**StoryReels**

**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**

**B.Srikanth 13QH1A0507**

**M. Sai Krishna 13QH1A0523**

**CH . Praveen 13QH1A0508**

**Sk Riyaz Roshan 13QH1A0549**

**Under the guidance of**

**Ch.Ravi Kumar**

**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)**

**StoryReels**

**By**

**B.Srikanth 13QH1A0507**

**M. Sai Krishna 13QH1A0523**

**CH . Praveen 13QH1A0508**

**Sk Riyaz Roshan 13QH1A0549**

**Under the Esteemeed guidance of**

**Assoc.Prof G. Soma Sekhar**

**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)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

This is to certify the project report entitled “**Story Reels**” 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.

**B.Srikanth 13QH1A0507**

**M. Sai Krishna 13QH1A0523**

**CH . Praveen 13QH1A0508**

**Sk Riyaz Roshan 13QH1A0549**

**INTERNAL GUIDE Head of Department**

**CH RAVI KUMAR Prof. SOMA SEKHAR**

**Asst Prof Head of Department**

**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.

Signature: Signature:

Name: B. Srikanth Name: CH. Praveen

Roll No.: 13QH1a0507 Roll No.: 13QH1a0508

Date: Date:

Signature: Signature:

Name: M.Sai Krishna Name: Sk. Riyaz Roshan

Roll No.: 13QH1A0523 Roll No: 13QH1A0549

Date: Date:

**ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the mini project undertaken during final year of B.Tech. We owe special debt of gratitude to CH Ravi Kumar Sir,Assistant Professor of Computer Science & Engineering Department Holy Mary Institute of Technology for his constant support and guidance throughout the course of our work.

We take this opportunity to acknowledge the contribution of Professor Soma Shekar sir, Head of Department, Computer Science & Engineering Dept and all the faculty members of Computer Science & Engineering Department for their full support and assistance during the development of the project.

We would like to express our special thanks to our Principal Reddappa garu and Management of Holy Mary Institute of Technology for giving us moral support and providing us infrastructure to complete the project.

We also like to express our gratitude towards our parents/guardians & siblings for their kind co-operation and encouragement which helped us in completion of this project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:

Name: B. Srikanth

Roll No.: 13QH1a0507

Date:

Signature:

Name: M.Sai krishna

Roll No.: 13QH1A0523

Date:

Signature:

Name: CH. Praven

Roll No.: 13QH1A0508

Date:

Signature:

Name: SK Riyaz Roshan

Roll No: 13QH1A0549

Date:

**INDEX**

**PAGE NO**

**ABSTRACT**

**1.INTRODUCTIION 09**

1.1 Introduction

**2. LITERATURE SURVEY 11**

2.1 Existing System with drawbacks

2.2 Proposed System with advantages

**2. REQUIREMENT SPECIFICATION 13**

2.1 Requirement specification

2.2 Functional requirement

**17**

**3. DESIGN**

3.1UML Diagrams

**4. SURVEY OF TECHNOLOGY 24**

4.1 .NET Framework

4.2 Visual C#

4.3 ASP.NET

4.4 SQL Server

**5. IMPLEMENTATION 36**

5.1Technology Used

5.2 Coding

**75**

**6. RESULT**

**7. TESTING 83**

7.1 Introduction

7.2 Test cases

**89**

**08. CONCLUSION**

**09. SCOPE OF PROJECT AND FUTURE ENHANCEMENT 90**

**LIST OF DIAGRAMS**

**1.Use Case Diagram**

**2.Class Diagram**

**3.SEQUENCE DIAGRAM**

**4.COLLABORATION DIAGRAM**

**5. ACTIVITY DIAGRAM**

**6. .NET FRAMEWORK IN CONTEXT**

**7. ARCHITECTURE OF C# IN .NET**

**LIST OF OUTPUT SCREENS**

**SCREENSHOT 1:Register.aspx**

**SCREENSHOT 2:AboutUs.aspx**

**SCREENSHOT 3: ContactUs.aspx**

**SCREENSHOT 4: Login.aspx**

**SCREENSHOT 5:UserHome.aspx**

**SCREENSHOT 6:UserProfile.aspx**

**SCREENSHOT 7:Poststories.aspx**

**SCREENSHOT 8:StoriesSearch.aspx**

**SCREENSHOT 9:Queries.aspx**

**SCREENSHOT 10:view.aspx**

**SCREENSHOT 11:answered.aspx**

**ABSTRACT**

Travelling has been one of the basic and normal activities in every person’s life. In most of the times travelling may involve in exploring new and almost unknown places, resulting in a situation where a person needs to relay on the third party to guide him through the trip. Hence there is a need for an application in which a user can explore through a set of place and locations in such places, without any involvement of another person to guide him through. This project focuses on giving the user a partial experience of using a social networking web application, and the aspect to entertain and educate the user on such new locations by providing set of stories posted by some other users regarding those particular locations. User can go through different stories posted by other users eventually gaining an overview of that location before visiting. User can ask queries about that location, and can reply to the queries posted by others if the user has previously visited that place. The posts are sorted according to the places in such a way that will be helpful for the user while searching for a place or location. User can also follow that particular place so as to get recent stories that other users have posted. User can also share the stories about the places which he visited previously making the trip entertaining and memorable.

# 1.INTRODUCTION

* 1. INTRODUCTION:

Travelling or visiting various places in holidays these days has become one of the primary things for the Human Beings. Travelling involves in exploring new and almost unknown places, resulting in a situation where a person needs to relay on the third party to guide him through the trip. Hence there is a need for an application in which a user can explore through a set of place and locations in such places, without any involvement of another person to guide him through. Computers have become part of the life for accessing almost any kind of information. Life in this generation is full of technological advancement and in this technological age it is very difficult for any organization to survive without utilizing technology. This project focuses on giving the user a partial experience of using a social networking web application, and the aspect to entertain and educate the user on such new locations by providing set of stories posted by some other users regarding those particular locations. User can go through different stories posted by other users eventually gaining an overview of that location before visiting. User can ask queries about that location, and can reply to the queries posted by others if the user has previously visited that place. The posts are sorted according to the places in such a way that will be helpful for the user while searching for a place or location. User can also follow that particular place so as to get recent stories that other users have posted. User can also share the stories about the places which he visited previously making the trip entetaining and memorable.

# The system mainly consists of two modules. They are

1. Admin Module
2. User Module

**Admin Module:**

Admin Module is a portal where Admin can manage or organize the web application. Admin is provided with a User Id and password where the Admins can login and change the things in the web application. Admin have the right to modify the web application at any time. He also can remove the posts or accounts of the user’s if the Admins finds that the posts or pictures are offensive.

**User Module:**

User module is one of the two modules we provide in our application. It is a portal in which the user performs user’s desired tasks of his interest which are provided by our application. User is provided by respective user Id and password so the user can log into the application. This module consists of set of tabs providing respective functionality to the user in which he may select a desired function. The tabs like Post Stories, Queries, View Queries, Answers, etc are present in the user module offering their respective functionality. The Post Stories tab is where a user can post a desired story about that particular location of users interest. The Queries, View Queries, and Answer tabs provide the functionality to post, view, answer the particular queries respectively. The search Stories tab is main tab in a user module where a user can search for a location view the respective posts and follow that particular location if he is interested so that the resulted posts can be displayed on the users home accordingly. This module is the main part of our application through which user can decide or judge our application.

**LITERATURE SURVEY**

**Existing system with Drawbacks**

In the existing field of systems we don’t have any separate social networking web application for Travelling.

Travelers have to refer each and every site before they plan their journey.

The users should maintain records which they referred from various sites.

Separate registers are maintained for same city in different sites which is risky for travelers.

**Drawbacks**

* The information or memories of visited places cannot be saved in travel site.
* Data security is less.
* Records are more prone to getting lost or misplaced.
* Reports are produced by various sites, that will be an unformatted manual and unreliable

**2.3 Proposed system with advantages**

This web application enables the end users to register online, select the places they want to visit from the menu and save their valuable memories. By just selecting the place that the user want to visit by using this application makes travelling easy. By gaining the knowledge from both the social networking and travelling sites this application has been designed. This application is a little grasp from the sites mentioned above which gives us privilege to access them in one single application.

* It reduces the burden of referring sites.
* Easier maintenance of application & data.
* Improved access to information because of networking.
* Graphically oriented, highly interactive user interface.
* All records can be accessed exclusively by the admin. The admin has the rights to modify any record.
* It provides data security.

**Advantages:-**

* System maintains details of the places to be visited, with a clear information of the particular city.
* System maintains detail of packages, visited stories and special memories.
* System provides various notes related to service to facilitate the users.
* System maintain backup of database, import data, export data to avoid loss of data.
* The proposed system fulfills all the functions needed by the user and the administrator.

**3. REQUIREMENTS SPECIFICATION**

**3.1Requirement Specifcations**

**Hardware requirements:**

* Processor : INTEL P4 or above
* Hard Disk : 40 GB.
* RAM : 2 GB
* Processor Speed : 1.5 GHZ

**Software requirements:**

* Operating system : - Windows XP/7/8/10
* Coding Languages : ASP.NET,C# Programming
* Data Base : SQL Server or any compatible Database

Hardware specifications of the system that is used in the project is

* System: ACER E1-532
* RAM: 4 GB
* Processor Speed: 1.54 GHZ
* Hard Disk: 500 GB

Software specifications of the system that is used in the project is

* **Operating system:** WINDOWS 7
* **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 uer 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 2007:** We use Microsoft Office Word 2007 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.

**3.2Functionality Requirements**

**Functional Requirements:**

Functional requirements are associated with specific functions; tasks or behaviors the system must support. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements. Because non-functional requirements tend to be stated in terms of constraints on the results of tasks which are given as functional requirements (e.g., constraints on the speed or efficiency of a given task), a task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement. That is the approach taken in this work. It can be helpful to think of non-functional requirements as adverbially related to tasks or functional requirements:

**Non-Functional Requirements:**

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. In general, functional requirements define what a system is supposed to *do* whereas non-functional requirements define how a system is supposed to *be*. Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements". Qualities, aka. Non-functional requirements, can be divided into two main categories.

Execution qualities, such as security and usability, are observable at run time. Evolution qualities, such as testability, maintainability, extensibility and scalability, are embodied in the static structure of the software system.

**Reliability:**

The services provided by our application are reliable, we ensure privacy to the documents uploaded by the users and they are only used for the purpose of verification.

**Extensibility:**

The application is widely extensible, where we have included services like the login id to check the message in the profile in case of loss of the messages or mobiles. Also many call routing mechanisms can be included.

**Efficiency:**

The system functions in an efficient manner with proper acknowledgements and responses at high speed.

**4. DESIGN**

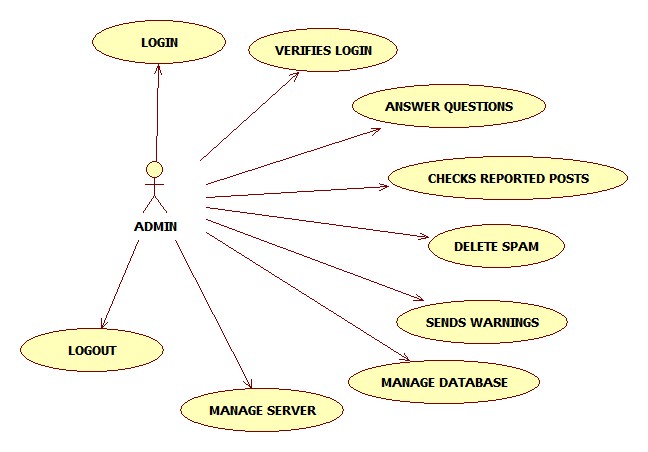
**4.1 UML Diagrams**

The Unified Modelling Language (UML) is a Standard language for specifying, visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed.

