL, OpenGL ES and OpenAL. It runs on Windows, Linux, Mac OS X, BSD, Android and iOS. It can be used standalone or integrated into a GUI.

Windows Presentation Foundation (WPF)

a graphical subsystem for rendering user interfaces, developed by Microsoft. It also contains a 3D rendering engine. In addition, interactive 2D content can be overlaid on 3D surfaces natively. It only runs on Windows operating systems.

Proprietary 3D graphics

Unity

a cross-platform game engine developed by Unity Technologies and used to develop video games for PC, consoles, mobile devices and websites

Image processing

AForge.NET

a computer vision and artificial intelligence library. It implements a number of image processing algorithms and filters. It is released under the LGPLv3 and partly GPLv3 license. Majority of the library is written in C# and thus cross-platform. Functionality of AForge.NET has been extended by the Accord.NET library.

Accord.NET

another computer vision and artificial intelligence library, available under the Gnu Lesser General Public License, version 2.1. It is mainly written in C#.

Graphical user interface

Gtk#

C# wrappers around the underlying GTK+ and GNOME libraries, written in C and available on Linux, MacOS and Windows.

Windows Forms (WinForms)

Microsoft GUI framework. The original Microsoft implementation runs on Windows operating systems and provides access to Windows User Interface Common Controls by wrapping the Windows API in managed code. The alternative Mono implementation is open source and cross-platform (it runs on Windows, Linux, Unix and OS X). It is mainly compatible with the original implementation but not completely. The library is written in C# in order to avoid Windows dependence.

At the Microsoft Connect event on December 4, 2018, Microsoft announced releasing of Windows Forms as open-source project on GitHub. It is released under the MIT License. Windows Forms has become available for projects targeting the .NET framework. However, the framework is still available only on Windows platform and the Mono incomplete implementation of WinForms remains the only cross-platform implementation.

Windows Presentation Foundation (WPF)

a graphical subsystem for rendering user interfaces in Windows-based applications by Microsoft. It is based on DirectX and employs XAML, an XML-based language, to define and link various interface elements. WPF applications can be deployed as standalone desktop programs or hosted as an embedded object in a website.

At the Microsoft Connect event on December 4, 2018, Microsoft announced releasing of WPF as open-source project on GitHub.[20] It is released under the MIT License. Windows Presentation Foundation has become available for projects targeting the .NET framework. However, the system is still available only on Windows platform.

Windows UI Library (WinUI)

a set of Microsoft UI controls and features for the Universal Windows Platform (UWP). At the Microsoft Connect event on December 4, 2018, Microsoft announced releasing of WinUI as open-source project on GitHub.[20] WinUI has become available for projects targeting the .NET framework. It is released under the MIT License. However, the library is still available only on Windows platform.

Xamarin.Forms

a cross-platform UI toolkit for development of native user interfaces that can be run on macOS, iOS, Android, and Universal Windows Platform apps

.NET Multi-platform App UI (.NET MAUI)

a cross-platform UI toolkit announced in May 2020 that originated as a fork of Xamarin.Forms and that can run on Android, iOS, Linux, macOS, Tizen, and Windows. .NET MAUI will run on .NET 6 and later. The source code is licensed under MIT License and available on GitHub.

Quality assurance

NUnit

an open source unit testing framework for .NET, written in C# and thus cross-platform. It is one of many programs in the xUnit family. Licensed under MIT License.

Object–relational mapping

.NET Framework natively provides utilities for object–relational mapping through ADO.NET, a part of .NET stack since .NET 1.0. In addition, a number of third-party object–relational libraries have emerged, especially in earlier years of the .NET development, in order to fill some perceived gaps of the framework.

As the framework has evolved, additional object–relational tools were added, such as the Entity Framework included with the .NET Framework 3.5. LINQ to SQL was also introduced with .NET 3.5. This somehow reduced significance and popularity of third-party object–relational libraries.

Entity Framework

an open-source object–relational mapping (ORM) framework for ADO.NET. It was a part of .NET Framework, but since Entity framework version 6 it is separated from .NET framework.

NHibernate

NHibernate is an object–relational mapper for the .NET platform.

PROS OF DOT NET FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT

Code Reusing

This framework works on object-oriented programming which removes unnecessary codes and includes less coding for the ASP.NET developers. The Dot Net framework consists of reusable code and several reusable components. This turns into less time and subsequently less cost to develop applications.

Flexible Deployment

With features like no-impact applications, controlled code sharing, private components, partially trusted code, and side-by-side versioning, the Dot Net framework makes deployment easier post-development. The code execution environment helps safe code execution for reduced conflicts in software deployment and versioning which minimize performance issues of scripted or interpreted environments.



Use for Service-Oriented Architecture (SOA)

The Dot Net framework is often used for Web Services, which is a solution for implementing an SOA strategy. Through Web Services, applications that are designed in different programming languages or platforms can communicate and transfer data using standard Internet protocols.

Security and Safety

Since the beginning, Dot Net development services have been preserved as a challenging technological framework in numerous organizations. From government organizations to MNCs, the dot net development structure has worked as an associate info security framework.

Furthermore, for software development, ASP.NET framework and net APIs like SQL Server and Azure DocumentdB, a formidable technological system will be utilized for several industries, along with, finance, digital marketing, and education.

The Dot Net framework provides enhanced application security as internet software is developed by ASP.NET which contains Windows validation and configuration. There are two advanced features – Managed code and CLR which provide an additional safety option like code access security and role-based security.

Use across platform and language

Many Dot Net Development Companies allow developers to build applications for browsers, desktops, mobile phone browsers, or an application operating on the personal organizations. The .NET framework is promoted as a language-independent system, which means that development can take place in several compliant languages that involve managed C++, C#, VB.NET, IronPython, Visual COBOL, IronRuby, and much more.

Vast and Viable

When it comes to the quality of the Dot Net framework, it offers additional versatility and flexible mobile application development. In fact, the wide range of communities of the .NET framework has developed numerous projects to foster the connection between websites and mobile devices.

Wireless Universal Resource File (WURFL), a structural database for mobile phones which has the name, type, sort and variant of the browser, dimension of the screen, and many other additional features is the flagship project for this set of Dot Net framework.

This collection of data provides ASP.NET developers and a mobile and web development company additional information regarding the mobile traffic of their website. This database can be easily incorporated into any Dot Net project.

In general, it's eminent that Dot Net supported mobile applications deliver unbelievable user experiences and assist all kinds of companies in enhancing the use across different Platforms and Languages.

MVC

ASP.NET MVC development services offers robust, patterns-based progress to build dynamic sites that permits clean segregation of considerations, which delivers full control over markup for amiable, agile development. Dot Net MVC development includes several options that modify fast, TDD-friendly development for building refined applications that use the most recent internet standards.

Advanced UI Control

The Microsoft Dot Net framework offers high user interface (UI) controls. Not just it has an intense set of in-built UI controls. But it also facilitates third-party UI controls like Telerik. Telerik application provides options like drag-and-drop that help lesser development time with larger user expertise and improved visual effects and program.

Cross-platform Support

When it comes to web application development, we need to ensure that it is supported by all the platforms. The latest ASP.NET core software is a cross-platform development framework that allows to build easy web application that runs smoothly on OS like Windows, Mac, and Linux.

For example, Flutter Development framework is used to build an iOS app, and then the same principle can be utilized to create an Android application.

Cloud-Based Web Application Development Support

The ability of the Dot Net framework to precede all kinds of XML documents and write in any type of format of the file with pace and ease gives various routes for integration.

Integration with Legal Systems

The ability of the Dot Net framework to precede all kinds of XML documents and write in any type of format of the file with pace and ease gives various routes for integration.

