International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

A Detailed Review on ASP.NET(MVC)

K.Ashok Reddy, MS.Kokila

Computer Engineering, Presidency University, Bengaluru, Karnataka
Assistant Professor - CSE, Presidency University, Bengaluru, Karnataka

Abstract:

Microsoft offers ASP.NET MVC (Model-View-Controller), a framework for creating dependable, expandable, and maintainable web applications. Model, View, and Controller are the three primary parts of the application according to the MVC architectural pattern.

The data and business logic of the application is represented by the Model. It contains the data access layer, communicates with databases or other data sources, and handles data validation and manipulation. Developers may effectively manage and interface with the database with the aid of technologies like Entity Framework and SQL Server.
The View is in charge of showing the user interface to them. The structure, layout, and visual appearance of the web pages are defined by HTML, CSS, and JavaScript code. The standard view engine is Razor.
The Model and the View are connected through the Controller. It responds to user requests, analyses input, and chooses which view to display. The Controller facilitates clear and user-friendly URLs by mapping incoming URLs to particular actions through the use of routing.

To improve user experience and development productivity, ASP.NET MVC applications frequently make use of additional tools and technologies. The well-known integrated development environment (IDE), Visual Studio, offers a wealth of capabilities for managing projects and editing code. The package manager NuGet makes it simple to install and manage dependencies and third-party libraries.

Keywords— **C#, SQL SERVER MANAGEMENT STUDIO, IDE, ORM**

I. INTRODUCTION

Microsoft created the well-known online application framework known as ASP.NET MVC (Model-View-Controller). It provides a methodical and effective way to create scalable, dynamic web applications. Model, View, and Controller are the three main parts of the program that are divided into ASP.NET MVC's adoption of the MVC architectural pattern.

The data and business logic of the application is represented by the Model. It includes the data access layer, communicates with databases or other sources of data, and manages tasks like data processing and validation. This division makes code organization and maintenance cleaner and simpler.

The View is in charge of showing the user interface to them. The HTML, CSS, and JavaScript code that makes up this system defines the organization, structure, and aesthetic components of web pages. ASP.NET MVC makes use of the
Using the Razor view engine, which provides seamless server-side integration of C#-written code with the view, it is possible to generate dynamic content and bind data.

Between the Model and the View, the Controller serves as a middleman. It takes user requests and processes them, carries out required tasks, and chooses which view to render. Through routing techniques, the Controller assigns incoming URLs to certain tasks, encouraging the use of tidy and approachable URLs that improve the user experience as a whole.

Projects using ASP.NET MVC make use of a variety of extra tools and technologies to speed up development and improve functionality. The integrated development environment (IDE) called Visual Studio comes with robust tools for managing projects and editing code. The installation and administration of dependencies and third-party libraries are made easier by the NuGet package manager.

In ASP.NET MVC applications, front-end frameworks like Bootstrap and JavaScript libraries like jQuery are frequently utilized to create aesthetically pleasing and interactive user interfaces. Automated tests may be easily created with the help of unit testing frameworks like NUnit or MSTest, verifying the dependability and quality of the code.

2. Introduction to C#

Microsoft created the flexible and cutting-edge programming language C# (pronounced "C sharp"). It was first launched as a component of the.NET framework in the early 2000s and has since grown to be one of the most used languages for creating a variety of applications, including those for the web, desktop, mobile, and gaming platforms.

The following are some of the main attributes and traits of C#:

Programming that emphasizes objects and classes is called object-oriented programming (OOP), and C# is predominantly an object-oriented language. It supports fundamental OOP ideas like polymorphism, inheritance, and encapsulation, enabling programmers to create modular, reusable programs.

Strongly Typed: Since C# is a statically typed language, variables must be explicitly defined at compile time along with their data types. This exact typing improves efficiency and security by catching problems earlier.

Garbage collection: C# uses a garbage collector to automatically manage memory. It automates memory allocation and deallocation, freeing developers from the laborious task of manually managing memory and lowering the possibility of memory leaks and associated issues.

Cross-Platform Development: With the release of .NET Core, it is now possible to create cross-platform programs in C# that work on Windows, macOS, and Linux. Because of this, programmers can write code just once and deploy it.

Rich Standard Library: C# comes with a robust standard library called the Framework Class Library (FCL) that includes a variety of pre-built classes and APIs for everyday activities like file I/O, networking, database access, and more. This comprehensive library streamlines routine programming chores and expedites development.