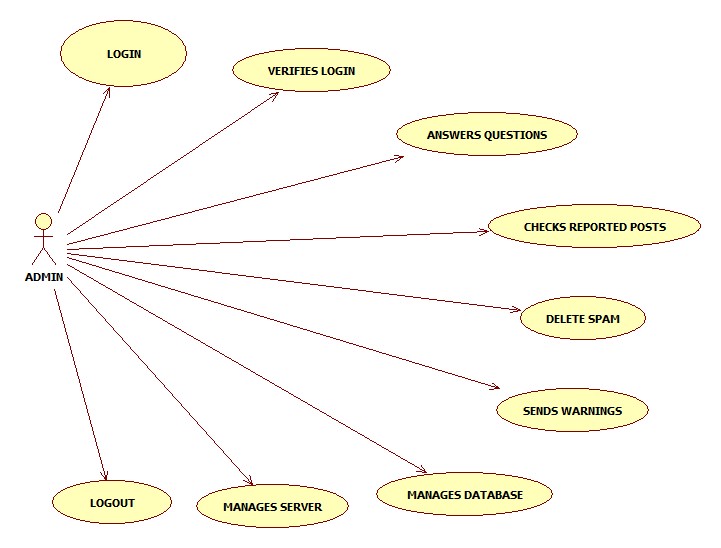
**USE CASE DIAGRAM**

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

**User Use case Diagram**



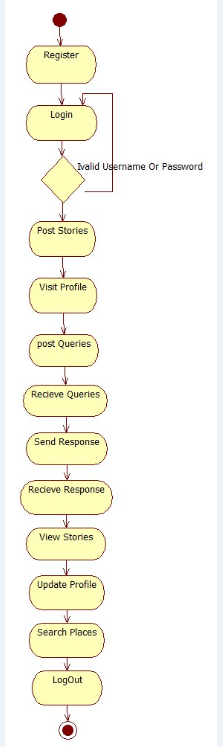
**Admin Use Case Diagram**



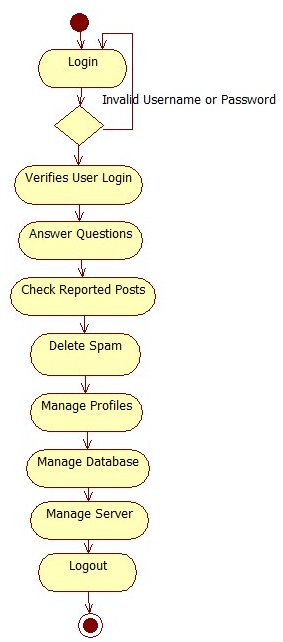
**ACTIVITY DIAGRAM**

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.

**User Activity Diagram**

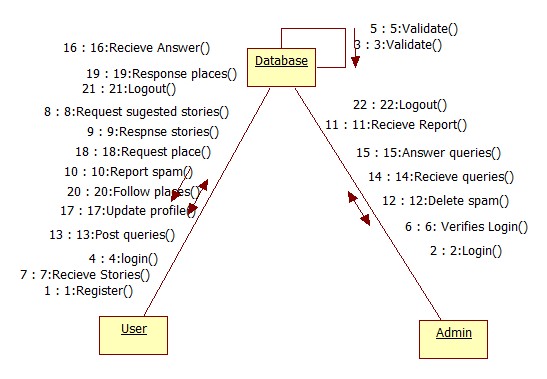


**Admin Activity Diagram**

****

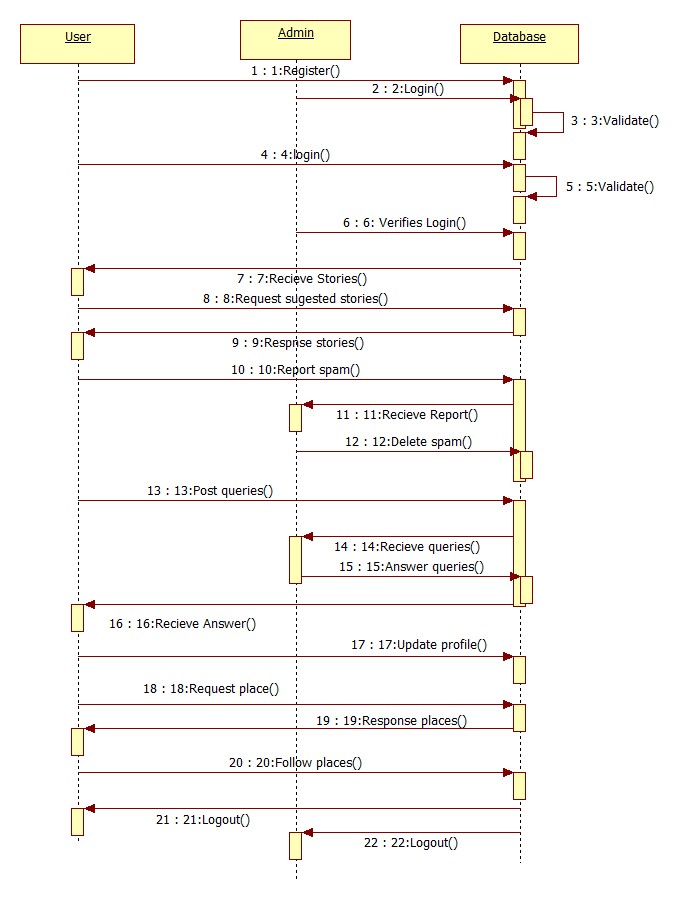
**Collaboration Diagram**

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.

****

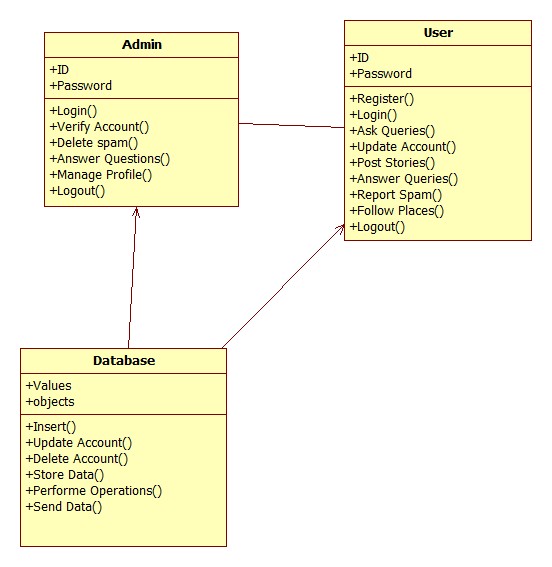
**Sequence Diagram**

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.

****

**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.



**5. SURVEY OF TECHNOLOGY**

**5.1 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.

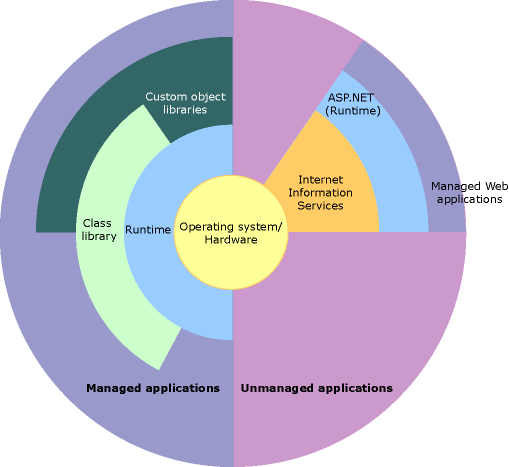
The .NET Framework consists of the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread 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).

**5.2 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.



Architectureof *C# in .Net*

**5.3 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. I5n 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.

**5.5 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. IMPLEMENTATION**

It is the process of bringing developed system of revised system into operational use. If the implementation phase is not carefully planned and controlled, it can lead to many problems. Thus proper implementation is essential to provide a reliable system to meet managerial requirements.

6.1 TECHNOLOGY USED

The application is completely written in .net .This integrates all existing bank and provides business solutions. This application acts as standard interface between the customers and all the banks. We use sql server as a back end for storing database.

6.1.1: Dot Net:

The .NET Framework (pronounced dot net) is a proprietary, partially open source freeware software framework developed by Microsoft that runs primarily on Microsoft Windows.

6.1.2: What is Dot Net?

The .NET is the technology from Microsoft, on which all other Microsoft technologies will be depending on in future. It is a major technology change, introduced by Microsoft, to catch the market from the SUN's Java. Few years back, Microsoft had only VC++ and VB to compete with Java, but Java was catching the market very fast. With the world depending more and more on the Internet/ Web and java related tools becoming the best choice for the web applications, Microsoft seemed to be losing the battle. Thousands of programmers moved to java from VC++ and VB. To recover the .market, .Microsoft announced .NET.

But Microsoft has a wonderful history of starting late but catching up quickly. This is true in case of .NET too. Microsoft put their best men at work for a secret project called Next Generation Windows Services (NGWS)., under the direct supervision of Mr. Bill Gates.

The outcome of the project is what we now know as .NET. Even though .NET has borrowed most of its ideas from Sun's J2EE, it has really outperformed their competitors.

Microsoft's VC++ was a powerful tool. But it was too complex. It has too many data types, and developers had to learn many libraries including Windows SDK, MFC, ATL, COM etc. There were many data type compatibility issues while exchanging data between different layers. Visual Basic was too easy, and many serious programmers hated it just for that reason. Even though Visual basic was very easy to use, it was not very flexible to develop serious applications. SUN's Java became a very good choice for these reasons. It had the flexibility and power of C++ and at the same time easy enough to catch the attention of VB programmers.

6.1.3: The structure of Dot Net:

• It is a platform for application developers.

• It is a Framework that supports Multiple Language and Cross language integration.

• IT has IDE (Integrated Development Environment).

• Framework is a set of utilities or can say building blocks of your application system.

• .NET Framework provides GUI in a GUI manner.

• .NET is a platform independent but with help of Mono Compilation System (MCS). MCS is a middle level interface.

• .NET Framework provides interoperability between languages i.e. Common Type System (CTS).

• .NET Framework also includes the .NET Common Language Runtime (CLR), which is responsible for maintaining the execution of all applications developed using the .NET library.

The .NET Framework consists primarily of a gigantic library of code.

Class Library:

.NET Framework includes a set of standard class libraries. The class library is organized in a hierarchy of namespaces. Most of the built-in APIs are part of either system or Microsoft namespaces. These class libraries implement a large number of common functions, such as file reading and writing, graphic rendering, database interaction, and XML document manipulation, among others. .NET class libraries are available to all CLI compliant languages. .NET Framework class library is divided into two parts: Framework Class Library (FCL) and Base Class Library (BCL).

FCL includes a small subset of the entire class library and is the core set of classes that serve as the basic API of CLR. Classes in and some classes.inSystem.dll and System.core.dll are part of FCL. FCL classes are available in .NET Framework as well as its alternative implementations including .NET Compact Framework, Microsoft Silverlight and Mono.

BCL is a superset of FCL and refers to the entire class library that ships with .NET Framework. It includes an expanded set of libraries, including Windows Forms, ADO.NET, ASP.NET, Language Integrated Query (LINQ), Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF) and Workflow Foundation (WF). BCL is much larger in scope than standard libraries for languages like C++, and comparable in scope to standard libraries of Java.

.NET Core:

.NET Core is a free and open-source partial implementation of the .NET Framework. It consists of CoreCLR and CoreFX, which are partial forks of CLR and BCL respectively.NET Core comes with an improved JIT compiler, called RyuJIT.

Assemblies:

Compiled CIL code is stored in CLI assemblies. As mandated by the specification, assemblies are stored in Portable Executable (PE) file format, common on Windows platform for all DLL and EXE files. Each assembly consists of one or more files, one of which must contain a manifest bearing the metadata for the assembly. The complete name of an assembly (not to be confused with the file name on disk) contains its simple text name, version number, culture, and public key token. Assemblies are considered equivalent if they share the same complete name, excluding the revision of the version number.

A private key can also be used by the creator of the assembly for strong naming. The public key token identifies which private key an assembly is signed with. Only the creator of the keypair(typically .NET developer signing the assembly) can sign assemblies that have the same strong name as a previous version assembly, since the creator is in possession of the private key. Strong naming is required to add assemblies to Global Assembly Cache.

C++/CLI:

Microsoft introduced C++/CLI in Visual Studio 2005, which is a language and means of compiling Visual C++ programs to run within the .NET Framework. Certain portions of the C++ program still run within an unmanaged Visual C++ Runtime, while specially modified portions

are translated into CIL code and run with the .NET Framework's CLR.

Assemblies compiled using the C++/CLI compiler are known as mixed-mode assemblies, since they contain native and managed code within the same DLL, Such assemblies are also difficult to reverse engineer, since .NET de compilers such as .NET Reflector only reveal the managed code.

Security:

.NET Framework has its own security mechanism with two general features: Code Access Security (CAS), and validation and verification. CAS is based on evidence that is associated with a specific assembly. Typically the evidence is the source of the assembly (whether it is installed on the local machine or has been downloaded from the intranet or Internet). CAS uses evidence to determine the permissions granted to the code. Other code can demand that calling code be granted a specified permission. The demand causes CLR to perform a call stack walk: every assembly of each method in the call stack is checked for the required permission; if any assembly is not granted the permission a security exception is thrown.

Managed CIL byte code is easier to reverse-engineer than native code, unless obfuscated. NET decompile programs enable developers with no reverse-engineering skills to view the source code behind unobfuscated .NET assemblies. In contrast, apps compiled to native machine code are much harder to reverse-engineer and source code is almost never produced successfully, mainly because of compiler optimizations and lack of reflection. One concern is over possible loss of trade secrets and the bypassing of license control mechanisms. To mitigate this, Microsoft has included Dotfuscator Community Edition with Visual Studio .NET since 2002.[b] Third-party obfuscation tools are also available from vendors such as vmware, V.i. Labs, Xenocode, Red Gate Software. Method-level encryption tools for .NET code are available from vendors such as Safe Net.

