Design of Web Application using .Net and MVC Framework

Syed Mustaqeem Uddin
Information Science and Engineering
Presidency University
Bangalore, India
syedmustaqeemuddin441@gmail.com

***Abstract*—An application model representing the elements of an MVC app is defined by ASP.NET Core MVC. For the purpose of changing how MVC elements behave, read and alter this model. To determine which classes are considered controllers, which methods on those classes are actions, and how parameters and routing behave, MVC by default adheres to a set of conventions. Create unique conventions and apply them globally or as attributes to adapt this behavior to the requirements of an app.**

1.2 Background information on .NET 6 Core MVC

Microsoft's ASP.NET web development framework enables programmers to create dynamic and responsive web applications. Since its initial release in 2002, it has undergone a number of iterations, including the introduction of ASP.NET Core in 2016.

Model-View-Controller (MVC) architecture is used by ASP.NET, which is based on the.NET framework. The Model, which represents the data and business logic, the View, which manages the user interface, and the Controller, which controls user input and orchestrates communication between the Model and View, are all separated into three interconnected parts by this design pattern.

A. Key Advantages:

1) Versatility: One of ASP.NET's key advantages is its adaptability. It is possible to use it to build a wide variety of online applications, from simple static websites to complex enterprise-level programmes. Additionally, it has built-in security features like authentication and authorization and supports a variety of programming languages including C#, Visual Basic, and F#.

2) Easy-to-use: The.NET 6 Core MVC has a lot of characteristics that make it an effective web development tool. It provides programmers with an easy-to-use model-view-controller (MVC) architecture, which makes it simple to separate concerns and build applications with a clear structure. Thanks to dependency injection, it is also possible to manage dependencies and develop testable code.

3) Cross-Platform Development: Another crucial aspect of.NET 6 Core MVC is its support for cross-platform programming. Developers may use.NET 6 Core MVC to construct applications for Windows, Linux, and macOS, as well as for mobile devices and the web. It may therefore be used to develop programmes that function across several platforms.

4) Considering everything,.NET 6 Core MVC is a reliable and flexible framework for building web apps. Because of its cross-platform features, modular architecture, and support for dependency injection, it is typically chosen by developers who need to construct scalable and high-performance applications..

Model View Controller (MVC)

The Model-View-Controller (MVC) architectural pattern separates an application into three main groups of components: Models, Views, and Controllers. This pattern helps to achieve separation of concerns. Using this pattern, user requests are routed to a Controller which is responsible for working with the Model to perform user actions and/or retrieve results of queries. The Controller chooses the View to display to the user, and provides it with any Model data it requires.

1) Model: An MVC application's model depicts the application's current state as well as any necessary business logic or processes. Business logic and any implementation logic for maintaining the application's state should be contained in the model.

2) View: The application's user interface is represented by this component. It consists of files for rendering HTML, CSS, and JavaScript so that users may see the data.

3) Controller: The interaction between the model and the view is controlled by this component. It takes user input, updates the model as necessary, and refreshes the view as a result.

.NET 6 Core MVC purpose and applications.

With.NET 6 Core MVC, developers will have access to a robust and adaptable framework for creating web applications on the Microsoft.NET platform. The framework has a number of features, including a modular architecture, support for dependency injection, and cross-platform development capabilities, that make it simple to create scalable and high-performance applications.

The creation of enterprise-level web applications is one of.NET 6 Core MVC's key applications. Support for caching, performance monitoring, and distributed tracing are among the many tools and features the framework offers for creating large-scale applications. This makes it the perfect option for companies that need to create intricate, scalable applications that can manage a lot of traffic.

ACKNOWLEDGMENTS

I extend my profound thanks to Sri. Y V Prasad Rao Sir, Programmer, Rail Wheel Factory, Bangalore. Special thanks to Sri. Buddha Appa Rao Sir, Rail Wheel Factory, Bangalore, for his valuable suggestions.

Finally, I thank my parents and friends for their moral support throughout my project. Their unwavering encouragement and understanding helped me to stay motivated and focused.

I acknowledge that any achievement does not depend solely on my individual efforts but also on the guidance, encouragement, and cooperation of intellectuals, elders, and friends.

REFERENCES

[1] ASP.NET 6 Core CRUD operations using SQL stored procedure and Dapper

[2] ASP.NET Core 6.0 - jQuery DataTable - update and delete data from table

[3] ASP.NET Core 6.0 - Using JQuery DataTables