**ARTICLE COMPARER**

**Mini Project report submitted in partial fulfillment of the requirements**

**for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**By**

**K. Varun Kumar (13QH1A0516)**

**K. Mohan Kumar (13QH1A0519)**

**N. Prabhakar (13QH1A0532)**

**R. Sai Prasad (13QH1A0543)**

**Under the guidance of**

**Mr. Janardhan**

**Asst.prof**

**

**HOLY MARY INSTITUTE OF TECHNOLOGY**

**(Recognized by A.I.C.T.E Affiliated to JNTU, Hyderabad).**

**Bogaram(v), keesara(m), R.R.Dist 501301**

**ARTICLE COMPARER**

**By**

**K. Varun Kumar (13QH1A0516)**

**K. Mohan Kumar (13QH1A0519)**

**N. Prabhakar (13QH1A0532)**

**R. Sai Prasad (13QH1A0543)**

**Under the Esteemed guidance of**

**Prof G. Soma Shekar**

**Head of the Department**

**Computer Science and Engineering**

**

**HOLY MARY INSTITUTE OF TECHNOLOGY**

**(Recognized by A.I.C.T.E Affiliated to JNTU, Hyderabad)**

**Bogaram(v), keesara(m), R.R.Dist 501301**

**HOLY MARY INSTITUTE OF TECHNOLOGY**

**(Approved by AICTE, New Delhi, Affiliated to JNTUH)**

**Hyderabad-500090**

**2016-2017**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify the Project report entitled “**Article Comparer**” is being submitted by the following student in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** from **Jawaharlal Nehru Technology University** Hyderabad , Telangana is record of bonafide work carried out during the academic year 2016-2017.

**K. Varun Kumar (13QH1A0516)**

**K. Mohan Kumar (13QH1A0519)**

**N. Prabhakar (13QH1A0532)**

**R. Sai Prasad (13QH1A0543)**

**INTERNAL GUIDE HEAD OF DEPARTMENT**

**Mr. JANARDHAN Prof. SOMA SHEKAR**

**Assistant Professor Head of Department of CSE Department of CSE**

**EXTERNAL EXAMINER**

**DECLARATION**

****

**HOLY MARY INSTITUTE OF TECHNOLOGY**

(Recognized by A.I.C.T.E Affiliated to JNTU, Hyderabad)

Bogaram(v), Keesara(m), R.R.Dist 501301.

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Name: K . Varun Kumar Name: K . Mohan Kumar

Roll No.: 13QH1A0516 Roll No.: 13QH1A0519

Date: Date:

Name: N. Prabhakar Name: R. Sai Prasad

Roll No.: 13QH1A0532 Roll No: 13QH1A0543

Date: Date:

**ACKNOWLEDGEMENT**

The successful completion of any task is not possible without proper suggestions guidance and environment. Combination of these three factors acts like backbone to my project titled **“ARTICLE COMPARER ”.**

We express our sincere thanks to our guide, Department of Computer Science and Engineering, **Mr. JANARDHAN** his valuable suggestions during our course period, timely help, guidance and providing us with most essential materials required for the completion of this work.

We express our since thanks to our HOD, Department of Computer Science and Engineering.

**Mr. G.SOMA SEKHAR** for her valuable suggestions during our course period, timely help, guidance and providing us with the most essential materials required for the completion of this work.

We would like to thank our principal, **Dr. B.S REDDAPPA** for providing support and simulating environment. We would like express my gratitude to the management of college name for providing me a pleasant environment and excellent and excellent laboratory facilities.

We special thanks to project coordinator **Ms. B.V. RAMANA** and all the faculty of CSE Department for giving the required information in doing my project work. Not to forget, I thank all the non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

We are thankful to all teaching and non-teaching staff of the Department of Computer Science & Engineering for the cooperation given for the successful completion of my project.

Finally we wish to convey my gratitude to my parents who fostered all the requirements and facilities that I need.

**ABSTRACT**

It is a 'wrongful appropriation' and 'stealing and publication' of another authors language, thoughts, ideas or expression .It is the act of taking another person's writing, conversation and even ideas and passing it off as your own. To avoid all such cases we use plagiarism checker. It shows the percentage of copied content which is not unique .It also shows the percentage of unique content. It also gives the percentage of similarity. Plagiarism has a wide spread problem in Assessment tasks in computing courses, students often plagiarise documents. Plagiarism is impractical many automated tools are available none has been applied to detect inter-lingual plagiarism, where text is copied from one. By this it is easy to evaluate the documents which are plagiarised. Recently, the problem of plagiarism is becoming an important issue in many debates in the field of education and technology. The wide use and availability of electronic resources makes it easy for students , authors and even academic people to access and use any piece of information and embed into user own work without proper citation. By using this article comparer we can create our new articles which will not match with already existed articles.

**INDEX**

**ABSTRACT PAGE NO**

**1. INTRODUCTION 09**

**2. LITERATURE SURVEY 10**

2.1 Introduction

2.2 Existing System

2.3 Proposed System

**3. REQUIREMENT SPECIFICATION 11**

3.1 Introduction

3.2 Requirement specification

3.3 Functional requirement

**4. DESIGN 13**

4.1 UML Diagrams

**5. IMPLEMENTATION 22**

**6. CODING 35**

**7. TESTING 55**

7.1 Test plan

7.2 Test cases

**8. OUTPUTS 60**

**9. CONCLUSION 66**

**10. FUTURE SCOPE 67**

**11. BIBLIOGRAPHY 68**

**LIST OF DIAGRAMS**

* USE CASE
* CLASS
* SEQUENCE
* COLLABORATION
* ACTIVITY

**LIST OF OUTPUT SCREENS**

* HOME PAGE
* REGISTRATION
* LOGIN
* USER HOME
* USER PROFILE
* FEEDBACK OF USER
* ADMIN HOME
* VIEW USERS
* UPLOAD FILES
* VIEW FILES
* ARTICLE COMPARER OUTPUT

**1. INTRODUCTION**

**About Project:**

Plagiarism is a 'wrongful appropriation' and 'stealing and publication' of another authors language, thoughts, ideas or expression and passing it off as your own . To avoid all such cases we use Article Comparer. It shows the percentage of copied content which is not unique. By this it is easy to evaluate the documents which are plagiarized.

**HOW IT WORKS:**

The content in the user file will be compared by the admin’s files content which are already stored in the database. Here we need a code for this and that code is used in this software for content comparison.

In our project we have two modules ‘User’ and ‘Admin’.

**1 .User Module:**

Firstly, the user needs to do registration to login into this site. After registration he will be allowed to use this software. The user will upload files and check for plagiarism. The user can update his/her profile.

**2 .Admin Module:**

The Admin work is to view the registered users and admin can delete the unwanted users. The admin will be keep on uploading files in the database. Admin will view the feedbacks given by the users and does the modifications to this software according to the users feedback.

**2. LITERATURE SURVEY**

**2.1 INTRODUCTION**

Plagiarism is a 'wrongful appropriation' and 'stealing and publication' of another authors language, thoughts, ideas or expression .It is the act of taking another person's writing, conversation and even ideas and passing it off as your own. To avoid all such cases we use Article Comparer. It shows the percentage of copied content which is not unique. By this it is easy to evaluate the documents which are plagiarised. Recently, the problem of plagiarism is becoming an important issue in many debates in the field of education and technology..By using this article comparer we can create our new articles which will not match with already existed articles.

**2.2 EXISTING SYSTEM**

There is a software called Plagiarism checker which is already existing in today’s world which is used to show the copied content or information. In that particular software we need to paste the content in the text box to check plagiarism. But there is no option to upload files and check.

**2.3 PROPOSED SYSTEM WITH FEAUTURE**

In the “Article Comparer” project we have an feature of uploading files. Here we need to upload the files which are in “.txt” and “.doc” format. No need of copying and pasting, directly we can upload files and check for plagiarism.

**3. REQUIREMENT SPECIFICATION**

**Software Requirements:**

* **Operating system :** WINDOWS 7,8,10.
* **Front end: MICROSOFT VISUAL STUDIO 2012:** Microsoft Visual Studio is an Integrated Development Environment from Microsoft .It can be used to develop console and graphical user interface applications along with windows forms applications , websites, web applications and web services in both native code together with managed code for all platforms supported by Microsoft Windows,.NET framework,.NET compact Framework and Microsoft Silver light .It also supports the c# that we use to create web site.
* **Back end: SQL SERVER 2008 R2:** In this SQL Server Management Studio is a software application first launched with the Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components with in Microsoft SQL Server.
* **Microsoft Office Word 2013:** We use Microsoft Office Word 2013 to do our documentation of this project. This is very important for us to do this project. We have use the feature of text box to draw the hierarchical chart to describe the various of subsystem, modules and sub-modules in the system. It also uses to check our spelling and grammar and justify all the words to make our document look nicer.

**Hardware requirements:**

* System **:** Any system
* Processor **:** Pentium & later
* Hard Disk **:** 80GB & above.
* RAM **:**  512MB & above
* System Type **:** 32- Bit Operating System

**4. DESIGN**

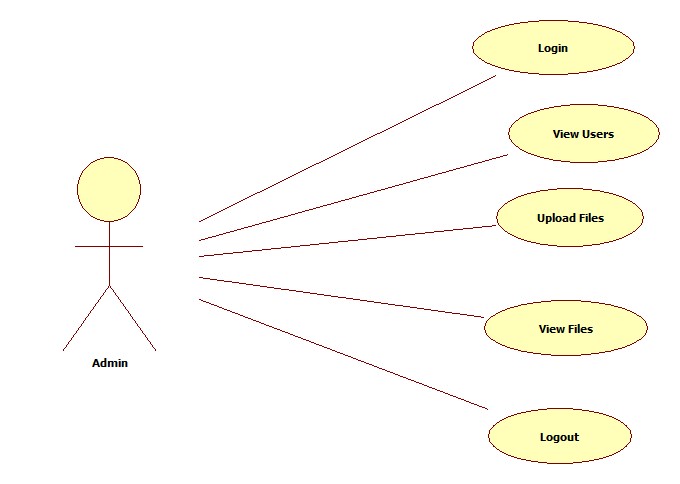
**4.1 UML DIAGRAMS**

The Unified Modeling Language (UML) is a graphical language for Visualizing , specifying , Constructing and Documenting of a software intensive system. The UML gives a standard way to write a system blue prints , covering conceptual things ,such as classes written in a specified programmed language ,Database Schemas and reusable software components.

