**CHAPTER 01**

**1. INTRODUCTION**

The College Tracking System is the Web Application which is used to search the various results for colleges in the single application. This Web application is built using DOTNET as the frontend and MySQL as the backend. It’s is a kind of search engine. A search engine, such as Google or Yahoo!, uses computer algorithms to search the Internet and identify items that match the characters and keywords entered by a user. Search engines are useful for finding information produced by governments, organizations, groups, and individuals. Examples of reliable information you can find through a search engine include freely available statistics published by a government agency or a freely available article published by a reputable news outlet.  This application can manage several activities such as Admin, Student Profile, Application Process, campus life, Careers & Majors, Student Lounge, Administration, Academics, Research and development cell, Admissions, Examinations, Foreign Relations, Photo Gallery, Affilated Colleges Portal, Training & Placement and Accounts & Management. This is very useful project for the colleges. The main aim of the project is to develop an application for manage all activities to search a college and make the things go easier.

Our college search tool lets you explore thousands of colleges throughout the Andhra Pradesh using specified criteria to help make your hunt for the perfect school much easier. You can search for a college by degree, location, student body size, special services, extracurricular activities and more.

If you were to visit the major college search websites, you would conclude that people have just a few considerations when choosing a college. Based on the search options common to eight of the most popular websites, students want to be able to pick a college based on its location, the location’s setting, size, tuition, selectivity, test scores, majors, whether it’s a community college or four-year institution,  and if it’s public or private.

The first question is whether or not these are the characteristics you should be using to pick a college.

Given that between financial aid and merit money few students pay full tuition, the tuition variable is pretty useless. A more specific average net cost or average financial aid award would be useful but isn’t found as a search option on most websites.

Public or private status is only moderately more useful than tuition since private colleges are likely to provide much more aid that public school. While there are instances where student have financial reasons to attend only public or private schools, it’s more likely to be used inappropriately as a cost filter.

Test scores fall into the necessary evil type information. Most public institutions use a combination of test scores and class rank for admission policies. Test scores are often used for calculating merit aid. Unfortunately, it isn’t always clear how a website is using the test score information in its search. Colleges are reporting 25th and 75th percentile scores so when you enter your score, which is it comparing it to? How close do the school’s scores have to be to yours to be considered a match?  What would be more useful, especially for those students looking for merit aid, is to know the 75th% scores and GPA.

Selectivity is another tricky category. Most people would take it to mean the percentage of students admitted to an institution. And that was true for all the websites I look at except for one.

Location setting would seem to be a useful variable. After all, students want to go to colleges where there lots of opportunities for entertainment and cultural activities outside of campus. At least they think they do but that’s another issue entirely. The problem is how schools are classified into settings.

That leaves college size, location, major, and whether it’s a four-year or two-year institution to choose from. Obviously, students need to be able to distinguish between a community college and a four-year institution. And it’s pretty hard for a search site to mess up selecting states to include in the search although one does.

Before you make a final decision as to where you would like to attend college, there are a variety of important factors to consider. Below we will outline some of these factors to help you determine which is most important to you.

**EXISTING SYSTEM:**

After completion of EAMCET Examination, Convener will declare the results and ranks to the students. After that, EAMCET rank holders should go to the help line centers for certificate verifications as per the schedule issued by convener. After completion of the certificate verification for all the rank holders, they will attend web based counseling for choosing their best college. Each student can opt for 3000+ options for getting admission into their selected college. After completion of this exercise from 1 to lank holder, then convener will release the college allotments to each and every rank holder who attend the web based counselling. After allotting college to rank holder, students should down load the allotment order and challan and should go to bank and pay the challan (as per the allotment order).

Along with the allotment order, challan and original certificates, student should go to allotted college and submit all certificates and challan college copy to admission officer of college. Then student should fill the application form manually. Here students may fill 3 to 4 applications manually, i.e. admission application form and anti-ragging form, transport , hostel application forms may also. All these forms should fill manually.

**PROPOSED SYSTEM:**

In this application all the work is maintained in a systematic mannet manner and managed in database. This reduces the paper work and human effort. it is easy to access all the information of colleges by using this website. In this students are able to search the information of colleges from any location.

**CHAPTER 02**

**LITERATURE SURVEY**

**2.1 SURVEY OF TECHNOLOGY**

**INTRODUCTION**

**2.1 .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.

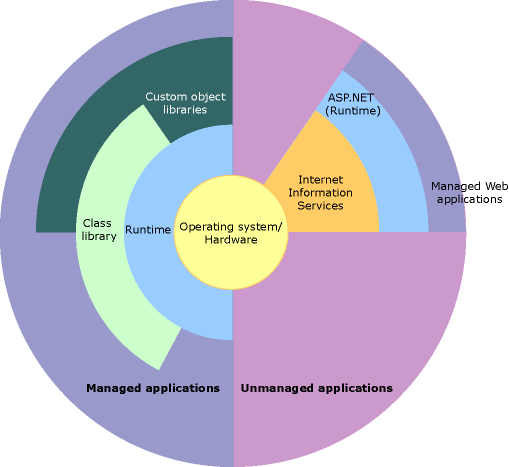
You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that promote security and robustness. In fact, the concept of code 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.



*Figure 2.1: .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 featuring 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).

**2.1.2 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.



*Figure 2.2: Architecture of C# in .Net*

**2.1.3 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 Flavours 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 advntage of the large number of components available for rapid application development.

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

**ASP.NET MVC**

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

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

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

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

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

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

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

**ASP.NET Compiler**

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

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

**2.1.4 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.

**ADVANTAGES:**

## Time Savings

## A search engine saves you time in two ways: by eliminating the need to find information manually, and by performing searches at high speeds. Without a search engine, you would have to look at sites one by one and pore over the contents of each carefully – a tedious prospect. A search site automatically compares your criteria to billions of Web pages and gives you results in a fraction of a second. You can perform dozens of searches in the course of a few minutes, altering the criteria as you narrow down results.

