

Design and implementation of web Application using .Net Framework

Asst. Lecturer Rabab J.Mohsin^{*}

Abstract

Due to the recent growth in the importance of Web applications, and their adoption as a main alternative for computer applications, and since they provide a shortcut in resources, centralized processing and database, in addition to ease of maintenance and update of a single server rather than a number of computers could reach into thousands.

Therefore in this paper was focused on building web applications using .Net framework and the programming languages used are (C# & Java Script) being the two languages hard to penetrate and easily linked to the database, especially for applications that need giant databases and error free data retrieval. Note that the platform providing the .Net framework is a solid platform that has a wide range of security and compatibility capabilities.

In this paper, website of Ministry Of Higher Education And Scientific Research has been designed and built, with database design and implement ,using Microsoft SQL Server to build the structure of the system. In addition the report viewer is used to view the contents of the database that can be converted to Pdf and Excel.

Key words: .Net Framework, Web applications , SQL

^{*} College of Eng. Al-Mustansiriya University

1.Introduction [1]

Microsoft views .NET as a vision, a platform for the digital future. A more concrete and equally accurate way to think about this new initiative is to understand that .NET is a brand, one that Microsoft has applied to several different technologies. Some of these technologies are entirely new, providing new services and new possibilities. Others allow an updated approach to creating the kinds of Windows applications we know today. Still other parts of the .NET family are just new releases of existing technologies dressed up with the .NET brand. The most important new technology in .NET is Web services. As the name suggests, a Web service provides some function that can be invoked programmatically via the Web. Most technologies that carry the .NET brand have direct support for Web services in some way, yet .NET is more than just Web services. The technologies that Microsoft has placed under the .NET umbrella today include the following:

The .NET Framework: The .NET Framework is Microsoft's object-oriented programming platform that simplifies application development in highly distributed environments, including the Internet .It includes the Common Language Runtime (CLR) and the .NET Framework class library. The CLR is a standard foundation for building a range of new applications, while the .NET Framework class library provides standard implementations of many new services for CLR-based applications. Among the technologies in the library are ASP.NET, which is the next generation of Active Server Pages; ADO.NET, the next generation of ActiveX Data Objects; support for building and using Web services; and much more. Microsoft is also releasing a trimmed-down incarnation of the .NET Framework called the .NET Compact Framework. This version is intended for use in smaller devices, such as personal digital assistants (PDAs).

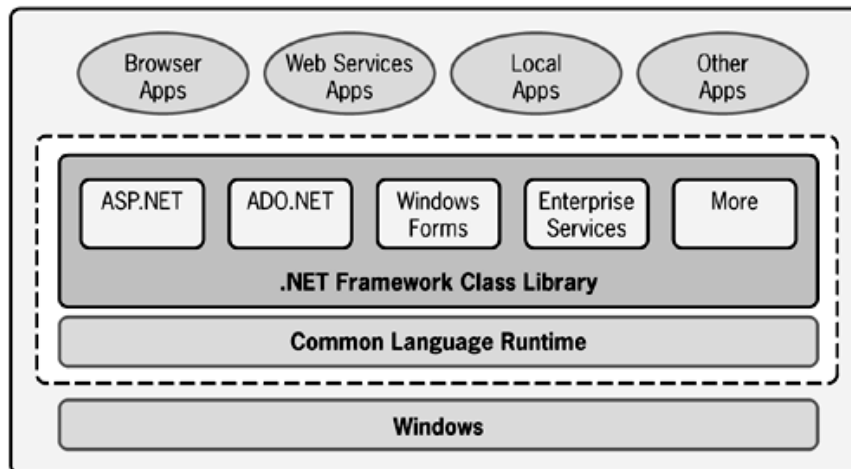
- **Visual Studio.NET:** Supports several programming languages that can be used with the .NET Framework. These languages include Visual Basic.NET, which is the next generation of Visual Basic; an enhanced version of

C++; and a wholly new language called C# designed explicitly for the .NET Framework.