Memory management:

CLR frees the developer from the burden of managing memory (allocating and freeing up when done); it handles memory management itself by detecting when memory can be safely freed. Instantiations of .NET types (objects) are allocated from the managed heap; a pool of memory managed by CLR. As long as there exists a reference to an object, which might be either a direct reference to an object or via a graph of objects, the object is considered to be in use. When there is no reference to an object, and it cannot be reached or used, it becomes garbage, eligible for collection.

.NET Framework includes a garbage collector (GC) which runs periodically, on a separate thread from the application's thread, that enumerates all the unusable objects and reclaims the memory allocated to them. It is a non-deterministic, compacting, mark-and-sweep garbage collector. GC runs only when a certain amount of memory has been used or there is enough pressure for memory on the system. Since it is not guaranteed when the conditions to reclaim memory are reached, GC runs are non-deterministic. Each .NET application has a set of roots, which are pointers to objects on the managed heap (managed objects). These include references to static objects and objects defined as local variables or method parameters currently in scope, as well as objects referred to by CPU registers. When GC runs, it pauses the application, and for each object referred to in the root, it recursively enumerates all the objects reachable from the root objects and marks them as reachable. It uses CLI metadata and reflection to discover the objects encapsulated by an object, and then recursively walk them. It then enumerates all the objects on the heap (which were initially allocated contiguously) using reflection. All objects not marked as reachable are garbage. This is the mark phase. Since the memory held by garbage is not of any consequence, it is considered free space. However, this leaves chunks of free space between objects which were initially contiguous. The objects are then compacted together to make used memory contiguous again. Any reference to an ect invalidated by moving the object is updated by GC to reflect the new location. The application is resumed after the garbage collection is over.

GC used by .NET Framework is also generational. Objects are assigned a generation; newly created objects belong to Generation 0. The objects that survive a garbage collection are tagged as Generation 1, and the Generation 1 objects that survive another collection are Generation 2 objects. .NET Framework uses up to Generation 2 objects. Higher generation objects are garbage collected less frequently than lower generation objects. This helps increase the efficiency of garbage collection, as older objects tend to have a longer lifetime than newer objects. Thus, by eliminating older (and thus more likely to survive a collection) objects from the scope of a collection run, fewer objects need to be checked and compacted.

Performance:

When an application is first launched, the .NET Framework compiles the CIL code into executable code using its just-in-time compiler, and caches the executable program into the .NET Native Image Cache. Due to caching, the application launches faster for subsequent launches, although the first launch is usually slower. To increase speed of the first launch, developers may use the Nativ Image Generator utility to manually compile and cache any .NET application, ahead-of-time.

The garbage collector, which is integrated into the environment, can introduce unanticipated delays of execution over which the developer has little direct control. "In large applications, the number of objects that the garbage collector needs to deal with can become very large, which means it can take a very long time to visit and rearrange all of them.

NET Framework provides support for calling Streaming SIMD Extensions (SSE) via managed code from April 2014 in Visual Studio 2013 Update 2. However, Mono has provided support for SIMD Extensions as of version 2.2 within the Mono.Simd namespace; before. Mono's lead developer Miguel de Icaza has expressed hope that this SIMD support will be adopted by CLR's ECMA standard.Streaming SIMD Extensions have been available in x86 CPUs since the introduction of the Pentium III. Some other architectures such as ARM and MIPS also have SIMD extensions. In case the CPU lacks support for those extensions, the instructions are simulated in software

**6.2 Coding**

**Login**

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.Data.SqlClient;

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

{

classprj cs = new classprj();

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

{

}

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

{

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

{

string qry = "select \* from register where EmailId='" + txtuname.Text+ "' and Password='" + txtpass.Text+ "'";

DataSet ds = cs.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["Email"] = ds.Tables[0].Rows[0][2].ToStrin**ABSTRACT ABSTRACT** g();

Session["Password"] = ds.Tables[0].Rows[0][3].ToString();

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

}

else

{

Response.Write("<script>alert('Invalid Login')</script>");

txtuname.Text = txtpass.Text = "";

}

}

else if (DropDownList1.SelectedItem.Text == "Admin")

{

if(txtuname.Text=="admin"&&txtpass.Text=="admin")

{

Response.Redirect("~/Admin/AdminHome.aspx");

}

}

}

}

**Register**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

classprj cs = new classprj();

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

{

}

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

{

string qry = "insert into register values('"+Txtname.Text+"','"+Txtemail.Text+"','"+Txtpass.Text+"','"+Txtage.Text+"','"+Txtdob.Text+"','"+RadioButtonList1.SelectedItem.Text+"','"+DropDownList1.SelectedItem.Text+"')";

int i = cs.inupdel(qry);

if (i > 0)

{

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

Txtname.Text = Txtemail.Text = Txtpass.Text = txtcpass.Text = Txtage.Text = Txtdob.Text = "";

}

else

{

Response.Write("<script>alert('registration failed')</script>");

}

**UserHome**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

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

{

Label2.Text = Session["name"].ToString();

}

}

}

}

**UserProfile**

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.Data.SqlClient;

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

{

classprj cs=new classprj();

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

{

if (!IsPostBack)

{

load();

}

}

public void load()

{

{

try

{

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

DataSet ds = cs.select(qry2);

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

{

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

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

//txtloc.Text = ds.Tables[0].Rows[0][9].ToString();

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

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

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

TextBox1.Text = ds.Tables[0].Rows[0][6].ToString();

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

}

else

{

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

}

}

catch (Exception ex)

{

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

}

}

}

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

{

{

try

{

string qry = "update Register set Password='" + txtpas.Text + "',Age='" + txtage.Text + "',State='" + txtstate.Text + "' where StudId='" + Session["id"].ToString() + "'";

int i = cs.inupdel(qry);

if (i > 0)

{

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

}

}

catch (Exception ex)

{

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

}

}

}

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

{

txtpas.ReadOnly = false;

}

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

{

txtage.ReadOnly = false;

}

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

{

txtstate.ReadOnly = false;

}

}

**ViewQueries**

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

classprj cs = new classprj();

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

{

if (!IsPostBack)

{

bind();

UpdatePanel1.Visible = false;

}

}

public void bind()

{

string qry="select \* from feedback";

DataSet ds = cs.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

else

{

Response.Write("<script>alert('No Queries Present')</script>");

}

}

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

{

Button btn = (Button)sender;

GridViewRow gr = (GridViewRow)btn.NamingContainer;

Label lblid = (Label)gr.FindControl("Label4");

Label lbluserid = (Label)gr.FindControl("Label7");

Label lblname = (Label)gr.FindControl("Label5");

Label lblmsg = (Label)gr.FindControl("Label6");

ViewState["id2"]= lblid.Text;

ViewState["userid"] = lbluserid.Text;

ViewState["name1"] = lblname.Text;

TextBox1.Text = lblmsg.Text;

UpdatePanel1.Visible = true;

}

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

{

try

{

string qry = "insert into AnsFeed values('" + ViewState["id2"].ToString() + "','" + ViewState["userid"].ToString() + "','" + Session["id"].ToString() + "','" + TextBox1.Text + "','" + TextBox2.Text + "')";

int i =cs.inupdel (qry);

if (i > 0)

{

string message = "alert('answered')";

ScriptManager.RegisterClientScriptBlock((sender as Control), this.GetType(), "alert", message, true);

TextBox1.Text = TextBox2.Text = "";

}

else

{

string message = "alert('Error while answering')";

ScriptManager.RegisterClientScriptBlock((sender as Control), this.GetType(), "alert", message, true);

}

}

catch (Exception ex)

{

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

}

}

}

**Search Story**

protected void ddlstate\_SelectedIndexChanged(object sender, EventArgs e)

{

if(ddlstate.SelectedItem.Text=="TG")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Hyderabad");

ddlcity.Items.Add("Warangal");

ddlcity.Items.Add("Karimnagar");

ddlcity.Items.Add("Nizambad");

ddlcity.Items.Add("SuryaPet");

}

else if(ddlstate.SelectedItem.Text=="AP")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("VIZAG");

ddlcity.Items.Add("Vijayawada");

ddlcity.Items.Add("Tirupathi");

}

else if (ddlstate.SelectedItem.Text == "BIHAR")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Patna ");

ddlcity.Items.Add("Badalpura ");

ddlcity.Items.Add("Nohsa ");

}

else if (ddlstate.SelectedItem.Text == "DELHI")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Agra");

ddlcity.Items.Add("Aligarh");

ddlcity.Items.Add("Allahabad");

}

else if (ddlstate.SelectedItem.Text == "MAHARASHTRA")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Mumbai");

ddlcity.Items.Add("Pune");

ddlcity.Items.Add("Nagpur");

}

}

protected void ddlcity\_SelectedIndexChanged(object sender, EventArgs e)

{

if (ddlstate.SelectedItem.Text == "TG")

{

if (ddlcity.SelectedItem.Text == "Hyderabad")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("--SELECT--");

ddllocality.Items.Add("Ameerpet");

ddllocality.Items.Add("Secunderabad");

ddllocality.Items.Add("Uppal");

ddllocality.Items.Add("Ghatkesar");

ddllocality.Items.Add("LB Nagar");

}

else if (ddlcity.SelectedItem.Text == "Warangal")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("--SELECT--");

ddllocality.Items.Add("Huzurabad");

ddllocality.Items.Add("Vijayawada");

ddllocality.Items.Add("Tirupathi");

}

else if (ddlcity.SelectedItem.Text == "Karimnagar")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("--SELECT--");

ddllocality.Items.Add("Patna ");

ddllocality.Items.Add("Badalpura ");

ddllocality.Items.Add("Nohsa ");

}

}

else if(ddlstate.SelectedItem.Text=="AP")

{

if (ddlcity.SelectedItem.Text == "VIZAG")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("--SELECT--");

ddllocality.Items.Add("ABC");

ddllocality.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "Vijayawada")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("EFG");

ddllocality.Items.Add("IJK");

ddllocality.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "Tirupathi")

{

ddllocality.Items.Clear();

ddllocality.Items.Add("OPG ");

ddllocality.Items.Add("ZZZ ");

ddllocality.Items.Add("AAA ");

}

}

}

**USERHOME.aspx (After Following)**

using System.Web.UI.WebControls;

using System.Data.SqlClient;

using System.Collections;

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

{

classprj obj = new classprj();

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

{

Label2.Text = Session["name"].ToString();

bind();

}

public ArrayList al = new ArrayList();

public void bind()

{

string qry = "select Location from post where Status='" + Session["id"].ToString() + "' and Follow='Follow'";

DataSet ds = obj.select(qry);

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

{

for (int i = 0; i < ds.Tables[0].Rows.Count; i++)

{

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

string qry1 = "select TOP 3 \* from post where Location='" + c.ToString() + "' and Status='Empty' order by Date1 DESC";

DataSet ds2 = obj.select(qry1);

if (ds2.Tables[0].Rows.Count > 0)

{

GridView1.DataSource = ds2;

GridView1.DataBind();

}

else

{

// GridView1.Visible = false;

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

}

}

}

else

{

// GridView1.Visible = false;

}

}

}

**Searchstories.aspx**

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

using System.Data;

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

{

classprj obj = new classprj();

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

{

}

protected void ddlstate\_SelectedIndexChanged(object sender, EventArgs e)

{

if (ddlstate.SelectedItem.Text == "Telangana")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Hyderabad");

ddlcity.Items.Add("Warangal");

ddlcity.Items.Add("Karimnagar");

ddlcity.Items.Add("Nizambad");

ddlcity.Items.Add("SuryaPet");

}

else if (ddlstate.SelectedItem.Text == "AndhraPradesh")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("VIZAG");

ddlcity.Items.Add("Vijayawada");

ddlcity.Items.Add("Tirupathi");

}

else if (ddlstate.SelectedItem.Text == "Kerala")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("kochin");

ddlcity.Items.Add("thiruvananthapuram");

ddlcity.Items.Add("aluva");

}

else if (ddlstate.SelectedItem.Text == "DELHI")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Agra");

ddlcity.Items.Add("Aligarh");

ddlcity.Items.Add("Allahabad");

}

else if (ddlstate.SelectedItem.Text == "TamilNadu")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Chennai");

ddlcity.Items.Add("velluru");

ddlcity.Items.Add("madhurai");

}

}

protected void ddlcity\_SelectedIndexChanged(object sender, EventArgs e)

{

if (ddlstate.SelectedItem.Text == "Telangana")

{

if (ddlcity.SelectedItem.Text == "Hyderabad")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("Ameerpet");

ddllocation.Items.Add("Secunderabad");

ddllocation.Items.Add("Uppal");

ddllocation.Items.Add("Ghatkesar");

ddllocation.Items.Add("LB Nagar");