## Relevance

When a search engine scans a website, it scores the content for relevance to particular search words. For example, the site, "Joe's Pizza" scores higher for the search term "pizza restaurant" than the site "Acme Pizza Dough Distributors." Although search companies keep their ranking formulas secret, they take into consideration factors such as repetition of related words and links from other sites. A search engine sorts its results page by relevance to your criteria, with the score in descending order. You see the highest-scoring results at the top; as you move down the list, websites become less relevant.

## Free Access

Some search engines, such as LexisNexis, specialize in legal or other specialized, scholarly information; these sites charge a fee to use their services. Google, Bing and Yahoo pay for their operations through advertising; searches are free to the user, without restrictions for the information you seek, the time spent on the site or the number of searches you perform. Although this benefits all users, it is a particular advantage for students, job seekers and others of limited means.

## Comprehensive

Search engines scan the entire Web and keep comprehensive data on every page they catalog.

Because search companies hold so much data, they help you find obscure sites about which you would not otherwise know. Search results are more likely to give you too much information rather than too little.

## Advanced Search

In addition to keywords, search engines let you use advanced search options to refine your results. These options help make your searches more flexible and sophisticated. For example, to exclude results containing a certain word, type a minus sign before the word. To look for an exact phrase, surround it with quotation marks. When you want to search only a specific site or group of sites, type "site:" without quotes followed by the site's Web address. You can, for example, search for mentions of beer on Facebook by typing, "beer site:www.facebook.com." To search all college and university sites for nuclear physics, use, "nuclear physics site:.edu."

Before you make a final decision as to where you would like to attend college, there are a variety of important factors to consider. Below we will outline some of these factors to help you determine which is most important to you.

**Search College by Location:**

Clarifying where you would like to go to college can help to narrow down your search using our advanced tool.

For some students, remaining close to home and having the option to return from time to time is a significant factor. For others, having a short trip home for an occasional visit is not as important, and some may even prefer to attend school out-of-state. Using our college search tool, you can create a list of the schools in your desired area.

**Search College by Degree:**

Different schools specialize in different areas of study, which is why it is a good idea to put together a list of possible majors you are interested in pursuing before utilizing our college search tool. Once you have established a number of these potential areas of study, you can then search colleges by the degree you are interested in ultimately obtaining.

Many schools offer several notable programs, so when performing your [search for a college by degree](http://www.collegeview.com/collegesearch/areaofstudy.jsp), you may notice that, if it is a similar area of study, some of the same schools might appear on your list more than once. Remember to take note of these schools as well as other institutions which appeal to you while using our college search tool. A number of college-bound students are unaware of the area of study they eventually pursue before they go to college, which is why it is a good idea to keep your options open when you search for a college by degree.

[**Search College by Student Body Size**](http://www.collegeview.com/collegesearch/studentbodysize.jsp)**:**

Another great option offered from our college search tool is the ability to find schools based on student body size. Colleges range in size from a few hundred students to several thousand students, and one of the important factors to consider when deciding which school to attend is the size. Determining the size of the college or university you wish to attend will allow you to concentrate on only those schools which meet your wants and needs.

[**Private vs. Public College**](http://www.collegeview.com/collegesearch/public_private.jsp)**:**

Have you decided whether you would like to [attend a private vs. public college](http://www.collegeview.com/collegesearch/public_private.jsp)? Both types of institutions have their advantages and disadvantages. Typically, private colleges are more expensive than public colleges due to the fact that they receive much less taxpayer money and are forced to rely more heavily on tuition for financial support.

The advantage of a private vs. public college, however, is that they are generally smaller and more able to offer individualized attention.

While public colleges tend to be larger and less expensive than private colleges, this is not always the case. Using our college search tool, you can create a private vs. public college list to examine the cost, student body size and more of different schools to help you decide which type of school is right for you.

[**Search for College by Special Services**](http://www.collegeview.com/collegesearch/specialservices.jsp)

The special services offered from different schools are another important factor to consider. These services include the following:

* Career counselling
* Student employment
* Internships (with pay)
* Internships (without pay)
* Job placement
* Study abroad (for credit)
* Study abroad (non-credit)
* Children’s day care

Students who come from smaller towns may feel overwhelmed and out of place at a large university. Students from larger cities, on the other hand, may feel as if there are not enough diversity or available activities at a smaller school.

Using our free college search tool, you have the ability to choose your ideal student body size, whether it is less than 500 students, greater than 15,000 students, or anything in between.

Our [customizable college search tool](http://www.collegeview.com/collegesearch/specialservices.jsp) lets you select any combination of these special services to create a tailored list of schools which best meet your particular needs.

Determining which of these special services are offered is important for a variety of reasons. Many students do not know which career path they will pursue once they have completed their college education. A career counselor can help you make this decision. Also, internships serve as a great way to obtain valuable experience before pursuing your chosen career. It is a good idea to reveal which schools offer internships, both with and without pay.

The more criteria you select when using our college search tool, the more customized your list becomes, and the easier it becomes to narrow down your top selections.

[**More Criteria of Our College Search Tool**](http://www.collegeview.com/collegesearch/advSearch.jsp)

Our college search tool lets you select from even more criteria than those listed above. These criteria include the following:

* City size
* School type
* Religious affiliation
* Student body type
* Ethnicity
* Athletics
* Disabilities

We want to make your search for the right college as easy as possible. Ultimately deciding on one school can be a difficult task. Let us help. [Use our free college search tool](http://www.collegeview.com/collegesearch/) today and begin your hunt for the perfect school for you.