Quick development

The Dot Net framework allows the fast and simple development of applications. Microsoft .NET applications are easily deployed on any Microsoft server. The Dot Net framework facilitates one-time setup creation to deploy applications faster.

Our Strategic delivery locations help to push the latest technology to their full potential.

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

CONS OF DOT NET FRAMEWORK FOR BUSINESS APPLICATION DEVELOPMENT

Legacy

Despite Core 1.0 had been released on summer 2016 and Core 3.0 is planned to be released on the second half of 2019, many CMS and eCommerce solutions are still based on a non-Core version of .NET that makes .NET Core features and advantages usage impossible.

It will take a few years till existing products based on .NET Core will be fully updated.

Learning Curve

Though there are a lot of .NET developers on the market, ASP.NET

Core framework is a huge step forward comparing to ASP.NET

Framework and there are many changes and new concepts. It requires time and extra efforts for developers to learn this new tool.

Indeed, I should admit that a new version is very welcomed by developers, and they are usually quite motivated to get a handle on it.

Windows-based Development Tools

Microsoft released Visual Studio for Mac. But there are many other development tools that work better with Windows, and it takes some time till they also become cross-platform.



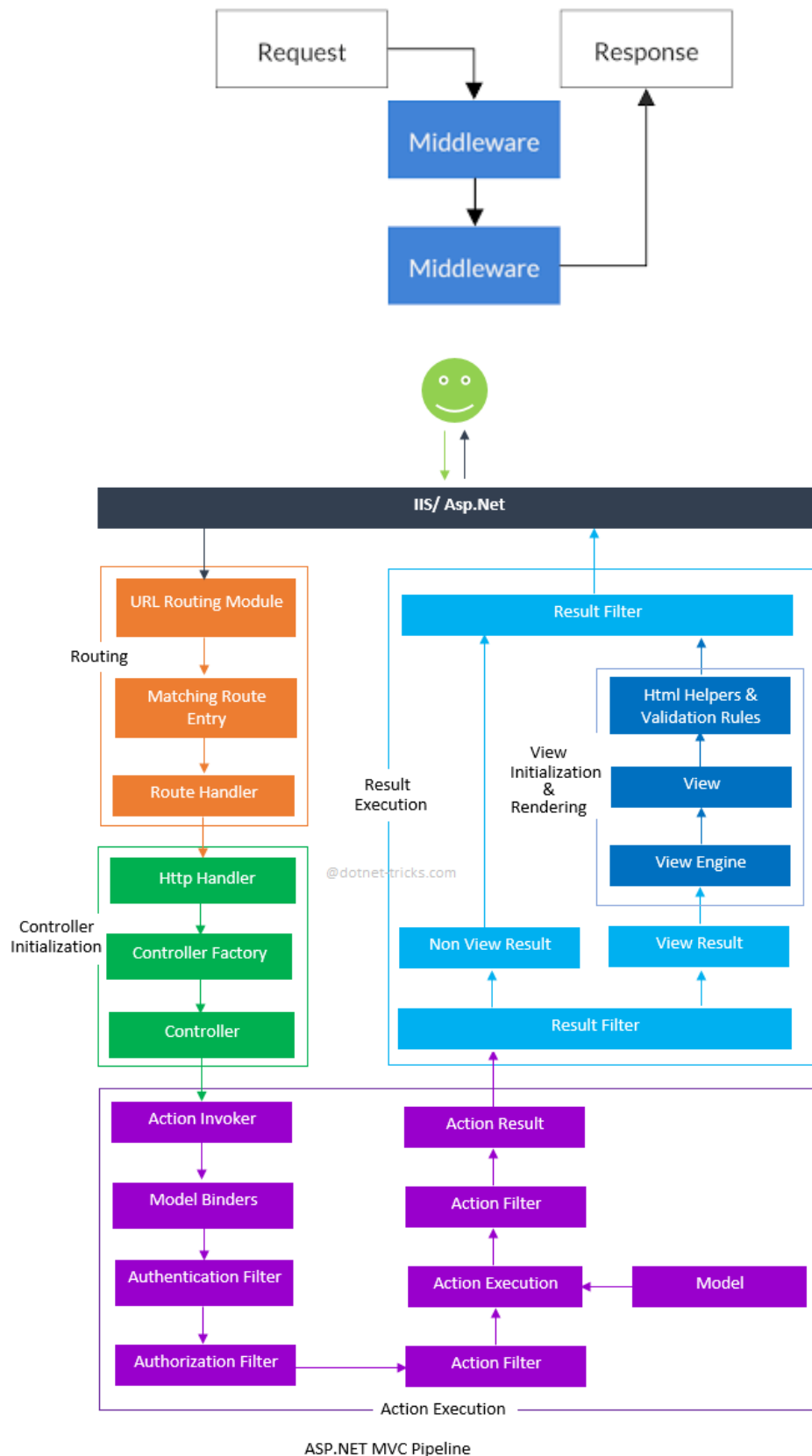
.NET PIPELINE

The ASP.NET Core request processing pipeline consists of a sequence of middleware components that are going to be called one after the other.

- Each middleware component can perform some operations before and after invoking the next component using the next method. A middleware component can also decide not to call the next middleware component which is called short-circuiting the request pipeline.
- The middleware component in asp.net core has access to both the incoming request and the outgoing response.
- The most important point that we need to keep in mind is the order in which the middleware components are added in the Configure method of the Startup class defines the order in which these middleware components are going to be invoked on requests and the reverse order for the response. So, the order is critical for defining the security, performance, and functionality of the application.

BUILDING CUSTOM MIDDLEWARE

- Middleware is software that's assembled into an app pipeline to handle requests and responses.
- ASP.NET Core provides a rich set of built-in middleware components, but in some scenarios, someone might want to write a custom middleware. This topic describes how to write convention-based middleware.



.NET FRAMEWORKS:

- .NET Framework is the original implementation of .NET. It supports running websites, services, desktop apps, and more on Windows.
- The two major components of .NET Framework are the Common Language Runtime and the .NET Framework Class Library.
 1. The Common Language Runtime (CLR) is the execution engine that handles running applications. It provides services like thread management, garbage collection, type-safety, exception handling, and more.
 2. The Class Library provides a set of APIs and types for common functionality. It provides types for strings, dates, numbers, etc. The Class Library includes APIs for reading and writing files, connecting to databases, drawing, and more.
- .NET applications are written in the C#, F#, or Visual Basic programming language. Code is compiled into a language-agnostic Common Intermediate Language (CIL). Compiled code is stored in assemblies—files with a .dll or .exe file extension.
- When an app runs, the CLR takes the assembly and uses a just-in-time compiler (JIT) to turn it into machine code that can execute on the specific architecture of the computer it is running on.

DATABASE MODELING, AND CONFIGURATION

Database configuration

Application configuration files contain settings that are specific to a particular application. For example, an ASP.NET application can have one or more web.config files, and a Windows application can have an optional app.config file. Configuration files share common elements, although the name and location of a configuration file vary depending on the application's host.



Database modeling

A data model is basically a visual representation that describes the connections between different data points and structures stored in an information system. The structure determines how the data is stored and how the system will access it. Efficient data modeling software is needed for an appropriate structure of the data. When conducting due diligence about leveraging the data to fuel business decisions, the integrity of said data is a critical prerequisite.

Each instance of a database is more or less identical. Relationships and rules are designed and programmed into the database by a modeler. An ideal programmer for this job must grasp this and formulate a plan to carry out the task accurately and efficiently.

The Three Layers of a Data Model

Data is going to be structured using a data warehouse modeling tool into one of the following three distinct layers, each with specific placement and function. Let's delve into each layer separately:

Conceptual Data Model

This is the primary level of the model that defines the structure of the data according to the business requirements. It focuses on business-oriented attributes, entries, and relationships.