The .NET Framework class library provides standard code for common functions. Among the most important technologies provided in the .NET Framework class library are the following:

- **ASP.NET:** The next generation of Active Server Pages for building Web-accessible applications. A key feature of this new technology is strong support for building applications that use and expose Web services.
- **ADO.NET:** The next generation of ActiveX Data Objects for accessing data stored in relational database management systems (DBMSs) and in other formats.
- **Windows Forms:** A standard set of classes for building Windows GUIs in any .NET Framework programming language.
- **Enterprise Services:** Standard classes for accessing COM+ services such as transactions and object pooling.

The .NET Framework class library contains much more than this short list indicates. Among the other services it provides are support for creating and working with XML documents, services for remote access, and mechanisms for interoperating with COM-based applications. (Figure 1) illustrates the main components of the .NET Framework and how they relate to one another.

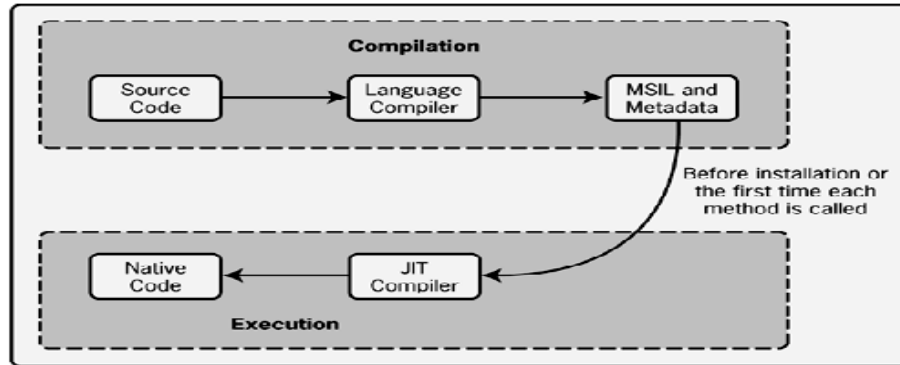


(Figure 1) The .NET Framework consists of the Common Language Runtime and the .NET Framework class library.

As the figure shows, the .NET Framework can be used to create many different types of applications. The choices include browser-accessible applications; Web services applications, applications that display just a local Windows GUI, and others. Because all of the services in the .NET Framework class library are built on the CLR, applications can combine them as needed. A browser application built using ASP.NET, for example, might use ADO.NET to access stored data and Enterprise Services to perform distributed transactions.

2. The CLR supports many different programming languages[2]

No matter what language it's written in, all managed code is compiled into Microsoft Intermediate Language (MSIL) rather than a machine-specific binary. MSIL is a set of CPU-independent instructions for performing typical operations such as loading and storing information and calling methods. Each DLL and EXE in an assembly contains MSIL rather than processor-specific code. Installing a .NET Framework application on your system really means copying to your disk files that contain MSIL rather than a machine-specific binary. When the application is executed, MSIL is transformed into native code before it's executed. Managed code is always compiled first into MSIL,(Figure 2) illustrates the process of compiling and executing managed code. Source code written in VB.NET, C#, Managed C++, or another language that targets the CLR is first transformed into MSIL by the appropriate language compiler. As the figure shows, the compiler also produces metadata that's stored in the same file as the MSIL. Before execution, this MSIL is compiled into native code for the processor on which the code will run. By default, each method in a running application is compiled the first time that method is called. Because the method is compiled just in time to execute it, this approach is called just-in-time (JIT) compilation.



(Figure 2) All managed code is compiled first to MSIL, and then translated into native code before execution.

3.CLR-Based Languages[3]

One of the goals of the .NET Framework is to let developers work in the language of their choice. To allow this, the CLR can support many different programming languages. Those languages often have a good deal in common, since the CLR defines the core semantics for each one, but a developer can still choose the language that feels most natural to him. Visual Studio.NET, Microsoft's primary tool for building .NET Framework applications, supports four CLR-based languages: Visual Basic.NET, C#, C++, and JScript.NET.