Language compatibility: C# works well with others. NET-based languages including Visual Basic.NET and F#. This enables programmers to use code and libraries created in many languages inside the same project.

Language Constructs: C# offers several language constructs, such as lambda expressions, events, delegates, control structures (if-else, loops), exception handling, and more. These characteristics improve the flexibility, readability, and expressiveness of the code.

Powerful Integrated Development Environments (IDEs) offer strong support for C# development.

like Visual Studio Code and Microsoft Visual Studio. These IDEs provide functions including project management, debugging, code auto-completion, and integration with other programming tools.

Community and Ecosystem: The developer community for C# is thriving and active, and there is a tonne of online resources, forums, and libraries at their disposal. A variety of frameworks and technologies are included in the ecosystem, including ASP.NET for web development, Xamarin for mobile app development, and Unity for game development.

Every time a new version of C# is released, new features and improvements are added. Due to its adaptability, performance, and tight connection with the .NET ecosystem, it is still a preferred option among developers.

3.IDE(Independent Development Environment):

When working on ASP.NET MVC projects, developers need an Integrated Development Environment (IDE). An all-encompassing environment for programming, testing, and deploying programs is offered by IDEs. They provide a selection of features to boost output, simplify project administration, and speed up the development process. Here is a list of popular IDEs for ASP.NET MVC development:

The most well-liked and feature-rich IDE for ASP.NET MVC programming is Microsoft Visual Studio. It offers a wide range of tools and features, including a visual designer for developing user interfaces, debugging tools, project templates, version control integration, and code editing with IntelliSense. A wide range of add-ons and extensions for Visual Studio is

available to expand functionality and support different technologies.

A portable, cross-platform code editor is called Visual Studio Code (VS Code). Because of its adaptability and extensibility, it has grown in popularity among developers. For ASP.NET MVC development, VS Code includes several extensions that include IntelliSense, debugging assistance, Git integration, and task automation. It works well with command-line tools and is very customizable.

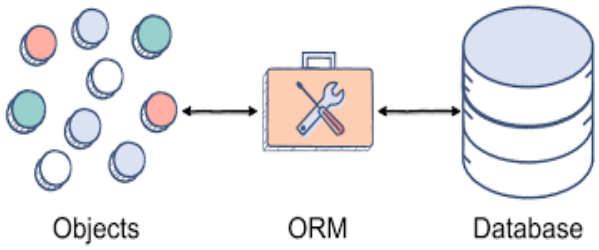
JetBrains Rider is a cross-platform integrated programming environment (IDE) made especially for .NET development, including ASP.NET MVC projects. It provides sophisticated coding support, refactoring tools, debugging capabilities, and version control system integration. A streamlined development process is offered by Rider, which also supports a wide range of technologies used in ASP.NET MVC projects.

Lightweight and adaptable code editor Sublime Text is renowned for its quickness and ease of use. It offers a large selection of plugins and extensions that may be tailored to enable ASP.NET MVC development, although lacking some of the integrated functionality of full-fledged IDEs. Developers that choose a simple environment with lots of customization love Sublime Text.

Atom: GitHub created Atom, an open-source, adaptable code editor. It offers a cutting-edge interface that is simple to use and supports a variety of languages, including C# for ASP.NET MVC development. The needs of ASP.NET MVC developers can be satisfied by the many community-developed packages and themes that Atom offers.

These IDEs give programmers strong tools and functionality to effectively design, test, and deploy ASP.NET MVC applications. Every IDE has unique strengths, and developers may have different preferences. The best IDE to use will depend on a person's preferences, the needs of the project, and the ecosystem of tools and technologies they are most at ease using.

4.ORM(OBJECT-RELATIONAL MAPPING):OVERVIEW



Object-Relational Mapping (ORM) is a technique used in ASP.NET MVC and other web application frameworks to bridge the gap between the object-oriented programming (OOP) paradigm used in application code and the relational database model used for data storage. ORM frameworks automate the mapping between database tables and application objects, simplifying data access and manipulation.

In ASP.NET MVC, Entity Framework (EF) is the commonly used ORM framework. Here is an overview of ORM and its role in ASP.NET MVC:

Object-Relational Mapping (ORM):
ORM is a programming technique that allows developers to work with relational databases using objects and classes instead of writing low-level SQL queries. It provides a layer of abstraction between the application code and the database, enabling developers to focus on business logic and object-oriented design.