Characteristics

- Provides organization-wide coverage of the business concepts
- They cater to a specific business audience.

- The conceptual layer is built independently of any hardware specs, storage capacity, or software limitations. The focus is to represent the data as viewed in the real world.

Logical Data Model

This layer is more complex and structured than the conceptual layer. It contains information about how the model should be implemented by defining the structure and relationships of the data elements. The main advantage of the logical model is that it provides a firm foundation for the logical and physical data model.

Characteristics

- The logical model lists the project's requirements, but it could also integrate with other data models depending on the scope.
- It is designed and developed independently from DBMS.
- The data types of the data elements have a precise length.

Physical Data Model

The physical layer illustrates how to implement a data model in a database management system. It streamlines the implementation methodology in the form of tables, indexes, partitioning, etc. The physical data model diagram helps visualize the entire structure of the database.

Characteristics

- The physical model lists the needs of a single project, but depending on the scope of the project, it could also integrate with other physical models.
- This model contains the table of relationships between the data, addressing the nullability and cardinality of said relationships.
- It is designed and developed specifically for a specific DBMS version, the technology used for the project, required data storage, and location.
- For the sake of exact representation of the data, all columns must have a precise data type, default values, and length.
- Columns should have exact data types, lengths, and default values.
- Primary keys and foreign keys, access profiles, indexes, and authorization are pre-defined.

.NET DIFFERENT LIBRARIES AND THEIR WORKING

Class libraries are the shared library concept for .NET. They enable to componentize useful functionality into modules that can be used by multiple applications. They can also be used as a means of loading functionality that is not needed or not known at application startup. Class libraries are described using the .NET Assembly file format.



There are three types of class libraries that can be used:

- **Platform-specific class libraries** have access to all the APIs in each platform (for example, .NET Framework on Windows, Xamarin iOS), but can only be used by apps and libraries that target that platform.
- **Portable class libraries** have access to a subset of APIs and can be used by apps and libraries that target multiple platforms.
- **.NET Standard class libraries** are a merger of the platform-specific and portable library concept into a single model that provides the best of both.

Platform-specific class libraries

Platform-specific libraries are bound to a single .NET platform (for example, .NET Framework on Windows) and can therefore take significant dependencies on a known execution environment. Such an environment exposes a known set of APIs (.NET and OS APIs) and maintains and exposes expected state (for example, Windows registry).

Developers who create platform-specific libraries can fully exploit the underlying platform. The libraries will only ever run on that given platform, making platform checks or other forms of conditional code unnecessary (modulo single sourcing code for multiple platforms).

Platform-specific libraries have been the primary class library type for the .NET Framework. Even as other .NET implementations emerged, platform-specific libraries remained the dominant library type.

Portable class libraries

Portable libraries are supported on multiple .NET implementations. They can still take dependencies on a known execution environment; however, the environment is a synthetic one that's generated by the intersection of a set of concrete .NET implementations. Exposed APIs and platform assumptions are a subset of what would be available to a platform-specific library.

Choose a platform configuration when creating a portable library. The platform configuration is the set of platforms that is needed to be supported (for example, .NET Framework 4.5+, Windows Phone 8.0+). The more platforms are opted to support, the fewer APIs and fewer platform assumptions one can make, the lowest common denominator. This characteristic can be confusing at first, since people often think "more is better" but find that more supported platforms result in fewer available APIs.

.NET Standard class libraries

.NET Standard libraries are a replacement of the platform-specific and portable libraries concepts. They are platform-specific in the sense that they expose all functionality from the underlying platform (no synthetic platforms or platform intersections). They are portable in the sense that they work on all supporting platforms.

.NET Standard exposes a set of library contracts. .NET implementations must support each contract fully or not at all. Each implementation, therefore, supports a set of .NET Standard contracts. The corollary is that each .NET Standard class library is supported on the platforms that support its contract dependencies.

.NET Standard does not expose the entire functionality of .NET Framework (nor is that a goal), however, the libraries do expose many more APIs than Portable Class Libraries.

The following implementations support .NET Standard libraries:

- .NET Core
- .NET Framework
- Mono
- Universal Windows Platform (UWP)

- For more information, see .NET Standard.

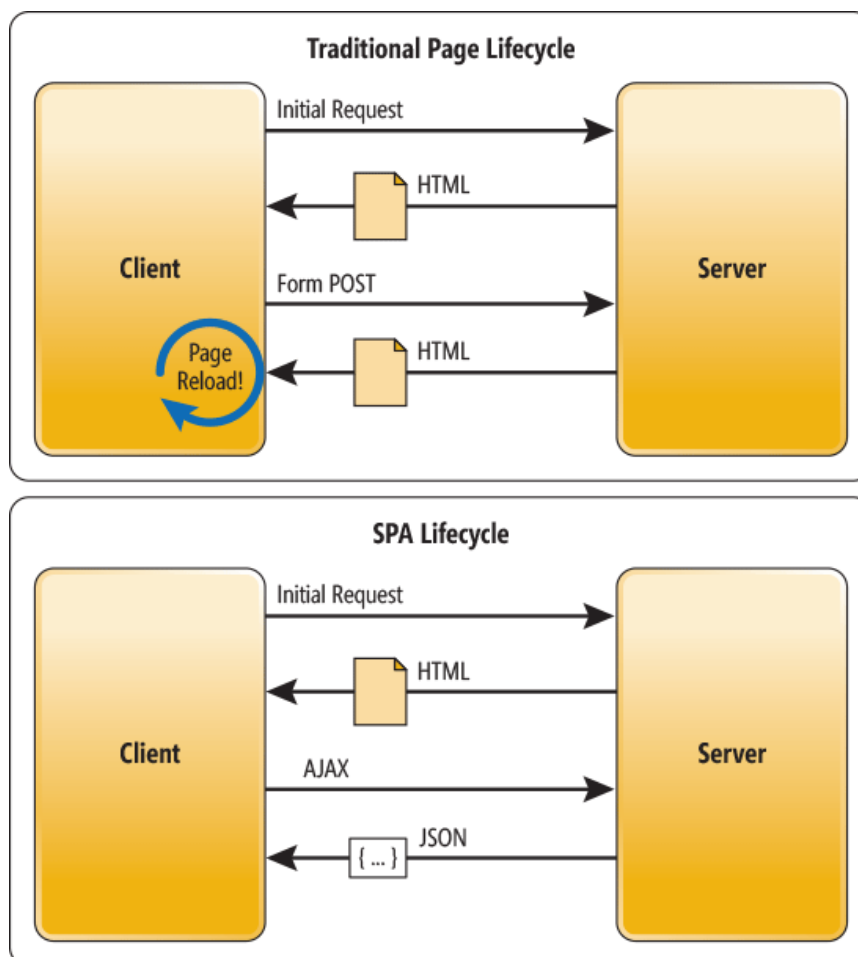
Mono class libraries

Class libraries are supported on Mono, including the three types of libraries described previously. Mono is often viewed as a cross-platform implementation of .NET Framework. In part, this is because platform-specific .NET Framework libraries can run on the Mono runtime without modification or recompilation. This characteristic was in place before the creation of portable class libraries, so was an obvious choice to enable binary portability between .NET Framework and Mono (although it only worked in one direction).

SINGLE PAGE APPLICATION IN .NET

Single-Page Applications (SPAs) are Web apps that load a single HTML page and dynamically update that page as the user interacts with the app.

SPAs use AJAX and HTML5 to create fluid and responsive Web apps, without constant page reloads. However, this means much of the work happens on the client side, in JavaScript. For the traditional ASP.NET developer, it can be difficult to make the leap. Luckily, there are many open source JavaScript frameworks that make it easier to create SPAs.