**CHAPTER 03**

**REQUIREMENT SPECIFICATION**

**3.1 SOFTWARE AND HARDWARE REQUIRMENTS**

* Operating system : - Windows 8.
* Coding Language : ASP.NET,C# Programming
* Data Base : SQL Server

**Software specifications**

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

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

**Hardware requirements:**

• Processor : INTEL P4 or above

• Hard Disk : 40 GB.

• RAM : 2 GB

• Processor Speed : 1.54 GHZ

**CHAPTER 04**

**DESIGN**

**4.1 UML DIAGRAMS**

The Unified Modelling Language (UML) is a graphical language for visualizing, specifying, constructing and documenting of a software intensive system. The UML gives a standard way to standard way to write a system blueprints, covering conceptual things, such as classes written in a specified programmed language, database schemas and reusable software components.

There are types of diagrams

* Class Diagram
* Object Diagram
* Use case Diagram
* Sequence Diagram
* Collaboration Diagram
* State chart Diagram
* Activity Diagram
* Component Diagram
* Deployment Diagram

**4.1.1 Use Case Diagram**

* Use case is a description of set of sequence of actions that a system performs that yields an observable results. These are important in organizing the behaviors of a system.
* Use case is used to structure the behavioral things in a model.
* A Use case is realized by collaboration.
* Graphically , a use case is rendered as a ellipse with solid lines, usually including only its name

**ACTOR**- also referred to as roles. Name and stereotype of an actor can be changed in its properties lab

**Use Cases-** these can have extension points.

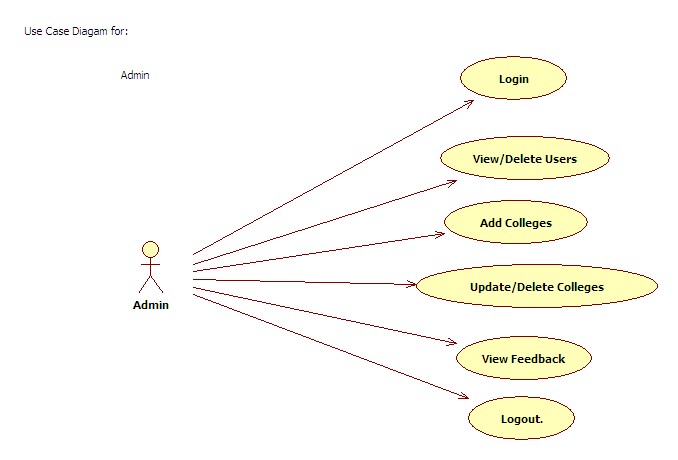


Fig:4.1.1 - Use case design for Admin

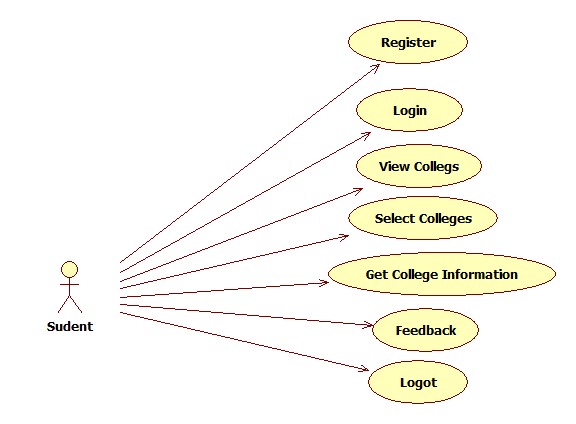


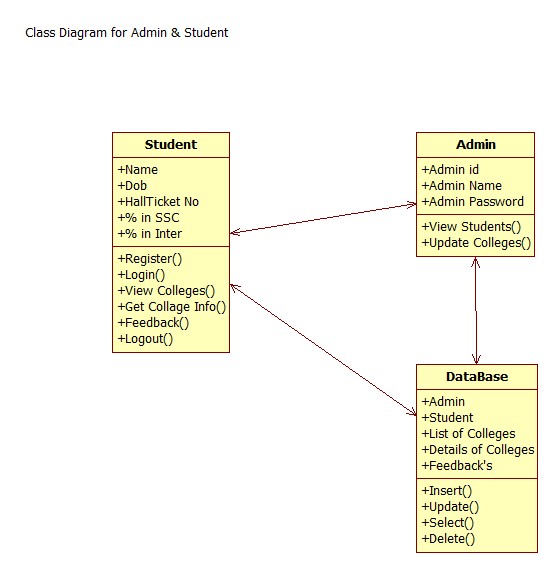
Fig:4.1.2 - Use case design for student

**4.1.2 CLASS DIAGRAM**

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing , describing documenting different aspects of a system but also for constructing executable code of the software application. The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented system because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaboration and constraints. It is also known as ***structural diagram***.

Fig:4.1.3 - Class Diagram for Admin & Student



**4.1.3 SEQUENCE DIAGRAM**

A Sequence diagram is an interaction diagram that emphasizes the time ordering of messages. A Sequence diagram shows a set of objects and messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other thing such as collaboration, components and nodes. These are used to illustrate the dynamic view of a system. It is isomorphic, which means taking any one of them in to consideration, and can be transformed to another.

A Sequence diagram shows, as parallel vertical lines (*Life lines*), a different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the in which they occur.

This allows the specification of simple runtime scenarios in a graphical manner. In order to show the sequence of time ordering of messages involved in the working of PHC a sequence diagram was used.