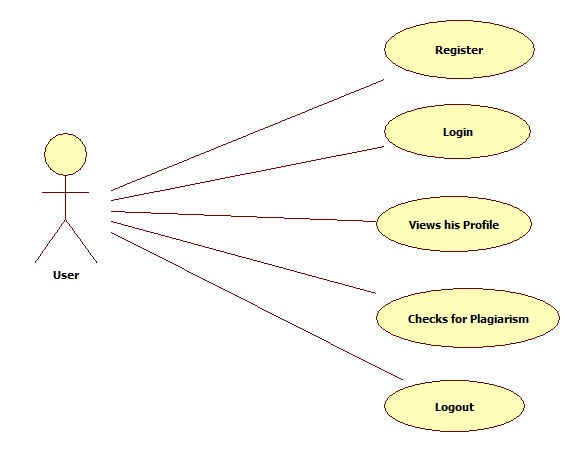
**Use Case Diagram**

Use case: Use case diagrams are one of the five diagrams in the UML for modelling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modelling the dynamic aspects of systems). Use case diagram are central to modelling the behaviour of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.

**Admin Use Case Diagram**

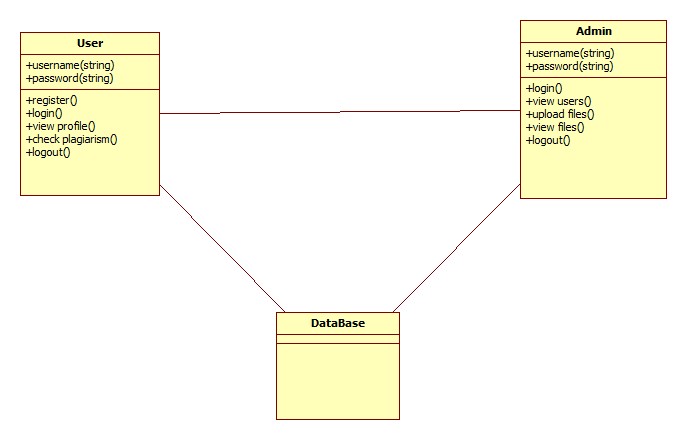
****

**User Use Case Diagram**



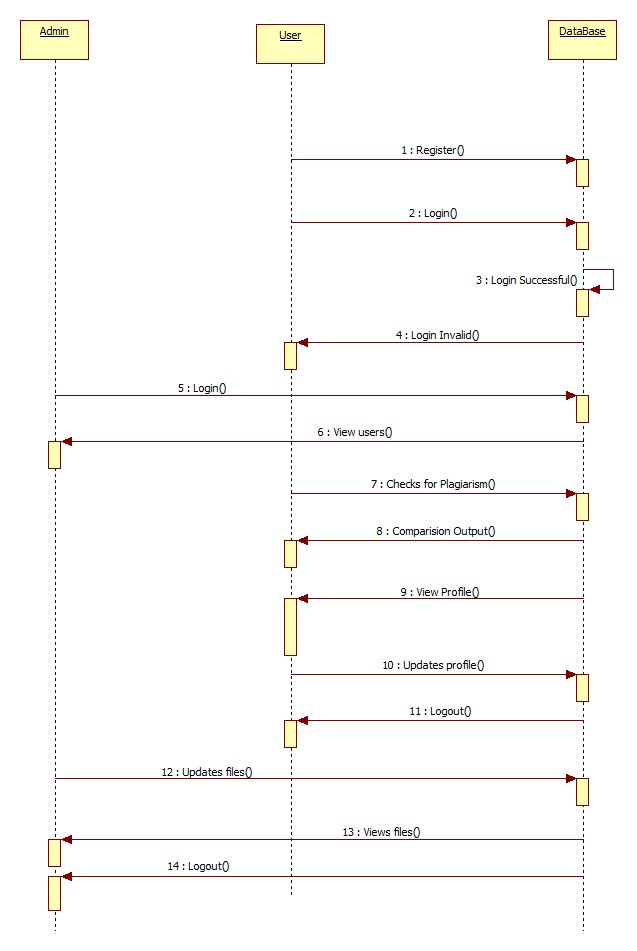
**Class Diagram**

A “Class Diagram” shows a set of classes, interfaces and collaborations and their relationships. These diagrams are most common diagram in modelling object oriented systems.

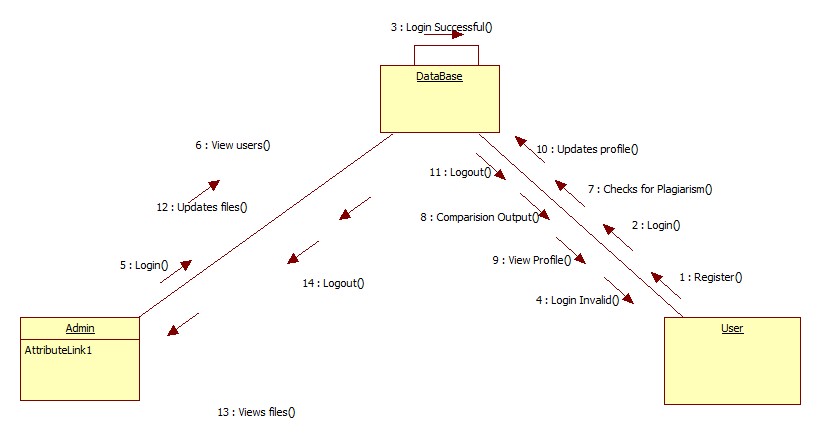


**Sequence Diagram**

A sequence diagram is an interaction diagram. From the name it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another. Interaction among the components of a system is very important from implementation and execution perspective. So Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

**Collaboration Diagram**

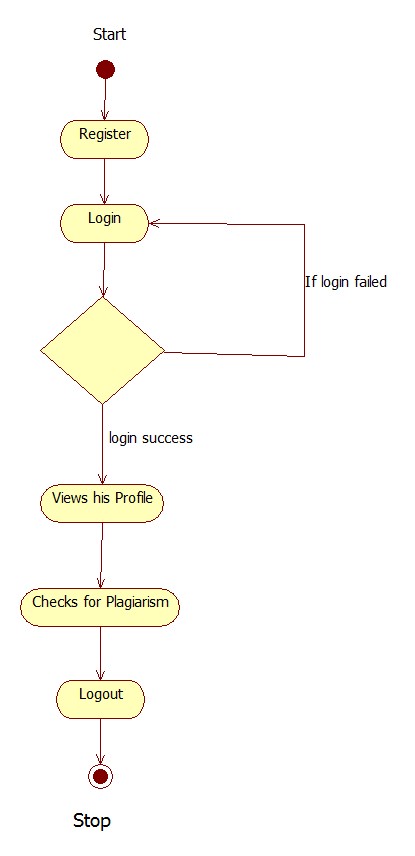
Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links. The purpose of collaboration diagram is similar to sequence diagram. But the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.



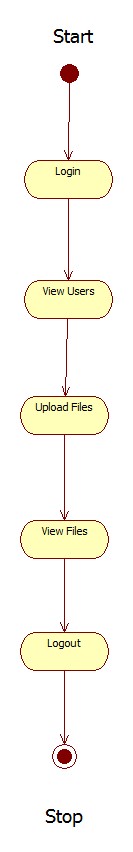
**Activity Diagram**

Activity diagram describes the flow of control in a system. So it consists of activities and links. The flow can be sequential, concurrent or branched. Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system. Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

**Admin Activity Diagram**



**User Activity Diagram**

****

**5. IMPLEMENTATION**

**.Net Framework:-**

The purpose of testing is a technology that supports building and running the next generation of applications and XML Web services. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

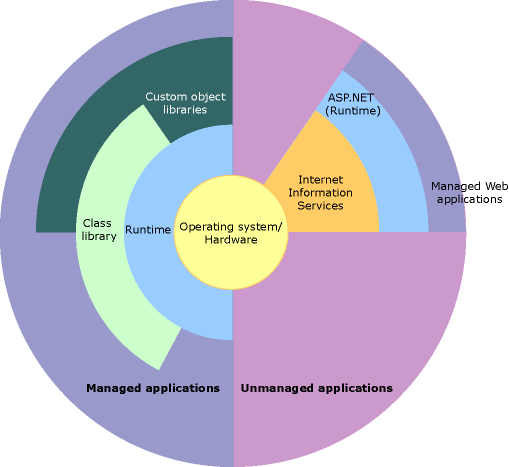
The .NET Framework consists of the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable ASP.NET applications and XML Web services.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Windows Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the overall system. The illustration also shows how managed code operates within a larger architecture.



*NET Framework in context*

## Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type-and-code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft SQL Server and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

## .NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

* Console applications.
* Windows GUI applications (Windows Forms).
* Windows Presentation Foundation (WPF) applications.
* ASP.NET applications.
* Windows services.
* Service-oriented applications using Windows Communication Foundation (WCF).
* Workflow-enabled applications using Windows Workflow Foundation (WF).

**Visual C#**

C# is an elegant and type-safe object-oriented language that enables developers to build a variety of secure and robust applications that run on the .NET Framework. You can use C# to create Windows client applications, XML Web services, distributed components, client-server applications, database applications, and much, much more. Visual C# provides an advanced code editor, convenient user interface designers, integrated debugger, and many other tools to make it easier to develop applications based on the C# language and the .NET Framework.

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 nullable 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 iterators, which enable implementers of collection classes to define custom iteration behaviors that are simple to use by client code. Language-Integrated Query (LINQ) expressions make the strongly-typed query a first-class language construct.

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. Methods that override virtual methods in a parent class require the **override** keyword as a way to avoid accidental redefinition. In C#, a struct is like a lightweight class; it is a stack-allocated type that can implement interfaces but does not support inheritance.

In addition to these basic object-oriented principles, C# makes it easy to develop software components through several innovative language constructs, including the following:

* Encapsulated method signatures called *delegates*, which enable type-safe event notifications.
* Properties, which serve as accessors for private member variables.
* Attributes, which provide declarative metadata about types at run time.
* Inline XML documentation comments.
* Language-Integrated Query (LINQ) which provides built-in query capabilities across a variety of data sources.

If you have to interact with other Windows software such as COM objects or native Win32 DLLs, you can do this in C# through a process called "Interop." Interop enables C# programs to do almost anything that a native C++ application can do. C# even supports pointers and the concept of "unsafe" code for those cases in which direct memory access is absolutely critical.

The C# build process is simple compared to C and C++ and more flexible than in Java. There are no separate header files, and no requirement that methods and types be declared in a particular order. A C# source file may define any number of classes, structs, interfaces, and events.

