**CALIFORNIA STATE UNIVERSITY-LOS ANGELES**

**CALIFORNIA 90032-8530**



**A**

**PROJECT Report**

**On**

**WEB SERVICE PERFORMANCE USING**

**.NET PROGRAMMING**

**Submitted By**

|  |
| --- |
| CHITTAKSH KHADSE RAKESH SAVANI |
| [305058538] [304454961] |
|  |
| VIDUSHI DIKSHIT PARISHA DALAL |
| [305064011] [305058915] |
|  |
| PRIYANKA MUPPURI MRUNALINI KUMBHALKAR |
| [304823420] [304822900] |
|  |
| KRISH BHANUSHALI SABHYA GUPTA |
| [305053962] [305082861] |
|  |
| MANISH PUROHIT BHAGYESHREE GAIKWAD |
| [304449059] [304360204] |
|  |

**Under the Guidance of**

**PROF .Dr. JIANG GUO**

**Department of Computer Science**

**College Of Engineering**

**Computer Science & Technology**

**January 2016 – March**

**ABSTRACT**

Web Service Performance is very useful in projects where implementation of the Web Service has not started or is not finished for use yet or where you for some other reasons can't access the Web Services live.

A [Web Service](http://en.wikipedia.org/wiki/Web_service) is a way to communicate between two devices over the Internet.  A user (client) of a Web Service invokes remote procedure calls on a Web Service that provides some data or other services.  The user of a Web Service is typically another application.  The application can be written in any language and on any platform.

A Web Service provider will provide a [WSDL](http://www.w3.org/TR/wsdl) (Web Services Description Language) file that fully describes the Web Service.  The WSDL file is an XML document that has all of the information needed to access and use a Web Service.

The WSDL file describes all of the calls that can be made on the Web Service, the parameters and return values for each call, and the address of the service.  Most programming languages provide a way to translate a WSDL file into code that can be used to make calls to the Web Service.

The working of the web service can be explained by the pre-defined functions from the web service of the weather forecasting. Some of the functions include the selection of URL and method with some of the given parameters or no parameters.

The web services being selected can be compared in the form of a graph. It selects the number of entries to be compared. The graph selects the type of web service to be compared and the number of repetition calls of the web service. The output can be seen in user friendly and interactive fashion, which can be easily interpreted by a user

This project is being achieved in ASP.Net and IIS. The database used for storing the data is SQL Server. The efficiency of the web service is being measured in Milliseconds and is compared in a graph by different web services for a given number of repetition c

Chapter 1

Introduction

1. **INTRODUCTION**

**1.1 WEB SERVICES**

The term *Web services* describes a standardized way of integrating Web-based [applications](http://www.webopedia.com/TERM/A/application.html) using the [XML](http://www.webopedia.com/TERM/X/XML.html), [SOAP](http://www.webopedia.com/TERM/S/SOAP.html), [WSDL](http://www.webopedia.com/TERM/W/WSDL.html) and [UDDI](http://www.webopedia.com/TERM/U/UDDI.html) [open](http://www.webopedia.com/TERM/O/open.html) [standards](http://www.webopedia.com/TERM/S/standard.html) over an Internet [protocol](http://www.webopedia.com/TERM/P/protocol.html) backbone. Web services are self-contained, modular, distributed, dynamic applications that can be described, published, located, or invoked over the network to create products, processes, and supply chains. These applications can be local, distributed, or web-based. Web services are built on top of open standards such as TCP/IP, HTTP, Java, HTML, and XML.

Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. Unlike traditional [client/server](http://www.webopedia.com/TERM/C/client_server_architecture.html) models, such as a Web [server](http://www.webopedia.com/TERM/S/server.html)/Web page system, Web services do not provide the user with a [GUI](http://www.webopedia.com/TERM/G/GUI.html). Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

The basic web services platform is XML + HTTP. All the standard web services work using the following components

* SOAP (Simple Object Access Protocol)
* UDDI (Universal Description, Discovery and Integration)
* WSDL (Web Services Description Language)

# 1.2 ASP.NET

**ASP.NET** is an [open-source](https://en.wikipedia.org/wiki/Open_source)[server-side](https://en.wikipedia.org/wiki/Server-side_scripting) [web application framework](https://en.wikipedia.org/wiki/Web_application_framework) designed for [web development](https://en.wikipedia.org/wiki/Web_development) to produce [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page). It was developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) to allow [programmers](https://en.wikipedia.org/wiki/Programmer) to build dynamic [web sites](https://en.wikipedia.org/wiki/Web_site), [web applications](https://en.wikipedia.org/wiki/Web_application) and [web services](https://en.wikipedia.org/wiki/Web_service).

ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development in ASP.NET. There are two basic methodologies for Web Forms, a web application format and a web site format. Web applications need to be compiled before deployment, while web sites structures allows the user to copy the files directly to the server without prior compilation.

**1.2.1 ASP.NET MVC**

ASP.NET MVC targets developers who are interested in patterns and principles like [test-driven development](http://en.wikipedia.org/wiki/Test-driven_development), [separation of concerns](http://en.wikipedia.org/wiki/Separation_of_concerns), [inversion of control](http://en.wikipedia.org/wiki/Inversion_of_control) (IoC), and [dependency injection](http://en.wikipedia.org/wiki/Dependency_injection) (DI). This framework encourages separating the business logic layer of a web application from its presentation layer.

By dividing the application into the [model (M), views (V), and controllers (C)](http://en.wikipedia.org/wiki/Model-view-controller), ASP.NET MVC can make it easier to manage complexity in larger applications. With ASP.NET MVC, you can have multiple teams working on a web site because the code for the business logic is separate from the code and mark-up for the presentation layer — developers can work on the business logic while designers work on the mark-up and JavaScript that is sent to the browser.

With ASP.NET MVC, you work more directly with HTML and HTTP than in Web Forms. Web Forms tends to hide some of that by mimicking the way you would program a Win Forms or WPF application. For example, Web Forms can automatically preserve state between HTTP requests, but you have to code that explicitly in MVC. The MVC model enables you to take complete control over exactly what your application is doing and how it behaves in the web environment.

**1.3 INTERNET INFORMATION SERVICES**

**Internet Information Services** (**IIS**, formerly **Internet Information Server**) is an extensible [web server](https://en.wikipedia.org/wiki/Web_server) created by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) for use with [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) family.[[2]](https://en.wikipedia.org/wiki/Internet_Information_Services#cite_note-2) IIS supports [HTTP](https://en.wikipedia.org/wiki/HTTP), [HTTPS](https://en.wikipedia.org/wiki/HTTPS), [FTP](https://en.wikipedia.org/wiki/File_Transfer_Protocol), [FTPS](https://en.wikipedia.org/wiki/FTPS), [SMTP](https://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol) and [NNTP](https://en.wikipedia.org/wiki/Network_News_Transfer_Protocol).

The Web Server (IIS) role includes Internet Information Services (IIS) 7, which is a unified Web platform that integrates IIS, ASP.NET, Windows Communication Foundation, and Windows SharePoint Services. IIS 7 lets you share information with users on the Internet, an intranet, or an extranet. Windows Server® 2008 delivers IIS 7.0, which is also included with some editions of Windows Vista

**1.3.1 IIS AND VISUAL STUDIO**

A local Internet Information Services (IIS) Web site project is an IIS Web application on your computer. Visual Studio communicates with the Web site project by using the HTTP protocol.

Visual Studio provides two conceptual models for working with Web applications: the Web site project model, and the Web application project model.

While both options allow you to create Web applications and publish them to an IIS server, they do have significant differences in how the corresponding ASP.NET application is built and deployed. Some of the differences between the two models are:

* The Web site project option allows you to connect directly to a local IIS Web site, or to a remote IIS Web site by using a network share, FTP, or Front Page Server Extensions. With the Web site project, you work directly with the content of your IIS Web site and there is no project file.
* The Web application project option requires the source application files to be located on the local file system, or, on a network share. However, you can subsequently publish the Web application to a remote IIS Web site by using a network share, FTP, or Front Page Server Extensions.

**1.3.2 IIS AND SQL SERVER**

**Microsoft SQL Server** is a [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) developed by [Microsoft](https://en.wikipedia.org/wiki/Microsoft). As a [database server](https://en.wikipedia.org/wiki/Database_server), it is a software with the primary function of storing and retrieving data as requested by other applications which may run either on the same computer or on another computer across a network (including the Internet).