DEPLOYMENT OF .NET PROJECT

The .NET Framework provides several basic features that make it easier to deploy the applications:

No-impact applications.

This feature provides application isolation and eliminates DLL conflicts. By default, components do not affect other applications.

Private components by default.

By default, components are deployed to the application directory and are visible only to the containing application.

Controlled code sharing.

Code sharing requires to explicitly make code available for sharing instead of being the default behavior.

Side-by-side versioning.

Multiple versions of a component or application can coexist, choose which versions to use, and the common language runtime enforces versioning policy.

XCOPY deployment and replication.

Self-described and self-contained components and applications can be deployed without registry entries or dependencies.

On-the-fly updates.

Administrators can use hosts, such as ASP.NET, to update program DLLs, even on remote computers.

Integration with the Windows Installer.

Advertisement, publishing, repair, and install-on-demand are all available when deploying the application.

Enterprise deployment.

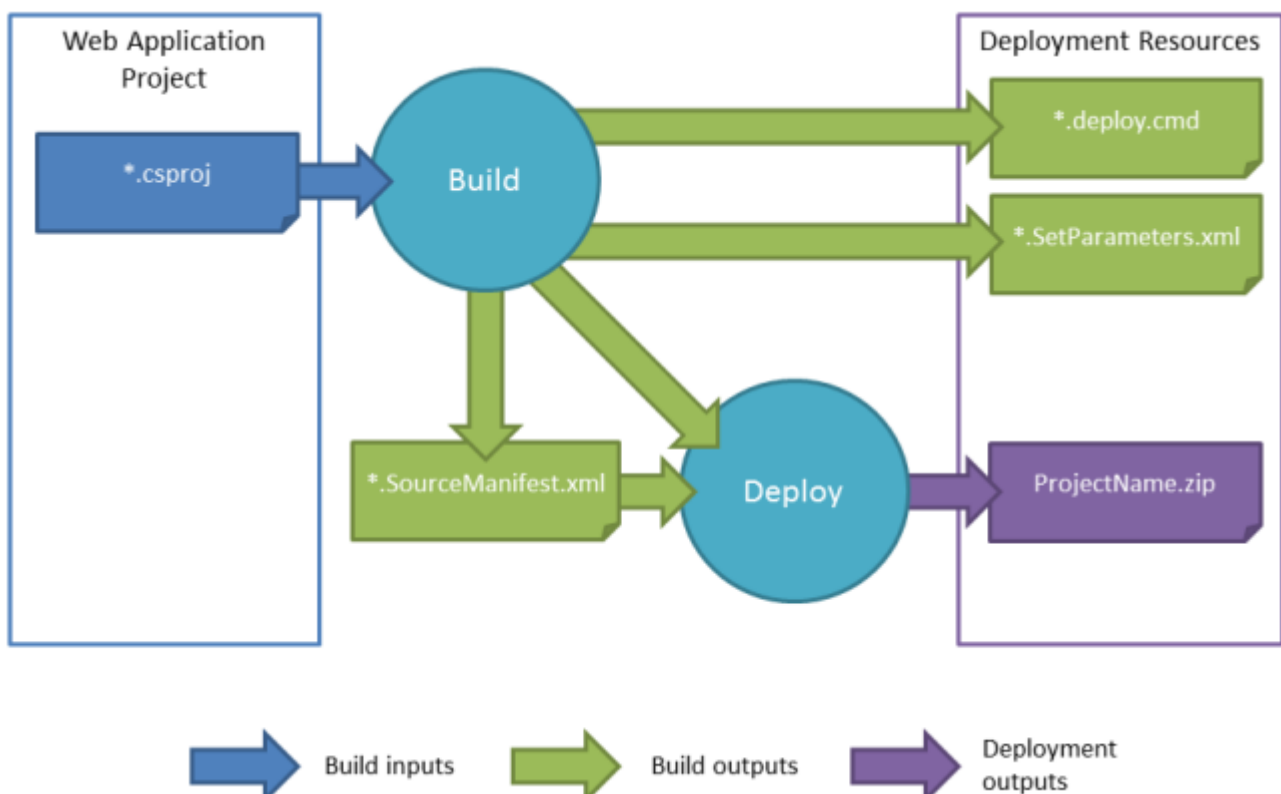
This feature provides easy software distribution, including using Active Directory.

Downloading and caching.

Incremental downloads keep downloads smaller, and components can be isolated for use only by the application for low-impact deployment.

Partially trusted code.

Identity is based on the code instead of the user, and no certificate dialog boxes appear.



PROVIDING HIGH, MODERN WEB AUTHENTICATION AND AUTHORIZATION SECURITY

Authentication in ASP.NET

There are two closely interlinked concepts at the heart of security for distributed applications - authentication and authorization. Authentication is the process of obtaining some sort of credentials from the users and using those credentials to verify the user's identity. Authorization is the process of allowing an authenticated user access to resources. Authentication is always preceding to Authorization; even if the application lets anonymous users connect and use the application, it still authenticates them as being anonymous.

Authentication providers

Assuming IIS passes a request to ASP.net, what happens next? The answer depends on the configuration of ASP.net itself. The ASP.net architecture includes the concept of an authentication provider a piece of code whose job is to verify credentials and decide whether a particular request should be considered authenticated. Out of the box ASP.net gives a choice of three different authentication providers.

- The windows Authentication provider lets someone authenticate users based on their windows accounts. This provider uses IIS to perform the authentication and then passes the authenticated identity to the code. This is the default provided for ASP.net.
- The passport authentication provider uses Microsoft's passport service to authenticate users.
- The forms authentication provider uses custom HTML forms to collect authentication information and lets user uses his own logic to authenticate users. The user's credentials are stored in a cookie for use during the session.

Windows authentication and IIS

If one select windows authentication for his ASP.NET application, he also have to configure authentication within IIS. This is because IIS provides Windows authentication. IIS gives him a choice for four different authentication methods:

Anonymous, basic digest, and windows integrated

- If one selects anonymous authentication, IIS doesn't perform any authentication, any one is allowed to access the ASP.NET application.
- If one selects basic authentication, users must provide a windows username and password to connect. However, this information is sent over the network in clear text, which makes basic authentication very much insecure over the internet
- If one selects digest authentication, users must still provide a window username and password to connect. However, the password is hashed before it is sent across the network. Digest authentication requires that all users be running

We offer full stack development and trainings on all the latest Microsoft and Non-Microsoft technologies.

Cognitive Convergence

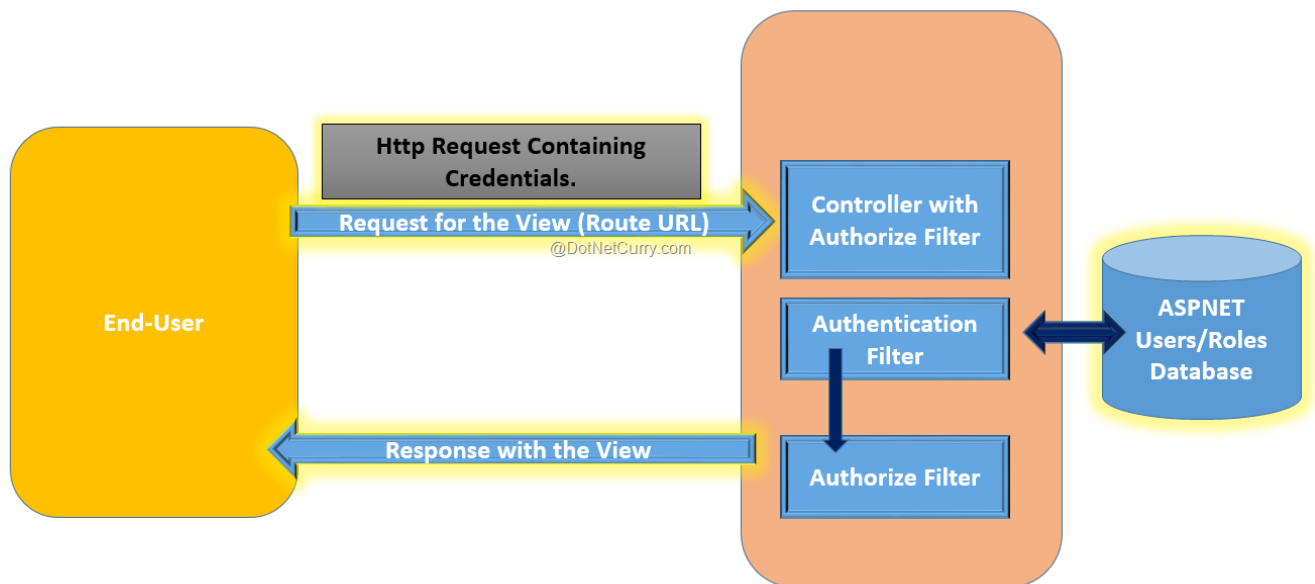
<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Internet Explorer 5 or later and that windows accounts to store in active directory.