**.NET Framework Platform Architecture for C#**

C# programs run on the .NET Framework, an integral component of Windows that includes a virtual execution system called the common language runtime (CLR) and a unified set of class libraries. The CLR is the commercial implementation by Microsoft of the common language infrastructure (CLI), an international standard that is the basis for creating execution and development environments in which languages and libraries work together seamlessly.

Source code written in C# is compiled into an intermediate language (IL) that conforms to the CLI specification. The IL code and resources, such as bitmaps and strings, are stored on disk in an executable file called an assembly, typically with an extension of .exe or .dll.

When the C# program is executed, the assembly is loaded into the CLR, which might take various actions based on the information in the manifest. Then, if the security requirements are met, the CLR performs just in time (JIT) compilation to convert the IL code to native machine instructions. The CLR also provides other services related to automatic garbage collection, exception handling, and resource management. The following diagram illustrates the compile-time and run-time relationships of C# source code files, the .NET Framework class libraries, assemblies, and the CLR.



*Architecture of C# in .Net*

**ASP.Net**

ASP.NET is a unified Web development model that includes the services necessary for you to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications you have access to classes in the .NET Framework. You can code your applications in any language compatible with the common language runtime (CLR), including Microsoft Visual Basic and C#. These languages enable you to develop ASP.NET applications that benefit from the common language runtime, type safety, inheritance, and so on.

If you want to try ASP.NET, you can install Visual Web Developer Express using the Microsoft Web Platform Installer, which is a free tool that makes it simple to download, install, and service components of the Microsoft Web Platform. These components include Visual Web Developer Express, Internet Information Services (IIS), SQL Server Express, and the .NET Framework. All of these are tools that you use to create ASP.NET Web applications. You can also use the Microsoft Web Platform Installer to install open-source ASP.NET and PHP Web applications.

**The Three Flavors of ASP.NET: Web Forms, MVC, and Web Pages**

ASP.NET offers three frameworks for creating web applications: ASP.NET Web Forms, ASP.NET MVC, and ASP.NET Web Pages. All three frameworks are stable and mature, and you can create great web applications with any of them.

Each framework targets a different audience or type of application. Which one you choose depends on a combination of your web development experience, what framework you’re most comfortable with, and which is the best fit for the type of application you’re creating. All three frameworks will be supported, updated, and improved in future releases of ASP.NET.

**ASP.NET Web Forms (.aspx pages)**

The Web Forms framework targets developers who prefer declarative and control-based programming, such as Microsoft Windows Forms (WinForms) and WPF/XAML/Silverlight. It offers a WYSIWYG designer-driven (drag-and-drop) development model, so it's popular with developers looking for a rapid application development (RAD) environment for web development. If you’re new to web programming and are familiar with the traditional Microsoft RAD client development tools (for example, for Visual Basic and Visual C#), you can quickly build a web application without having expertise in HTML and JavaScript.

Web Forms works well for small teams of Web developers and designers who want to take advantage of the large number of components available for rapid application development. In general, creating a Web Forms application requires less programming effort than creating the same application by using the ASP.NET MVC framework. The components (the [Page](https://msdn.microsoft.com/en-us/library/system.web.ui.page.aspx) class, controls, and so on) are tightly integrated and usually require less code than ASP.NET MVC applications. However, Web Forms is not just for rapid application development. There are many complex commercial apps and app frameworks built on top of Web Forms.

**ASP.NET MVC**

ASP.NET MVC targets developers who are interested in patterns and principles like 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 markup for the presentation layer — developers can work on the business logic while designers work on the markup 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 WinForms 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.

MVC was designed to be extensible, providing power developers the ability to customize the framework for their application needs. In addition, the ASP.NET MVC source code is available under an [OSI license](http://www.opensource.org/licenses).

**ASP.NET Web Pages (.cshtml and .vbhtml files)**

ASP.NET Web Pages targets developers who want a simple web development story, along the lines of PHP. In the Web Pages model, you create HTML pages and then add server-based code to the page in order to dynamically control how that markup is rendered. Web Pages is specifically designed to be a lightweight framework, and it's the easiest entry point into ASP.NET for people who know HTML but might not have broad programming experience — for example, students or hobbyists. It's also a good way for web developers who know PHP or similar frameworks to start using ASP.NET.

Like Web Forms, Web Pages is oriented toward rapid development. Web Pages provides components called helpers that you can add to pages and that let you use just a few lines of code to perform tasks that would either be tedious or complex. For example, there are helpers to display database data, add a Twitter feed, log in using Facebook, add maps to a page, and so on.

**ASP.NET Compiler**

All ASP.NET code is compiled, which enables strong typing, performance optimizations, and early binding, among other benefits. Once the code has been compiled, the common language runtime further compiles ASP.NET code to native code, providing improved performance.

ASP.NET includes a compiler that will compile all your application components including pages and controls into an assembly that the ASP.NET hosting environment can then use to service user requests.

**SQL SERVER**

SQL Server is a client / server Relational Database Management System (RDBMS) which has been developed by Microsoft and is made up several different client and server programs that make up the entire product.

**SQL Server Database Engine (Server)**

SQL Server's database engine is the primary server application of the SQL server package. Its main functions are:

* Provide reliable storage
* Rows of data are stored as pages, each 8kb in size.
* Provide a means to rapidly access the data; this is done through utilizing indexes both clustered and non-clustered to search for data, which removes the need for all data to be scanned from the database tables.
* Ensure consistent access to the data; Consistent access basically means only allowing one client to modify/changing the data at any one time
* Implement security; Microsoft SQL Server has multiple security levels Server Level, database level and database object level. Access to the server can be controlled by a Username or password or through Windows security in LAN/Networked environment.
* Enforce data integrity; ensure the data stays consistent.

The ITS Systems Database team offers standards-based, professionally managed SQL Server database systems including hardware, software, and system administration for University of Texas customers. The Microsoft SQL Server database servers may be used for websites, commercial applications, and custom applications designed and built by the customer. The central Microsoft SQL Server database servers are shared to minimize cost and ensure efficient use of the service resources. This service provides three database environments to support the full application development life cycle. The included environments are Development, Quality Assurance (QA), and Production. This service includes database/system administration, database backups and recovery, and monitoring.

**Features**

* Microsoft SQL Server 2012 environment with Always On Availability Group configuration
* 10 gigabytes of storage space on the central Microsoft SQL Server database servers
* Daily production backups (at minimum) with a four-week retention
* Ability to safely store [Confidential Data](http://security.utexas.edu/policies/extended-cat-1) in databases
* Access to the ITS Systems Microsoft SQL Server team for advice and troubleshooting

In the Fully Managed infrastructure, ITS owns, manages, and supports the SQL Server hardware and host server software resources. The Fully Managed ITS shared SQL Server service is recommended for departments, colleges, or other groups that require SQL Server database service. ITS provides the hardware, software, and staff resources to manage the central SQL Server infrastructure. It is also recommended for those who may be using Confidential Data in their databases.

**6. CODING**

**Code for Registration Form (User):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

public partial class Register : System.Web.UI.Page

{

Class1 o = new Class1();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

string qry = "insert into Article\_Comparer values('" + TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + TextBox6.Text + "','" + RadioButtonList1.SelectedItem.Text + "','" + TextBox7.Text + "','" + DropDownList1.SelectedItem.Text + "')";

int i = o.inupdel(qry);

if (i > 0)

{

Response.Write("<script>alert('Registeration Successful')</script>");

TextBox2.Text = TextBox3.Text = TextBox4.Text = TextBox5.Text = TextBox6.Text = RadioButtonList1.SelectedItem.Text = TextBox7.Text = DropDownList1.SelectedItem.Text = " ";

}

else

{

Response.Write("<script>alert('Registeration Unsuccesful')</script>");

}

}

catch (Exception exe)

{

Response.Write("<script>alert('" + exe.Message + "')</script>");

}

}

}

**Code for Login Form (User):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Data;

public partial class Login : System.Web.UI.Page

{

Class1 o = new Class1();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

if (DropDownList1.SelectedItem.Text == "User")

{

string qry = "select \* from Article\_Comparer where Name='"+TextBox1.Text+"' and Password='"+TextBox2.Text+"'";

DataSet ds = o.select(qry);

if (ds.Tables[0].Rows.Count > 0)

{

Session["id"] = ds.Tables[0].Rows[0][0].ToString();

Session["name"] = ds.Tables[0].Rows[0][1].ToString();

Session["pswd"] = ds.Tables[0].Rows[0][7].ToString();

Response.Redirect("~/User/UserHome.aspx");

}

else

{

Response.Write("<script>alert('Invalid UserName or Password !!')</script>");

}

}

**Code for Profile & Update (User):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

public partial class User\_Profile : System.Web.UI.Page

{

Class1 o = new Class1();

protected void Page\_Load(object sender, EventArgs e)

{

if (!IsPostBack)

{

load();

}

}

public void load()

{

try

{

string qry2 = "select \* from Article\_Comparer where Id='" + Session["id"].ToString() + "'";

DataSet ds = o.select(qry2);

if (ds.Tables[0].Rows.Count > 0)

{

TextBox2.Text = ds.Tables[0].Rows[0][1].ToString();

TextBox3.Text = ds.Tables[0].Rows[0][2].ToString();

TextBox4.Text = ds.Tables[0].Rows[0][3].ToString();

TextBox5.Text = ds.Tables[0].Rows[0][4].ToString();

TextBox6.Text = ds.Tables[0].Rows[0][5].ToString();

TextBox7.Text = ds.Tables[0].Rows[0][7].ToString();

TextBox8.Text = ds.Tables[0].Rows[0][8].ToString();

}

else

{

Response.Write("<script>alert('There is No Data')</script>");

}

}

catch (Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "')</script>");

}

}

protected void Button2\_Click(object sender, EventArgs e)

{

try

{

string qry = "update Article\_Comparer set Address='" + TextBox3.Text + "',Mobile='" + TextBox4.Text + "',Email='" + TextBox5.Text + "',D\_O\_B='"+TextBox6.Text+"',Password='"+TextBox7.Text+"',State='"+TextBox8.Text+"' where Id='" + Session["id"].ToString() + "'";

int i = o.inupdel(qry);

if (i > 0)

{

Response.Write("<script>alert('Updated Succesfully')</script>");

}

}

catch (Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "')</script>");

}

}

protected void LinkButton2\_Click(object sender, EventArgs e)

{

TextBox3.ReadOnly = false;

}

protected void LinkButton1\_Click(object sender, EventArgs e)

{

TextBox4.ReadOnly = false;

}

protected void LinkButton5\_Click(object sender, EventArgs e)

{

TextBox5.ReadOnly = false;