4. Features provided by C#[4]

The list of features provided by C# is nearly identical to the list just given for Visual Basic.NET. Among those features are the following:

- Support for single implementation inheritance
- Method overriding
- Support for exception handling
- Full multithreading
- The ability to define explicitly interfaces directly in C#
- Support for properties and events
- Support for attributes as in VB.NET

- **Garbage collection**

C# also provides a few features that aren't available in VB.NET, including the following:

- **Operator overloading, allowing a class to define its own unique meaning for standard operators such as the plus sign**
- **The ability to write code that directly accesses specific memory addresses, sometimes referred to as unsafe code**

4.The Entity Data Model[5]

The Entity Data Model (EDM) is intended for developing rich data centric applications. It extends the classic relational model with concepts from the E-R domain. The central concepts in the EDM are entities and associations. Entities represent top-level items with identity, while Associations are used to relate (or, describe relationships between) two or more entities. An important aspect of EDM is that it is value-based like the relational model (and SQL), rather than object/reference-based like C# (CLR). Several object programming models can be easily layered on top of the EDM. Similarly, the EDM can map to one or more DBMS implementations for persistence. The EDM and Entity SQL represent a richer data model and data manipulation language for a data platform and are intended to enable applications such as CRM and ERP, data-intensive services such as Reporting, Business Intelligence, Replication and Synchronization, and data-intensive applications to model and manipulate data at a level of structure and semantics that is closer to their needs.

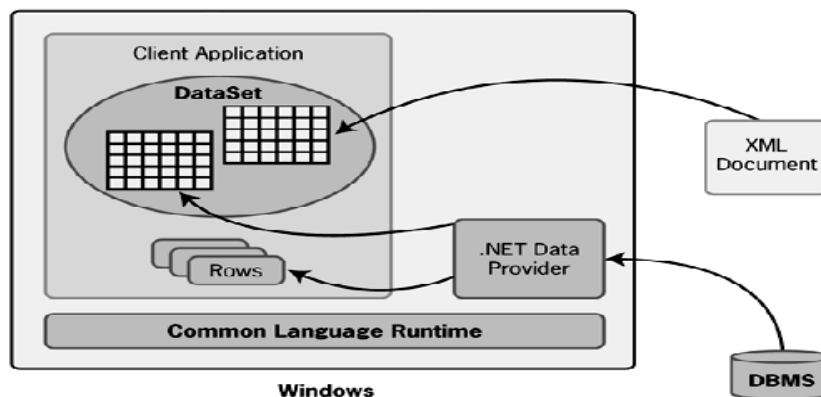
5.ADO.NET[1]

Because it contains the types that implement ADO.NET, System Data is one of the most important namespaces in the .NET Framework class library. Like its predecessor ADO, ADO.NET lets applications work with data stored in a DBMS. Apart from this fundamental similarity, however, ADO and ADO.NET don't have much in common. Instead, ADO.NET focuses on allowing managed code to work with

data in a way that makes sense for a world increasingly dominated by Web-based applications.

ADO.NET lets applications access stored data

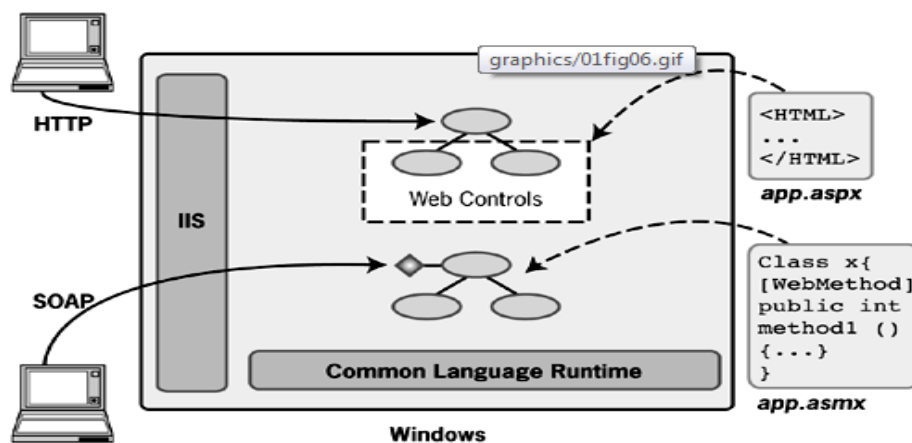
(Figure 3) illustrates how an application can use ADO.NET to access stored information. As the figure shows, access to a DBMS relies on a .NET data provider, written as managed code. .NET data providers that allow access to SQL Server, Oracle, and other DBMSs exist today, and they allow a client application to issue commands against the DBMS and examine any results those commands return. The result of a SQL query, for example, can be examined in two ways. Applications that need only read the result a row at a time can do this in a straightforward way, marching directly through what a query returns one record at a time. Applications that need to do more complex things with the result of a query, such as send it to a browser or store it on disk, can instead have the query's result packaged inside a Dataset object. Applications that need only read the result a row at a time can do this in a straightforward way, marching directly through what a query returns one record at a time. Applications that need to do more complex things with the result of a query, such as send it to a browser or store it on disk, can instead have the query's result packaged inside a Dataset object.