Entity Framework (EF): EF is Microsoft's ORM framework for ASP.NET MVC. Creating object-oriented code based on the database schema makes database access and manipulation simpler. Several database providers are supported by EF, including SQL Server, MySQL, PostgreSQL, and SQLite.

Mapping configurations are used by ORM frameworks like EF to specify how database tables and columns relate to application objects and properties. Without the need for human SQL coding, EF automatically generates SQL queries to extract and persist data when the mapping is specified.

The EF framework supports both the code-first and database-first methods. With Code-First, developers use classes and annotations to specify the object model and relationships of their application, while EF builds the matching database schema. Developers who use the Database-First methodology begin with an existing database schema and generate the corresponding object model in the application.

CRUD Procedures:

To use application objects to conduct Create, Read, Update, and Delete (CRUD) actions on the database, ORM frameworks offer APIs. To retrieve and manipulate data, developers can utilize LINQ (Language-Integrated Query) or SQL-like queries, producing code that is cleaner and easier to maintain.

Entity Framework and other ORM frameworks make database interactions in ASP.NET MVC projects simpler and require less boilerplate code. They offer a more object-oriented approach and abstract away the SQL query complexities, freeing developers to concentrate on application architecture and business logic.

5. Overview of the SQL Server Management Studio

Microsoft offers the SQL Server Management Studio (SSMS), a graphical user interface (GUI) application for managing and controlling SQL Server databases. It functions as a complete environment for creating, setting up, and keeping track of SQL Server instances and databases. The main attributes of SQL Server Management Studio are listed below:

Database management: SSMS enables database administrators and developers to carry out a variety of database management-related tasks. Tools are provided for building and maintaining databases, tables, views, stored procedures, functions, and other database objects. Database settings, security permissions, and performance can all be configured by administrators.

A powerful tool for creating and running SQL queries against SQL Server databases is the Query Editor in SSMS. It provides debugging tools, syntax highlighting, code snippets, and IntelliSense (code completion). The Query Editor allows developers to interface directly with the database by supporting complicated queries, joins, subqueries, and other SQL features.

SSMS users can create scripts for a variety of database objects and processes. Tables, stored procedures, and other objects can all be scripted for backup, migration, or deployment needs. PowerShell scripting is also supported by SSMS, making it possible to automate routine administrative activities.

Database administration functions include backup and restore operations, database migration, database mirroring, replication settings, and log file management. These features are all provided by SQL Server Management Studio. Using built-in reports and performance monitoring tools, administrators may carry out duties like making maintenance schedules, administering SQL Server Agent jobs, and keeping an eye on server performance.

Data Import and Export: SSMS provides tools for transferring data between SQL Server databases and other data sources by importing and exporting it. It offers a wizard-based interface for defining data transfer settings and mapping columns and supports several formats, including CSV, Excel, and XML.

Analysis and reporting tools are available in SSMS for use with SQL Server data. Both SQL Server Analysis Services (SSAS) and SQL Server Reporting Services (SSRS) projects can be created and run with its assistance. SQL Server Analysis Services and SQL Server Reporting Services are tools that developers and analysts can use to design and deploy multidimensional cubes, produce reports, and perform data analysis.

Security and Permissions: Through the use of SSMS, administrators can control security settings and permissions at several different levels, including the server, database, and object levels. To maintain data integrity and restrict access to private information, administrators can create and manage users, roles, and permissions.

A strong tool that makes it easier to manage and create SQL Server databases is SQL Server Management Studio. For managing and optimizing SQL Server instances, databases, and related components, it offers an intuitive user interface, a robust feature set, and a wide range of capabilities.

A well-liked online application platform for creating scalable, secure, and dynamic web applications is ASP.NET (MVC). With support for a variety of web technologies and tools, including C#, Visual Studio IDE, and object-relational mapping (ORM) frameworks, it offers a reliable architecture for creating sophisticated applications. Model, View, and Controller are the three main parts of the program that are divided into three by the MVC design pattern used in ASP.NET. This allows for a clear separation of concerns and simpler code maintenance. Routing, caching, authentication, and authorization are just a few of the capabilities and functions that ASP.NET (MVC) offers. These features and functionalities speed up development and enhance application performance.

In addition, SQL Server Management Studio is a thorough tool offered by Microsoft for managing and administering SQL Server databases, offering a variety of features for database management, query editing, scripting, automation, administration, data import and export, analysis and reporting, and security and permissions. Overall, these tools and technologies offer a strong and effective framework for creating and maintaining online applications.