}

protected void LinkButton3\_Click(object sender, EventArgs e)

{

TextBox7.ReadOnly = false;

}

protected void LinkButton4\_Click(object sender, EventArgs e)

{

TextBox8.ReadOnly = false;

}

}

**Code for Article Comparision (User):**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

using System.IO;

using System.Text.RegularExpressions;

using Microsoft.Office.Interop.Word;

public partial class User\_CheckPlagiarism : System.Web.UI.Page

{

public float co, co1;

Class1 obj = new Class1();

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

string txt = System.IO.Path.GetExtension(FileUpload1.PostedFile.FileName);

if (txt.ToUpper() == ".TXT")

{

if (FileUpload1.HasFile)

{

string jag = "";

string jag1 = "";

string fname = "~/Files/" + FileUpload1.FileName;

string ss = FileUpload1.FileName;

FileUpload1.SaveAs(Server.MapPath(fname));

string mystring = File.ReadAllText(Server.MapPath("~/Files/" + FileUpload1.FileName));

int count = 0;

string kk = Regex.Matches(mystring.Trim(), "[a-zA-Z]").Count.ToString();

float u12 = Convert.ToInt64(kk);

//int u13 = mystring.Trim().Count();

//float u12 = (float)u13;

foreach (char c in mystring)

{

if (char.IsLetter(c))

{

count++;

}

}

string u11 = count.ToString();

int u = Convert.ToInt32(u11);

string o = "";

string[] sentences1 = Regex.Split(mystring.Trim(), @"(?<=[\.!\?])+");

foreach (string sentence2 in sentences1)

{

o = sentence2;

string x = "";

string qry = "select Content1 from FileUpload";

DataSet ds = obj.select(qry);

int h = ds.Tables[0].Rows.Count;

for (int i = 0; i < h; i++)

{

x += " " + ds.Tables[0].Rows[i][0].ToString();

}

string j = "";

string[] sentences = Regex.Split(x.Trim(), @"(?<=[\.!\?])+");

foreach (string sentence in sentences)

{

j = sentence;

if (o.Trim() == j.Trim())

{

jag += " " + o.ToString();

//co =Convert.ToInt16( jag.Trim().Count());

string kk1 = Regex.Matches(jag.Trim(), "[a-zA-Z]").Count.ToString();

co1 = Convert.ToInt64(kk1);

}

else

{

jag1 += "" + o.ToString();

}

// TextBox2.Text = jag1;

}

float j7 = Convert.ToInt64((co1 / u12) \* 100);

//int j7 = (co / u13) ;

TextBox1.Text = jag;

lblper.Text = j7.ToString();

}

}

else

{

Response.Write("<script>alert('Please Upload The File !!!!')</script>");

}

}

else if (txt.ToUpper() == ".DOCX" || txt.ToUpper() == ".DOC")

{

if (FileUpload1.HasFile)

{

string fname = "~/temp/" + FileUpload1.FileName;

string ss = FileUpload1.FileName;

FileUpload1.SaveAs(Server.MapPath(fname));

ApplicationClass wordApp = new ApplicationClass();

object file = Server.MapPath(fname);

object nullobj = System.Reflection.Missing.Value;

Document doc = wordApp.Documents.Open(ref file,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj);

Document doc1 = wordApp.ActiveDocument;

string mystring = doc1.Content.Text;

doc.Close(ref nullobj, ref nullobj, ref nullobj);

string jag = "";

string jag1 = "";

//string mystring = File.ReadAllText(Server.MapPath("~/temp/" + FileUpload1.FileName));

int count = 0;

string kk = Regex.Matches(mystring.Trim(), "[a-zA-Z]").Count.ToString();

float u12 = Convert.ToInt64(kk);

//int u13 = mystring.Trim().Count();

//float u12 = (float)u13;

foreach (char c in mystring)

{

if (char.IsLetter(c))

{

count++;

}

}

string u11 = count.ToString();

int u = Convert.ToInt32(u11);

string o = "";

string[] sentences1 = Regex.Split(mystring.Trim(), @"(?<=[\.!\?])+");

foreach (string sentence2 in sentences1)

{

o = sentence2;

string x = "";

string qry = "select Content1 fromFileUpload";

DataSet ds = obj.select(qry);

int h = ds.Tables[0].Rows.Count;

for (int i = 0; i < h; i++)

{

x += " " + ds.Tables[0].Rows[i][0].ToString();

}

string j = "";

string[] sentences = Regex.Split(x.Trim(), @"(?<=[\.!\?])+");

foreach (string sentence in sentences)

{

j = sentence;

if (o.Trim() == j.Trim())

{

jag += " " + o.ToString();

//co =Convert.ToInt16( jag.Trim().Count());

string kk1 = Regex.Matches(jag.Trim(), "[a-zA-Z]").Count.ToString();

co1 = Convert.ToInt64(kk1);

}

else

{

jag1 += "" + o.ToString();

}

// TextBox2.Text = jag1;

}

float j7 = Convert.ToInt64((co1 / u12) \* 100);

//int j7 = (co / u13) ;

TextBox1.Text = jag;

lblper.Text = j7.ToString();

}

}

else

{

Response.Write("<script>alert('Please Upload The File !!!!')</script>");

}

}

else

{

Response.Write("<script>alert('Please Upload The File !!!!')</script>");

}

}

catch (Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "')</script>");

}

}

}

**Code for Uploading Files (Admin):**

using Microsoft.Office.Interop.Word;

using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

public partial class Admin\_UpdateFiles : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

Class1 obj=new Class1();

protected void Button1\_Click(object sender, EventArgs e)

{

try

{

string txt = System.IO.Path.GetExtension(FileUpload1.PostedFile.FileName);

if (txt.ToUpper() == ".TXT")

{

if (FileUpload1.HasFile)

{

string filename1 = "~/Files/" + FileUpload1.FileName;

FileUpload1.SaveAs(Server.MapPath(filename1));

string f1 = FileUpload1.FileName;

string sr = File.ReadAllText(Server.MapPath("~/Files/" + FileUpload1.FileName));

if (sr != null)

{

string qry1 = "select Content1 from FileUpload";

DataSet ds = obj.select(qry1);

if (ds != null)

{

int h = ds.Tables[0].Rows.Count;

int dm = 0;

Boolean am = true;

for (int i = 0; i < h; i++)

{

string a = ds.Tables[0].Rows[i][0].ToString();

if (a.ToString() == sr.ToString())

{

Response.Write("<script>alert(' Already Exit !!!')</script>");

dm++;

am = false;

}

}

if (dm != 1)

if (am != false)

{

string s1 = sr.ToString();

string qry = "insert into FileUpload values('" + f1.ToString() + "','" + filename1.ToString() + "','" + TextBox1.Text + "','" + s1 + "')";

int j = obj.inupdel(qry);

if (j > 0)

{

Response.Write("<script>alert('File Uploaded Succesfully')</script>");

}

else

{

Response.Write("<script>alert('Not Yet Uploaded !!!')</script>");

}

//TextBox1.Text = sr.ToString();

//string qry = "insert into Dedup values('" + ss + "','" + TextBox1.Text + "')";

//int x = obj.inupdel(qry);

}

}

}

}

}

else if (txt.ToUpper() == ".DOCX" || txt.ToUpper() == ".DOC")

{

if (FileUpload1.HasFile)

{

string filename2 = "~/Files/" + FileUpload1.FileName;

FileUpload1.SaveAs(Server.MapPath(filename2));

string f2 = FileUpload1.FileName;

ApplicationClass wordApp = new ApplicationClass();

object file = Server.MapPath(filename2);

object nullobj = System.Reflection.Missing.Value;

Document doc = wordApp.Documents.Open(ref file,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj,

ref nullobj);

Document doc1 = wordApp.ActiveDocument;

string sr1 = doc1.Content.Text;

if (sr1 != null)

{

string qry1 = "select Content1 from FileUpload";

DataSet ds = obj.select(qry1);

if (ds != null)

{

int h = ds.Tables[0].Rows.Count;

int dm = 0;

Boolean am = true;

for (int i = 0; i < h; i++)

{

string a = ds.Tables[0].Rows[i][0].ToString();

if (a.ToString() == sr1.ToString())

{

Response.Write("<script>alert(' Already Exit !!!')</script>");

dm++;

am = false;

}

}

if (dm != 1)

if (am != false)

{

string s1 = sr1.ToString();

string qry = "insert into FileUpload values('" + f2.ToString() + "','" + filename2.ToString() + "','" + TextBox1.Text + "','" + s1 + "')";

int j = obj.inupdel(qry);

if (j > 0)

{

Response.Write("<script>alert('File Uploaded Succesfully')</script>");

}

else

{

Response.Write("<script>alert('Not Yet Uploaded !!!')</script>");

}

//TextBox1.Text = sr.ToString();

//string qry = "insert into Dedup values('" + ss + "','" + TextBox1.Text + "')";

//int x = obj.inupdel(qry);

}

} }

catch (Exception ex)

{

Response.Write("<script>alert('" + ex.Message + "')</script>");

}

} }

**7. TESTING**

**7.1 Introduction**

The purpose of testing is to discover errors in the system. Testing is the process of trying to discover every conceivable fault or weakness in a system by implementing using testing strategies. A process of executing a program with the explicit intention of finding errors, that is making the program fail. Testing is the process of detecting errors. Testing performs a very critical role for quality assurance and for ensuring the reliability of software. The results of testing are used later on during maintenance also. Psychology of Testing.

The aim of testing is often to demonstrate that a program works by showing that it has no errors. The basic purpose of testing phase is to detect the errors that may be present in the program. Hence one should not start testing it the intent of showing that a program works but the intent should be to show that a program does not work. Testing is a process of executing a program in order to find out the errors in the program.

**7.2 TESTING STRATEGIES**

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. System Testing

**7.2.1 UNIT TESTING**

Unit Testing is done on individual modules in our project as they are completed and become executable. It is confined only to the designer's requirements.

Each module can be tested using the following two Strategies:

**Black Box Testing:**

In our system the black box testing can be implemented by testing the compilation and run-time errors.

**White Box testing:**

In our system the test-cases for each case are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. White box testing has been:

* Every logical decision is found by checking the validations “please enter valid password” “enter valid details”.
* In our project the each if loop is checked weather it is satisfying its needs.

**7.2.2 INTEGRATION TESTING**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**7.2.3 VALIDATION TESTING**

The system can be validated; whenever the input is given into the system it checks the validations for each module.Since, our system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

* Enter the correct password.
* Enter the valid details.
* Fill all the fields etc.