}

else if (ddlcity.SelectedItem.Text == "Warangal")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("Huzurabad");

ddllocation.Items.Add("Vijayawada");

ddllocation.Items.Add("Tirupathi");

}

else if (ddlcity.SelectedItem.Text == "Karimnagar")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("kabham");

ddllocation.Items.Add("srisailam");

ddllocation.Items.Add("kajipet");

}

}

else if (ddlstate.SelectedItem.Text == "AndhraPradesh")

{

if (ddlcity.SelectedItem.Text == "VIZAG")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "Vijayawada")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "Tirupathi")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "Delhi")

{

if (ddlcity.SelectedItem.Text == "Agra")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "Alighar")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "Allahadabad")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "Kerala")

{

if (ddlcity.SelectedItem.Text == "kochin")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "thiruvananthapuram")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "aluva")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "TamilNadu")

{

if (ddlcity.SelectedItem.Text == "chennai")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "velluru")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "madhurai")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

}

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

{

try

{

string qry1 = "select \* from post where State='" + ddlstate.SelectedItem.Text + "' and City='" + ddlcity.SelectedItem.Text + "' and Location='" + ddllocation.SelectedItem.Text + "' and Status='" + Session["Id"].ToString() + "'";

DataSet ds1 = obj.select(qry1);

if (ds1.Tables[0].Rows.Count > 0)

{

string qry = "select \* from post where State='" + ddlstate.SelectedItem.Text + "' and City='" + ddlcity.SelectedItem.Text + "' and Location='" + ddllocation.SelectedItem.Text + "' and Follow='Follow'and Status='" + Session["Id"].ToString() + "'";

DataSet ds = obj.select(qry);

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

{

string qry6 = "select \* from post where Location='" + ddllocation.SelectedItem.Text + "' and Status='Empty'";

DataSet ds6 = obj.select(qry6);

if (ds6.Tables[0].Rows.Count > 0)

{

GridView1.DataSource = ds6;

GridView1.DataBind();

lblmsg.Text = ddllocation.SelectedItem.Text;

Button2.Text = "Unfollow";

}

else

{

}

}

else

{

string qry7 = "select \* from post where State='" + ddlstate.SelectedItem.Text + "' and City='" + ddlcity.SelectedItem.Text + "' and Location='" + ddllocation.SelectedItem.Text + "' and Follow='Unfollow'and Status='" + Session["Id"].ToString() + "'";

DataSet ds7 = obj.select(qry7);

if (ds7.Tables[0].Rows.Count > 0)

{

string qry6 = "select \* from post where Location='" + ddllocation.SelectedItem.Text + "' and Status='Empty'";

DataSet ds6 = obj.select(qry6);

if (ds6.Tables[0].Rows.Count > 0)

{

GridView1.DataSource = ds6;

GridView1.DataBind();

lblmsg.Text = ddllocation.SelectedItem.Text;

Button2.Text = "Follow";

}

}

}

}

else

{

string qry = "select \* from post where State='" + ddlstate.SelectedItem.Text + "' and City='" + ddlcity.SelectedItem.Text + "' and Location='" + ddllocation.SelectedItem.Text + "' ";

DataSet ds = obj.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

lblmsg.Text = ddllocation.SelectedItem.Text;

Button2.Text = "Follow";

}

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)

{

string qry1 = "select \* from post where State='" + ddlstate.SelectedItem.Text + "' and City='" + ddlcity.SelectedItem.Text + "' and Location='" + ddllocation.SelectedItem.Text + "' and Status='" + Session["Id"].ToString() + "'";

DataSet ds1 = obj.select(qry1);

if (ds1.Tables[0].Rows.Count > 0)

{

string a = ds1.Tables[0].Rows[0][6].ToString();

if (a == "Unfollow")

{

string qry2 = "update post set Follow='Follow' where Status='" + Session["Id"].ToString() + "' and Location='" + ddllocation.SelectedItem.Text + "' ";

int j = obj.inupdel(qry2);

if (j > 0)

{

Button2.Text = "Unfollow";

}

else

{

}

}

else

{

string qry2 = "update post set Follow='Unfollow' where Status='" + Session["Id"].ToString() + "' and Location='" + ddllocation.SelectedItem.Text + "' ";

int j = obj.inupdel(qry2);

if (j > 0)

{

Button2.Text = "Follow";

}

else

{

}

}

}

else

{

string qry = "insert into post values('" + ddlstate.SelectedItem.Text + "','" + ddlcity.SelectedItem.Text + "','" + ddllocation.SelectedItem.Text + "','','','Unfollow','" + Session["Id"].ToString() + "','" + DateTime.Now.ToString("yyy-mm-dd") + "')";

int i = obj.inupdel(qry);

if (i > 0)

{

Button2.Text = "Unfollow";

}

else

{

}

}

}

}

**Post Query**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

classprj cs = new classprj();

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

{

if (!IsPostBack)

{

TextBox1.Text = Session["Name"].ToString();

}

}

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

{

try

{

string qry = "insert into feedback values('" + Session["Id"].ToString() + "','" + TextBox1.Text + "','" + TextBox2.Text + "')";

int i = cs.inupdel(qry);

if (i > 0)

{

Response.Write("<script>alert('query posted')</script>");

TextBox2.Text = "";

}

else

{

Response.Write("<script>alert('query not posted')</script>");

}

}

catch (Exception ex)

{

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

}

}

}

**Post Story**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

classprj cs = new classprj();

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

{

}

protected void ddlstate\_SelectedIndexChanged(object sender, EventArgs e)

{

if(ddlstate.SelectedItem.Text=="Telangana")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Hyderabad");

ddlcity.Items.Add("Warangal");

ddlcity.Items.Add("Karimnagar");

ddlcity.Items.Add("Nizambad");

ddlcity.Items.Add("SuryaPet");

}

else if(ddlstate.SelectedItem.Text=="AndhraPradesh")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("VIZAG");

ddlcity.Items.Add("Vijayawada");

ddlcity.Items.Add("Tirupathi");

}

else if (ddlstate.SelectedItem.Text == "Kerala")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("kochin");

ddlcity.Items.Add("thiruvananthapuram");

ddlcity.Items.Add("aluva");

}

else if (ddlstate.SelectedItem.Text == "DELHI")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Agra");

ddlcity.Items.Add("Aligarh");

ddlcity.Items.Add("Allahabad");

}

else if (ddlstate.SelectedItem.Text == "TamilNadu")

{

ddlcity.Items.Clear();

ddlcity.Items.Add("--SELECT--");

ddlcity.Items.Add("Chennai");

ddlcity.Items.Add("velluru");

ddlcity.Items.Add("madhurai");

}

}

protected void ddlcity\_SelectedIndexChanged(object sender, EventArgs e)

{

if (ddlstate.SelectedItem.Text == "Telangana")

{

if (ddlcity.SelectedItem.Text == "Hyderabad")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("Ameerpet");

ddllocation.Items.Add("Secunderabad");

ddllocation.Items.Add("Uppal");

ddllocation.Items.Add("Ghatkesar");

ddllocation.Items.Add("LB Nagar");

}

else if (ddlcity.SelectedItem.Text == "Warangal")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("Huzurabad");

ddllocation.Items.Add("Vijayawada");

ddllocation.Items.Add("Tirupathi");

}

else if (ddlcity.SelectedItem.Text == "Karimnagar")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("kabham");

ddllocation.Items.Add("srisailam");

ddllocation.Items.Add("kajipet");

}

}

else if (ddlstate.SelectedItem.Text == "AndhraPradesh")

{

if (ddlcity.SelectedItem.Text == "VIZAG")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "Vijayawada")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "Tirupathi")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "Delhi")

{

if (ddlcity.SelectedItem.Text == "Agra")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "Alighar")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "Allahadabad")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "Kerala")

{

if (ddlcity.SelectedItem.Text == "kochin")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "thiruvananthapuram")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "aluva")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

else if (ddlstate.SelectedItem.Text == "TamilNadu")

{

if (ddlcity.SelectedItem.Text == "chennai")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("--SELECT--");

ddllocation.Items.Add("ABC");

ddllocation.Items.Add("BCD");

}

else if (ddlcity.SelectedItem.Text == "velluru")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("EFG");

ddllocation.Items.Add("IJK");

ddllocation.Items.Add("MNP");

}

else if (ddlcity.SelectedItem.Text == "madhurai")

{

ddllocation.Items.Clear();

ddllocation.Items.Add("OPG ");

ddllocation.Items.Add("ZZZ ");

ddllocation.Items.Add("AAA ");

}

}

}

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

{

try

{

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

if (txt.ToUpper() == ".PNG" || txt.ToUpper() == ".JPG" || txt.ToUpper() == ".JPEG")

{

if (FileUpload1.HasFile)

{

string filename1 = "~/Images1/" + FileUpload1.FileName;

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

string f1 = FileUpload1.FileName;

string qry = "insert into post values('" + ddlstate.SelectedItem.Text + "','" + ddlcity.SelectedItem.Text + "','" + ddllocation.SelectedItem.Text + "','" + filename1 + "','" + TextBox1.Text + "','Follow','Empty','" + DateTime.Now.ToString("yyyy-mm-dd") + "')";

int i = cs.inupdel(qry);

}

else

{

Response.Write("<script>alert('plese uplode file')</script>");

}

}

else

{

Response.Write("<script>alert('plese uplode only image file')</script>");

}

}

catch (Exception ex)

{

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

}

}

}

**Answered Query**

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

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

{

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

{

if(!IsPostBack)

{

bind();

}

}

classprj cs = new classprj();

public void bind()

{

try

{

string qry = "select \* from AnsFeed where PresentId='" + Session["id"].ToString() + "'";

DataSet ds = cs.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

else

{

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

}

}

catch (Exception ex)

{

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

}

}

}

**Admin View Posts**

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.Data.SqlClient;

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

{

classprj cs = new classprj();

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

{

if (!IsPostBack)

{

bind();

}

}

public void bind()

{

string qry = "select \* from post";

DataSet ds = cs.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

else

{

Response.Write("<script>alert('No Queries Present')</script>");

}

}

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

{

Button img = (Button)sender;

GridViewRow gr = (GridViewRow)img.NamingContainer;

Label lbl = (Label)gr.FindControl("Label2");

string qry = "delete from post where FieldId='" + lbl.Text + "'";

int i = cs.inupdel(qry);

if (i > 0)

{

Response.Write("<script>alert('Deleted Sucesfully !!')</script>");

bind();