If your application runs on a Windows-based intranet, you might be able to use Windows integrated authentication for database access. Integrated security uses the current Windows identity established on the operating system thread to access the SQL Server database. You can then map the Windows identity to a SQL Server database and permissions.

If SQL Server is on a different computer than the Web server, the Windows identity must be able to flow across the network to the remote instance of SQL Server. (Windows networks that have been configured appropriately with Kerberos authentication are able to do this.) However, depending on the settings in the [identity](https://msdn.microsoft.com/en-us/library/72wdk8cc.aspx) configuration element, the Windows identity established on the operating system thread for ASP.NET applications may not be able to flow properly to the remote SQL Server.

Chapter 2

System Analysis

1. **SYSTEM ANALYSIS**

System analysis phase deals with user requirements, which is to fulfill the user’s requirement that is known during the problem definition phase. To carry out a system analysis, analysts must develop an understanding of the problem domain. The choice of appropriate software for the smooth implementation of the project is accomplished by this system analysis.

**2.1 HARDWARE REQUIREMENTS**

The minimum/recommended configuration required for this project for the system will be:

* Processor- x64- architecture or compatible dual core 1.5 GHz processor.
* Memory- 4GB RAM
* Hard Disk: 10GB of hard disk available space.

**2.2 SOFTWARE REQUIREMENTS**

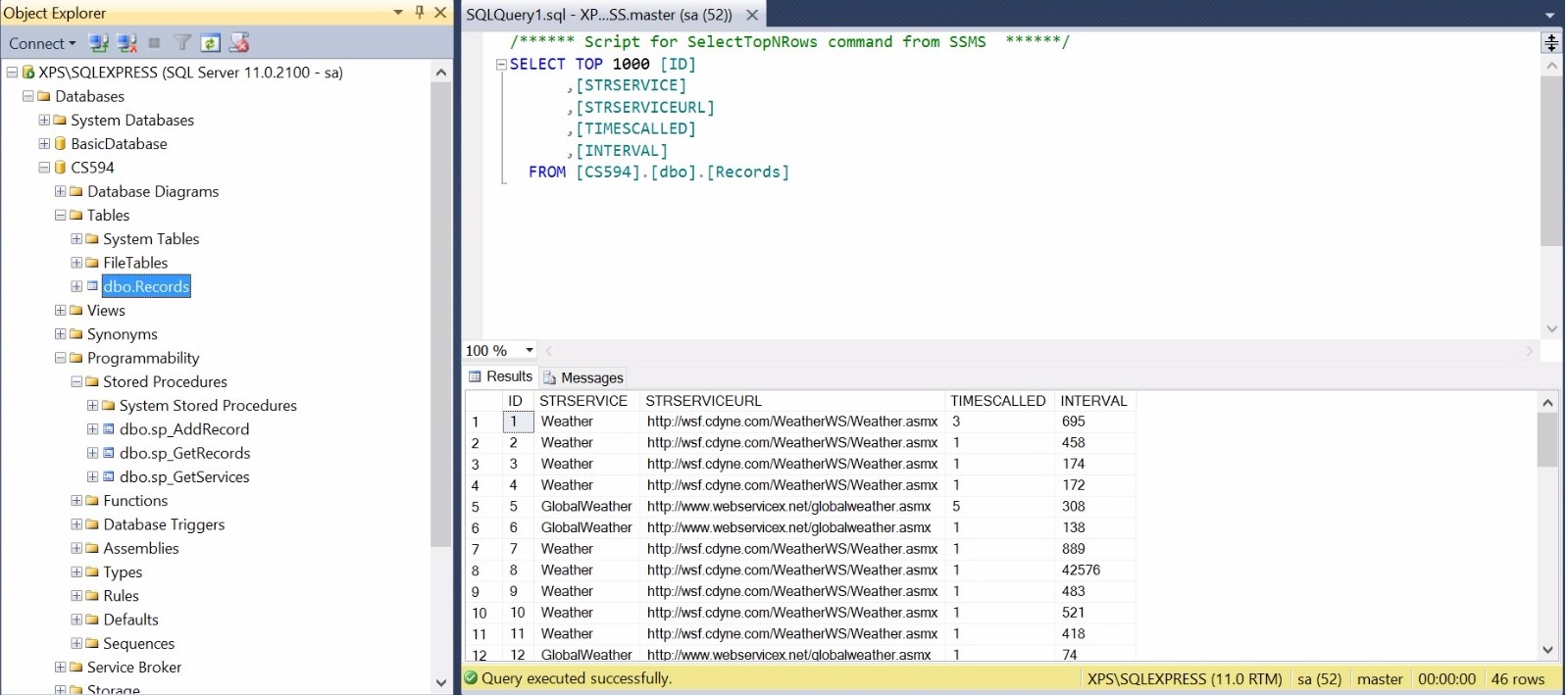
* Microsoft Visual Studio 2015
* SQL Server 2012
* Information Internet Services(IIS)