- If one selects windows integrated authentication, passwords never cross the network. Users must still have a username and password, but the application uses either the Kerberos or challenge/response protocols authenticate the user.



Passport authentication

Passport authentication lets user to use Microsoft's passport service to authenticate users of his application. If the users have signed up with passport, and he configure the authentication mode of the application to the passport authentication, all authentication duties are offloaded to the passport servers.

Forms authentication

Forms authentication provides with a way to handle authentication using own custom logic with in an ASP.NET application. The following applies if one choose forms authentication.

1. When a user requests a page for the application, ASP.NET checks for the presence of a special session cookie. If the cookie is present, ASP.NET assumes the user is authenticated and processes the request.
2. If the cookie isn't present, ASP.NET redirects the user to a web form one provides
3. One can carry out whatever authentication, checks he like in his form. When the user is authenticated, he indicates this to ASP.NET by setting a property, which creates the special cookie to handle subsequent requests.

Configuring Authorization

After the application has authenticated users, you can proceed to authorize their access to resources. But there is a question to answer first: Just who is the user to whom your are grating access? It turns out that there are different answers to that question, depending on whether you implement impersonation. Impersonation is a technique that allows the ASP.NET process to act as the authenticated user, or as an arbitrary specified user.

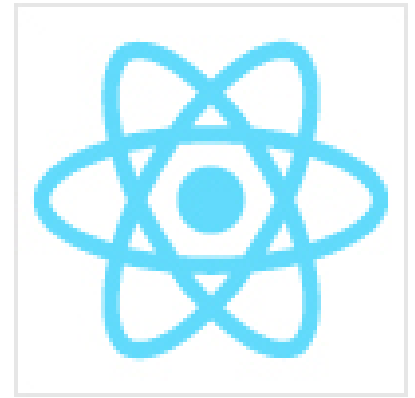
BEST FRONT-END FRAMEWORKS

React

React is the open-source framework developed and created by Facebook. This framework is the best UI framework of 2022, used by a majority of the frontend developers according to Stack Overflow Developer's survey 2021.

Its team was concerned with procuring excellent performance by forming a useful UI. The primary intention was to fix code maintainability problems due to the constant addition of features in the app.

A front-end framework, React stands apart because of its virtual Document Object Model (DOM), which presents its excellent functionality. It is a perfect framework for those who expect high traffic and require a steady platform to manage it.



Furthermore, this framework is user-friendly for new developers – the guides also aid to deal with any complications that happen during the learning process.

Pros

- Saving time while re-utilizing components
- Virtual DOM enhances both the experience of the users and the work of the developer
- An open-source library with a diversity of tools
- The steady code is supplied by one direction data movement

Cons

- Absence of documentation due to significant pace development
- The comparatively long learning curve
- The developers find it challenging to comprehend the complications of JSX

When to use:

React is utilized for developing rich user interface, particularly when you need to build single-page apps. It is the most robust front end framework when you require to make an interactive interface with less time since it supports reusable components.

When not to use:

When you do not have hands-on knowledge with JavaScript, React is not the suggested alternative. Similarly, for inexperienced developers, the JSX learning curve is severe.

Angular

The list of best front end development frameworks is incomplete without mentioning Angular. It is a simple and straightforward framework that is based on TypeScript.

Formally released in 2016, the Angular framework was established by Google to link the gap between the mounting demands of technology and conventional notions that displayed the results.



In contrast to React, Angular is exclusive with its two-way data binding trait. It means that there is actual time synchronization between view and model, where any alteration in the model replicates promptly on view and vice versa.

If your scheme contains developing web or mobile apps, Angular is a perfect choice. As an Angular development agency in Canada, our web developers build progressive web applications and single page applications with the framework.

Companies like Xbox, BMW, Blender, and Forbes are some of the best applications developed with Angular.

In comparison of Angular vs React, Angular is not simple to learn. However, there is countless documentation accessible; they are either too composite or perplexing to read.

Pros

- Making the coding procedure easier due to its refactoring services and enhanced navigation
- The component-based pattern of Angular sanctions forms a user interface with single components
- Large Ecosystem
- Angular Material reorganizes Material Design interface production
- High Performance

Cons

- Angular complication
- Relocating legacy schemes from AngularJS to Angular
- The CLI documentation is not fairly defined
- The learning effort

When to use:

Angular enlarges the execution of browser-based apps by vigorously modernizing the contents in less time since it exercises two-way data binding. For an enterprise based and active web app, utilizing Angular is apt.

When not to use:

Angular is a comprehensive solution as a front-end framework. If you need to develop apps with inadequate scopes, you will not be capable of employing the resources that Angular furnishes.

Vue.js

Nowadays, one of the simplest frameworks is Vue.js. It is worthy of eliminating the intricacies that Angular software developers face.

It has a small size and presents two main benefits – visual DOM and component-based. It also employs two-way data binding. This frontend framework is versatile, and it aids you with various tasks when building web applications.

From constructing web and mobile applications to progressive web apps (PWA), it can manage both dynamic and simple developments with simplicity.

The difference between Vue and React is that Vue is a JS framework while React is a JS library. So Vue is more suitable for large projects. Although Vue is



created to tackle complexities and enhance app performance, it is not very popular amongst the industry's giants yet. Similarly, comparing Angular vs Vuejs, Vue improves the performance and usability of Angular.

Nevertheless, Xiaomi, Alibaba, Reuters, and 9Gag are the users of this framework. Vue.js continues to propagate in terms of adoptions regardless of fewer takers from Silicon Valley.

Pros

- Extensive and detailed documentation
- Simplicity and clarity
- Browser dev tools extensions
- Code reusability and simple integration

Cons

- Reduced developers' community
- Flexibility leads to code irregularities

When to use:

Vue.js is one of the most popular frontend frameworks today owing to its simplicity and flexibility. It lets you scheme the whole thing from scratch and is efficacious in building massive projects as well. Apt for progressive web apps, dynamic web apps, and complex projects with a scalable and efficient architecture.

When not to use:

If you consider the support community would be there to respond to the intricacies, Vue.js is not the right track to go. Likewise, the applications necessitating steady components are not appropriate to be fabricated with Vue since the framework has presented difficulties with the firmness of parts.

Ember.js

Ember.js is a component-based framework developed in 2011. It presents two-way data binding, which is similar to Angular. It is intended to manage the growing demand for contemporary technologies impeccably.

You can build multifaceted web and mobile applications with Ember.js and expect its effective design to handle the concerns.