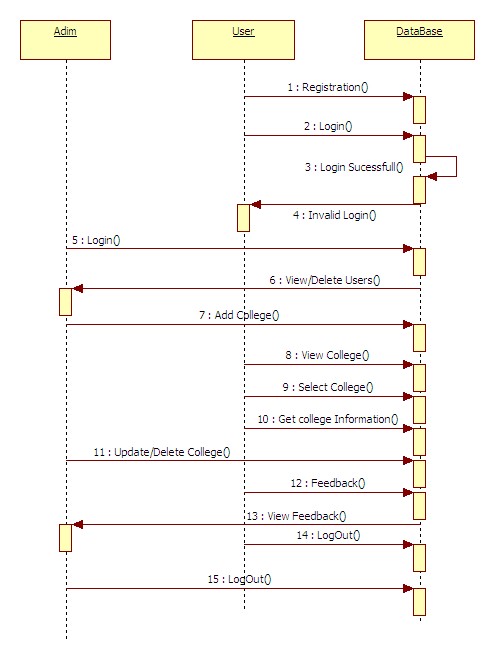


Fig:4.1.4 – Sequence Diagram

**4.1.4 COLLABORATION DIAGRAM**

A Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Collaboration diagrams are isomorphic, meaning that you can take one and transform it in to another

Graphically, collaboration is rendered as an ellipse with dashed lines, usually including only its name

**4.1.5 ACTIVITY DIAGRAMS**

Activity diagrams are graphical representations of work flows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes. Activity diagrams show the overall flow of control.

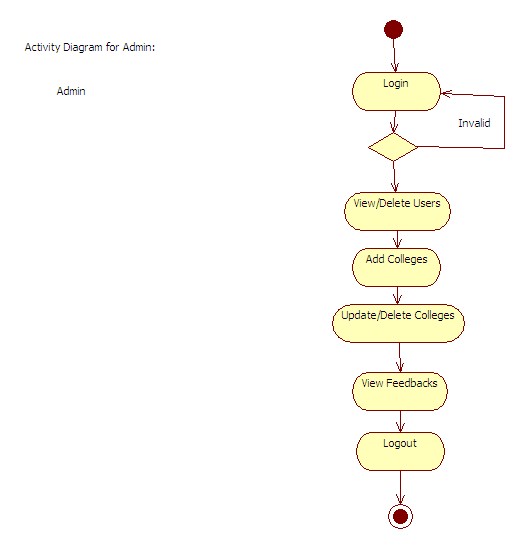


Fig:4.1.5 - Activity Diagram for Admin

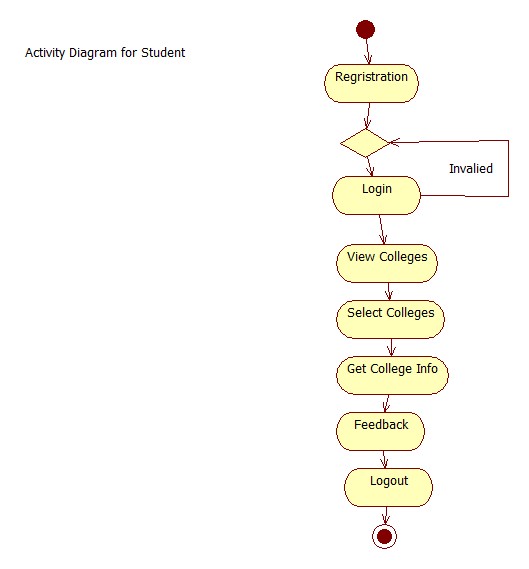


Fig:4.1.6 - Activity Diagram for Student

**CHAPTER-5**

**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.

The application is completely written .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.

**5.1: TECHNOLOGY USED:**

**5.1.1: 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.

**5.1.2: 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.

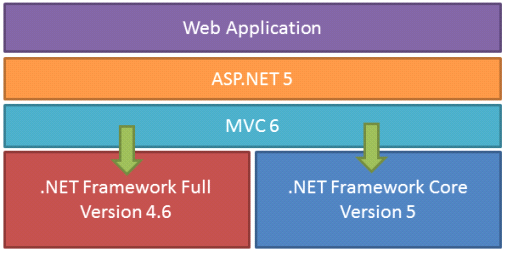
The outcome of the project is what we now know as .NET. Even though .NET has borrowed most of its ideas from Sun's J2EE, it has really outperformed their competitors.

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.

**5.1.3: 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](https://en.wikipedia.org/wiki/Standard_library) class libraries. The class library is organized in a hierarchy of [namespaces](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/List_of_CLI_languages). .NET Framework class library is divided into two parts: [Framework Class Library](https://en.wikipedia.org/wiki/Framework_Class_Library) (FCL) and [Base Class Library](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/.NET_Compact_Framework), [Microsoft Silverlight](https://en.wikipedia.org/wiki/Microsoft_Silverlight) and [Mono](https://en.wikipedia.org/wiki/Mono_(software)).

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](https://en.wikipedia.org/wiki/Windows_Forms), [ADO.NET](https://en.wikipedia.org/wiki/ADO.NET), [ASP.NET](https://en.wikipedia.org/wiki/ASP.NET), [Language Integrated Query](https://en.wikipedia.org/wiki/Language_Integrated_Query) (LINQ), [Windows Presentation Foundation](https://en.wikipedia.org/wiki/Windows_Presentation_Foundation) (WPF), [Windows Communication HYPERLINK "https://en.wikipedia.org/wiki/Windows\_Communication\_Foundation"Foundation](https://en.wikipedia.org/wiki/Windows_Communication_Foundation) (WCF) and [Workflow Foundation](https://en.wikipedia.org/wiki/Workflow_Foundation) (WF). BCL is much larger in scope than standard libraries for languages like [C++](https://en.wikipedia.org/wiki/C%2B%2B), and comparable in scope to [standard libraries of Java](https://en.wikipedia.org/wiki/Java_Class_Library).

**.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]](https://en.wikipedia.org/?title=.NET_Framework) 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 *G*eneration *0*. The objects that survive a garbage collection are tagged as *G*eneration *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 Native 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.