Chapter 3

Database Design

**3. DESIGN AND SPECIFICATION**

* 1. **TABLE**

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | STRSERVICE | STRSERVICE | TIMESCALLED | INTERVAL |
| 1 | Weather | http://wsf.cdyne.com/WeatherWS/Weather.asmx | 3 | 695 |
| 2 | GlobalWeather | http://www.webservicex.net/globalweather.asmx | 5 | 308 |

Used SQL Server to create a database, the table consisted of four columns STRSERVICE which tells us the name of the service called, STRSERVICEURL consisted of the URL of the service called. TIME SCALLED column kept the track of how many times a particular web service is requested and INTERVAL kept the record of the time taken by each web service requested.

Chapter 4

Implementation

1. **IMPLEMENTATION**

**4.1 WEB SERVICE DESCRIPTION LANGUAGE (WSDL):**

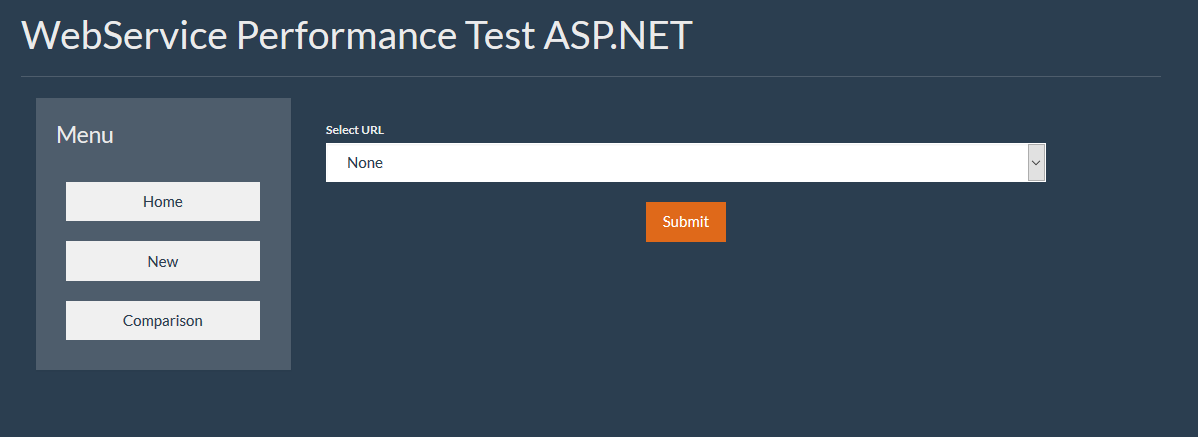
Web Services Description Language (WSDL) is a format for describing a Web Services interface. It is a way to describe services and how they should be bound to specific network addresses.

