2.1 INTRODUCTION TO FRONT END

Microsoft Visual C# 2005, pronounced C sharp, is a programming language designed for building a wide range of applications that run on the .NET Framework. C# is simple, powerful, type-safe, and object-oriented With its many innovations, C# enables rapid application development while retaining the expressiveness and elegance of C-style languages. Visual Studio supports Visual C# with a full-featured Code Editor, project templates, designers, code wizards, a powerful and easy-to-use debugger, and other tools. The .NET Framework class library provides access to a wide range of operating system services and other useful, well-designed classes that speed up the development cycle significantly.

C# syntax is highly expressive, yet it is also simple and easy to learn. The curly-brace syntax of C# will be instantly recognizable to anyone familiar with C, C++ or Java. Developers who know any of these languages are typically able to begin to work productively in C# within a very short time. C# syntax simplifies many of the complexities of C++ and provides powerful features such as null able value types, enumerations, delegates, lambda expressions and direct memory access, which are not found in Java. C# supports generic methods and types, which provide increased type safety and performance, and iterations, which enable implementers of collection classes to define custom iteration behaviours that are simple to use by client code. As an object-oriented language, C# supports the concepts of encapsulation, inheritance, and polymorphism. All variables and methods, including the Main method, the application's entry point, are encapsulated within class definitions. A class may inherit directly from one parent class, but it may implement any number of interfaces. Method that override virtual methods in a parent class require the override keyword as a way to avoid accidental redefinition. In C#, a strut is like a lightweight class; it is a stack allocated type that can implement interfaces but does not support inheritance.

Why ASP.NET?

Since 1995, Microsoft has been constantly working to shift its focus from Windows-based platforms to the Internet. As a result, Microsoft introduced ASP (Active Server Pages) in November 1996. ASP offered the efficiency of ISAPI applications along with a new level of simplicity that made it easy to understand and use. However, ASP script was an interpreted script and consisted unstructured code and was difficult to debug and maintain. As the web consists of many different technologies, software integration for Web development was

complicated and required to understand many different technologies. Also, as applications grew bigger in size and became more complex, the number of lines of source code in ASP applications increased dramatically and was hard to maintain. Therefore, an architecture was needed that would allow development of Web applications in a structured and consistent way. The .NET Framework was introduced with a vision to create globally distributed software with Internet functionality and interoperability. The .NET Framework consists of many class libraries, includes multiple language support and a common execution platform. It's a very flexible foundation on which many different types of top class applications can be developed that do different things. Developing Internet applications with the .NET Framework is very easy. ASP.NET is built into this framework; we can create ASP.NET applications using any of the built-in languages.

Unlike ASP, ASP.NET uses the Common Language Runtime (CLR) provided by the .NET Framework. This CLR manages execution of the code we write. ASP.NET code is a compiled CLR code instead of interpreted code (ASP). CLR also allows objects written in different languages to interact with each other. The CLR makes development of Web applications simple.

Advantages Using ASP.NET

- ASP.NET drastically reduces the amount of code required to build large applications
- ASP.NET makes development simpler and easier to maintain with an event-driven, server-side programming model
- ASP.NET pages are easy to write and maintain because the source code and HTML are together
- The source code is executed on the server. The pages have lots of power and flexibility by this approach

- The source code is compiled the first time the page is requested. Execution is fast as the Web Server compiles the page the first time it is requested. The server saves the compiled version of the page for use next time the page is requested
- The HTML produced by the ASP.NET page is sent back to the browser. The application source code you write is not sent and is not easily stolen

- ASP.NET makes for easy deployment. There is no need to register components because the configuration information is built-in
- The Web server continuously monitors the pages, components and applications running on it. If it notices memory leaks, infinite loops, other illegal software or activities, it seamlessly kills those activities and restarts itself
- ASP.NET validates information (validation controls) entered by the user without writing a single line of code
- ASP.NET easily works with ADO .NET using data-binding and page formatting features
- ASP.NET applications run faster and counter large volumes of users without
- Performance problems.

C#.NET

C# (pronounced C Sharp) is a multi-paradigm programming language that encompasses functional, imperative, generic, object oriented and component oriented programming disciplines. It was developed by Microsoft as part of the .Net initiative and later approved as a standard by ECMA (ECMA-334) and ISO (ISO/IEC 23270). C# is intended to be a simple, modern, general-purpose, object-oriented programming language. It has an object-oriented syntax based on C++ and is heavily influenced by Java. It was initially named Cool, which stood for "C-like Object Oriented Language." However, in July 2000, when Microsoft made the project public, the name of the programming language was given as C#. The most recent version of the language is 3.0 which was released in conjunction with the .Net Framework 3.5 in 2007.

- C# is intended to be a simple, modern, general-purpose, object-oriented programming language.
- Because software robustness, durability and programmer productivity are important, the language should include strong type checking, array bounds checking, and detection of attempts to use uninitialized variables, source code and automatic garbage collection.
- The language is intended for use in developing software components that can take advantage of distributed environments.

- Programmer portability is very important, especially for those programmers already familiar with C and C++.
- C# is intended to be suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions. Although C# applications are intended to be economical with regard to memory and processing power requirements, the language is not intended to compete directly on performance and size with C.

2.2 FEATURES OF FRONT END

- Strongly typed data control.

  In ASP.NET you now have data control that can be strongly typed. You will get intelligence you just need to assign the item type property to a model that is going to be associated with the data controls used in your .asp pages.

- Enhanced support for asynchronous programming.

  ASP.NET provides excellent support in asynchronous programming you can now read and write HTTP requests and responses without the need of OS threads. Also, you have support for two new keywords await and sync.

- Support for web sockets.

  HTML 5 web sockets allow you to perform duplex communication between the client browser and the web server. ASP.NET provides support for web socket protocols.

- Support for improving paging in ASP.NET Grid View control.

  Paging support in ASP.NET Grid View control has been improved a lot. Allow Custom paging property provides great support for paging and sorting through large amounts of data efficiently.

ANDROID APPLICATION

Android is a software stack for mobile devices such as mobile telephones and tablet computers developed by Google Inc. and the Open Handset Alliance. Google purchased the initial developer of the software, Android Inc., in 2005. The unveiling of the Android