**SAMPLE CODE:**

**Registration:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

usingSystem.Data.SqlClient;

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

{

Class1 o = new Class1();

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

{

}

protected void Button1\_Click1(object sender, EventArgs e)

{

try

{

stringqry = "insert into Regristration values('" + txtName.Text + "','" + txtUname.Text + "','" + txtMid.Text + "','" + txtPw.Text + "','" + DropDownList1.SelectedItem.Text + "','" + txtDob.Text + "')";

inti = o.inupdel(qry);

if (i> 0)

{

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

txtName.Text = txtUname.Text = txtMid.Text = txtPw.Text = txtCpw.Text = DropDownList1.SelectedItem.Text = txtDob.Text = " ";

}

else

{

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

}

}

catch (Exception exe)

{

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

}

}

}

**Login:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

usingSystem.Data.SqlClient;

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

{

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

{

}

Class1 obj = new Class1();

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

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

{

try

{

string qry = "select \* from Regristration where Mailid='" + txtUid.Text + "' and Password='" + txtPw.Text + "'";

DataSet ds = obj.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["UserName"] = ds.Tables[0].Rows[0][2].ToString();

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

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

}

else

{

Response.Write("<script>alert('Invalid Email ID and Passowrd')</script>");

}

}

catch (Exception ex)

{

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

}

}

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

{

try

{

if (txtUid.Text == "admin" &&txtPw.Text == "admin")

{

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

}

}

catch (Exception)

{

Response.Write("<script>alert('Invalid Username and Password')</script>");

}

}

}

}

**Add College:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data.SqlClient;

usingSystem.Data;

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

{

Class1 o = new Class1();

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

{

}

intcn=1;

string s1;

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

{

try

{

inti=cbl1.Items.Count;

foreach (Object item in cbl1.Items)

{

if (cn - 1 <i)

{

s1 += item.ToString() + ",";

}

else

{

s1 += item.ToString();

}

cn++;

}

stringqry = "insert into ColgInfo values('" + ddl1.SelectedItem.Text + "','" + ddl2.SelectedItem.Text + "','" + txtLoc.Text + "','" + txtCName.Text + "','" + s1.ToString() + "','" + txtFee.Text + "','" + txtStaff.Text + "')";

int j = o.inupdel(qry);

if (j > 0)

{

Response.Write("<script>alert('College Added Succesful')</script>");

txtLoc.Text = txtCName.Text =txtFee.Text = txtStaff.Text = " ";

}

else

{

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

}

}

catch (Exception exe)

{

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

}

}

}

**Add Feedback:**

using System;

usingSystem.Collections.Generic;

usingSystem.Data;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

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

{

Class1 o = new Class1();

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

{

try

{

String qry = "Select CollegeName from ColgInfo";

DataSet ds = o.select(qry);

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

{

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

foreach (DataRowrd in ds.Tables[0].Rows)

{

ddlColg.Items.Add(rd["CollegeName"].ToString());

}

}

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

}

catch (Exception ex)

{

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

}

}

protected void ImageButton1\_Click(object sender, ImageClickEventArgs e)

{

try

{

stringqry="insert into Feedback values('"+Session["Id"].ToString()+"','"+ddlColg.SelectedItem.Text+"','"+txtname.Text+"','"+txtFeed.Text+"')";

inti = o.inupdel(qry);

if (i> 0)

{

}

}

catch (Exception)

{

throw;

}

}

}

**Class File:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Data.SqlClient;

usingSystem.Data;

/// <summary>

/// Summary description for Class1

/// </summary>

public class Class1

{

SqlConnectioncn = new SqlConnection("Data Source=.;Initial Catalog=CollegeTrackingSystem;Integrated Security=True");

publicintinupdel(string qry)

{

SqlCommandcmd = new SqlCommand(qry, cn);

cn.Open();

inti = cmd.ExecuteNonQuery();

cn.Close();

returni;

}

publicDataSet select(string qry)

{

SqlCommandcmd = new SqlCommand(qry, cn);

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataSet ds = new DataSet();

da.Fill(ds);

return ds;

}

publicint login(string qry)

{

SqlCommandcmd = new SqlCommand(qry, cn);

cn.Open();

inti = (int)cmd.ExecuteScalar();

cn.Close();

returni;

}

}

**Search College:**

using System;

usingSystem.Collections.Generic;

usingSystem.Data;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

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

{

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

{

}

protected void DropDownList1\_SelectedIndexChanged(object sender, EventArgs e)

{

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

{

ddlcity.Items.Clear();

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

ddlcity.Items.Add("Hyderabad");

}

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

{

ddlcity.Items.Clear();

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

}

}

Class1 o = new Class1();

protected void ImageButton1\_Click1(object sender, ImageClickEventArgs e)

{

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

DataSet ds = o.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

}

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

{

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

DataSet ds = o.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

}

}

**View College:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

usingSystem.Data.SqlClient;

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

{

Class1 o = new Class1();

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

{

if (!IsPostBack)

{

bind();

}

}

public void bind()

{

stringqry = "select \* from ColgInfo";

DataSet ds = o.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 Button\_Click(object sender, EventArgs e)

{

Button img = (Button)sender;

GridViewRow gr = (GridViewRow)img.NamingContainer;

Label lbl = (Label)gr.FindControl("Label1");

string qry = "delete from ColgInfo where Id='" + lbl.Text + "'";

inti = o.inupdel(qry);

if (i> 0)

{

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

bind();

}

else

{

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

}

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

**View User:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Web;

usingSystem.Web.UI;

usingSystem.Web.UI.WebControls;

usingSystem.Data;

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

{

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

{

if (!IsPostBack)

{

bind();

}

}

public void bind()

