ONLINE JOB SEARCH

By

SWETHA DEVA

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas
2008

Approved by:
Major Professor
Dr. Daniel Andresen

ABSTRACT

The aim of this project is to help students find a job that suits their profile. This provides a common platform for the job seekers to search for jobs on one website instead of searching them on multiple websites which highly reduces the time of searching for a suitable job. This website also provides a platform for the recruiters to post a job and search for the resume suitable to their job requirements.

This website allows the job seekers to build a resume using resume builder (using this students can design their resume online), search for a job (search is based on different selection criteria like location, salary, job type, company, category etc), check apply history (can go through the list of jobs applied), create a search agent according to their priorities through which they can be updated with all the latest jobs posted on the website.

This application also allows the recruiters to post a new job available in their organization, can search for resume and can schedule the interview if the person's profile matches with the job requirements posted by the recruiter.

This website is developed using ASP.NET 2005 and MS SQL SERVER 2005. The main goal in designing this website was to get familiar with .NET technology.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

I would like to thank my Major Professor Dr. Daniel Andresen for his constant help, encouragement and guidance throughout the project.

I would also like to thank Dr. Torben Amtoft and Dr. Mitchell L. Neilsen for their support and for graciously accepting to serve on my committee.

Dedication

I would like to dedicate this project to my parents Mr. Pratap Deva and Mrs. Manjula Deva for their continuous encouragement and support through all my difficult times.

1. INTRODUCTION

1.1 ONLINE JOB SEARCH

Online job search is a search engine that helps the user to find a right job that matches his profile. This web interface provides a user friendly search engine where the user can choose the category to search according to his priority there by reducing the response time. The search results are filtered by making an appropriate selection in the search criteria.

This application also provides a user friendly interface for both the users and the recruiters. Multiple options like resume builder, creating a search agent, checking the apply history are provided to the job seeker. User can also update his/her profile whenever needed. Recruiter can post a job, search for the candidate resume according to his job requirements, check the list of jobs he posted on to the website and can schedule the interview with the suitable candidate. All these features provided to job seeker and recruiter makes the interface user friendly and reduces the response time in searching for jobs for the job seekers and resume for the recruiters.

1.2 OBJECTIVE

The main objective to implement this web application is to get familiar with the technologies like ASP.NET 2.0 with C# and SQL SERVER 2005. This web application is implemented using ASP.NET 2.0 for the front end development and SQL SERVER 2005 as the database back end.

There are two major roles involved in this web application:

- Job Seeker
- Recruiter

Job seekers registered with this website can use all of the features listed below:

Build a Resume:

Job seeker can build a resume using the resume builder provided on this website. Resume builder creates a resume with all the information entered by the user.

Search for Jobs:

Job seeker can search for jobs using different search criteria there by reducing the total time in searching for the jobs matching his/her profile.

Create an Agent:

The registered user can create a search agent with his priorities so that this creates an additional filter for the jobs that match the user's profile.

View and Apply for a Job:

The user can view all the details of the job like salary, location of the job, job type etc and apply for the job if he/she is interested**.**

Job History:

The job seeker can check the list of jobs he/she applied for so that he can keep track of the list for future use if required.

Recruiters registered with this website can use all of the features listed below:

Post a Job

Recruiter can post a new job with all details like job title, number of job positions available, salary offered for that position etc.

Search for Resume

Recruiter can search for the candidate resume whose profile matches the job requirements posted by him.

Post History

Recruiter can also view the jobs posted by him so that he can keep track of the listing for his future use.

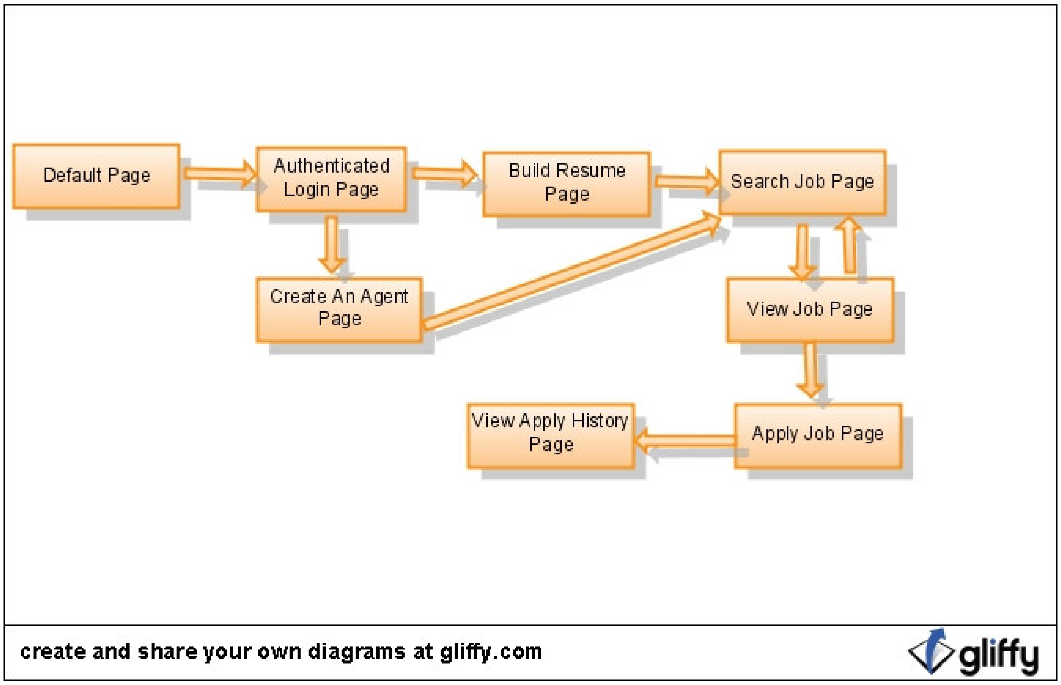
1.3 INTENDED AUDIENCE

The intended audiences for this website are the job seekers who search for job and also the recruiters who are looking for the candidate that matches the job profile he posted.