**Use of WSDL in web Service**:

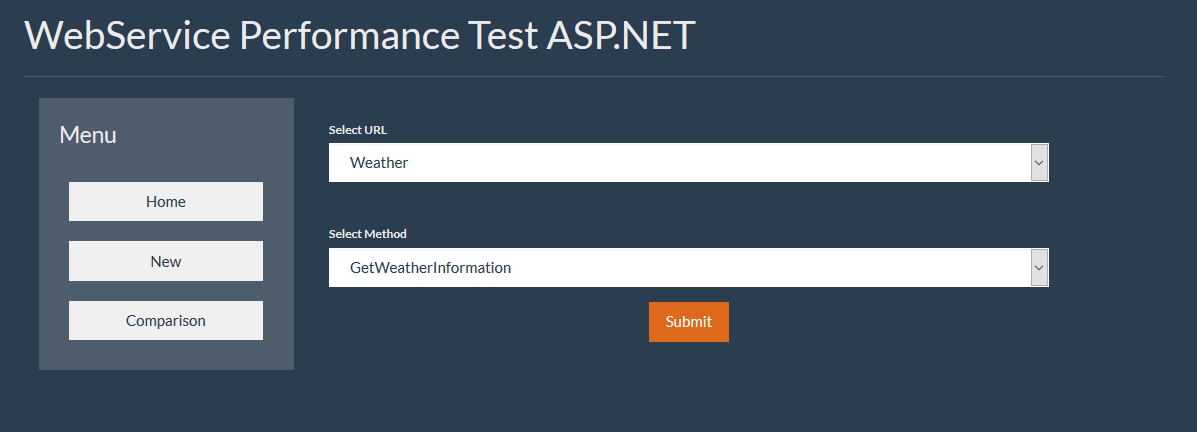
**WSDL** is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint.

**4.2 METHODS**:

**4.2.1 Home page:**

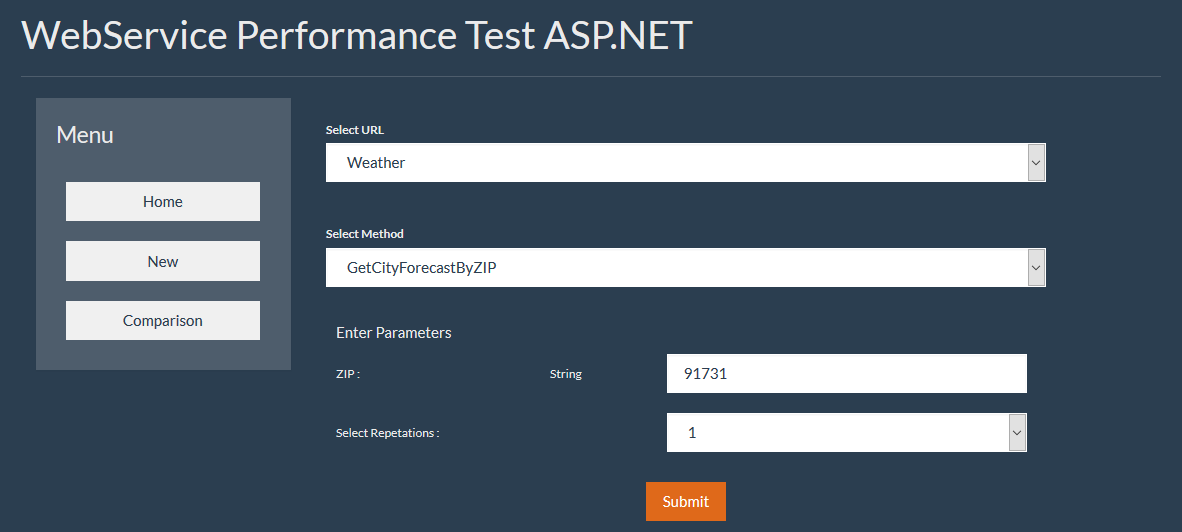


* Home page consists of a menu tab such as Home, New and Comparison.
* Select URL: It selects URL, which says weather and global weather.
* Weather: A WSDL service is called at the backend, which in turns calls, a web service. Weather consists of a URL, which calls a method and selects a type of URL.
* Global Weather: A URL is called at the backend and returns the service for the view. After selecting the type of URL a type of method is being selected.
* Method: This method information is called and provides us three functions. These functions include getting weather information by ZIP and getting city forecast by ZIP.



After selecting the above two methods, the web service asks for some of the parameters.

1. ZIP: Enter a ZIP code of the area whose weather is to be forecasted
2. String: Enter the name of the area of the forecasted weather.
3. Repetition: Select the number of repetition calls that has to be called to the server.