{

stringqry = "select \* from Regristration";

DataSet ds = o.select(qry);

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

{

GridView1.DataSource = ds;

GridView1.DataBind();

}

else

{

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

}

}

Class1 o = new Class1();

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

{

Button img = (Button)sender;

GridViewRow gr = (GridViewRow)img.NamingContainer;

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

string qry = "delete from Register where Id='" + lbl.Text + "'";

inti = o.inupdel(qry);

if (i> 0)

{

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

bind();

}

else

{

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

}

}

protected void GridView1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

}

**Web Configuration:**

<?xml version="1.0"?>

<!--

For more information on how to configure your ASP.NET application, please visit

http://go.microsoft.com/fwlink/?LinkId=169433

-->

<configuration>

<system.web>

<compilation debug="true" targetFramework="4.0"/>

</system.web>

<appSettings>

<add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>

</appSettings>

</configuration>

**CHAPTER-6**

**TESTING**

**6.1 Introduction**

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

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

**6.2 TESTING STRATEGIES**

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

1. Unit Testing

2. Integration Testing

3. Validation Testing

4. System Testing

**6.2.1 UNIT TESTING**

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

Each module can be tested using the following two Strategies:

**Black Box Testing:**

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

**White Box testing:**

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

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

**6.2.2 INTEGRATION TESTING**

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

**6.2.3 VALIDATION TESTING**

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

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

**6.2.4 SYSTEM TESTING**

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

**6.3 TEST CASES**

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

**Test Case Reports:**

**Test1:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | ACTUAL OUTPUT | | CONDITION |
| Registration | Registration Successfully | Registration successfully | | Pass |
| Login | Login successfully | Login successfully | | Pass |
| Search Colleges | Search successfully | Search Successful | | Pass |
| Student Feedback | Added successfully | Added successfully | | Pass |
| View Student | Viewed successfully | Viewed successfully | | Pass |
| Add Colleges | College Added successfully | | College Added successfully | Pass |
| Delete Colleges | Deleted Successfully | | Deleted successfully | Pass |
| View Feedback | Viewed successfully | | Viewed successfully | Pass |

T**able: 6.7.1 Positive Test Cases for Multi Banking Application of .NET documents**

**Test2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASES | EXPECTED OUTPUT | | ACTUAL OUTPUT | CONDITION |
| Registration | Registration Successfully | | Registration unsuccessfully | Fail |
| Login | Login successfully | | Login unsuccessfully | Fail |
| Search Colleges | Search successfully | | Search unsuccessfully | Fail |
| Student Feedback | Added successfully | | Added unsuccessfully | Fail |
| View Student | Viewed successfully | | Viewed unsuccessfully | Fail |
| Add Colleges | College Added successfully | CollegeAdded unsuccessfully | | Fail |
| Delete Colleges | Deleted Successfully | Deleted unsuccessfully | | Fail |
| View Feedback | Viewed successfully | Viewed unsuccessfully | | Fail |

**Table: 6.7.2 Negative Test Cases for a Multi Banking Application for .NET documents**

All the above validations on table have verified and they are successfully executed. The flow is tested in different possible conditions of testing stages and all conditions are modified when the condition failed in the process of testing stage and all the bugs in application has modified in the testing.