(Figure 3). ADO.NET allows .NET Framework applications to access data stored in DBMSs and XML documents.

6. ASP.NET[1]

Web namespace, ASP.NET is a major piece of the .NET Framework. The successor to the very popular Active Server Pages technology, ASP.NET brings a number of new features to ASP developers. Like traditional ASP applications, ASP.NET applications are built from one or more pages. Each page contains HTML and/or executable code. As (Figure 4) shows, however, ASP.NET allows the creation of two types of applications: traditional browser applications accessible via HTTP and HTML, and Web services applications accessible via SOAP.



(Figure 4). ASP.NET allows the creation of both browser applications and Web services applications.

ASP.NET is the successor to Active Server Pages. These two different kinds of applications use different kinds of pages. Applications that present traditional browser GUIs can be built by combining HTML and executable code in files with the extension .aspx. Applications that expose methods as Web services can be built from files with the extension .asmx, each of which contains only code. As (Figure 4) shows, all that's required to expose a method as a Web service from an .asmx page is to insert WebMethod before the method definition. Unlike Active Server Pages (ASP), ASP.NET is a compiled, .NET-

based platform. ASP.NET applications can be built using any .NET-compatible programming language. Also, because ASP.NET is built on top of the .NET Framework, your ASP.NET applications have access to the entire range of functionality provided by the framework. Among the major advantages of the ASP.NET environment are the following:[5]

□ASP.NET applications are fully compiled .NET applications, providing superior performance characteristics.

□ASP.NET supports WYSIWYG HTML editors and programming environments such as Visual Studio .NET. This enables you to be very productive when developing ASP.NET applications and enables you to leverage the many features of these tools.

□ASP.NET applications support extensive configuration capabilities based on XML configuration files.

□ASP.NET provides flexible, advanced, and easy-to-use application and session state management features that can be extended or replaced with custom schemes.

□ASP.NET implements multiple authentication and authorization schemes that can be extended or replaced with custom schemes.

ASP.NET provides two programming models that can be used to create Web applications: Web Forms and Web Services. Web Forms enable you to build formsbased Web applications using a technology called server controls, which enable you to create user interfaces from common UI elements such as text boxes, list boxes, and so on.

6. Website vs. Web Application[7]

The advent of the Internet led to the invention of new terms that are exclusively used to refer to things that you can do or get from the Internet. Website is one of the very first and it is used to refer to a location that hosts several pages that are often on the same topic. The site is accessed with the use of a URL (Uniform Resource Locator). On the other hand, a web application is a term used identifies a program or application that is run and used on separate computers.

A web application can exist in the Internet or across a local network, Intranet, VPN, among other things. When a web application is available in the Internet, it is often hosted as a separate page on a website. The site can also contain other materials that are not used by the web application but are often related to what the web application does.

Comparatively, a web application is more resource intensive compared to a website that does not contain a web application. Depending on the type and goal of the web application, it needs to process the data it gets as well as access databases. Although some of the more complicated sites can be as resource intensive, most are not. This is because most sites simply show information that are static and are not updated very often.