Nevertheless, the learning curve of Ember is one of its few deficiencies. This framework is one of the most challenging web UI frameworks to learn owing to its conventional and rigid structure.

Being comparatively new and unexplored, its developer community is trivial. Anyone with no restraint in devoting the time for acquiring its knowledge can learn it.



Pros

- Fastest development framework
- Well-managed

- Appropriate documentation
- Two-way data binding

Cons

- Slow updates and challenging syntax
- A minor community and less popular
- Substantial framework for small apps
- Difficult learning curve

When to use:

If you need to develop contemporary applications like LinkedIn with a responsive user interface, Ember.js is the right frontend framework. It comes with every mechanical front-end enablement, like seeing a more comprehensive range of applications because of outstanding routing delivered by Ember.js.

This framework positions as the complete frontend solution for an extensive project because it supplies good data binding, an equipped configuration, and custom properties to deliver the page as required.

When not to use:

Ember.js is probably not an apt choice with a smaller development team since the framework needs business logic and experience to resolve the difficulties. The preliminary cost may be more significant with Ember.js.

Similarly, for scripting simple Ajax functionalities and building user interfaces, the framework may not be suitable.

jQuery

jQuery is one of the initial frontend frameworks introduced in 2006. Regardless of its release date, what makes it stand apart is its significance in today's tech ecosystem.

Not only does jQuery present ease of use, but it also diminishes the requirement to script wide-ranging JavaScript codes.

Basically, jQuery is utilized to operate DOM and CSS manipulation on the frontend and to augment the interactivity and functionality of a website.



Though you are not able to build mobile apps with jQuery, current developments in jQuery Mobile have shown that the usage borders have been extended. Moreover, the latest development in this framework allows the developers to create native mobile apps with its HTML5 based UI scheme – jQuery Mobile.

Further, jQuery framework is browser-friendly and supports the browsers you aim to utilize.

Pros

- Conveying HTTP requests is streamlined
- Document Object Model (DOM) is flexible for accumulating or eliminating the components
- HTTP requests are made easy
- Simplifies dynamic content

Cons

- Numerous progressive substitutes are accessible other than jQuery
- APIs of the Document Object Model (DOM) is out-dated
- Reasonably sluggish working capability

When to use:

This web development framework is utilized in building desktop-based JavaScript apps. This framework retains the code crisp and fairly simple. It is employed in managing the events and executing animations.

When not to use:

When you are building a large-scale app, it is not feasible to utilize jQuery as it makes your application hefty by placing extra JavaScript code. This framework is not proficient in challenging the contemporary frameworks with the progressive enablement of JavaScript, fewer coding, and reusability of elements.

Semantic-UI

Semantic UI was invented by Jack Lukic and it got released in 2014. It is a new frontend framework that is supported by LESS and jQuery.

The objective of Semantic lies in authorizing the designers and developers by creating a language for sharing UI. It uses natural language that makes the entire code self-explanatory.

The framework is comparatively novel to the ecosphere. But with its striking user interface, simple functionalities, and features, it has turned out to be one of the most popular frontend frameworks among all others available in the market.

Fresher can simply make command as it permits a unified development procedure with incorporation to several third-party libraries.



Pros

- Has superior functionality
- Amongst the newest front-end frameworks
- Rich UI modules and receptiveness
- Less complex

Cons

- Not appropriate for learners with no knowledge of JavaScript
- Have to be skilled enough to build custom necessities
- Larger packages

When to use:

Semantic-UI is used for a super lightweight practice and smooth design of interactive user interfaces.

When not to use:

When you have a squad of beginners and are less knowledgeable with JavaScript, Semantic-UI framework is not suggested because it needs the aptitude to form customizations in the app without relying on the handy functions.

Backbone.js

Backbone.js is one of the easiest frameworks which permits you to build single-page applications quickly. It is a framework that is based on the MVC pattern. Parallel to a Controller, MVC architecture View sanctions the execution of the component logic.

This platform also lets you build projects that need dissimilar types of users, where the collections can be utilized to differentiate the models.



Consequently, irrespective of whether you aim to implement Backbone.js for backend or frontend, this is a perfect choice as its REST API compatibility delivers smooth synchronization amid the two.

Pros

- Fastest JavaScript frameworks
- Much easier to learn
- Have a lightweight framework

Cons

- Delivers uncomplicated tools to produce the app construction.
- The framework does not offer a handy structure
- Have to transcribe boilerplate code

When to use:

Backbone.js is utilized for vibrant applications like Trello. It permits developers to develop a client-side model, quicker updates, and reuse the coding. Henceforth, it is proficient in vigorously maintaining the client, conducting the updates, and continuous synchronization with the server.

When not to use:

Backbone.js offers a minimalist prerequisite for developing a web app as compared to the rest of the MVC client-side frameworks. Nevertheless, you can expand the functionalities with the provision of extensions and plugins.

Consequently, the team targeting for a comprehensive solution in one framework should not intend for Backbone.js.

Preact

Preact is a JavaScript library that defines itself as the quickest 3KB substitute to that of React containing a similar ES6 API. Preact presents the smallest possible virtual DOM abstraction in addition to DOM.

It develops on steady platform features and functions well with other available frontend and UI libraries. Preact has a small size but does



not compromise in speed, and enables developing complex dynamic web applications.

Pros

- Utilize the advantages of React-style elements in the old sites that are founded on the Backbone and jQuery
- Preact increases performance while building an app
- All these characteristics are executed in the React community

Cons

- It does not support the React prototype
- Preact does not deliver any support for synthetic event employment of React
- It still does not offer support for the context

When to use:

As it is known, Preact is a lightweight form of React so when it is about utilizing a lightweight framework then you should opt for Preact and not React.

When not to use:

Preact does not offer support to displaced functional elements. So when you are in such a requirement, then you must not use Preact.

Svelte

Svelte is one of the modern web frameworks for frontend development. This framework has made a change by placing the work into an accumulated step instead of tapping them in the browser, unlike the frameworks like Vue and React.

It transcribes the code to update the Document Object Model (DOM) in synchronization with the condition of the application.



Pros

- Scalable framework
- Improved reactivity
- It is lightweight, simple, and uses the prevailing JavaScript libraries
- Newest of all
- Quicker than any other framework, namely, React or Angular

Cons

- Lack of support
- It is not very popular yet
- Minor community
- Lack of tooling

When to use:

This framework is most suitable for small application projects having a small team. Because it does not have a larger support community, it is better if you do not use it for multifarious projects.

When not to use:

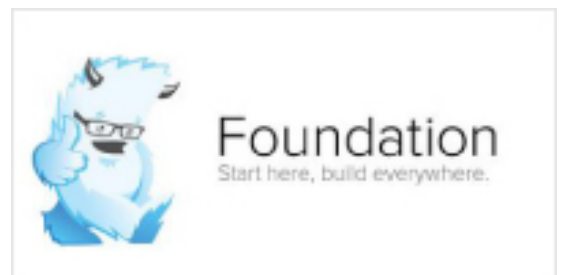
It is better if you do not use the Svelte framework for huge projects presently because the frameworks lack sufficient community as well as tooling.

It is difficult to search for answers to the questions or bugs that may be visible later in the development process because of the minor community.

Foundation

Invented by Zurb, Foundation is majorly intended for enterprise-level development of agile and responsive websites. It is advanced as well as challenging for web developers to begin developing frontend applications utilizing Foundation.

It presents GPU acceleration for swift mobile rendering features, smooth animations, and data-interchange attributes where it loads lightweight segments for heavy and mobile sections for bigger devices.