**Chapter 7**

**OUTPUTS:**

**SCREENSHOTS:**

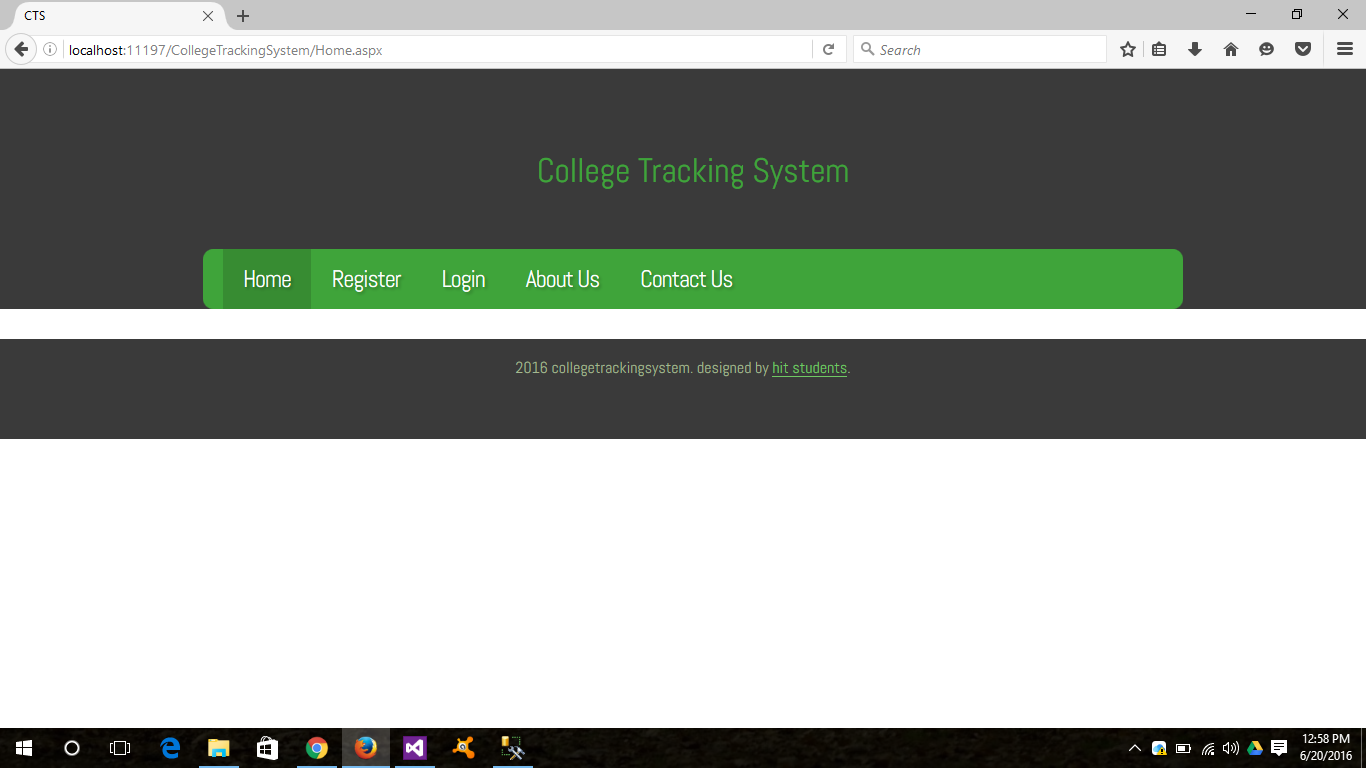


Fig:7.1.1 – Home Page

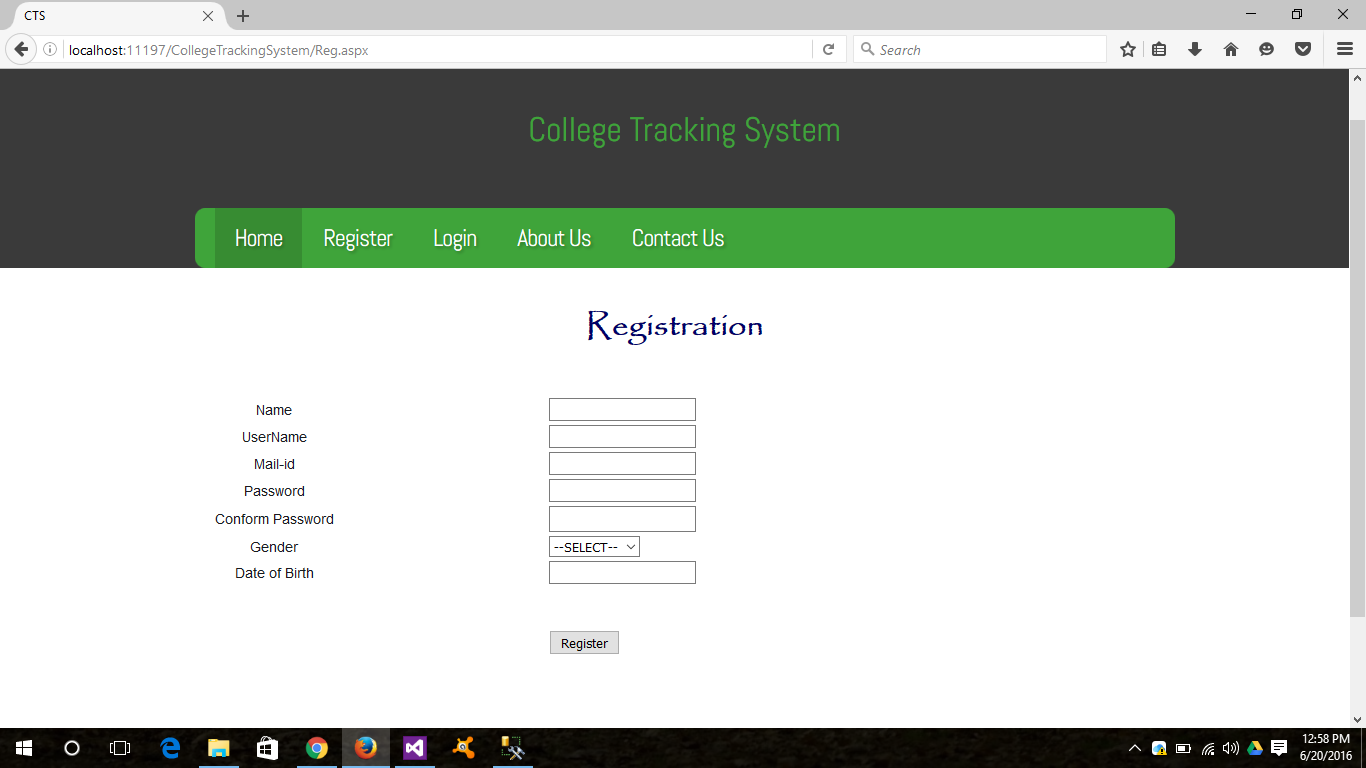