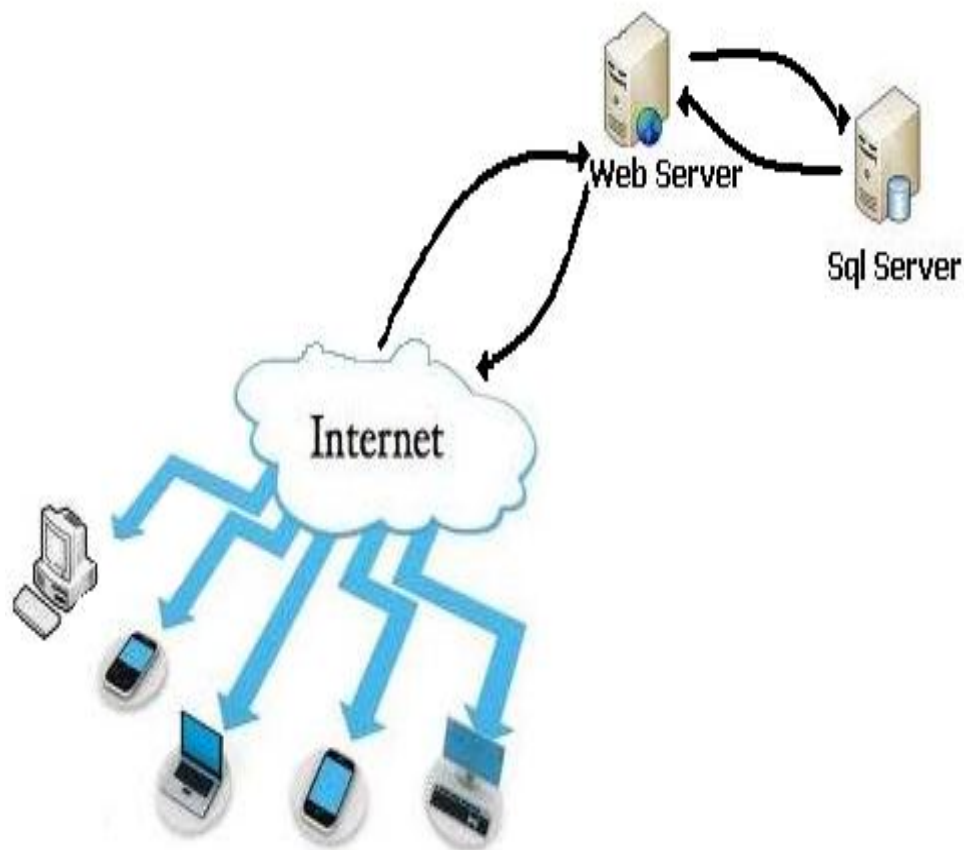
The same is also true when it comes to the difficulty of creating a website or a web application. Static websites can be coded as long as you know HTML (Hypertext Markup Language). With web applications, it is not enough to know HTML, the part that makes it an application is coded with a more difficult language that is known to programming languages. The list of languages includes Java, JavaScript, C#, DHTML, Silverlight, PHP, and AJAX. It is also necessary to know two or more of these languages in order to implement server side scripts that process the data and client side scripts that format the information on screen.

A web application is an application that is accessed over a network such as the Internet or an intranet. The term may also mean a computer software application that is coded in a browser-supported language (such as JavaScript, combined with a browser-rendered markup language like HTML) and reliant on a common web browser to render the application executable.[7]

Web applications are popular due to the ubiquity of web browsers, and the convenience of using a web browser as a client, sometimes called a thin client. The ability to update and maintain web applications without distributing and installing software on potentially thousands of client computers is a key reason for their

popularity, as is the inherent support for cross-platform compatibility. Common web applications include webmail, online retail sales, online auctions, wikis and many other functions.[8]

For this reason In this paper I will design and implement a web application (web of ministry of higher education of Iraq) connected to huge data base using Microsoft SQL server , the web application will be built with Microsoft visual studio using (C#, JavaScript ,HTML) languages .The architecture of the web application will be as shown in the figure below



(Figure 5)The architecture of system network

Diagram (1) illustrates the process of checking authentication of the IP address and domain permitted to the server that contains the code of the application and also to the data base server. It also illustrates how to receive a request from IIS and ASP.NET.