}

else

{

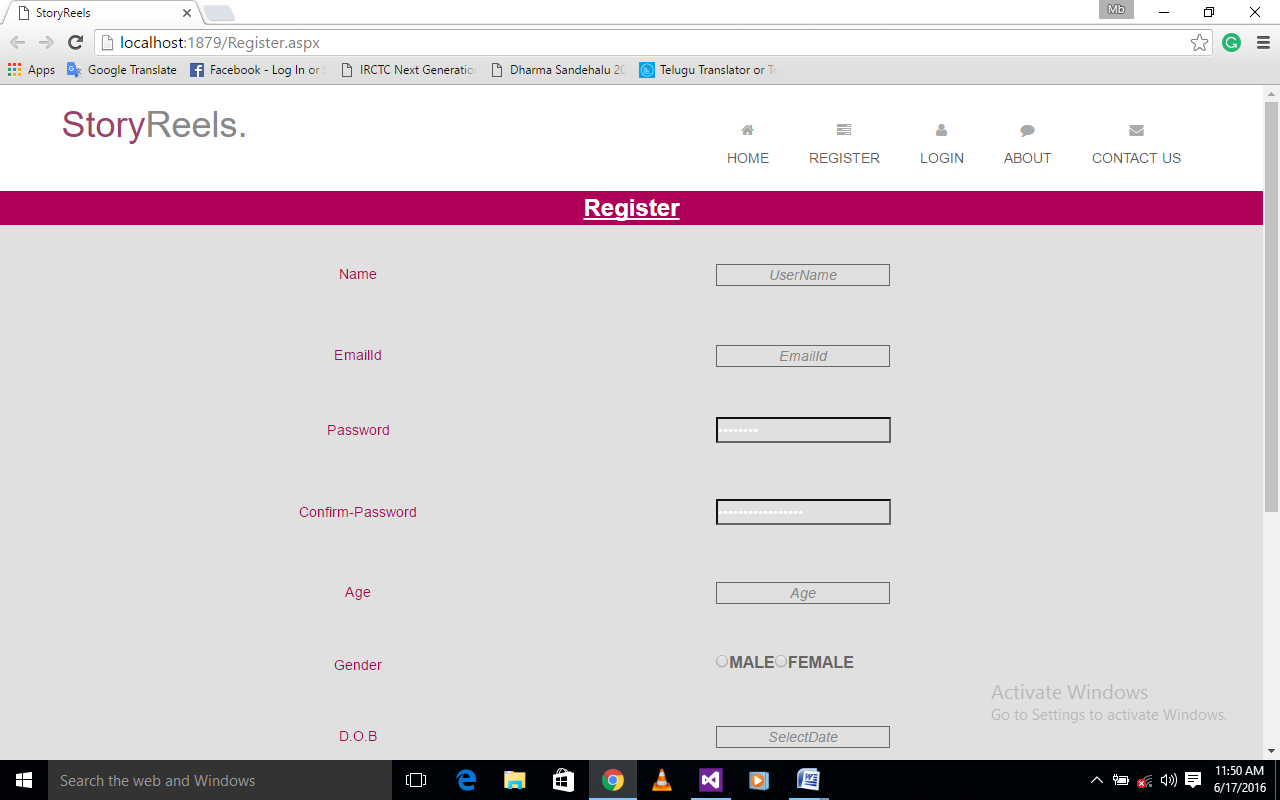
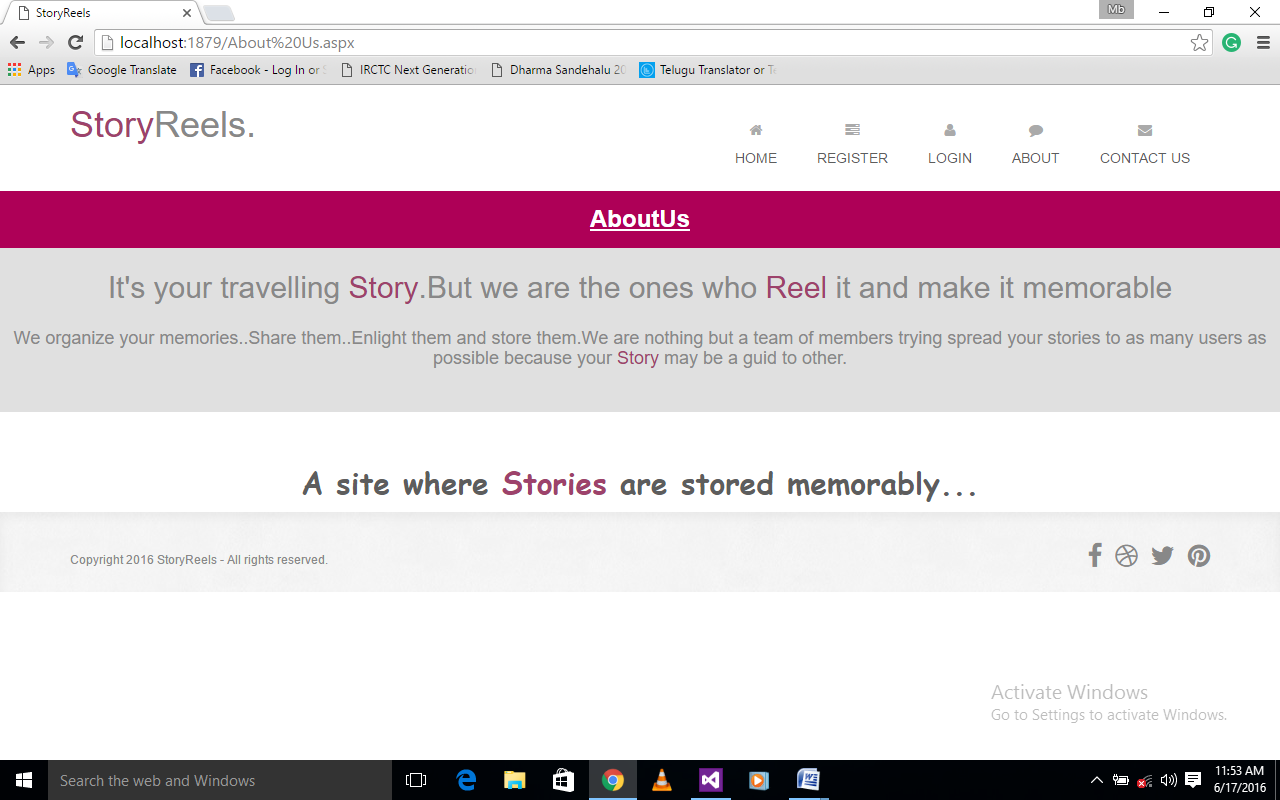
Response.Write("Not Yet Deleted !!");