**7.2.4 SYSTEM TESTING**

At last our system is tested that each module is checked before delivery to the user. Our aim is to satisfy the user the system meets all requirements of the client's specifications.

**7.3 TEST CASES**

The test-cases are used to validate the each module every time whenever user enters the input into the system then we can provide the results either pass or fail with expected result and actual result. Clinically defined (IEEE 829-1998) a test case is an input and an expected result. This can be as pragmatic as 'for condition x your derived result is y', whereas other test cases described in more detail the input scenario and what results might be expected. It can occasionally be a series of steps (but often steps are contained in a separate test procedure that can be exercised against multiple test cases, as a matter of economy) but with one expected result or expected outcome. The optional fields are a test case ID, test step or order of execution number, related requirement(s) depth, test category, author, and check boxes for whether the test is automatable and has been automated. Larger test cases may also contain prerequisite states or steps, and descriptions. A test case should also contain a place for the actual result. These steps can be stored in a word processor document, spreadsheet, database or other common repository. In a database system, you may also be able to see past test results and who generated the results and the system configuration used to generate those results. These past results would usually be stored in a separate table.

**Test Case Reports:**

**Test1:**

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | ACTUAL OUTPUT | CONDITION |
| If user uploads files for checking | Same content & copy percentage | Same content & copy percentage | Pass |
| If all login and registration are valid. | Login & Registration successful. | Login & Registration successful. | Pass |
| If Admin upload files. | Uploaded Successfully. | Uploaded Successfully. | Pass |

T**able: 7.7.1 Positive Test Cases for Article Comparer in .NET**

**Test2:**

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | ACTUAL OUTPUT | CONDITION |
| If user doesn’t enter the login details correctly. | Login successful | Invalid user name or password | fail |
| If user doesn’t upload file for checking. | copy percentage | Please upload file. | fail |
| If registration details are incorrect | Registration successful | Registration Unsuccessful | Fail |

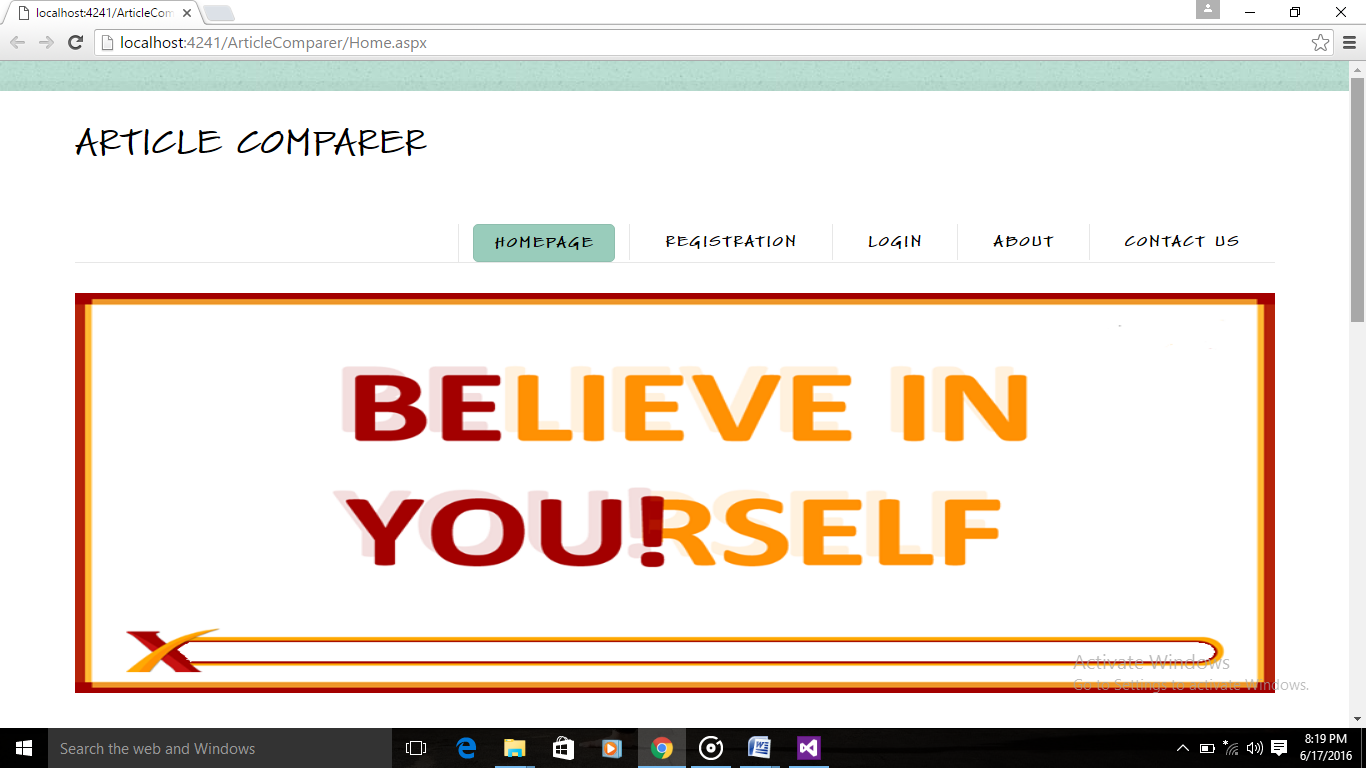
**Table: .7.2 Negative Test Cases for a Article Comparer in .Net**

All the above validations on table have verified and they are successfully executed. The flow is tested in different possible conditions of testing stages and all conditions are modified

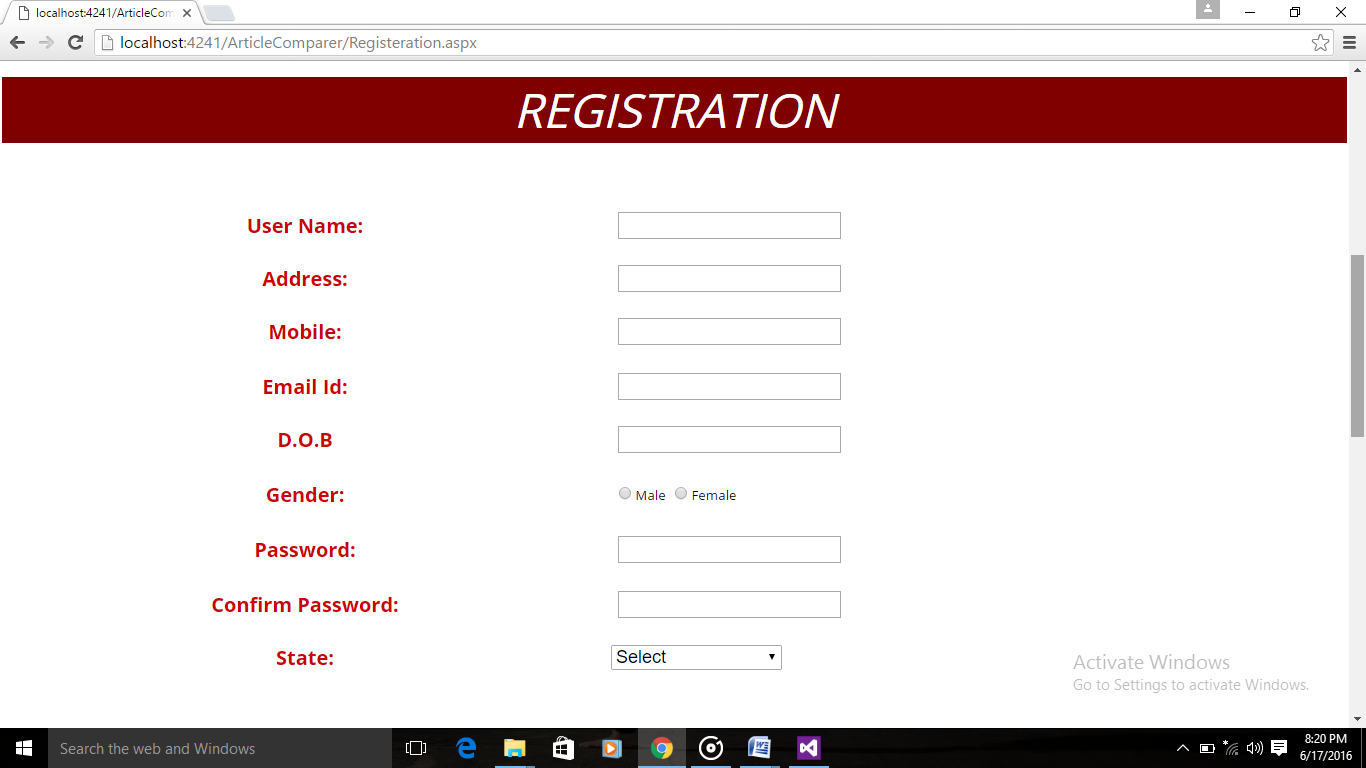
when the condition failed in the process of testing stage and all the bugs in application has modified in the testing.