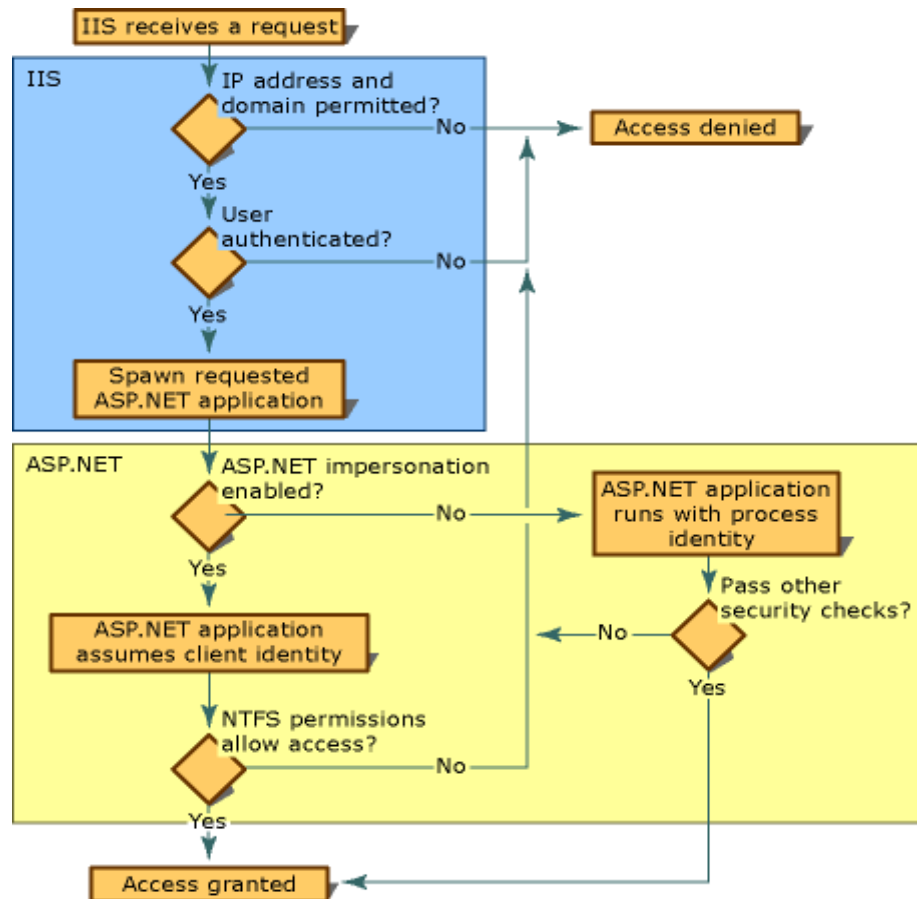


Diagram (1)[9]

While diagram (2) shows the mechanism of requesting the web application from the web server during the ASP and IIS.

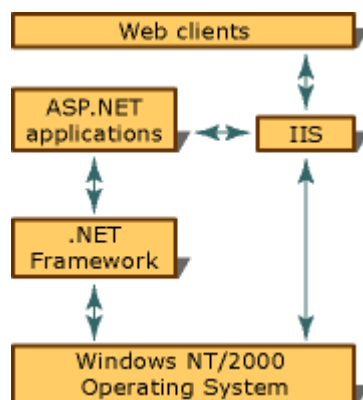
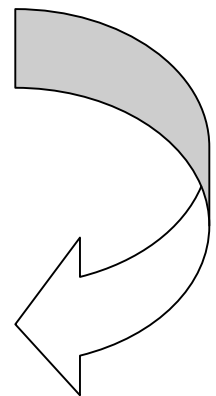
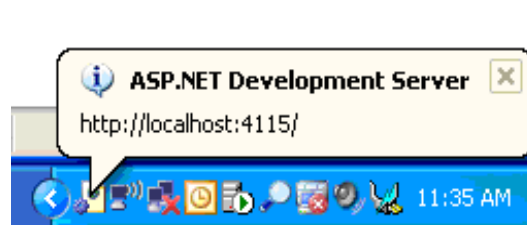


Diagram (2)[9]

ASP.NET Web application adds a tab named Web to the project properties list. This tab can be used to configure how a Web application is run and debugged. By default, ASP.NET Web application projects are configured to launch and run using the built-in ASP.NET Development Server on a random (HTTP) port on the machine. The following screenshots show the ASP.NET Development Server: the Task bar icon and a page in the browser where you can see the server's URL in the Address box as shown in figure (6)



Figure(6)