}

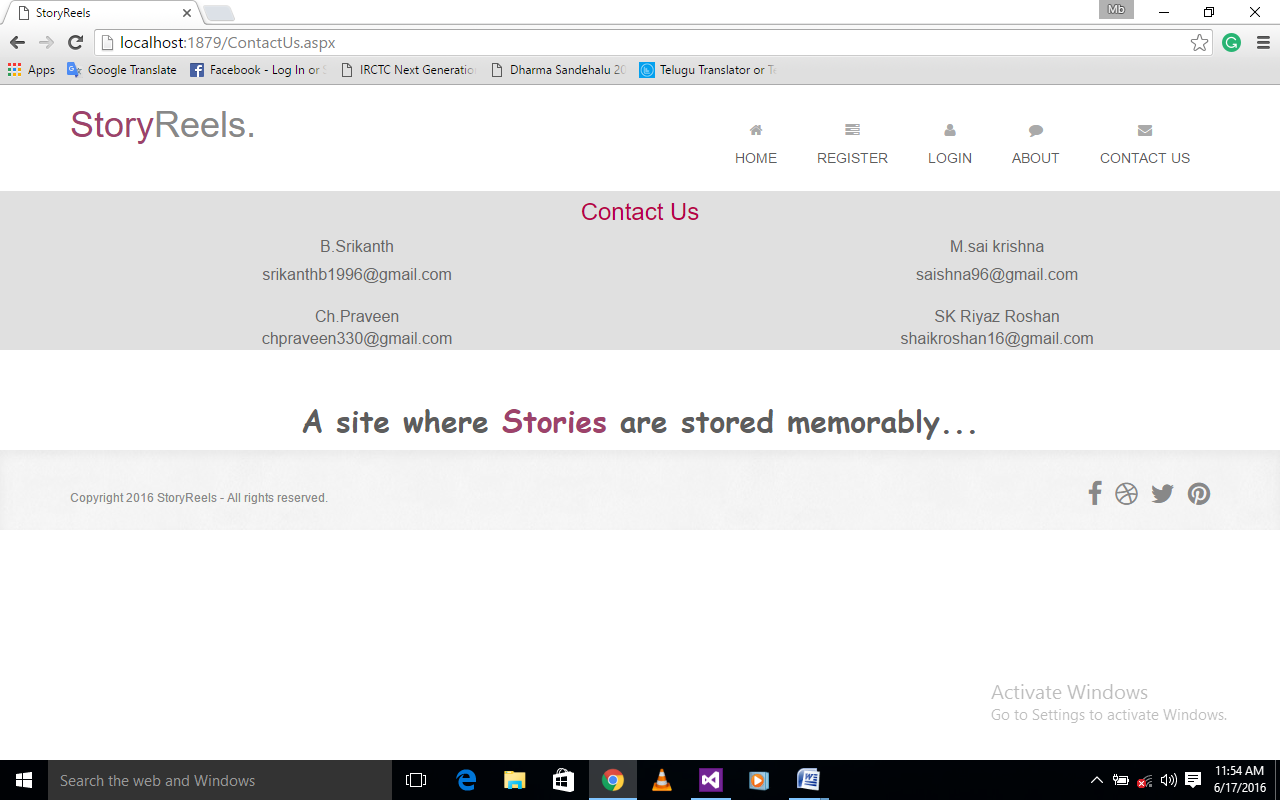
}

}

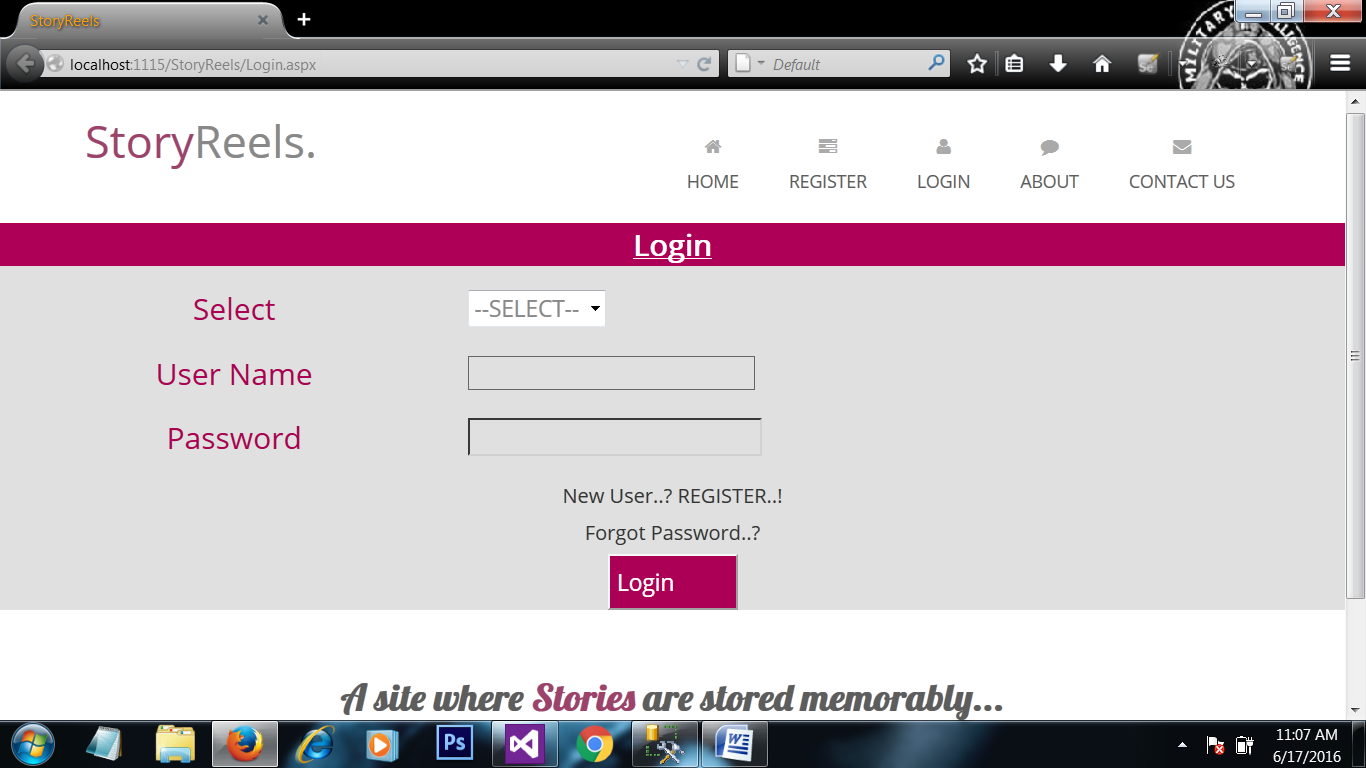
**7. Result**

**Register.aspxAboutus.aspx**

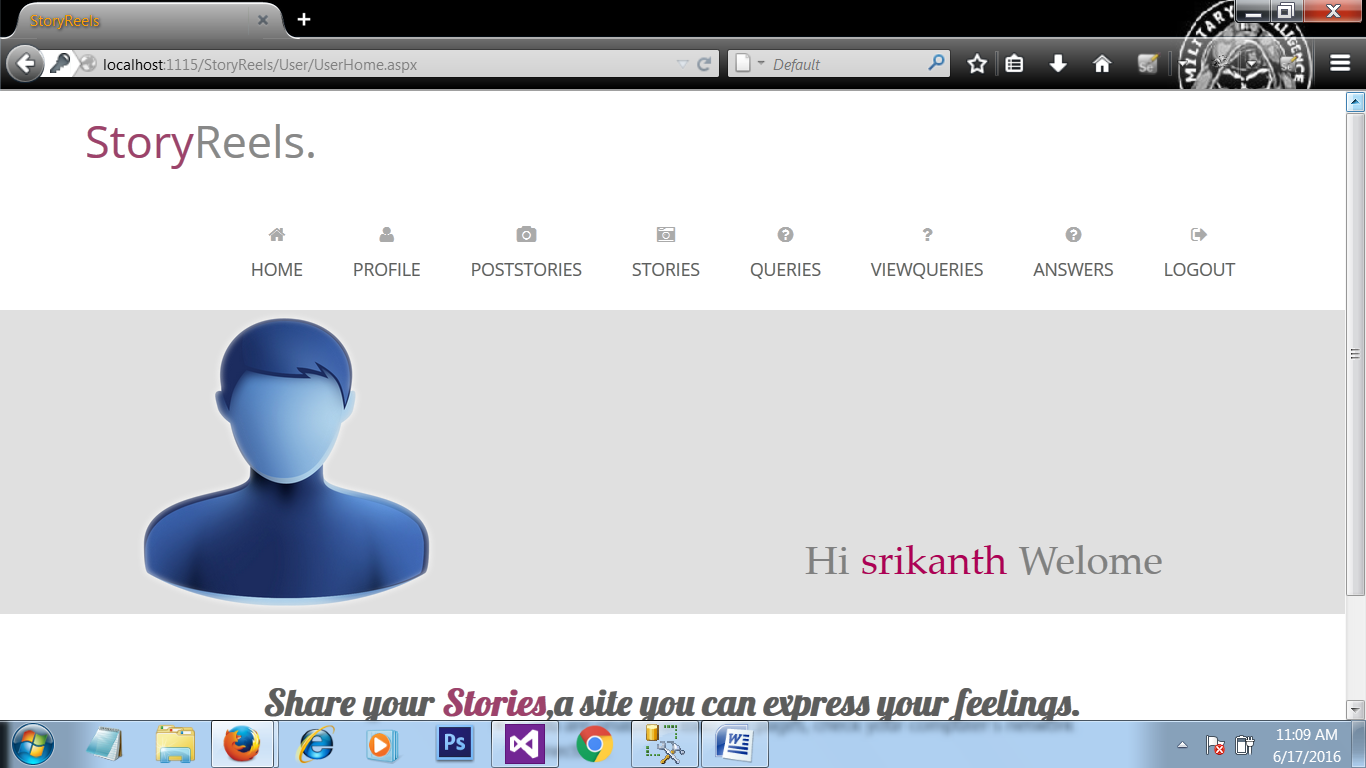
**Contactus.aspx**

****

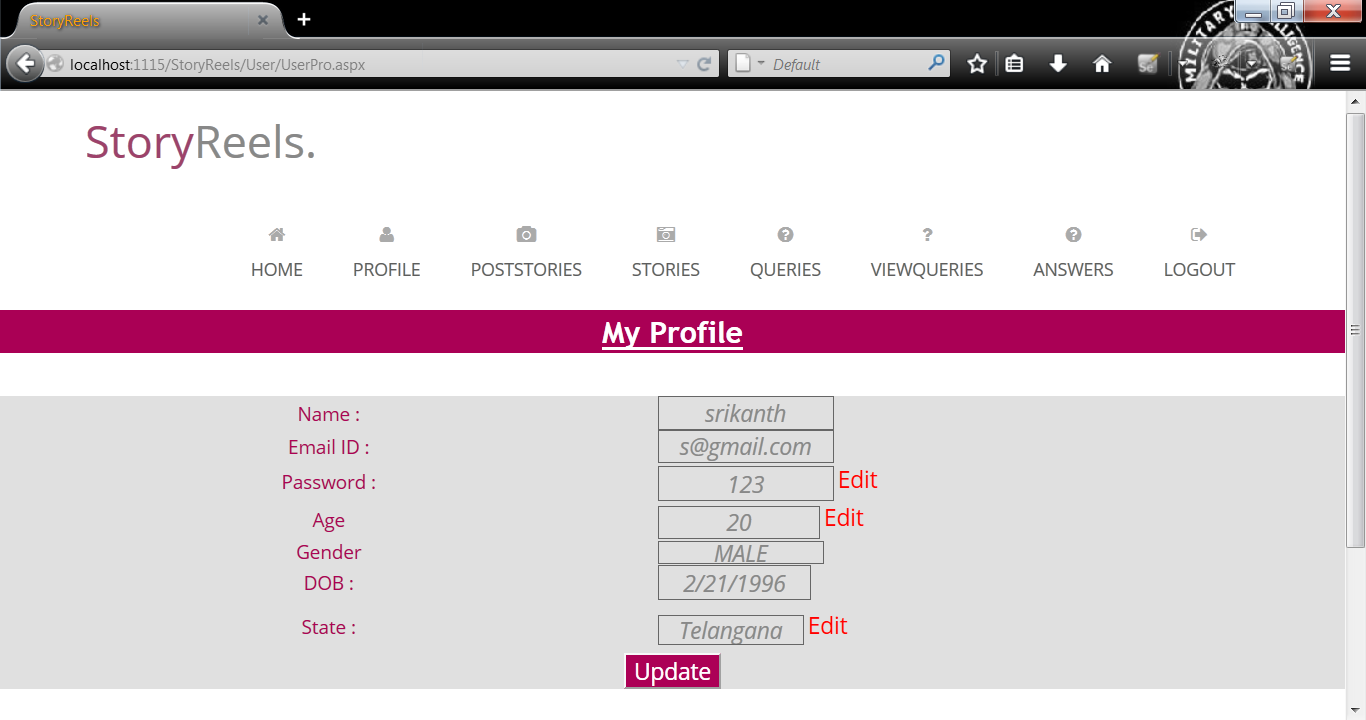
**Login.aspx**



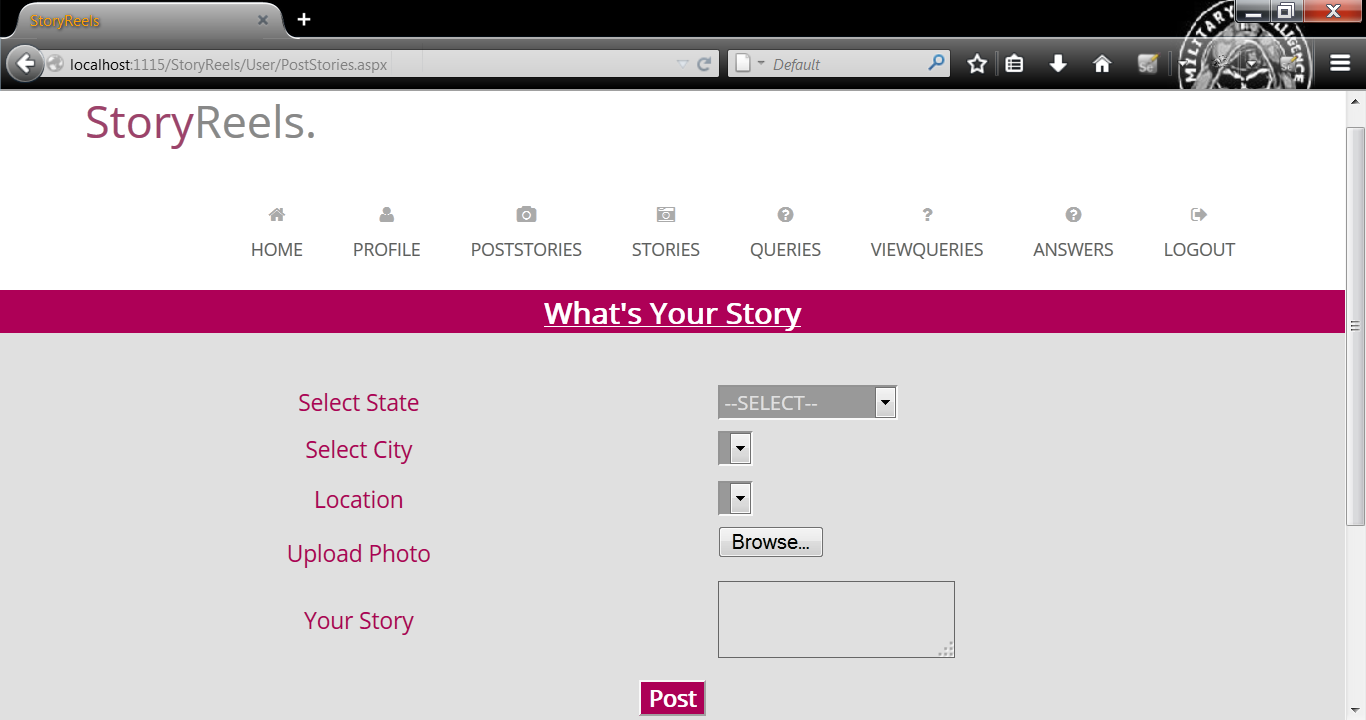
**UserHome.aspx**



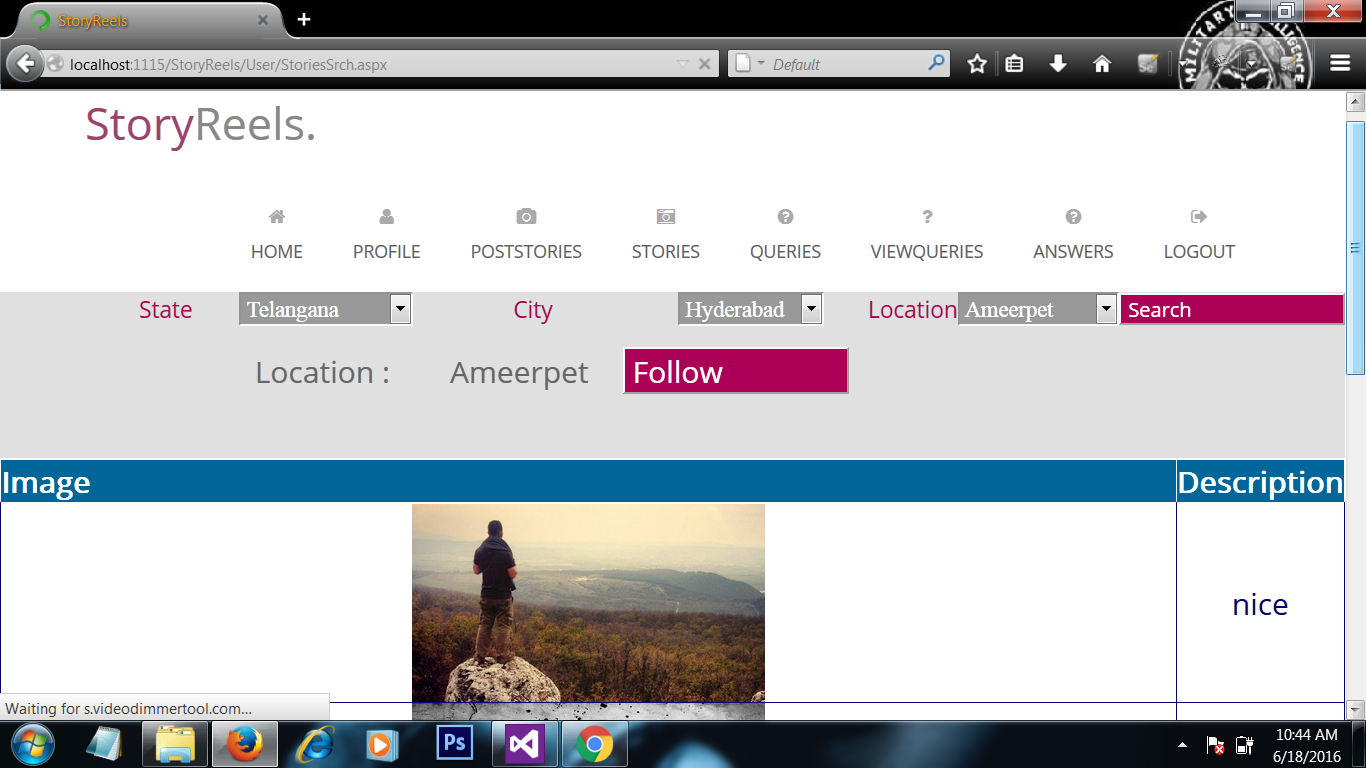
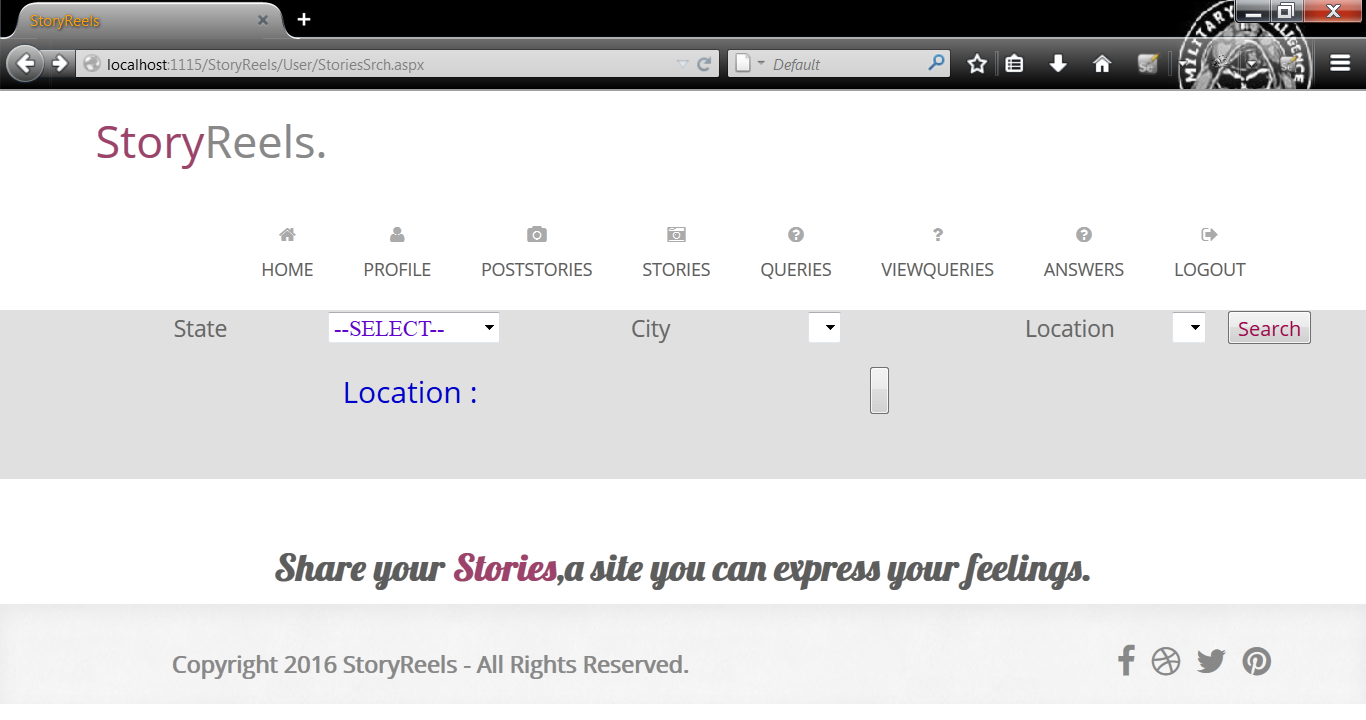
**UserProfile.aspx**