**8. OUTPUT SCREENS**

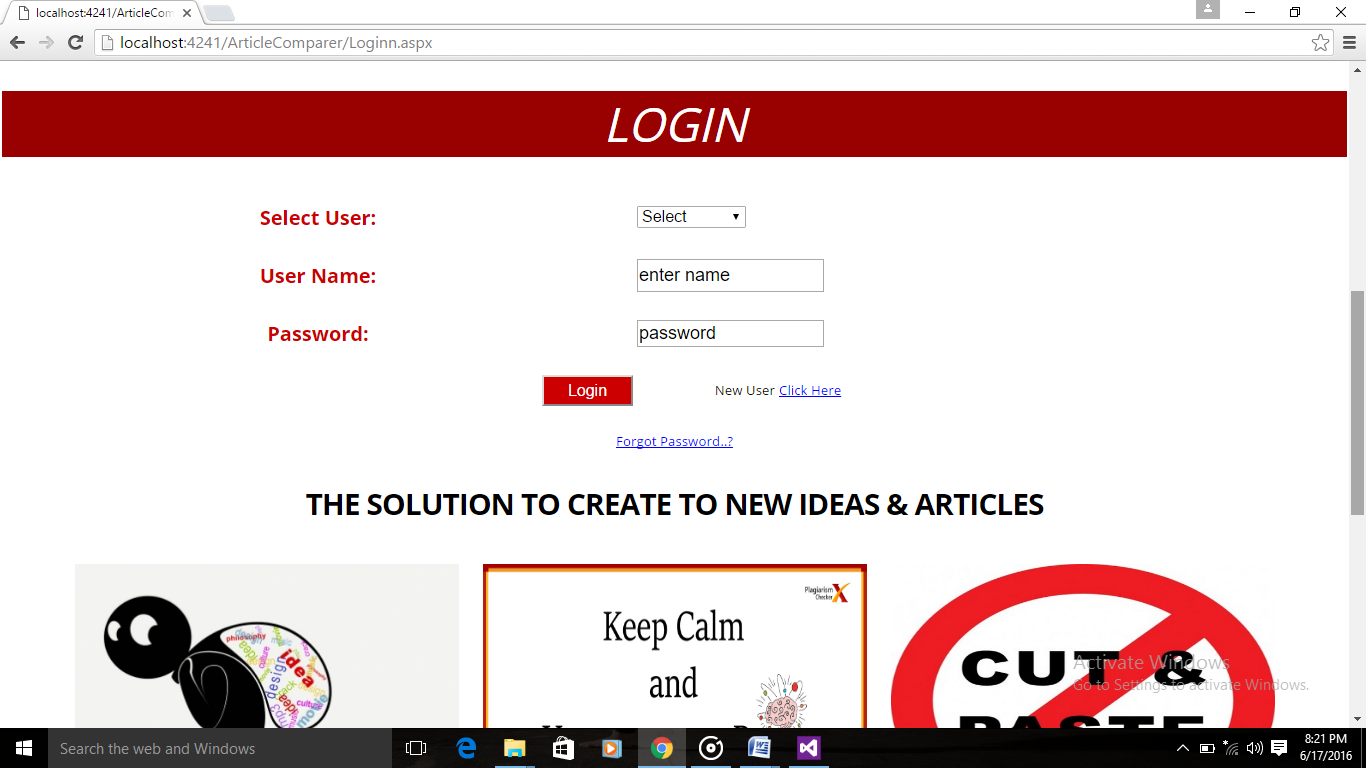
**Home Page:**

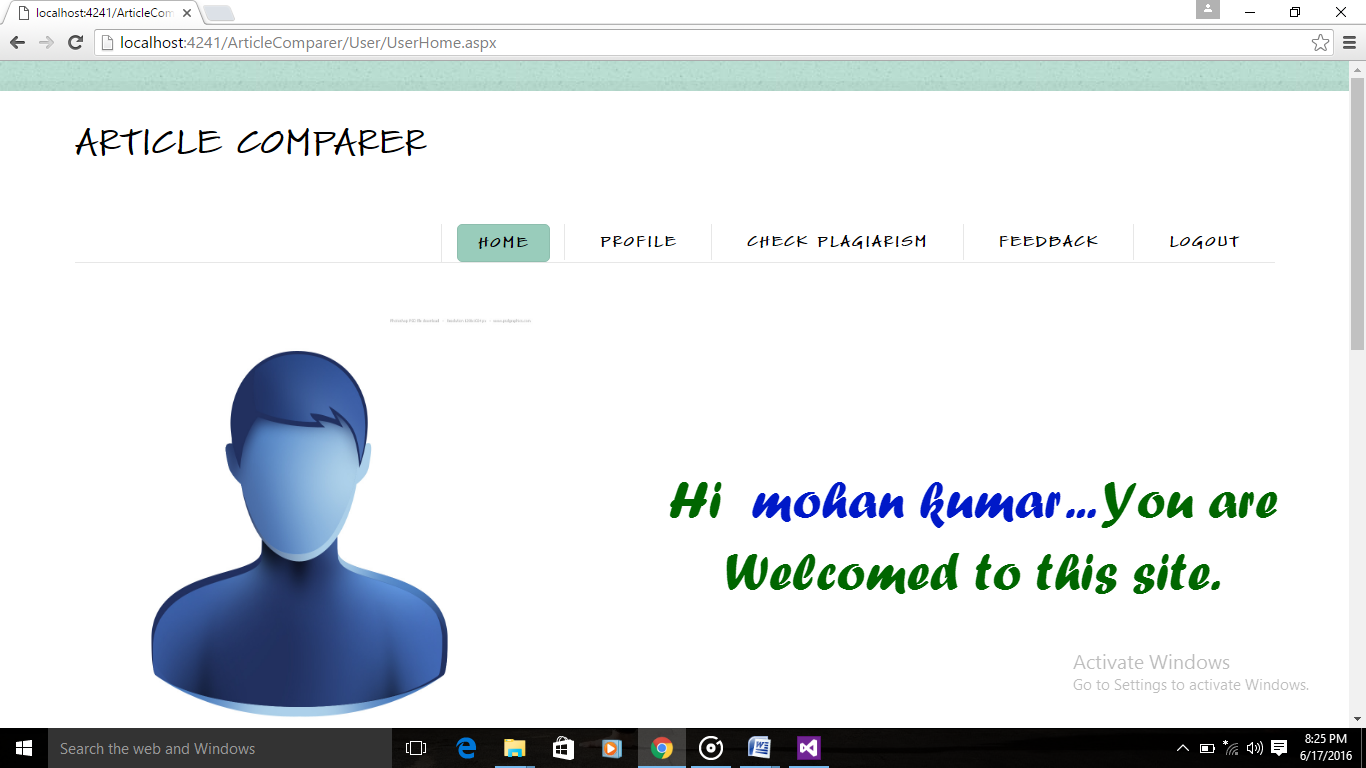
****

**Registration:**

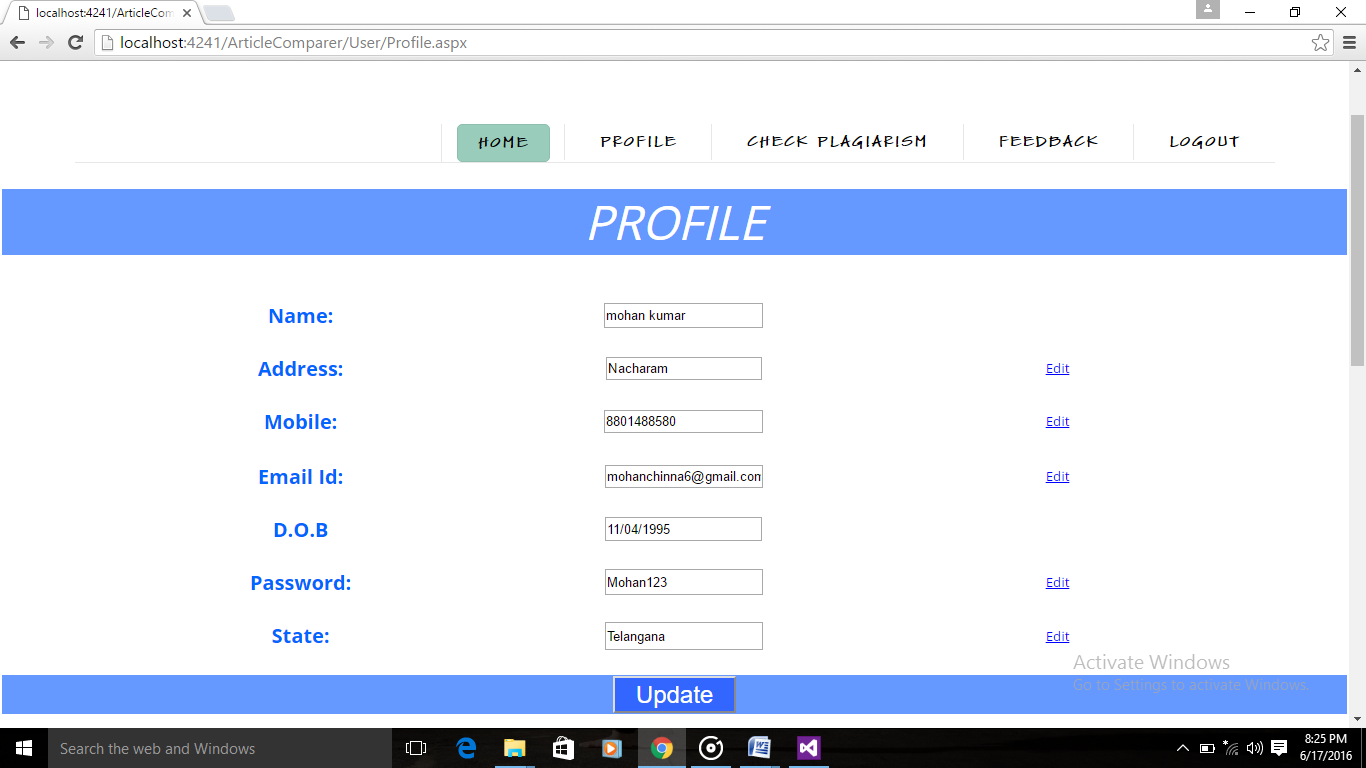
****

**Login:**

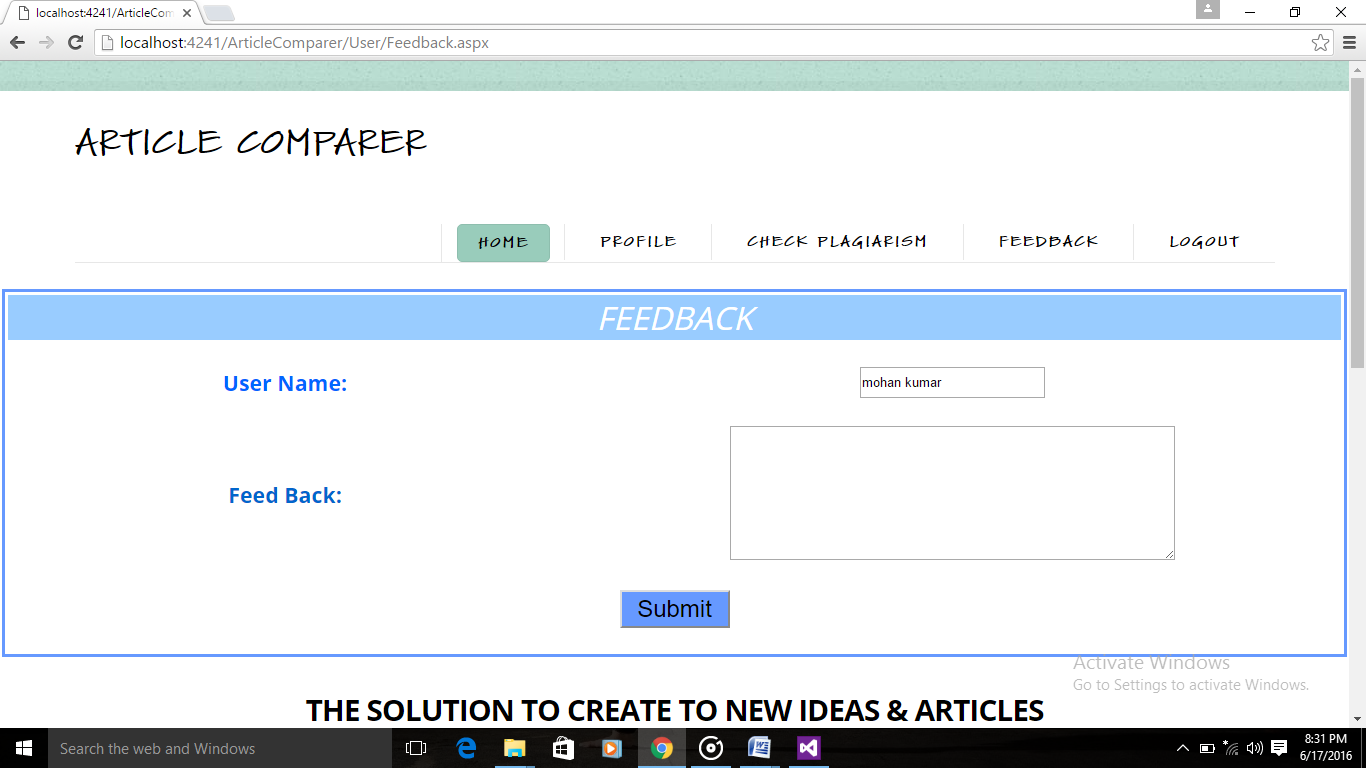
****

**User Home:**

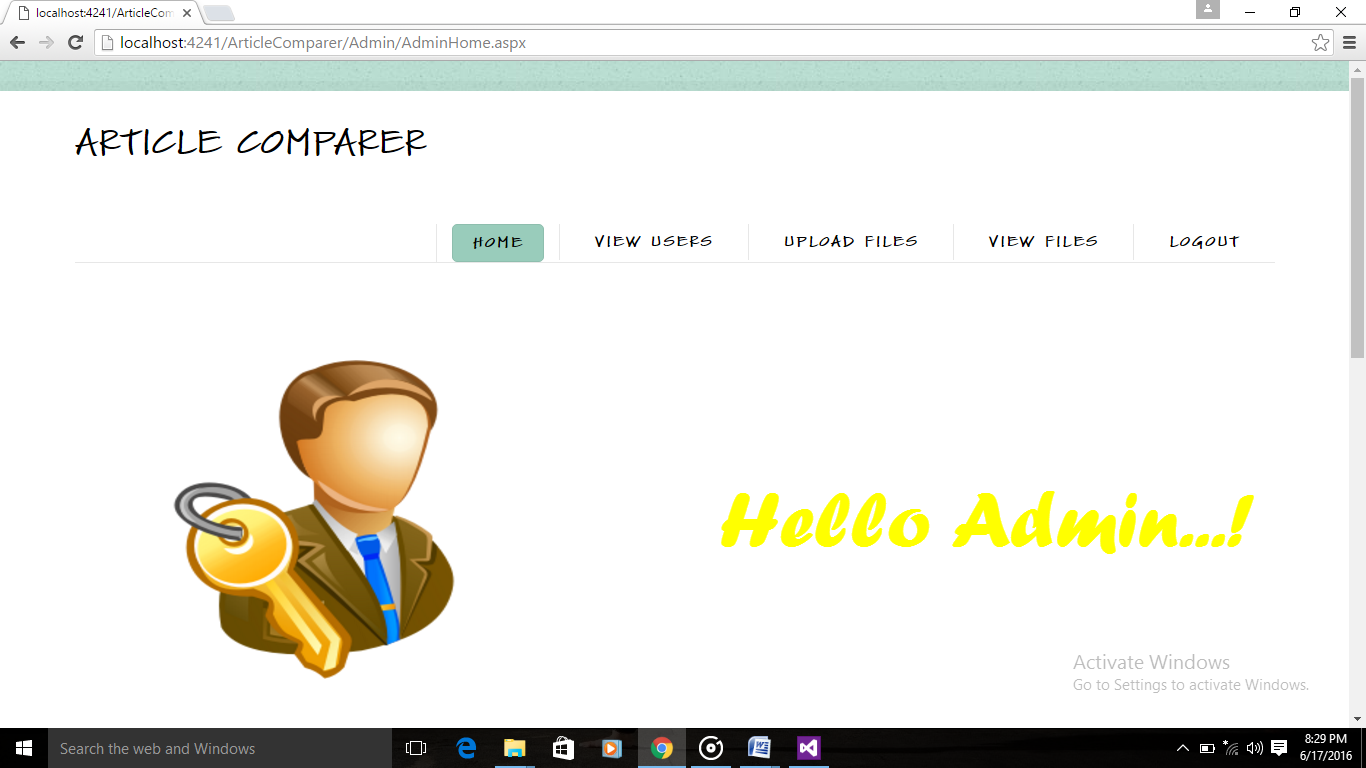
**User Profile:**

****

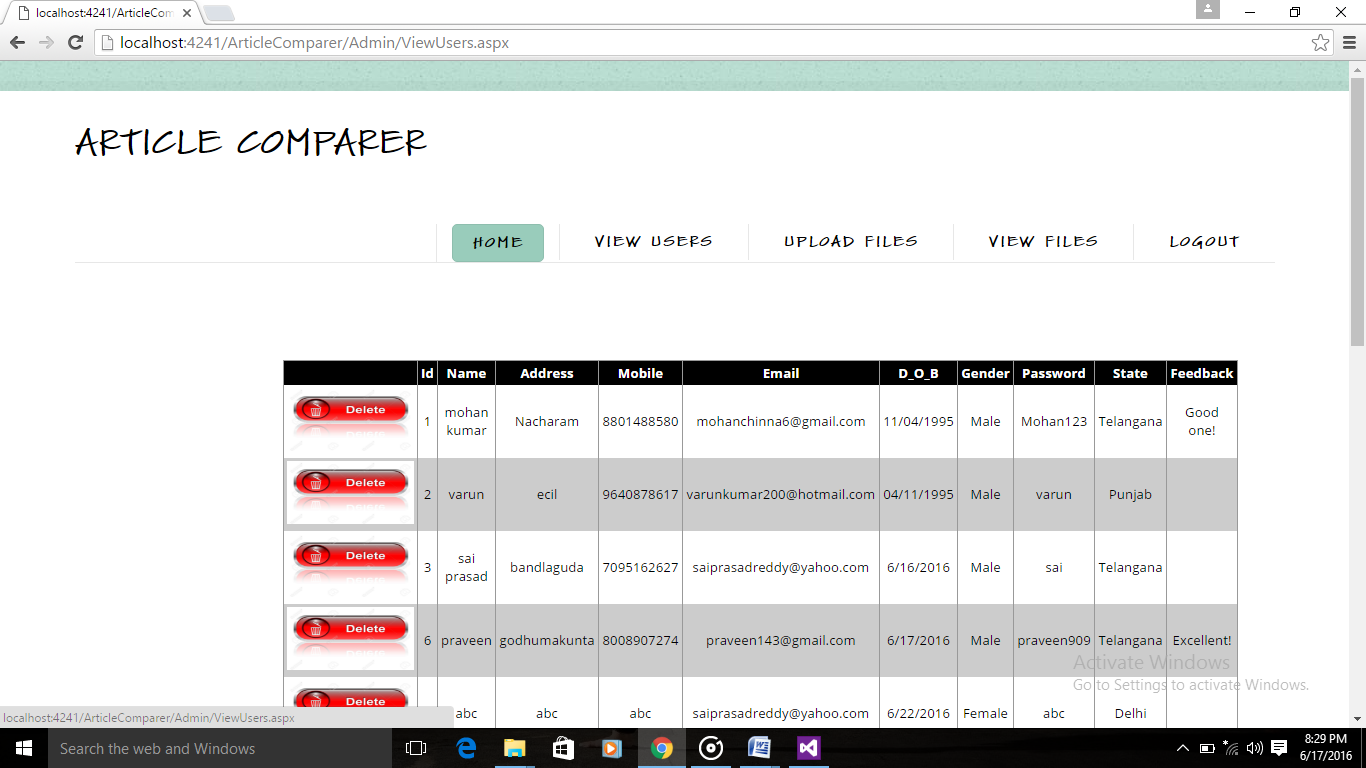
**Feedback of User:**

****

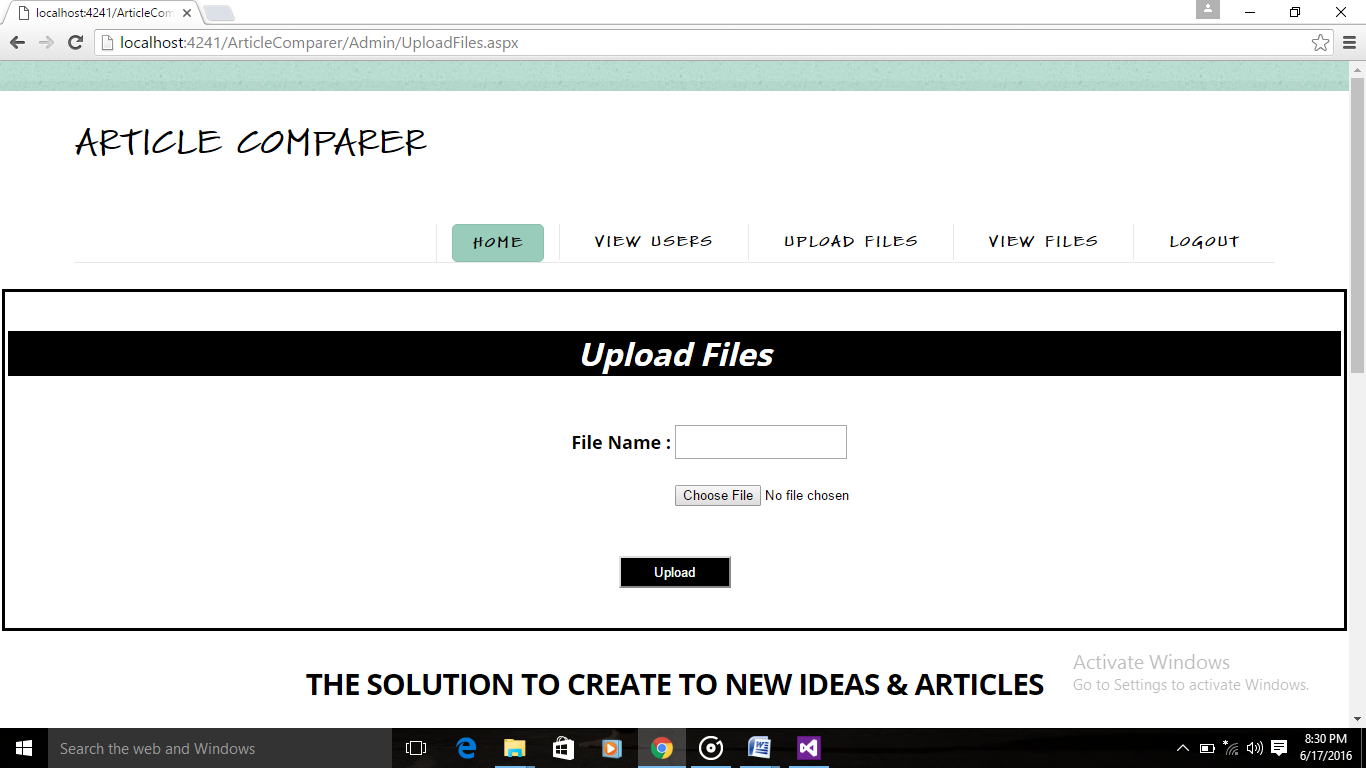
**Admin Home:**

****

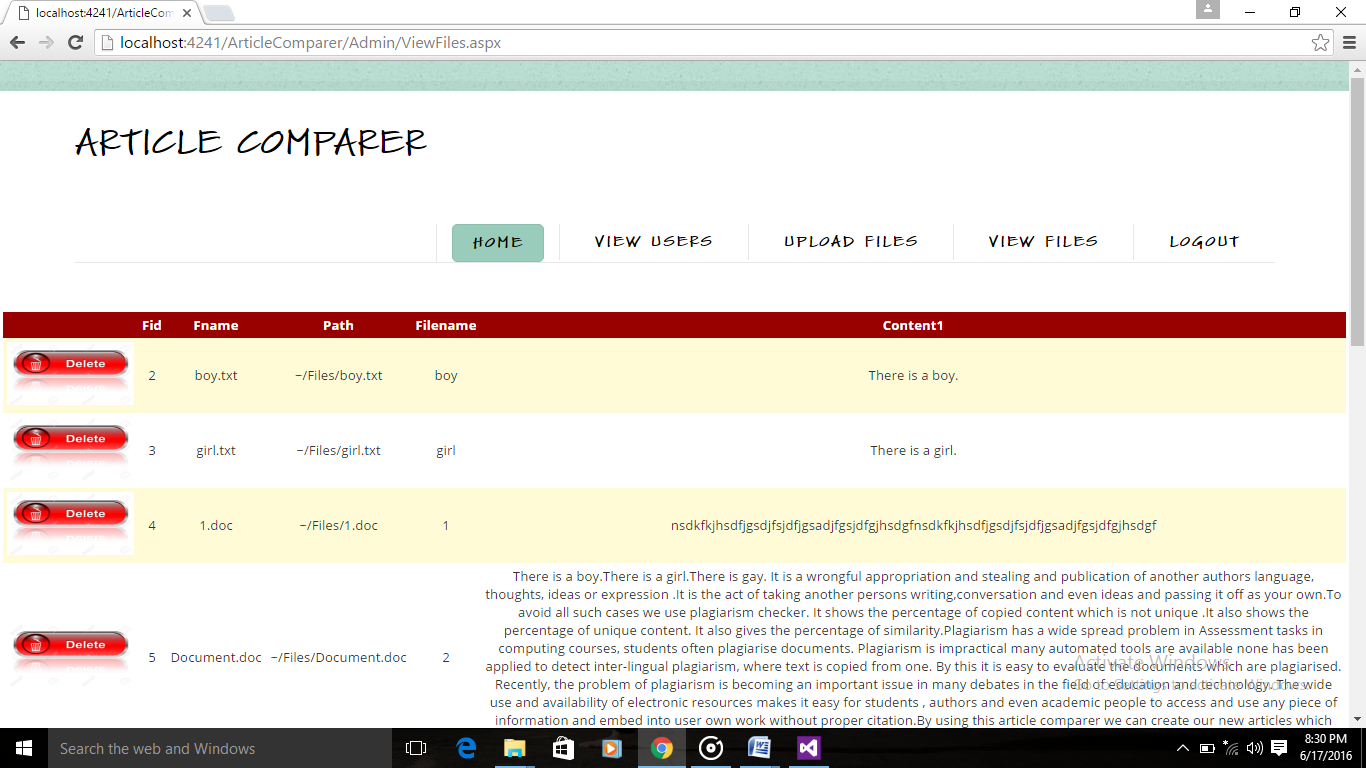
**View Users:**

****

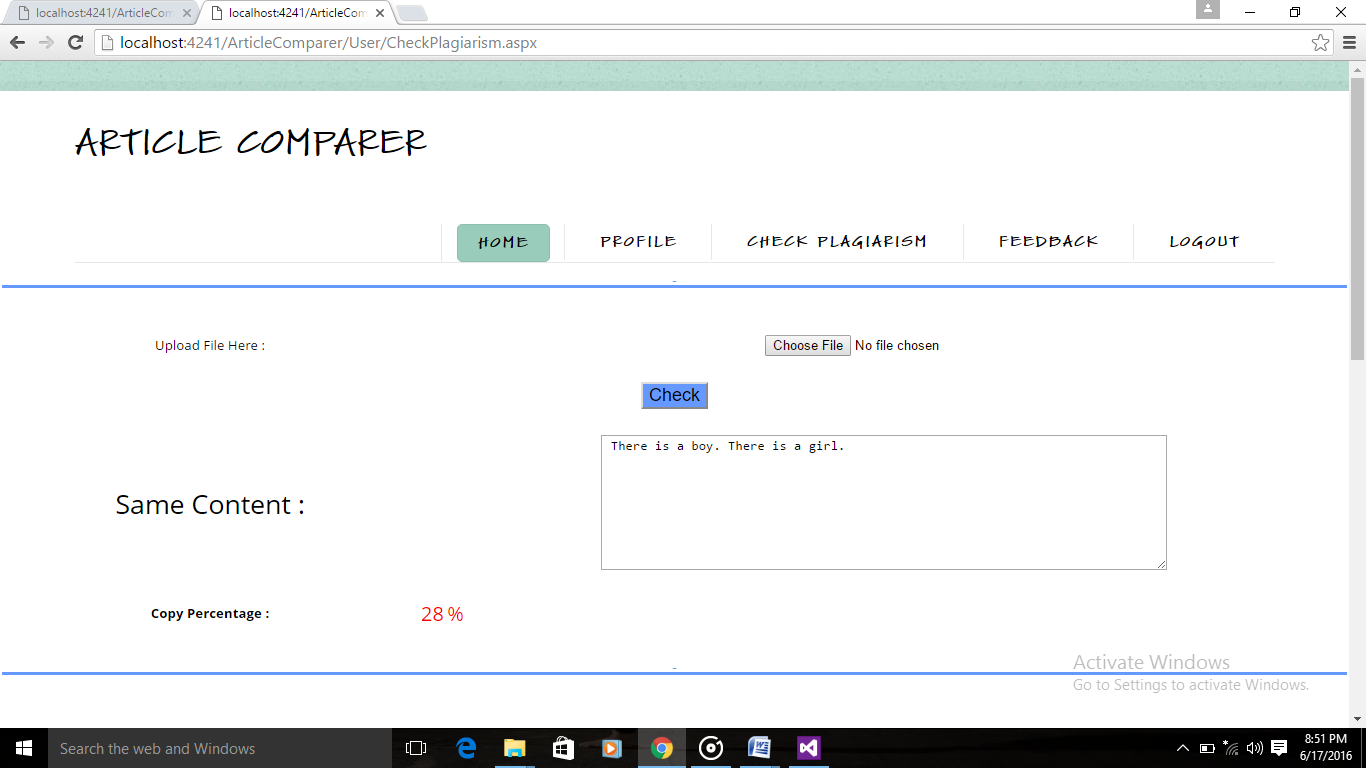
**Upload Files:**

****

**View Uploaded Files:**

****

**Article Comparer output:**

****

**9. CONCLUSION**

In conclusion we can say that, a plagiarizer steals the hard work of an individual and in simple words its cheating. We can prevent that and simplify by the process of “Article Comparer”.

With the widespread use of computers and advent of internet, it has become very easy to plagiarize the work of others. And it’s also quite easy to be accused of plagiarism. And that’s where article comparer is useful. Although internet engines can be used to look for plagiarized material, plagiarism software has