2. IMPLEMENTATION

The Online Job Search web application is implemented using Microsoft visual studio framework 2005 and Microsoft SQL Server 2005. The diagram below shows the overall page flow of the web application:

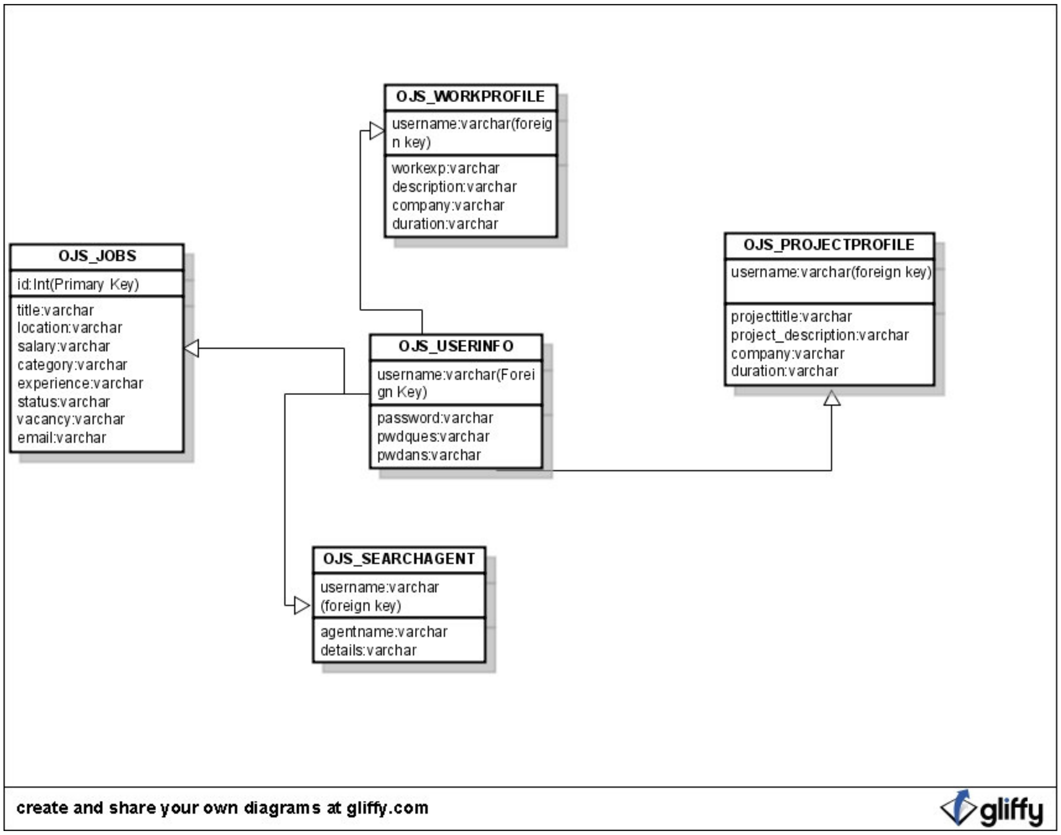
Initially user logs into the system and performs necessary operations as shown below:



The overall flow of the application is shown above. The user initially logs into the application and builds the resume and search for jobs according to his requirements. The user interface for the search page has options to select the search criteria, this filters the search based up on the selected options. Then he can apply for the jobs that match the user's profile. User can check the list of the jobs applied so that they can have the track of the jobs applied. User can also create the search agent based up on the requirements and can get remainders when ever there are new jobs matching the search agent criteria.

In the similar way recruiter logs into the application and can search for the resumes that match the job requirements posted by him. He can also post a new job on to the website. Recruiter can also keep track of the list of jobs he posted on the website. Recruiter can send a notification to the user whose profile matches to the job requirements he posted on to the website.

2.1 Database Schema:



The database schema diagram is shown above with all the tables used in this web application. It also depicts the relationships between the tables in the database. The OJS_USERINFO table has the primary key username which is used as the foreign key in all other tables to store the information pertaining to that specific user. The OJS_JOBS is the main table through which the signed up user can retrieve the data about the jobs posted on to the website.

Tables used in this application are listed below:

- OJS_JOBS
- OJS_WORKPROFILE
- OJS_PROJECTPROFILE
- OJS_USERINFORMATION
- OJS_AGENT

OJS_JOBS:

This table keeps track of all the attributes of jobs like job id, job title, number of job positions available, job description, job status, job requirements, salary, location of the job, type of the job etc. This table keeps track of all the new jobs posted on the website. Every job posted on this website is assigned a unique id so that there will not be any difficulty in retrieving the information regarding the jobs.

OJS_WORKPROFILE:

This table has all information about the work profile of the student who registered with the site. It records the work experience student has. The fields in this table include job title, organization name, description of the work and number years the user worked for.

OJS_PROJECTPROFILE:

This table has all information about the work profile of the student who registered with the site. It records the work experience student has. The fields in this table include project title, organization name and description of the project and number years the user worked on that project.

OJS_USERINFORMATION:

This table keeps track of all users registered with this site. This table records the user attributes like username, password, password related questions and answers. This table has all the details of a user. Every user registered with this site has a unique user id. This table keeps track of all the registered users with this website.

OJS_SEARCH_AGENT:

This table records the information about the search agent created by the user. Every search agent created has a unique id and it has all the priorities listed by the user. It retrieves the results based up on these felids when ever the user runs the search agent.

3. TECNOLOGIES USED

Here we discuss the details of technologies used to develop this web application. This application is developed using ASP.NET 2.0 with C# and MS SQL SERVER 2005.

3.1 Microsoft Visual Studio.NET 2.0:

Microsoft Visual Studio is an Integrated Development Environment (IDE) from Microsoft. It can be used to develop console and Graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code and managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework.

Visual Studio supports languages by means of language services, which allow any programming language to be supported (to varying degrees) by the code editor and debugger, provided a language-specific service has been authored. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#).

Visual Studio 2005 was upgraded to support all the new features introduced in .NET Framework 2.0, including generics and ASP.NET 2.0. The IntelliSense feature in Visual Studio was upgraded for generics and new project types were added to support ASP.NET web services. Visual Studio 2005 also includes a local web server, separate from IIS that can be used to host ASP.NET applications during development and testing. It also supports all SQL Server 2005 databases.