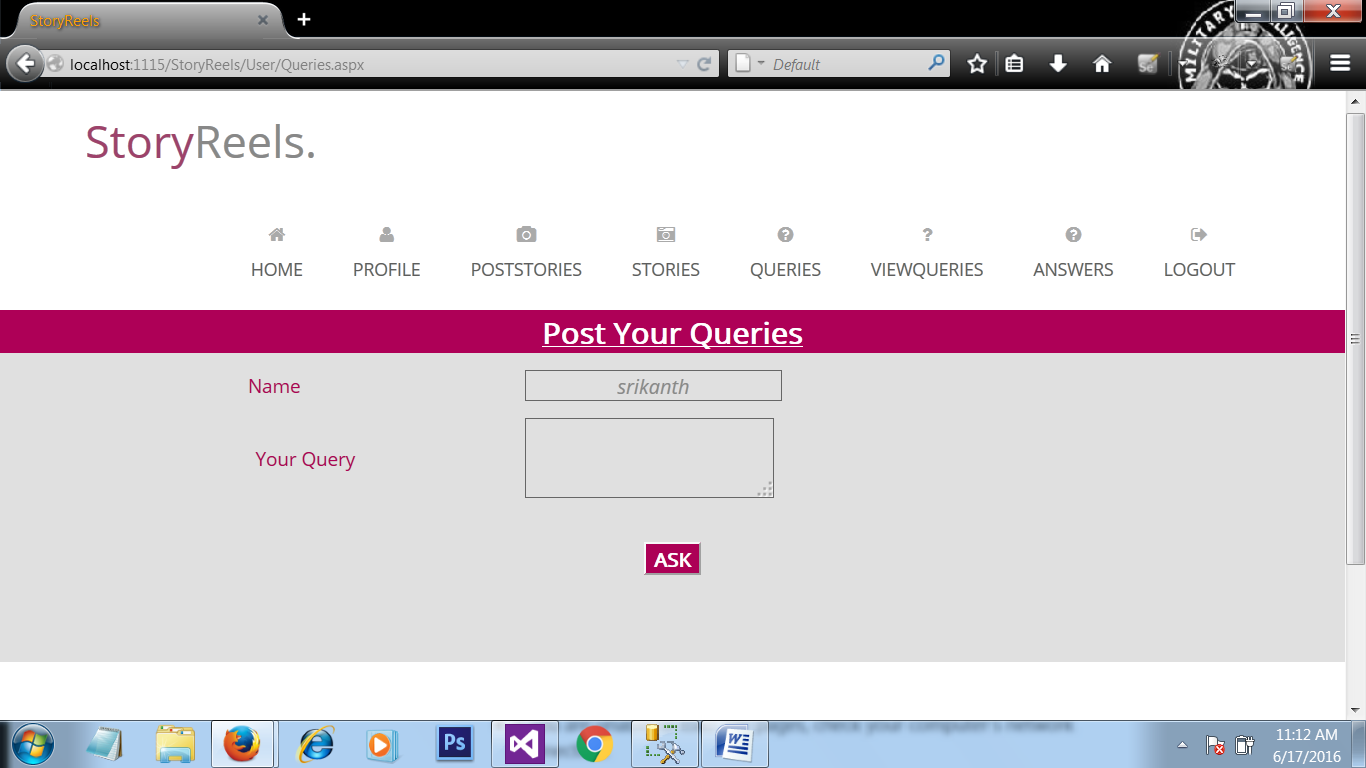
**PostStories.aspx**

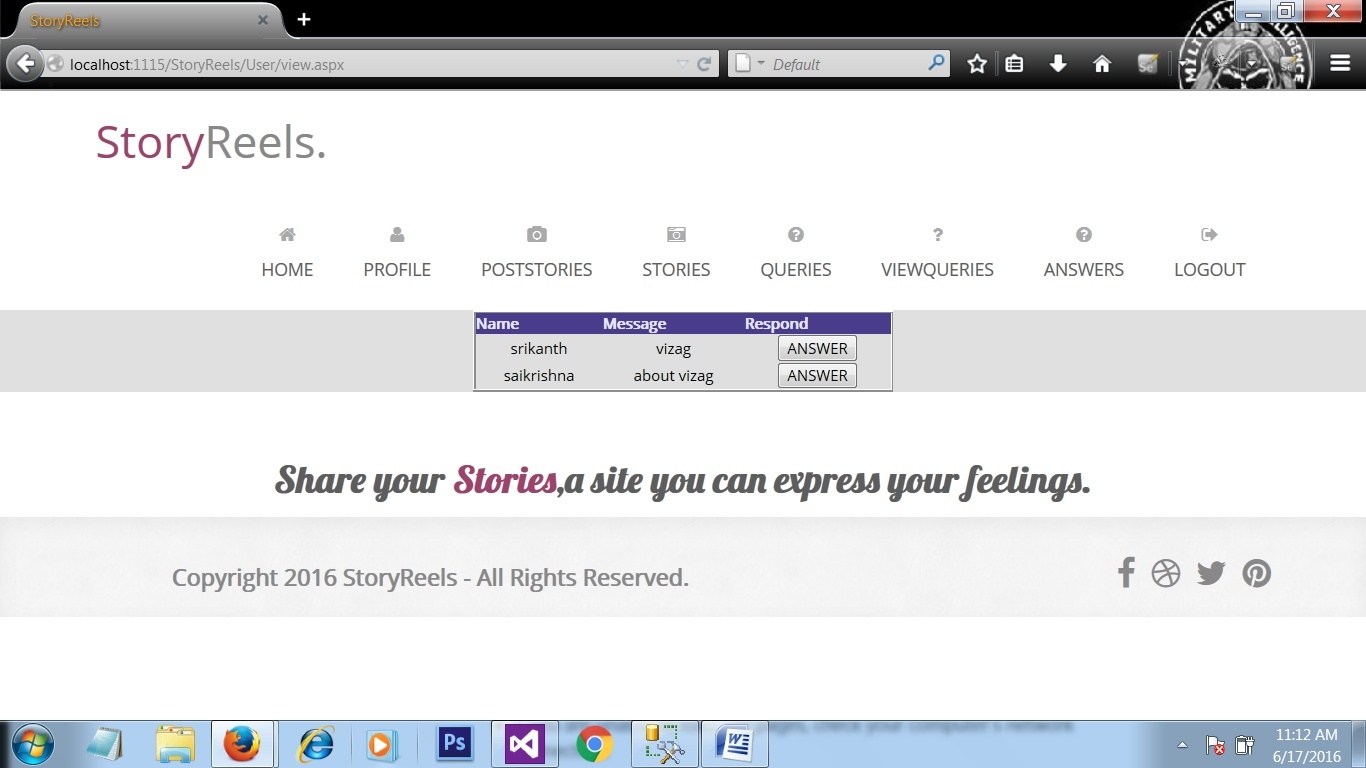


**StoriesSrch.aspx**

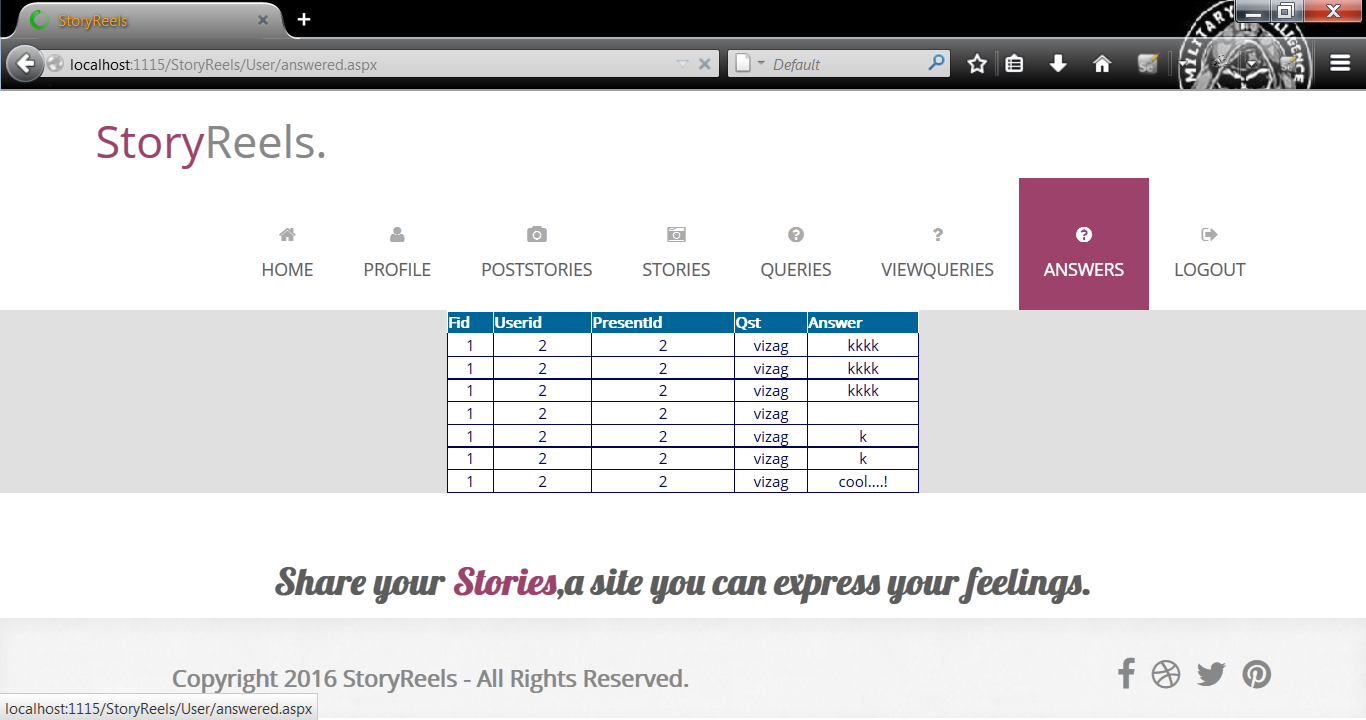


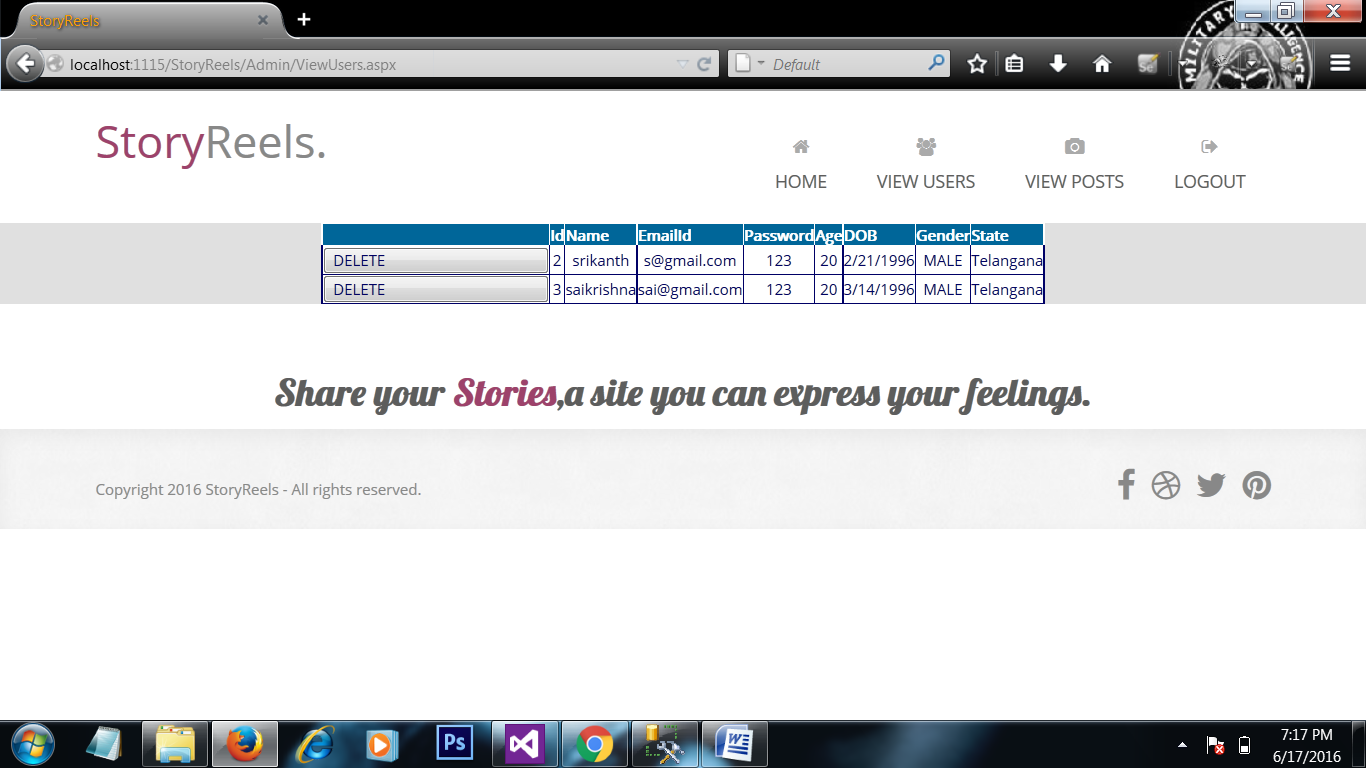
**Queries.aspx**

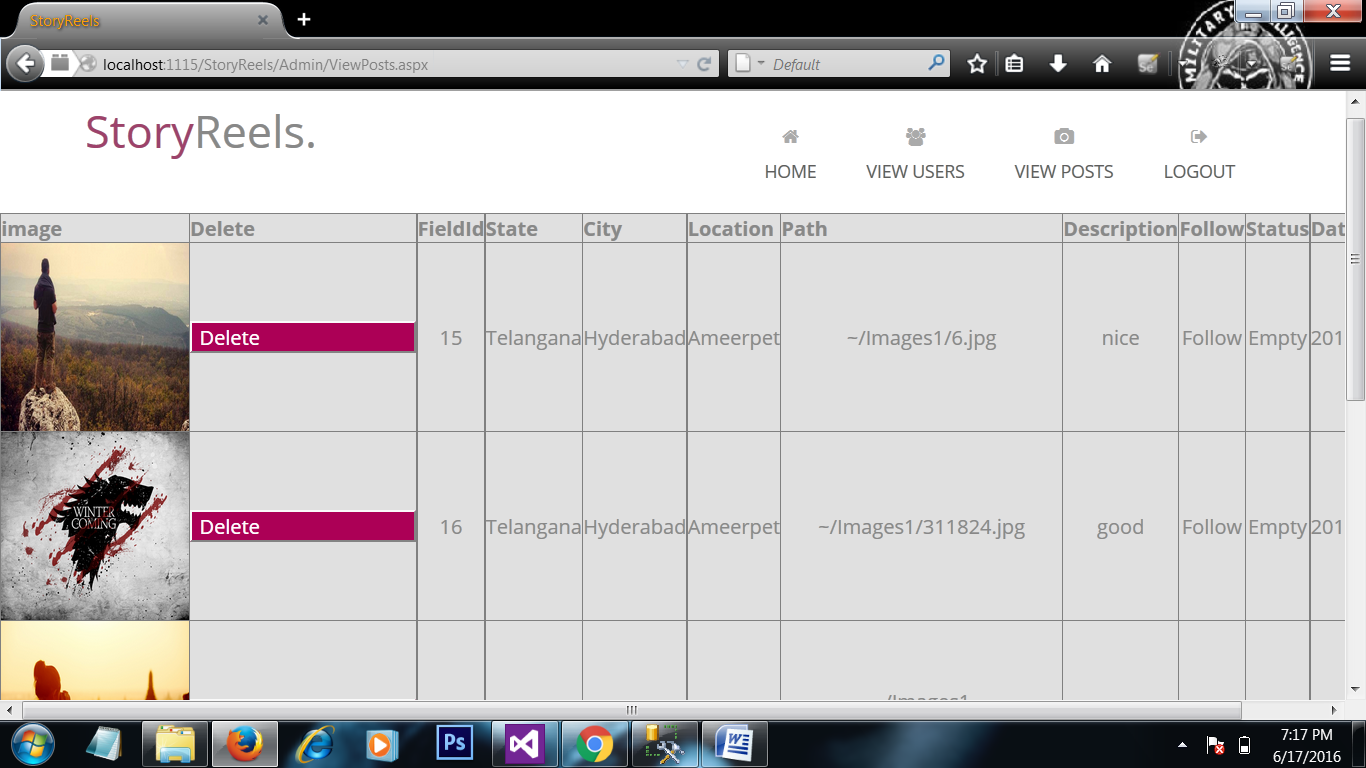


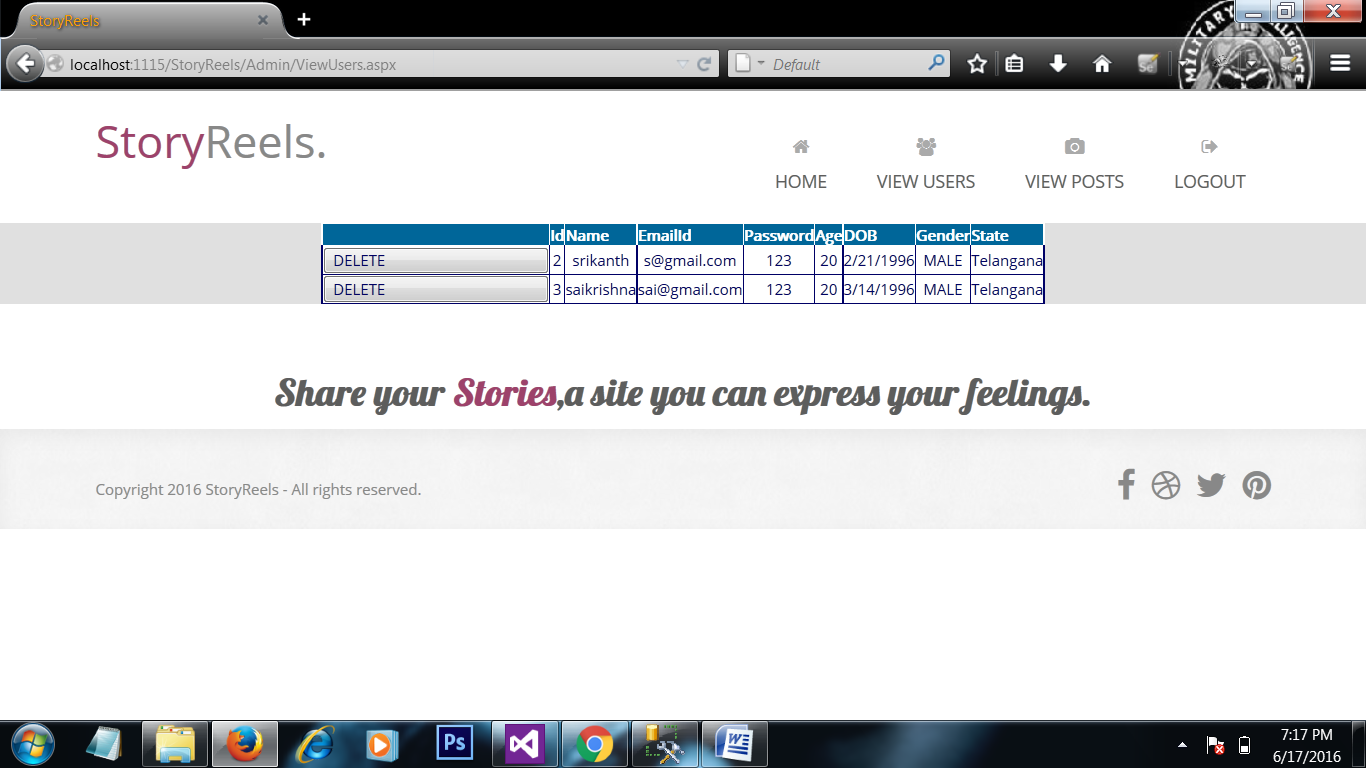
**View.aspx**

**Answered.aspx**





**Viewposts.aspx**

**viewUsers.aspx**

**8. TESTING**

**Testing:**

It is the process of testing the functionality and it’s the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as at undiscovered error. A successful test is one that uncovers an as at undiscovered error. Software testing is usually performed for one of two reasons.

* Defect Detection
* Reliability estimation

**Black Box Testing:**

The base of black box testing **s**trategy lies in the selection ofappropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system.

The following are the different types of techniques involved in black box testing. They are:

* Decision Table testing
* All pairs testing
* State transition tables testing
* Equivalence Partitioning

**Decision table testing:**

In this the variables are tested for both true and false cases. In this, All this type of such cases can be tested.

**All pairs Testing:**

In this all the input parameters which are be filled are checked for all possible combinations.

During different types of registration all the fields are available in the forms need to be filled by the users. If any one of the field is left unfilled an error message gets displayed. Throughout the code development phase, validation was not written for forms which led the user to leave some of the fields unfilled. By this test, testcase was identified and solved.

**State transition tables:**

By using this current state of the application and to which states the application will move based on input parameters.

**White Box testing:**

White box testing requires access to source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during unit testing phase.

In automation of PHC, the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

**Performance Testing:**

The performance of the application can be tested through this test. Since the number of code is less, performance is good.

**Database Testing:**

This test involves testing whether the values entered through the form gets stored in the database correctly or not.

**Acceptance Testing:**

Testing to verify a product meets customer specified requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtained.

**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.

**7.2 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.

**Test1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | ACTUAL OUTPUT | | CONDITION |
| Registration | Registration Successfully | Registration successfully | | Pass |
| Login | Login successfully | Login successfully | | Pass |