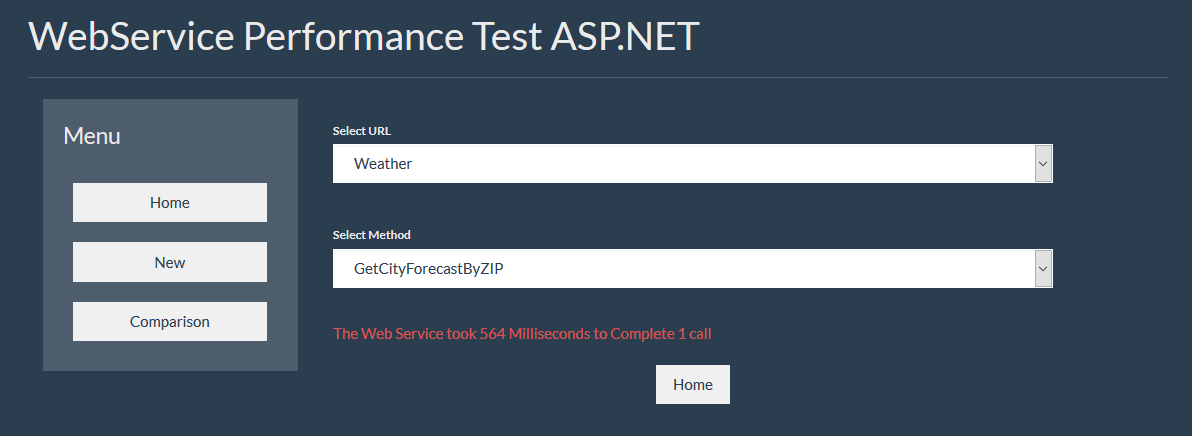


After clicking the Submit button, a display page can be seen where the output is given as how much time is taken by the web service in the given number of calls**.** Example: URL: Global Weather

ZIP: 91801

Repetition calls: 1

4. Output: “The web service took 559 Milliseconds to complete 1 call.”



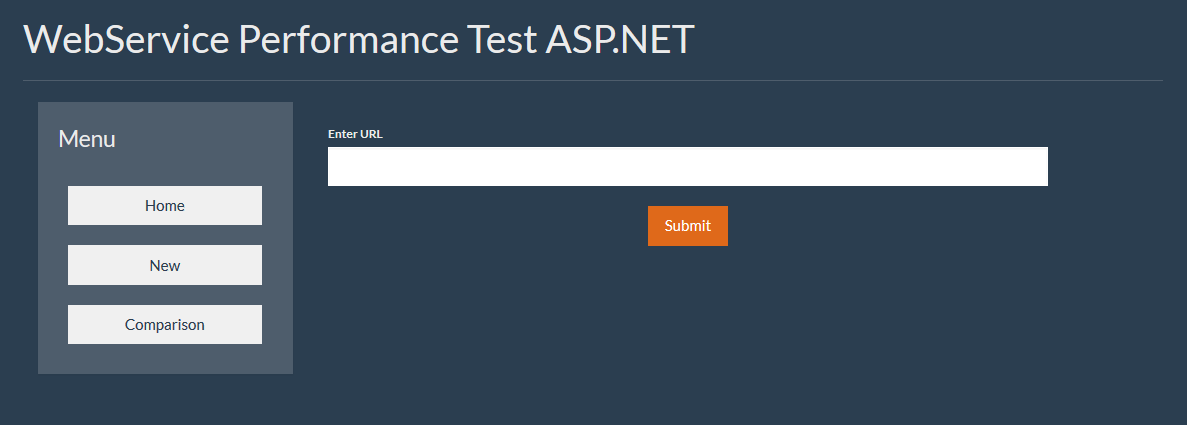
**4.2.2. New page:**

The functions used under this page are not pre-defined. This page asks for a new URL to enter. After entering the URL, it again asks for the type of method to be selected.

* Weather
* Global Weather

After clicking the submit button, the selection for parameters such as String or ZIP occurs.