3.2 ASP.NET:

ASP.NET aims for performance benefits over other script-based technologies (including Classic ASP) by compiling the server-side code to one or more DLL files on the web server. This compilation happens automatically the first time a page is requested (which means the developer need not perform a separate compilation step for pages). This feature provides the ease of development offered by scripting languages with the performance benefits of a compiled binary.

The ASPX and other resource files are placed in a virtual host on an Internet Information Services server. The first time a client requests a page, the .NET framework parses and compiles the file(s) into a .NET assembly and sends the response; subsequent requests are served from the DLL files. By default ASP.NET will compile the entire site in batches of 1000 files upon first request. If the compilation delay is causing problems, the batch size or the compilation strategy may be tweaked.

ASP.NET leverages the multi-language capabilities of the .NET Common Language Runtime, allowing web pages to be coded in VB.NET, C#, J#, Delphi.NET, Chrome etc.

ASP.NET encourages the programmer to develop applications using an event-driven GUI model, rather than in conventional web-scripting environments like ASP and PHP.

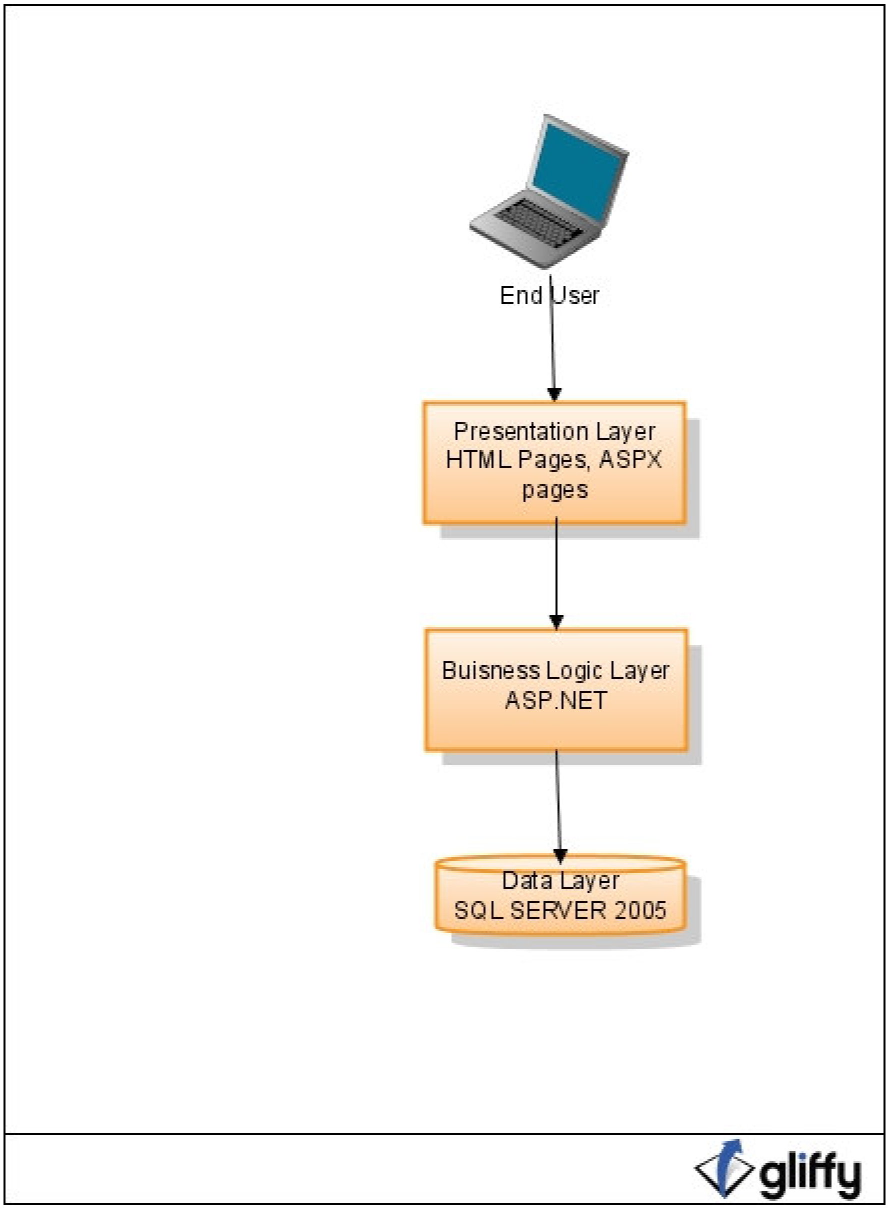
3.3 SQL SERVER 2005:

Microsoft SQL Server is a relational database management system (RDBMS), its primary query languages are MS-SQL and T-SQL. For relational data, T-SQL has been augmented with error handling features and support for recursive queries. SQL Server 2005 has also been enhanced with new indexing algorithms and better error recovery systems.

SQL Server 2005 also allows a database server to be exposed over web services using TDS packets encapsulated within SOAP requests. Online Job Search web application makes use of MS SQL SERVER 2005 to store the data and generate the required results.

4. System Architecture

The architecture of this web application is based on the three-tier architecture. The three layers present are: Presentation layer, Business Logic Layer, Data Access Layer.