Pros

- It lets you form exquisite looking websites
- Flexible grids
- Personalized user experience for dissimilar devices and media
- HTML5 form authentication library

Cons

- Fewer community forums and support platforms
- Contender framework like Twitter, Bootstrap is more prevalent than Foundation
- Reasonably hard to learn for beginners

When to use:

If you are aiming for styled open-source, CSS components, and mobile-friendly front end framework, Foundation is better than the rest of the other options.

When not to use:

Not preferable for beginners since it is difficult to alter the code and increases the complexity because of its customization competence.

DESIGN PRINCIPLE

Interoperability

Because computer systems commonly require interaction between newer and older applications, .NET Framework provides means to access functions implemented in newer and older programs that execute outside .NET environment. Access to Component Object Model (COM) components is provided in System.Runtime.InteropServices and System.EnterpriseServices namespaces of the framework. Access to other functions is via Platform Invocation Services (P/Invoke). Access to .NET functions from native applications is via reverse P/Invoke function.

Language independence

.NET Framework introduces a Common Type System (CTS) that defines all possible data types and programming constructs supported by CLR and how they may or may not interact conforming to CLI specification. Because of this feature, .NET Framework supports the exchange of types and object instances between libraries and applications written using any conforming .NET language.

Type safety

CTS and the CLR used in .NET Framework also enforce type safety. This prevents ill-defined casts, wrong method invocations, and memory size issues when accessing an object. This also makes most CLI languages statically typed (with or without type inference). However, starting with .NET Framework 4.0, the Dynamic Language Runtime extended the CLR, allowing dynamically typed languages to be implemented atop the CLI.

Portability

While Microsoft has never implemented the full framework on any system except Microsoft Windows, it has engineered the framework to be cross-platform, and implementations are available for other operating systems (see Silverlight and § Alternative implementations). Microsoft submitted the specifications for CLI (which includes the core class libraries, CTS, and CIL), C#, and C++/CLI to both Ecma International (ECMA) and International Organization for Standardization (ISO), making them available as official standards. This makes it possible for third parties to create compatible implementations of the framework and its languages on other platforms.

Security

.NET Framework has its own security mechanism with two general features: Code Access Security (CAS), and validation and verification. CAS is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the Internet). CAS uses evidence to determine the permissions granted to the code. Other code can demand that calling code be granted a specified permission. The demand causes CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

Our .NET developers understand project-specific needs
and help you identify the right protocol

Cognitive Convergence

<http://www.cognitiveconvergence.com>

+1 4242530744

shahzad@cognitiveconvergence.com

Memory management

CLR frees the developer from the burden of managing memory (allocating and freeing up when done); it handles memory management itself by detecting when memory can be safely freed. Instantiations of .NET types (objects) are allocated from the managed heap; a pool of memory managed by CLR. As long as a reference to an object exists, which may be either direct, or via a graph of objects, the object is considered to be in use. When no reference to an object exists, and it cannot be reached or used, it becomes garbage, eligible for collection.

Performance

When an application is first launched, the .NET Framework compiles the CIL code into executable code using its just-in-time compiler, and caches the executable program into the .NET Native Image Cache. Due to caching, the application launches faster for subsequent launches, although the first launch is usually slower. To speed up the first launch, developers may use the Native Image Generator utility to manually ahead-of-time compile and cache any .NET application.

ASP .NET STARTER KIT

If you're building an ASP.NET Web site you could probably use a good, solid code foundation to start with and build upon. Wouldn't it be nice to start with a complete site, make a few tweaks and customizations, and go live? The ASP.NET Starter Kits are packaged solutions that let you do just that. The five kits—Community, Reports, Commerce, Portal, and Time Tracker—supply full, reusable code that can be easily customized. In addition, there are a number of ISPs that support automatic deployment of ASP.NET Starter Kit Web sites, leaving you with little left to do when you have to get there fast. Here, the author introduces the ASP.NET Starter Kits and builds a community Web site with lots of advanced features such as ratings, user polls, upload quotas, change notifications, and themes.

The Commerce and Portal Starter Kits

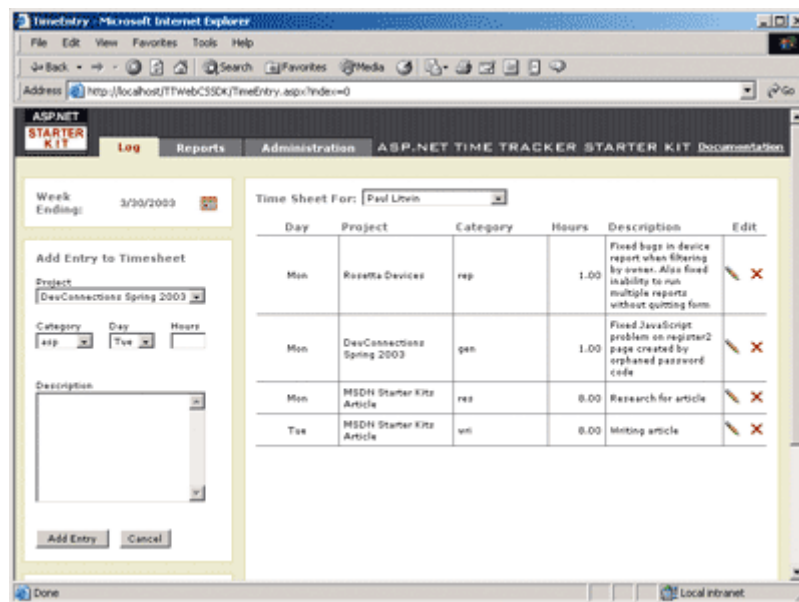
The Commerce Starter Kit creates an e-commerce storefront application for a fictitious spy company, IBuySpy. The application employs a three-tier architecture, caching, forms authentication, and a shopping cart. The Commerce Starter Kit also includes a B2B Web Service component. This is a great application to study and learn from, especially if you need to build an e-commerce application of your own.

The Portal Starter Kit creates the portal application for IBuySpy, the same fictitious spy company featured in the Commerce Starter Kit. The Portal Starter Kit features 10 portal modules: announcements, contacts, discussion, events, html/text, image, links, quicklinks, xml/xsl, and a custom module.

The Time Tracker Starter Kit

The Time Tracker Starter Kit (see Figure 1) is essentially a time and billing intranet application without the billing component. It is pretty straightforward. It has three tabs: Log, Reports, and Administration. You use the Administration tab to configure projects, the Log tab for logging hours, and the Reports tab for generating reports by project or individual.

The Time Tracker Starter Kit shows off a number of interesting ASP.NET architectural features including a three-tier design, Windows and forms authentication, dynamic chart generation using GDI+, and a mobile front end.



The Reports Starter Kit

The Reports Starter Kit differs from the other starter kits in that it is not a single cohesive application. Rather, it's a series of eight reports that you can read about, run against sample data, and examine the source code from. In addition, you can copy and paste the source into your own applications. This kit works similarly to the ASP.NET QuickStarts that ship with the Microsoft .NET Framework SDK and Visual Studio .NET. The reporting examples include simple tabular reports, two styles of hierarchical drill-down reports, cross tabs, and several GDI+ charts.

The Community Starter Kit

The Community Starter Kit may well be the star of the show. Without much effort or knowledge of how it works, you can use it to quickly put together a community site for your user group, club, family, or community. You could also use it to put together a simple news and information site. In some ways, it overlaps with the functionality provided by the Portal Starter Kit, but it focuses more on the noncommercial organization and is easier to set up and manage.



DATABASE WITH ASP .NET

The following list includes some of the more popular database systems that can be used in Visual Studio projects. The list is not exhaustive. For a list of third-party vendors that offer ADO.NET data providers that enable deep integration with Visual Studio tooling, see ADO.NET Data Providers.