access to more sources such as large databases that include periodicals and books that may not be available online. Plagiarism is used in universities and lots of other places. This software not only finds if the data is plagiarized or not but also tells the similarities and the percentage of similarities that exist. This can be really helpful to a professor, to a student or to a news reporter.

We developers develop software to make the day to day life of an individual easy. The basic aim of a software is to simplify the life of the person using it. And article comparer software does that. Its usefulness can be found in many fields and that is why its an important software that should be worked on and developed.

**10. FUTURE SCOPE**

In this project we have been done for the college level. In future we will be going to do for the university level and in the checking module also put more constraints because there is number of technical keywords which might be matched and also we are plan for connect our article comparer with www web server, if it happens then for admin no need to feed data in the databases it will redirect with the www web server so, it would be check the user request with globally**.**

**11. BIBILIOGRAPHY**

**Referred Sites:**

* **FOR .NET INSTALLATION**

[www.support.mircosoft.com](http://www.support.mircosoft.com)

* **FOR SQL**

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)

* **FOR C#.NET**

www.microsoftvirtualacademy.com

www.msdn.microsoft.com/en-IN/library/z1zx9t92.aspx

**Books for Reference:**

[1]C. Anley. Advanced sql injection in sql server applications. Technicalreport, Next Generation Security Software, Ltd, 2002.

[2]K. Bai, H. Wang, and P. Liu. Towards database ﬁrewalls. In DBSec2005.

[3]B. I. A. Barry and H. A. Chan. Syntax, and semantics-based signaturedatabase for hybrid intrusion detection systems. Security and CommunicationNetworks, 2(6), 2009.

[4]D. Bates, A. Barth, and C. Jackson. Regular expressions considered harmful in client-side xss ﬁlters. In Proceedings of the 19th international conference on World wide web, 2010.

[5]M. Christodorescu and S. Jha. Static analysis of executables to detect malicious patterns.[6]M. Cova, D. Balzarotti, V. Felmetsger, and G. Vigna. Swaddler: An Approach for the Anomaly-based Detection of State Violations in Web Applications. In RAID 2007.