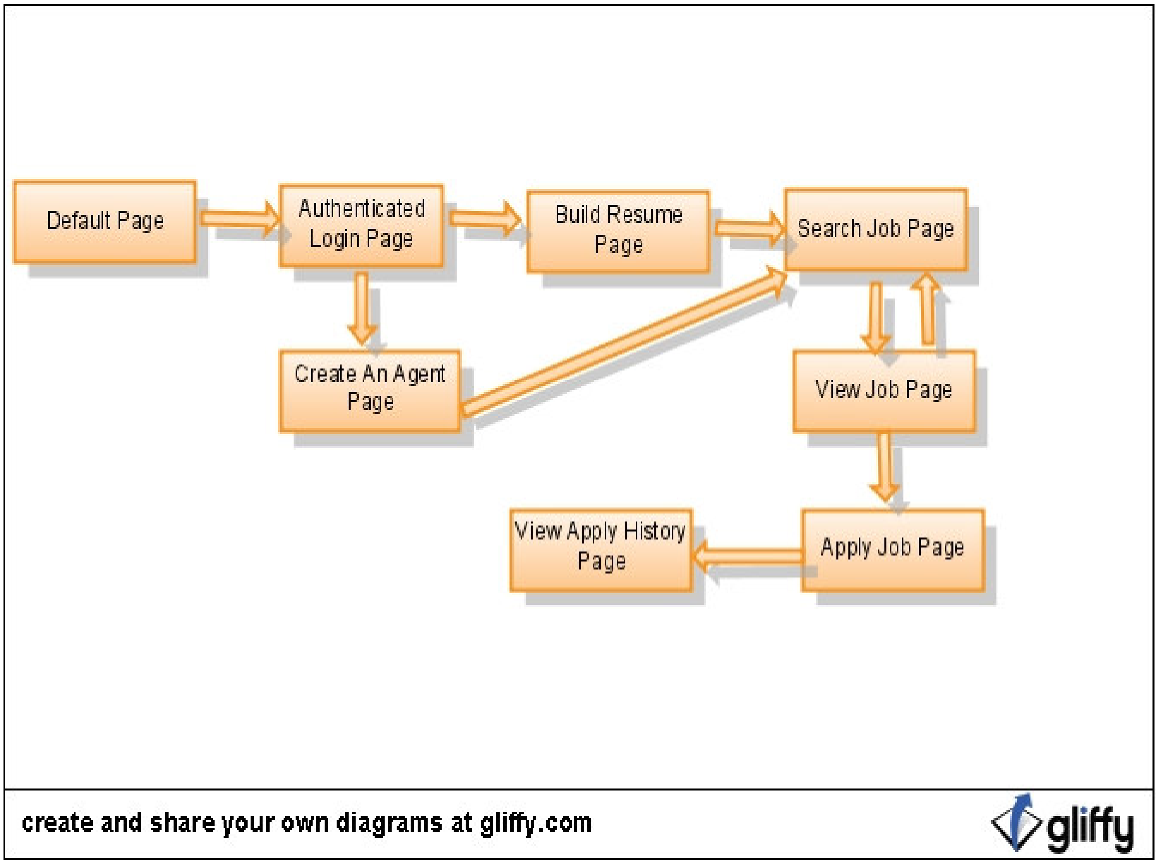
Each layer provides service abstraction to the layer above it. The data layer provides persistence services to the business logic layer and the business logic layer provides request / response services to the user interface layer.

4.1 Presentation Layer:

The presentation layer or the user interface layer contains all the asp.net forms and user controls. These pages are designed in the Microsoft Visual Studio 2005 IDE. The ASP.NET 2.0 version has a support for master pages which has all the common controls on the user web pages and only the content tab is loaded for different pages. I made use of master pages for the whole application which helped in reducing the amount of time spent on the designing of pages.

Users can interact with the application logic through this layer. The user interface is used to display content to the user. This layer takes input from the user, validates and passes them to the business logic layer for further processing. This layer can be more understood by looking at the page flow diagram below:

User Page Flow Diagram

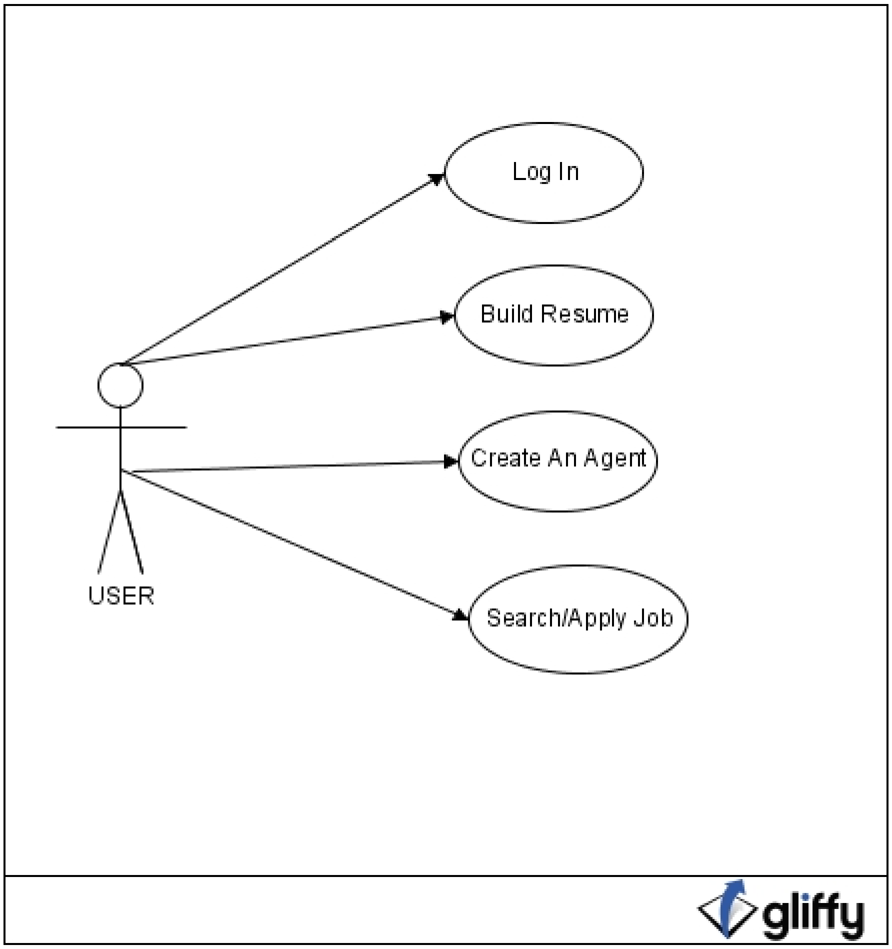


4.2 Business Logic Layer:

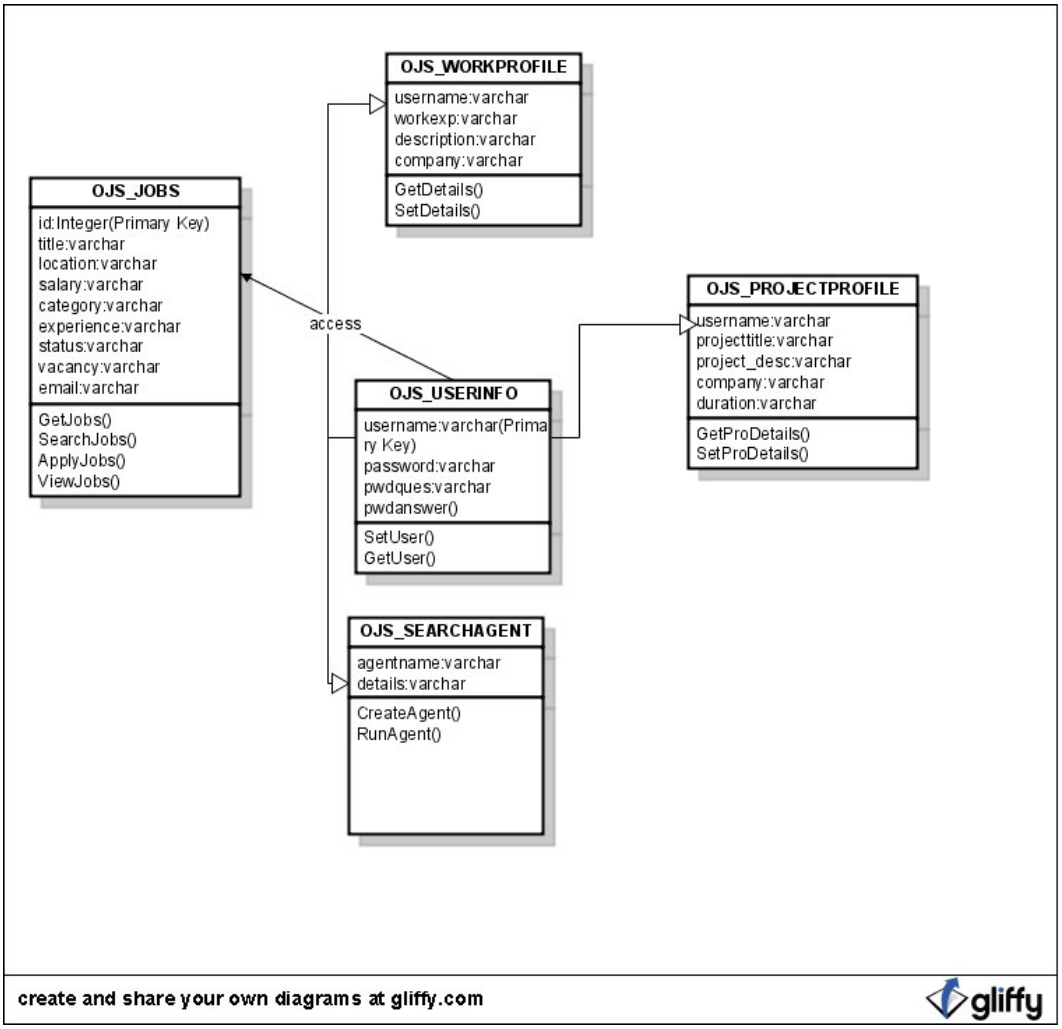
The major part of the application logic is involved here. This is done using the .NET framework. The class files manage the processing of the requests received from a client and for generating responses to be sent back to a client. This layer makes use of the Application Programmable Interface provided by the data layer. The middle layer or the business logic layer can be more understood by looking at the use-case diagram and class diagrams.

USECASE DIAGRAM:

Below is the use case diagram for the user. User can log in and create or build a resume using resume builder and the he can create a search agent with all his preferences. User can search for jobs, view jobs, apply for jobs. All these four major functionalities are depicted in the use case diagram below:



CLASS DIAGRAM FOR USER:



4.3 Data Access Layer:

The data layer manages the persistence mechanism of the application. This layer provides the interaction with the databases and also it provides functionality to the higher layers. This layer makes use of MS SQL SERVER 2005 in this specific web application.

5. Testing

In this we discuss the various tests performed on this web application. The major tests performed on this website are unit testing and performance testing.

5.1 PERFORMANCE TESTING:

Performance of a website is tested by applying some stress or increasing the load on the website. This testing is done using Apache JMeter. Apache JMeter is a 100% pure Java desktop application designed to load test functional behavior and measure performance. It was originally designed for testing Web Applications but has since expanded to other test functions.

Apache JMeter may be used to test performance both on static and dynamic resources (files, Servlets, Perl scripts, Java Objects, Data Bases and Queries, FTP Servers and more). It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types. You can use it to make a graphical analysis of performance or to test your server/script/object behavior under heavy concurrent load.

Using the JMeter I have taken increased the number of users using the page with the same Ramp-Up period and Loop count. Below is the table which lists all the different test cases I considered for testing the pages of this application.

Users	Ramp-up Period	Loop Count
100	5	1000
500	5	1000
1000	5	1000

The table below shows the tests results obtained after performing load testing on the website.

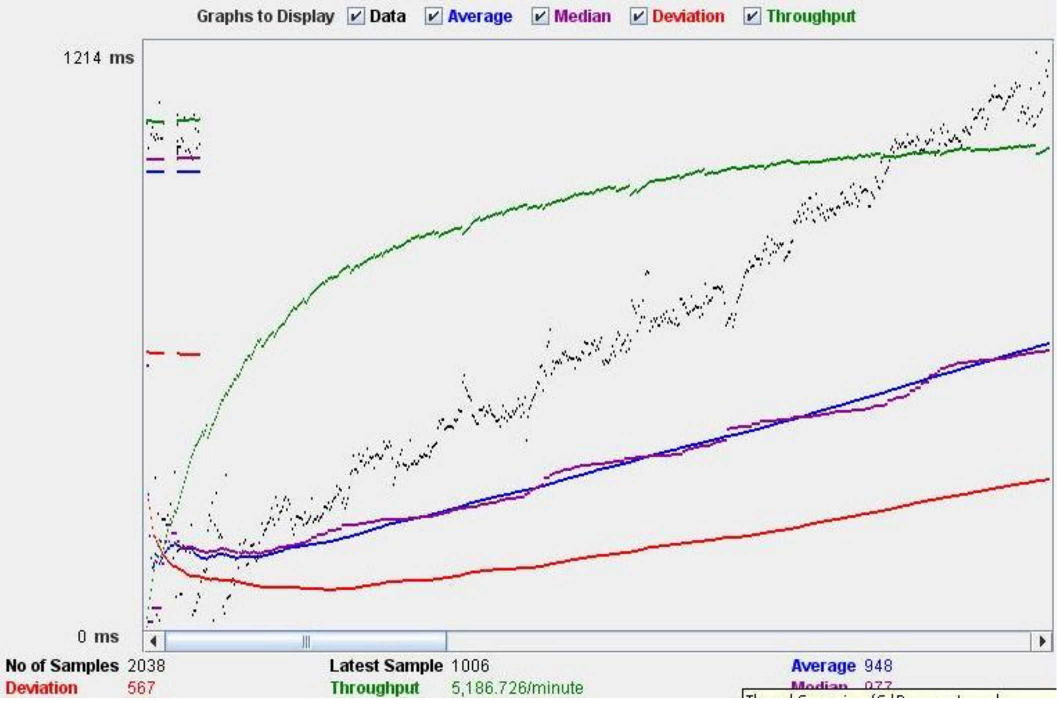
Users	Ramp-Up Period	Loop Count	Average Response	Throughput