Microsoft SQL Server

SQL Server is the Microsoft flagship database offering. SQL Server 2016 delivers breakthrough performance, advanced security, and rich, integrated reporting and analytics. It ships in various editions that are designed for different uses: from highly scalable, high-performance business analytics, to use on a single computer. SQL Server Express is a full-featured edition of SQL Server that is tailored for redistribution and embedding. LocalDB is a simplified edition of SQL Server Express that requires no configuration and runs in your application's process. You can download either or both products from the SQL Server Express download page. Many of the SQL examples in this section use SQL Server LocalDB. SQL Server Management Studio (SSMS) is a stand-alone database management application that has more functionality than what is provided in Visual Studio SQL Server Object Explorer. You can get SSMS from the previous link.



Oracle

You can download a paid or free edition of the Oracle database from the Oracle technology network page. For design-time support for Entity Framework and TableAdapters, you will need the Oracle Developer tools for Visual Studio. Other official Oracle products, including the Oracle Instant Client, are available through the NuGet Package Manager. You can download Oracle sample schemas by following the instructions in the Oracle online documentation.



MySQL

MySQL is a popular open-source database system that is widely used in enterprises and websites. Downloads for MySQL, MySQL for Visual Studio, and related products are at MySQL on Windows. Third parties offer various Visual Studio extensions and stand-alone management applications for MySQL. You can browse the offerings in the



NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution).

PostgreSQL

PostgreSQL is a free, open-source object relational database system. To install it on Windows, you can download it from the PostgreSQL download page. You can also build PostgreSQL from the source code. The PostgreSQL core system includes a C language interface. Many third parties provide NuGet packages for using PostgreSQL from .NET applications. You can browse the offerings in the NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution). Perhaps, the most popular package is provided by npgsql.org.



SQLite

SQLite is an embedded SQL database engine that runs in the application's own process. You can download it from the SQLite download page. Many third-party NuGet packages for SQLite are also available. You can browse the offerings in the NuGet Package Manager (Tools > NuGet Package Manager > Manage NuGet Packages for Solution).



Firebird

Firebird is an open-source SQL database system. You can download it from the Firebird download page. An ADO.NET data provider is available through the NuGet Package Manager.



USE CASES OF ASP .NET

Although many developers tend to favor ASP.NET over other types of web frameworks, it's important that you evaluate whether the use of ASP.NET will be beneficial to your specific business.



Business Use Cases ASP.NET Would Suit

The functionality and flexibility that the ASP.NET framework provides makes it very useful for businesses that need that functionality and flexibility. However, this is not the case for all companies. If you don't need it, you can use something a bit more simple to build your website. Keeping that in mind, the following are a few business use cases that are ideal for ASP.NET.

Any Internal Software Required to Deal with Data

If you're using existing software programs to collect, track and analyze data, whether it's employee data within your HR systems, time management systems, and accounting systems, or customer data within your sales and marketing systems, ASP.NET is one of the best options available to you. It allows you to display the most important of your data seamlessly within a web app. If you don't make use of your data to that great of a degree, the ASP.NET framework may not be necessary. Its ability to deal with complex and large data sets in a user-friendly way makes it particularly useful for companies.

Any Web Applications Which Must Generate Dynamic Content

A dynamic web page is a page on which different types of content is displayed every time it's browsed. The content changes based on the user's behavior, preferences, and interests. Take for example Facebook's news feed. This is a form of dynamic content. If you need to create web applications that generate dynamic content, the ASP.NET framework is the best option due to its dynamic data scaffolding and page templates features.

INDUSTRIES USE .NET IN THEIR BUSINESS

Samsung

The Samsung Group, a South Korean multinational manufacturing conglomerate, is based in Seoul, with factories worldwide. It is the largest South Korean family-owned business conglomerate and a popular provider of innovative products and services. Samsung has built a strong, loyal customer base by focusing on better user experience and lasting customer relationships.



Microsoft

Microsoft is well-known globally as a market leader that has a wide range of tech businesses, including hardware, software, cloud services, and digital products. This company gave birth to the .NET framework. While it's headquartered in Redmond, Washington, it also has offices in over 60 countries.



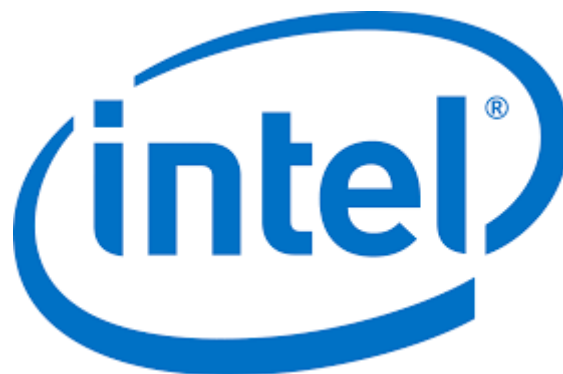
JP Morgan Chase

JPMorgan Chase is a corporation that handles investment banking and financial services. It is one of the world's leading financial institutions. The company has assets amounting to \$2.6 trillion. Its headquarterd in North America but operates in 60 countries with 240,000 employees around the world.



Intel

A leader in semiconductor chip design, Intel created the x86 series of microprocessors found in most personal computers. Although this company is more known in the computer hardware business, Intel also ventures into data centers and the internet of things. Intel is an American multinational technology company based in Santa Clara, California.



Cisco

This company is a multinational technological conglomerate with headquarters in San Jose, California. Cisco's leading products include Webex, OpenDNS, Jabber, and Jasper. It also designs, builds, and sells networking gear, software, telecommunications equipment, and other high-tech services and products.



Dell

Dell is an American company that's more commonly known for developing, selling, and maintaining computers and related goods and services. Unknown to many, though, this tech company also offers DevOps, Cloud, Edge, Workforce, and other software solutions and services. Dell's US corporate headquarters is in Round Rock, Texas.



Capgemini

Serge Kampf founded Capgemini in 1967. Capgemini is a consulting company and an information technology service provider based in Paris, France. Its range of services is in the space of cloud, cybersecurity, and artificial intelligence.



Intuit

Based in Mountain View, California, Intuit is a financial software company known for its products, TurboTax, Mint, and QuickBooks. These are accounting software for generating an income statement, cash flow statement, or taxable income statement, among many other features. Intuit has lobbied hard against the Internal Revenue Service (IRS) to offer free pre-filled forms to taxpayers.



GoDaddy

This company is an American publicly traded domain registrar and web hosting firm. GoDaddy had over 20 million customers in December 2021. The company currently hosts 82 million registered domain names through its platform.



Xero

Xero is an Australian-listed technology company with headquarters based in New Zealand. Its primary market is small businesses (SMBs) that need help in accounting and bookkeeping. As a software-as-a-service company, Xero offers cloud-based accounting tools and support to SMBs.



WHY CHOOSE .NET?

.NET is a leading technology and we've listed its main benefits:

- Unified software model;
- Simplified programming model;
- Package version control;
- Simplified deployment;
- Multi-platform;
- Automatic resource management;
- Type safety checking;
- Unified development environment;
- Ability to debug multilingual applications;
- Unified error handling model;
- Modern security model;
- Integration of programming languages;
- .NET core standardization.

CONCLUSION

As we have seen several advantages of the .NET framework for application development. The .NET development framework is a reliable, safe, and beneficial platform using which you can build ultimate business applications. It allows .NET developers to apply, expand, and scale excellent software, for both economic development and general business prosperity.

Asp.NET development services provide endless benefits for different problems such as security, memory management, phenomenal handling, etc. The .NET framework is an exciting and future dominating technology, appropriate for building robust web applications. Since most organizations search for a unique business solution to assist consumers in the best possible way, you can contact cognitive convergence.

Contact Us
Cognitive Convergence
<http://www.cognitiveconvergence.com>
+1 4242530744
info@cognitiveconvergence.com