Difference between ASP.NET MVC and WebForms:

Asp.Net Web Forms	Asp.Net MVC
Asp.Net Web Form follows a traditional event-driven development model.	Asp.Net MVC is lightweight and follows the MVC (Model, View, Controller) pattern-based development, model.
Asp.Net Web Form has server controls.	Asp.Net MVC has HTML helpers.
Asp.Net Web Form supports view state for state management on the client side.	Asp.Net MVC does not support view state.
Asp.Net Web Form has file-based URLs means the file name exist in the URLs must have its physical existence.	Asp.Net MVC has route-based URLs means URLs are divided into controllers and actions and it is based on the controller not on the physical file.
Asp.Net Web Form follows Web Forms Syntax	Asp.Net MVC follows customizable syntax (Razor as default)
In Asp.Net Web Form, Web Forms(ASPX) i.e. views are tightly coupled to Code behind(ASPX.CS) i.e. logic.	In Asp.Net MVC, Views and logic are kept separately.
Asp.Net Web Form has Master Pages for a consistent look and feel.	Asp.Net MVC has Layouts for a consistent look and feel.
Asp.Net Web Form has User Controls for code re-usability.	Asp.Net MVC has Partial Views for code re-usability.
Asp.Net Web Form has built-in data controls and is best for rapid development with powerful data access.	Asp.Net MVC is lightweight, provides full control over markup, and supports many features that allow fast & agile development. Hence it is best for developing an interactive web application with the latest web standards.
Asp.Net Web Form is not Open Source.	Asp.Net Web MVC is an Open Source.

MVC Vs ASP.NET

Let's see the key differences between MVC and ASP.Net

Key Factors	ASP.NET	ASP.NET MVC
PERFORMANCE	Offers better performance for simple and small applications	Offers better performance for complex and large applications
CROSS-PLATFORM SUPPORT	Limited cross-platform support	Cross-platform support runs on Windows, Linux, and macOS
CODE	Generates a lot of code automatically, with less control over the code	Offers more control over the code and allows for more customization and optimization

MAINTENANCE	This can lead to tight coupling and difficult maintenance in complex applications	Follows a modular and scalable architecture, easier to maintain in complex applications
SECURITY	Offers built-in security features, such as authentication and authorization	Offers built-in security features, but requires explicit implementation for some features
PORTABILITY	Limited portability	Cross-platform support can be deployed on multiple operating systems and cloud platforms