			Time(ms)	
100	5	1000	1200	2320/min
500	5	1000	2927	3249/min
1000	5	1000	3243	4859/min

Test Case Analysis:

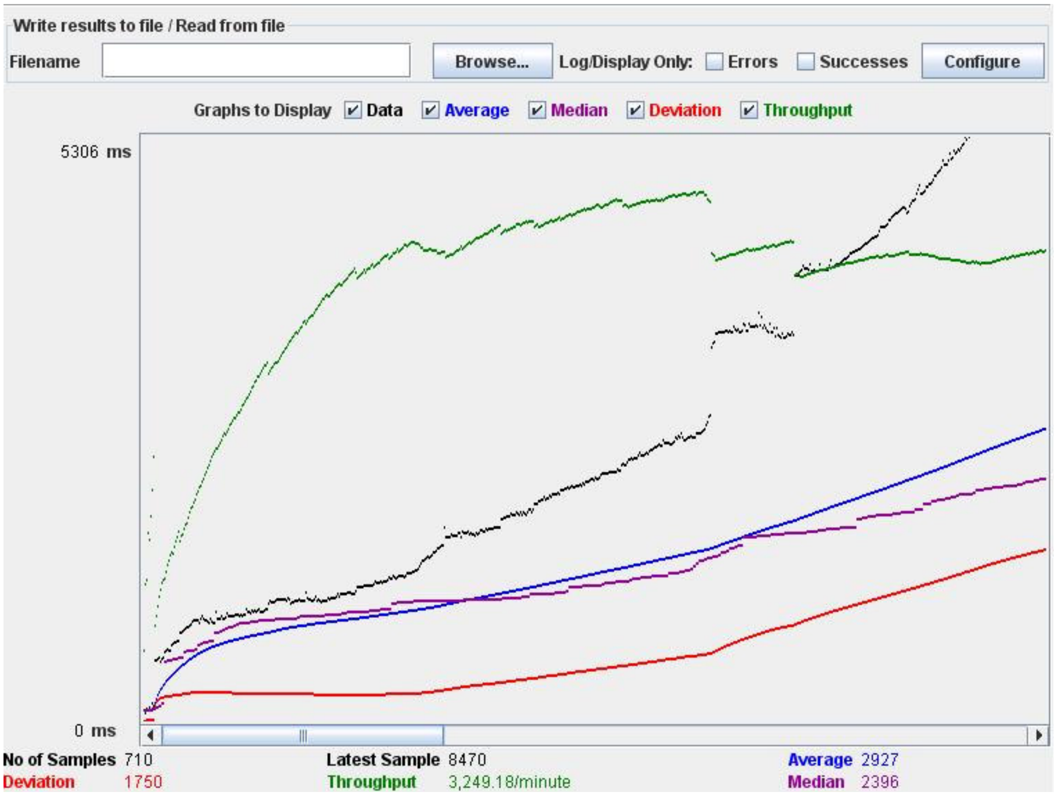
From the above observations and the graphical analysis it is clearly seen that the response time is increasing rapidly as the load on the website is increased.

Graphical Analysis for the first observation:



No of Samples 2038
Deviation 567
Latest Sample 1006
Throughput 5,186.726/minute
Average 948
Median 977

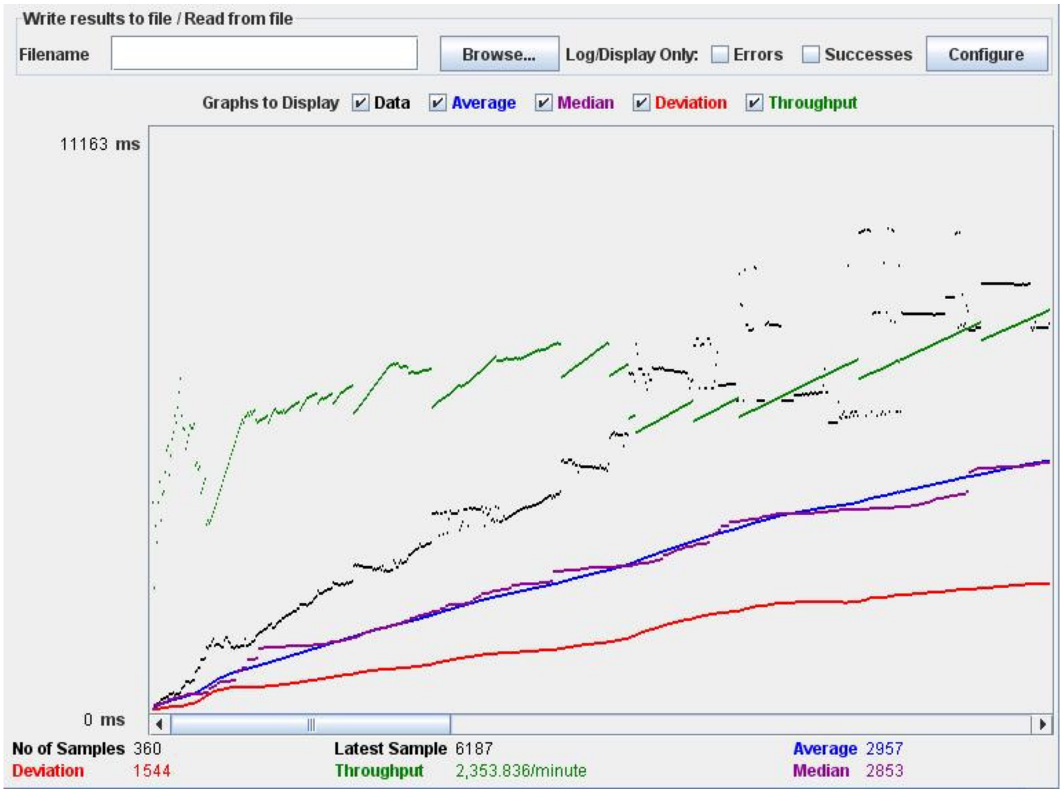
Graphical Analysis for the second observation:


Write results to file / Read from file
Filename
Browse...
Log/Display Only: □ Errors □ Successes
Configure
5306 ms
0 ms

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 710
Deviation 1750
Latest Sample 8470
Throughput 3,249.18/minute
Average 2927
Median 2396

Graphical Analysis for the third observation:


Write results to file / Read from file
Filename
Browse...
Log/Display Only: □ Errors □ Successes
Configure
11163 ms
0 ms

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 360
Deviation 1544
Latest Sample 6187
Throughput 2,353.836/minute
Average 2957
Median 2853

5.2 UNIT TESTING:

Unit Testing is a method of testing that verifies the individual units of source code are working properly. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or derived/child class.
I made sure that all the modules are tested during the development stage so even the basic entity of the source code undergone testing.

The major modules covered during the process of unit testing are: data retrieving, data submission and the navigation of the website.

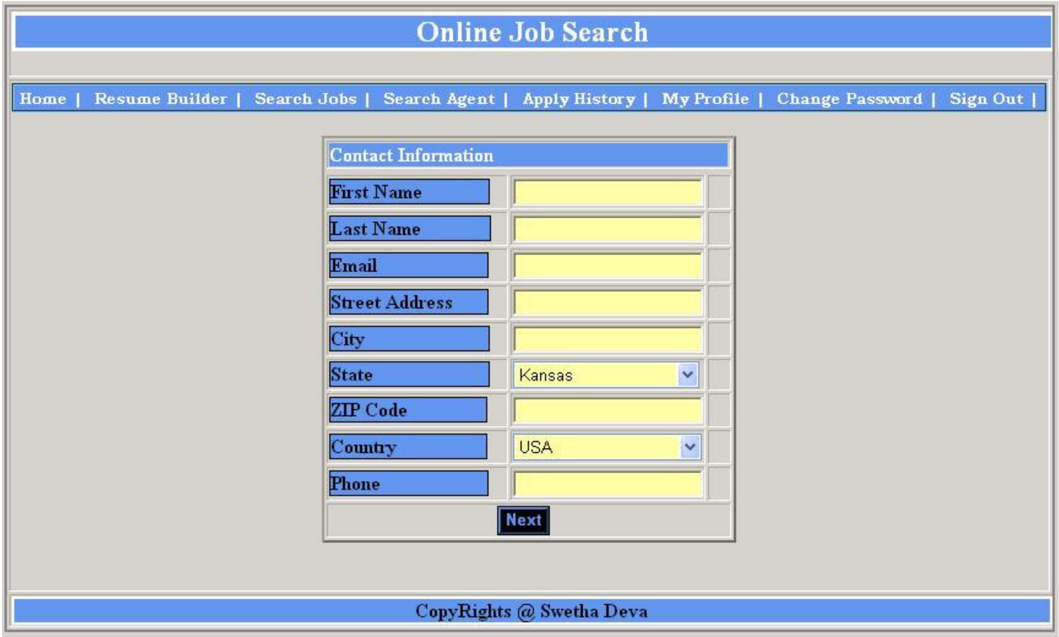
I made sure that all the data retrieving functionalities retrieve the data in a desired manner with out any loss of generated data from the database.

Similarly with the data submission for example when a user wants to sign up with the website his data is submitted up on his confirmation during this stage, I made sure that data is submitted properly with out any issues.

Now coming to the navigation module when ever there are some buttons or links on a page then I made sure that all the links and the link buttons redirect the user to the required page.

6. SCREEN SHOTS OF THE APPLICATION

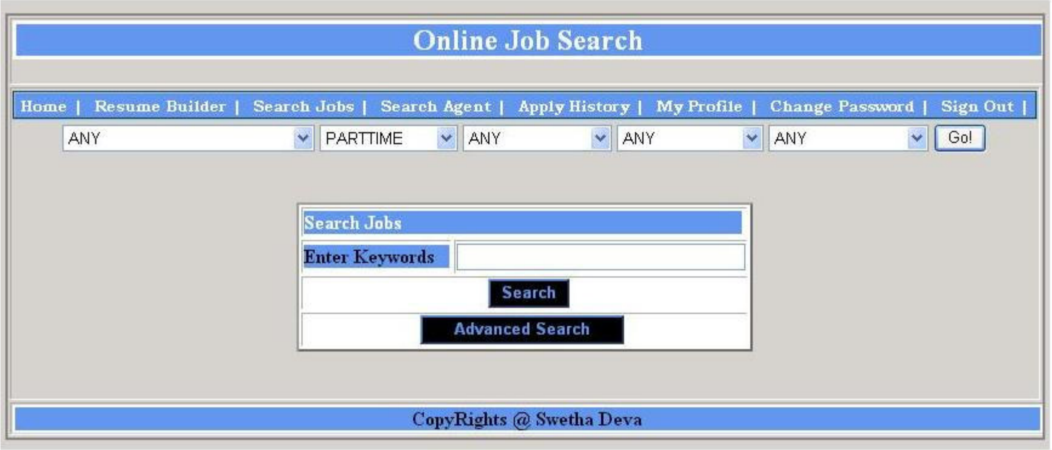
RESUME BUILDER VIEW 1:


Online Job Search
Home | Resume Builder | Search Jobs | Search Agent | Apply History | My Profile | Change Password | Sign Out |
Next
CopyRights @ Swetha Deva

Contact Information

First Name	
Last Name	
Email	
Street Address	
City	
State	Kansas
ZIP Code	
Country	USA
Phone	

SEARCH PAGE

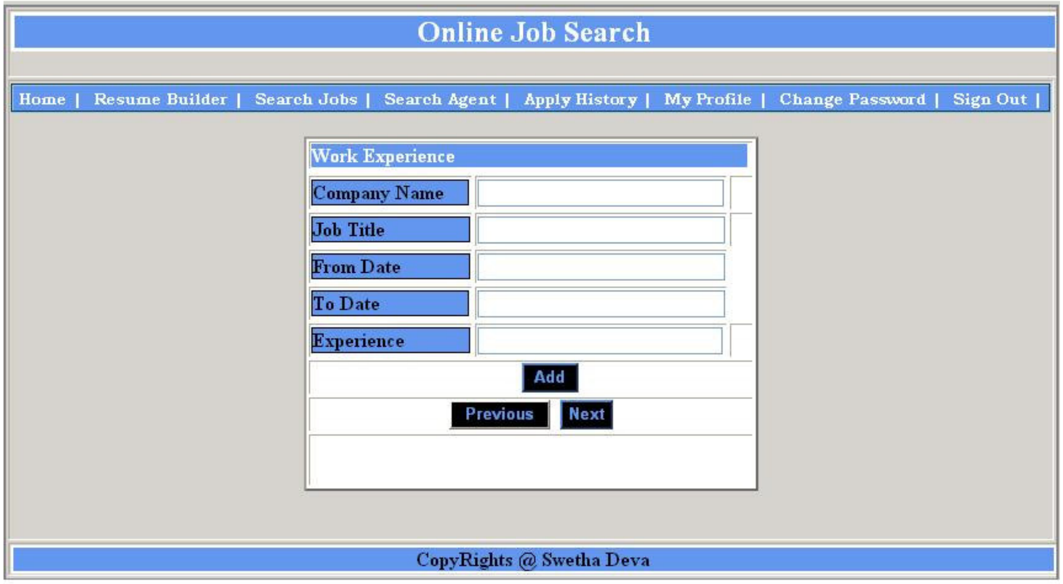

Online Job Search
Home | Resume Builder | Search Jobs | Search Agent | Apply History | My Profile | Change Password | Sign Out |
Go!
Search
Advanced Search
CopyRights @ Swetha Deva

ANY PARTTIME ANY ANY ANY

Search Jobs

Enter Keywords

RESUME BUILDER VIEW 2:


Online Job Search
Home | Resume Builder | Search Jobs | Search Agent | Apply History | My Profile | Change Password | Sign Out |
Add
Previous
Next

Work Experience

Company Name

Job Title

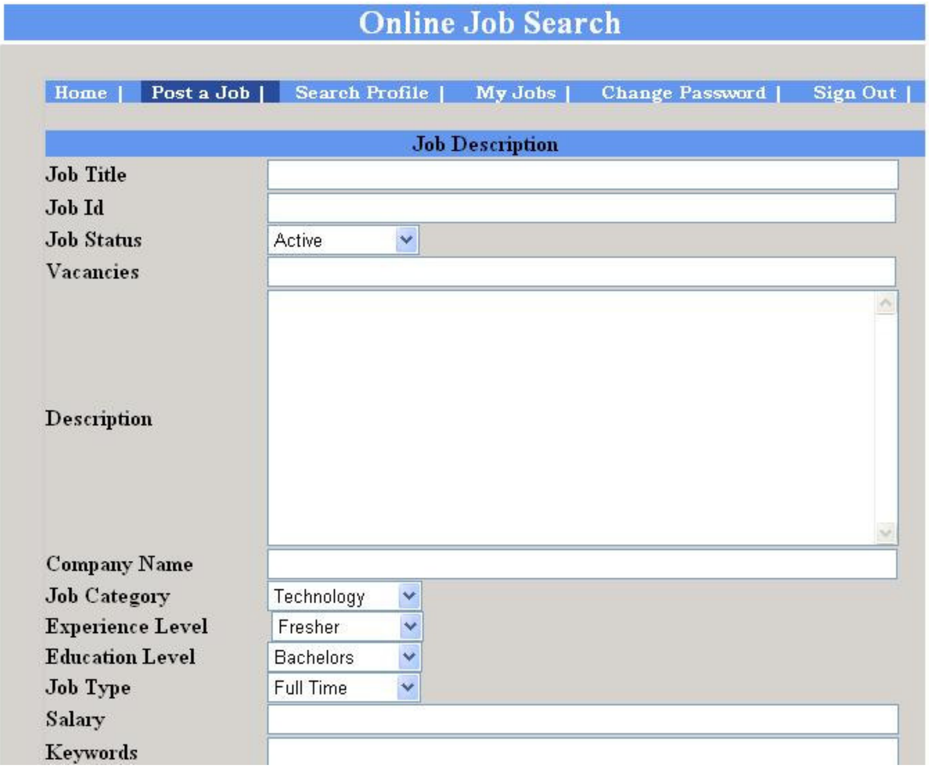
From Date

To Date

Experience

CopyRights @ Swetha Deva

RECRUITER – POST A JOB VIEW:


Online Job Search
Home | Post a Job | Search Profile | My Jobs | Change Password | Sign Out |

Job Description

Job Title

Job Id

Job Status: Active

Vacancies

Description

Company Name

Job Category: Technology

Experience Level: Fresher

Education Level: Bachelors

Job Type: Full Time

Salary

Keywords

7. CONCLUSION

This web application is designed in order to provide a simple and easy to use search engine for job seekers and also the recruiters. This application helped me to learn new concepts of ASP.NET and also SQL SERVER 2005.

This application has lot of database operations involved in it which helped me in learning various database related concepts also.

Reference:

http://en.wikipedia.org/wiki/Microsoft_Visual_Studio

http://en.wikipedia.org/wiki/Unit_testing

http://en.wikipedia.org/wiki/Microsoft_SQL_Server

http://www.asp.net/

http://www.csharp-corner.com/

http://jakarta.apache.org/jmeter/