- 81 -

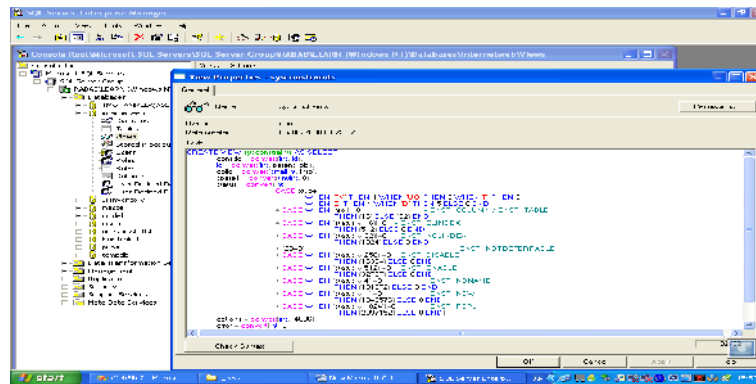


Figure (7)

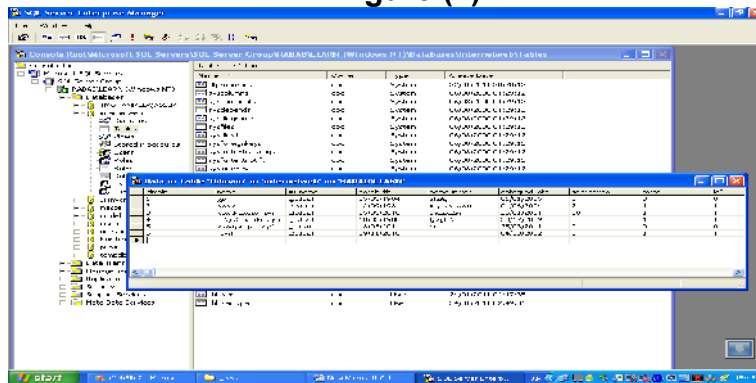


Figure (8) illustrate the main table that built in the SQL Server.

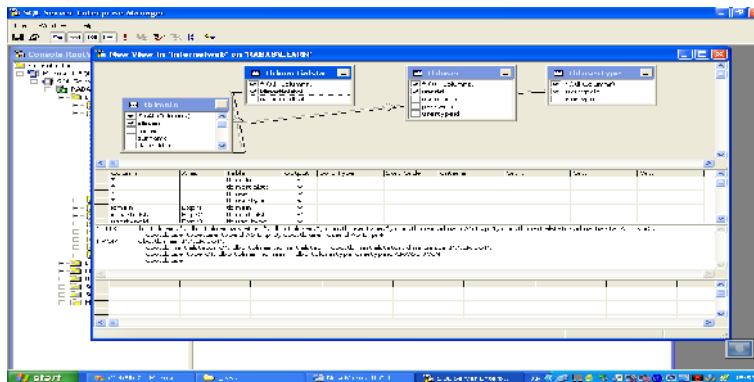


Figure (9)

Figure (9) illustrate the relationship between tables and how it can be connected to represent the viewer table to view in the report that will fetch from the SQL to view in the interface that built in Visual Studio 2005.

But from the side of visual studio can be connected to the SQL data base as shown in the figure (10) below

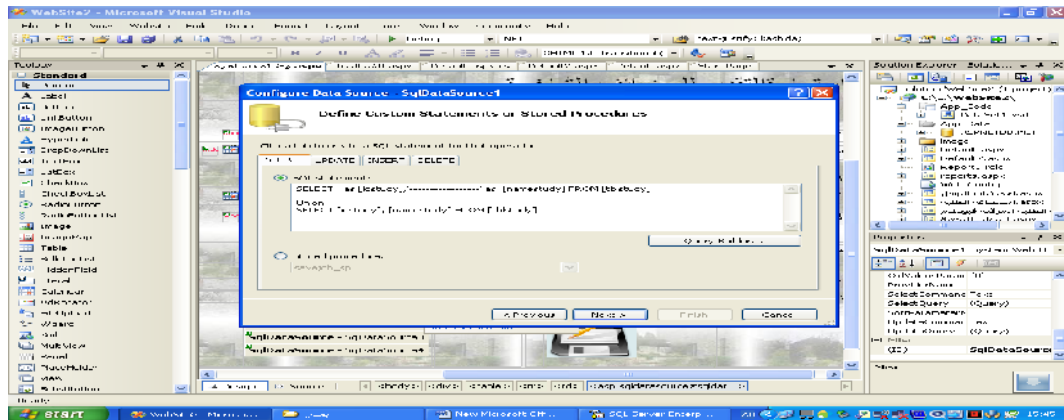


Figure (10)

The results can be fetched from the data base and display them in the web application by searching on the number of forms that are obtained during the save of form as shown in figure (11).