| Post Stories | Posted successfully | Posted Successfully | | Pass |
| Ask Queries | Query Posted successfully | Query Posted successfully | | Pass |
| Edit Profile | Profile Updated  Successfully | Profile Updated successfully | | Pass |
| Answer Queries | Answered successfully | | Answered successfully | Pass |

**Test2**

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | ACTUAL OUTPUT | CONDITION |
| Registration | Registration successfully | Registration Failed | Fail |
| Login | Login successfully | Invalid Username or Password | Fail |
| Post Stories | Posted successfully | Post Unsuccessfull | Fail |
| Ask Queries | Query Posted successfully | Query Not Posted | Fail |
| Answer Queries | Answered successfully | Not Yet Answered | Fail |

**8. CONCLUSION**

The project we developed is “Story Reels”. This system gives solutions to most of problems faced by the people who want to plan their journey or vacation. The system that we are going to introduce will address the problems accordingly.

This system is convenient, effective and easy there by, improving the performance of user’s who post in our web application. This system deals with database as an back end which is based on SQL server and its interface based on visual studio C#.

The system that we are going to introduce will address most of the problems that people are facing during the planning of their travel or vacation currently. The tasks that are now carried out currently will able to do with our system in more easy way. The data that are now kept in large physical files will be stored in the centralized database of the system. That will reduce the damages that can be happened to the data unexpectedly. Those features of introducing system will call upon the problems that we have encountered from the current system that is previlizing in the different places or locations now to make the tasks done at that user’s comfortability of planning their travell with much more efficiency.

It will also provide quality of service and customer satisfaction. Overall conclusion is that this is a fabulous online partial social web application.

**9. FUTURE SCOPE & ENHANCEMENTS**

**Future Scope:**

The administrator who is the owner of this web application can maintain all the details of different places , users and he can remove the user’s if he find’s any foul posts. Data Integrity and security has been provided. He can easily store and retrieve data whenever required.

The system implements the following functions: admin has the authority to change menu and has authority to view daily, weekly or monthly report on profits .Apart from these functions there may be some limitations in this system like synchronizing the photos of the locations you have added or Posted with the Maps.

**Enhancements:**

In future we will be working on providing the feature of synchronizing the photos of the locations you have posted with the Google Maps.