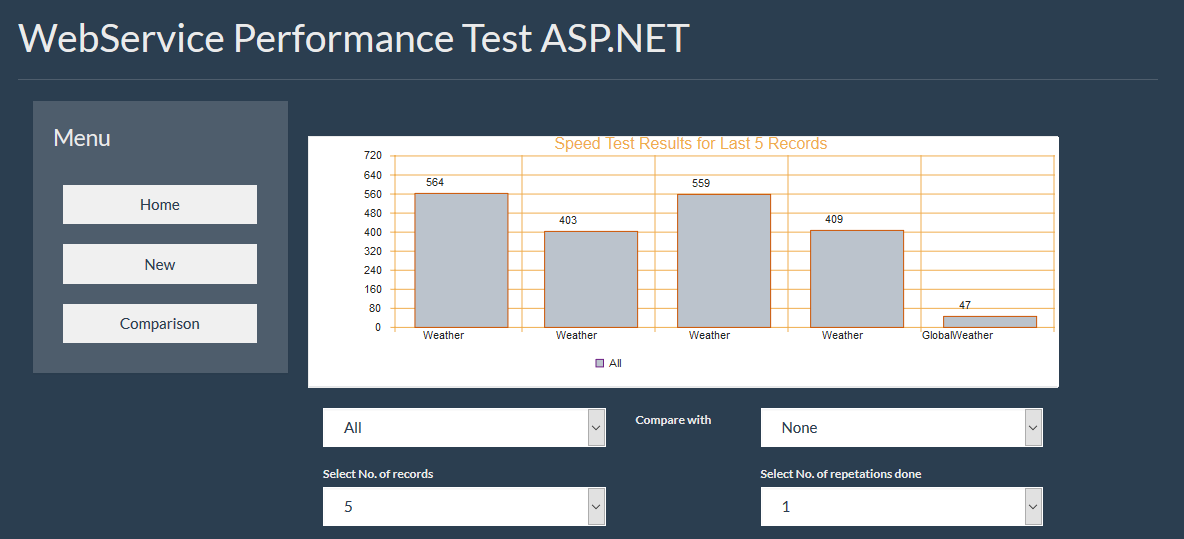
Entering the ZIP or name of the area. Prompts us to select the number of repetition calls that has been made to the web service.

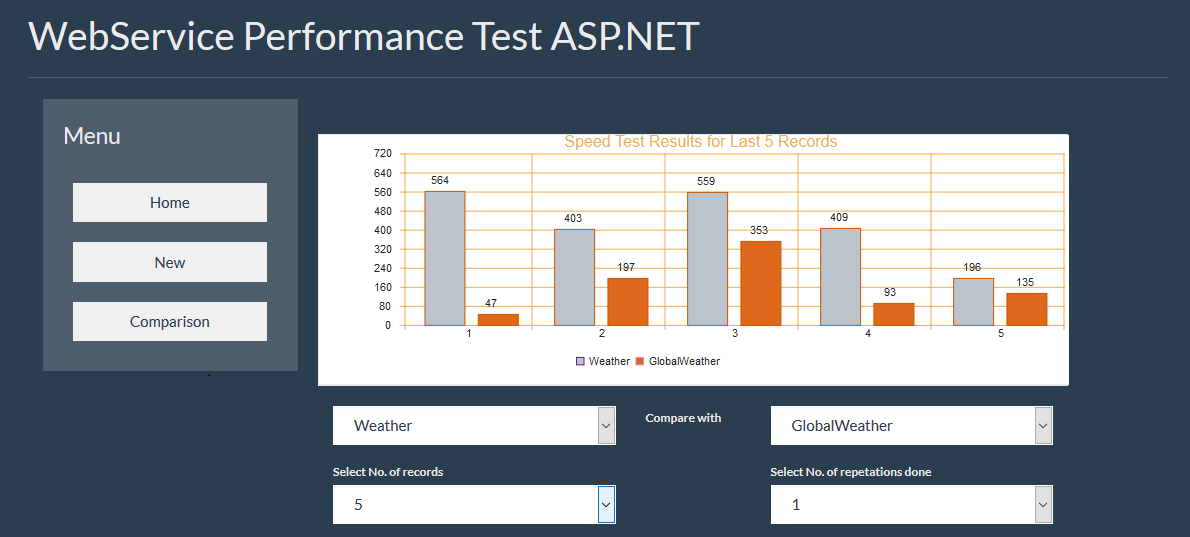


**4.2.3.** **Comparison:**

This is the most crucial part of the project, which compares the web services of different methods. This proves the efficiency of the web service and the time of each web service on a given number of calls.

* It prompts the user to enter and compare the types of services between Global weather and Weather.
* Selects a number of records: It selects the number of records to be displayed to show the comparison.
* Select number of repetition: Enters the number of repetition calls to the web service to be done for the comparison.