Fig.11

After that the form number can be entered in the field of search criteria and after that grid view will appear containing the information as shown in figure (12).



Figure (12)

For more details the techniques of report viewer will display the information with the ability of converting the format into (PDF or EXCEL) and download the data if needed as shown in the figure (13).



From her it can be exchange the format of report

Figure (13)

7. Conclusion:

The aim of my work is programming safety and comfort of the use of web application interfaces. The SQL Server code has stronger security than other RDBMS, SQL Server provides Security Development Lifecycle (SDL) and it is committed to serve the huge data storage that is needed by businesses worldwide with scalable and reliable technology like .NET Framework ,ASP and IIS specially when using a rich and powerful language() that is used in the programming of .NET platform and this combination between the SQL and Microsoft visual studio () will meet the high performance requirements and security of today's most demanding enterprises of building web application projects through high speed of data fetch and reliable project and object-oriented paradigm ,it is designed to give the blend of simplicity ,expressiveness and performance pushing beyond the limitation of other programming languages like Java, C, C++.

8. Reference:

1. David Chappell, “Understanding .Net”, 2005.
2. By Sathiaselvan & Sasikaladevi, “Programming With C# .Net”,2006.
3. Jesse Liberty, Brian MacDonald, Master the fundamentals of C# 3.0 , O'Reilly Media.
4.
http://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Visual_Basic_.NET
5. Atul Adya, José A. Blakeley, Sergey Melnik, S. Muralidhar, and the ADO.NET Team,” Anatomy of the ADO.NET Entity Framework”,2007.
6. Mridula Parihar and et al, “ASP.NET Bible”,2005.
7. <http://www.differencebetween.net/technology/internet/difference-between-website-and-web-application>
8. http://en.wikipedia.org/wiki/Web_application
9. <http://msdn.microsoft.com/en-us/default.aspx>

Abbreviations

1	SQL	Structured Query Language
2	DBMSs	database management systems
3	ASP	Active Server Pages
4	GUI	Graphic User Interface
5	COM	Component Object Model
6	XML	Extensible Markup Language
7	CLR	Common Language Runtime
8	ADO	ActiveX Data Object
9	PDAs	personal digital assistants
10	MSIL	Microsoft Intermediate Language
11	DLL	Data Link Language
12	JIT	just-in-time
13	VB	Visual Basic
14	HTTL	Hypertext Markup Language
15	IIS	Internet Information Server
16	IP	Internet Protocol

تصميم وتنفيذ تطبيقات الشبكة العنكبوتية باستخدام نطاق العمل .Net

م.م.رباب جاسم محسن*

المستخلص

نظرا للاهمية المتزايدة لتطبيقات الويب في الآونة الاخيرة و اعتمادها كبديل رئيسي للتطبيقات الحاسوبية لما توفره من اختصار في الموارد و مركزية في المعالجة و قاعدة البيانات , اضافة الى سهولة الصيانة و التحديث لخادم واحد بدلا من عدد من الحاسبات قد يصل الى الآلاف.

لذلك في هذا البحث تم التركيز على كيفية تصميم وبناء تطبيقات الشبكة العنكبوتية باستخدام نطاق العمل(.Net) واللغة البرمجية المستعملة لخلق التطبيقات هي (javascript , C#)كونهما لغتان صعبة الاختراق بالاضافة الى سهولة ربطها بانظمة قواعد البيانات ولا سيما التطبيقات التي تحتاج الى قاعدة بيانات عملاقة واستدعاء البيانات بدون مشاكل علما ان المنصة (Platform) التي تهين لنطاق عمل .Net. هي منصة رصينة بها عدة امكانيات من ناحية الامنية والتوافقية .

في هذا البحث تم تصميم وبناء موقع التعليم العالي والبحث العلمي مع قاعدة بيانات باستخدام (Microsoft SQL Server) لبناء هيكلية النظام كما تم استخدام عارض البيانات (Report viewer) لعرض محتويات قواعد البيانات التي يمكن تحويلها لصيغتي (Pdf,Excel)