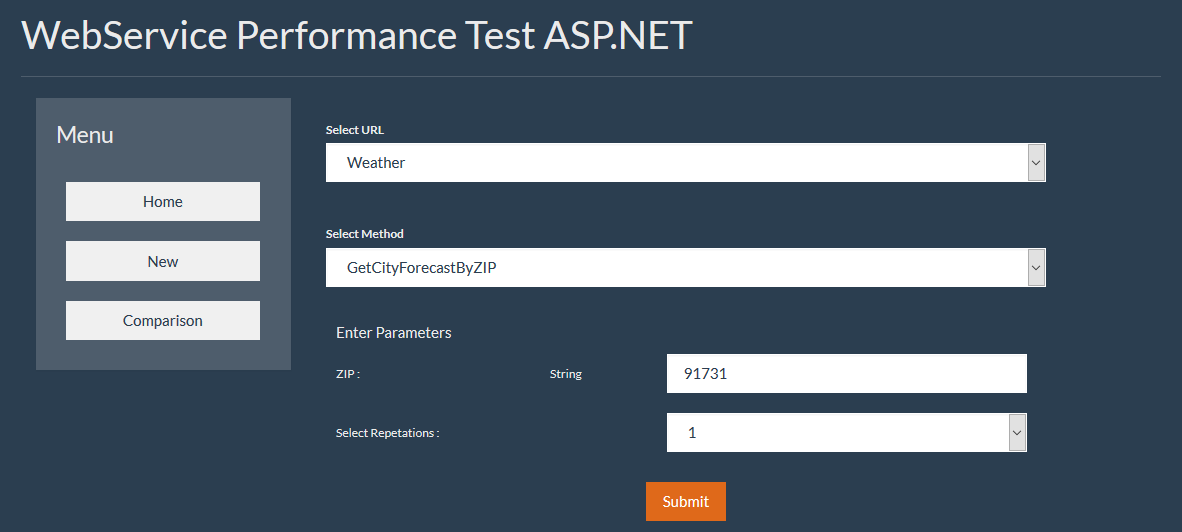


**4.3. PARAMETERS:**

The number of parameters is being decided by the URL and method. Web service is being called even if no parameter is being passed as an argument. It still asks for the number of repetition calls and displays the time taken by the web service.

Parameters to be entered:

* URL: Global Weather or weather
* Select Method: Weather forecast by Zip or City forecast
* String or ZIP
* Number of repetitions
* Time taken: time taken by each web service is being calculated on the basis of the parameters given in the web service. The response time is calculated in Milliseconds based on the number of repetition calls made by the user.



**4.4 DATABASE HANDLING:**

The database is executed using SQL Server 2012. The most preferred database used with ASP. NET is SQL Server since it is most compatible with the database because both are the products of Microsoft and best work with each other.

* One table is created for entering the data for the web service called “Records”.
* The attributes defined in the database are “URL”,”Method”,”String or ZIP”, “Repetition calls” and “time interval”.
* Time interval is being calculated on the given parameters and stored in the database. The display is given in Milliseconds.

Chapter 5

Conclusion and Future Scope

1. **CONCLUSION AND FUTURE SCOPE**

The main goal of the project is to display the performance of the web service and its efficiency. The goal is being achieved by the use of ASP. NET and IIS server with the help of database in SQL Server.

The working of the web service can be explained by the pre-defined functions from the web service of the weather forecasting. Some of the functions include the selection of URL and method with some of the given parameters or no parameters.

The web services being selected can be compared in the form of a graph. It selects the number of entries to be compared. The graph selects the type of web service to be compared and the number of repetition calls of the web service. The output can be seen in user friendly and interactive fashion, which can be easily interpreted by a user.

.

Chapter 6

References

1. **REFERENCES**
2. [http://www.service-architecture.com/articles/web- services/web\_services\_description\_language\_wsdl.html](http://www.service-architecture.com/articles/web-%20%09services/web_services_description_language_wsdl.html)
3. <http://www.codeproject.com/KB/webservices/>
4. <http://www.wcc.nrcs.usda.gov/web_service/AWDB_Web_Service_Tutorial.htm>
5. <https://msdn.microsoft.com/en-us/library/bsz5788z.aspx>