****

Fig:7.1.2 –Registration Page

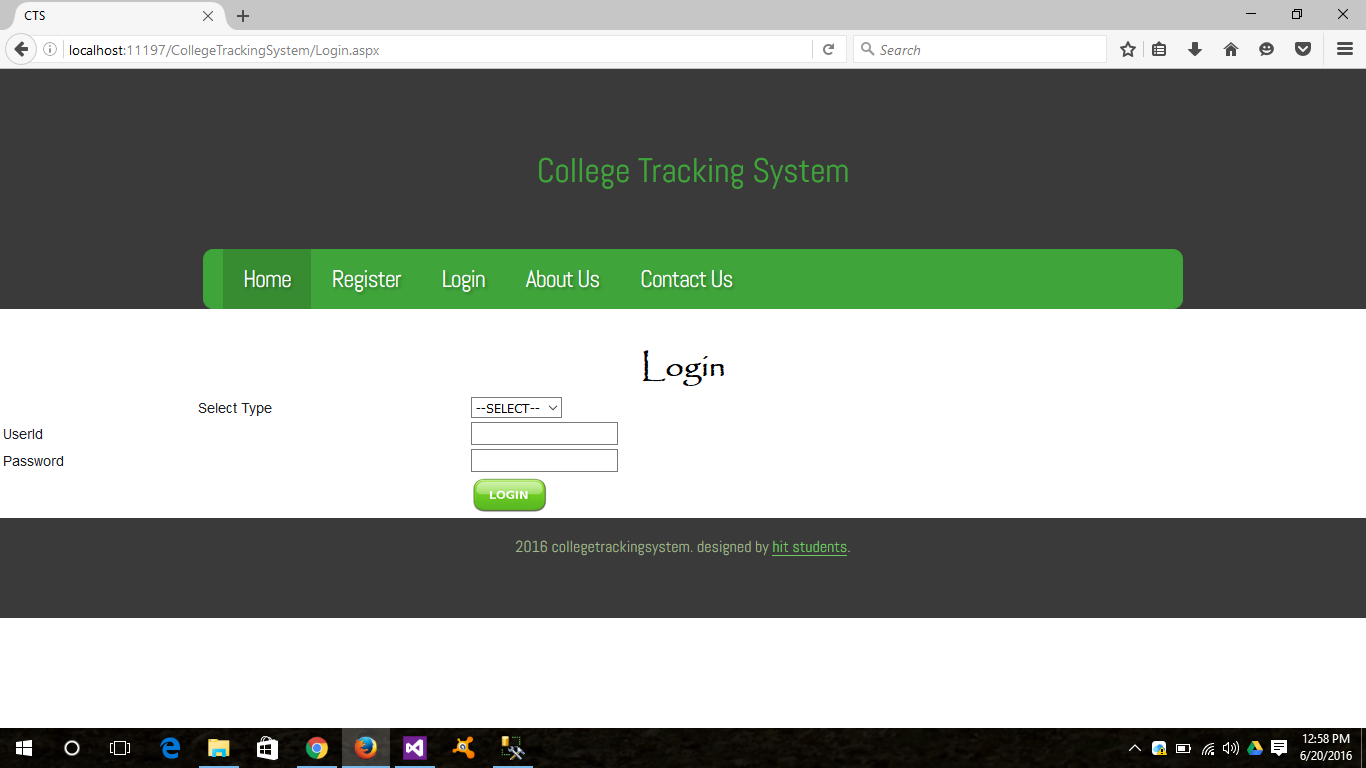


Fig:7.1.3 –Login Page

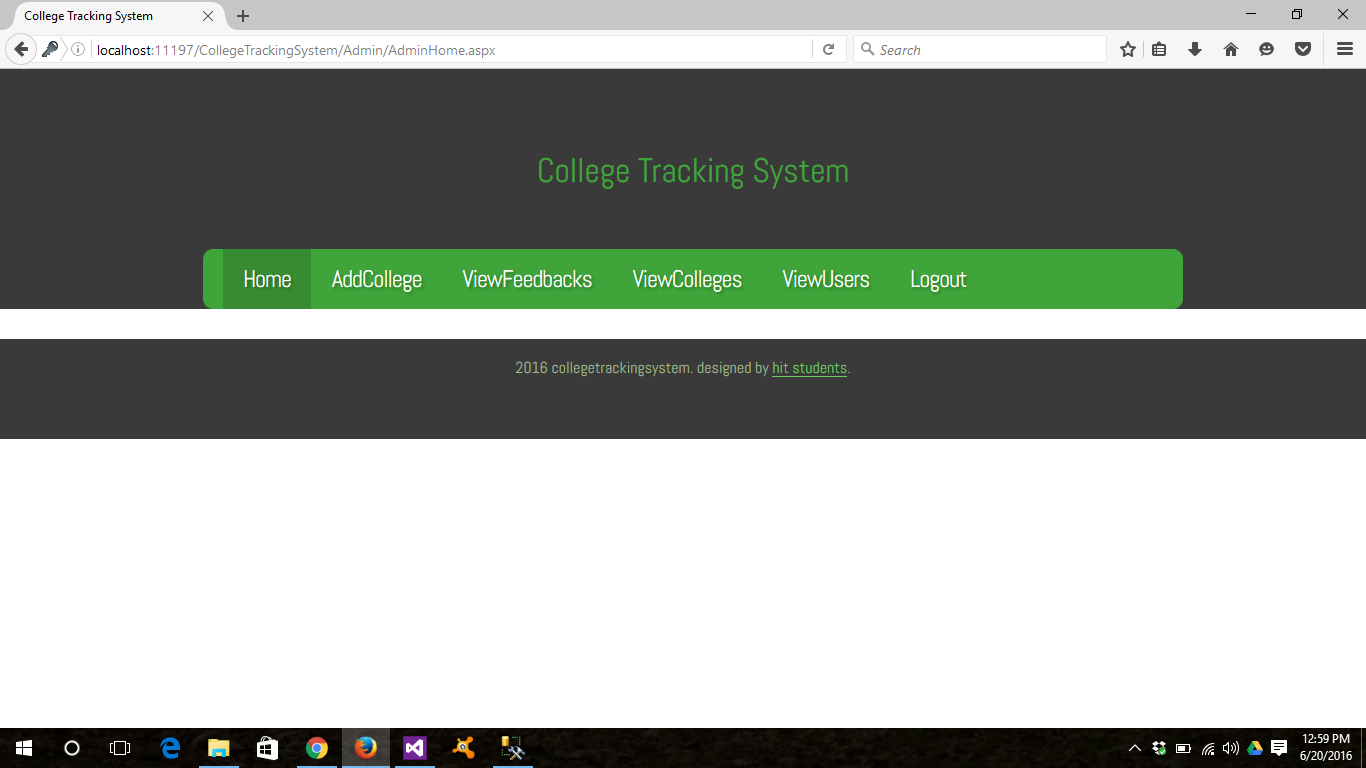


Fig:7.1.4 – Admin Home Page

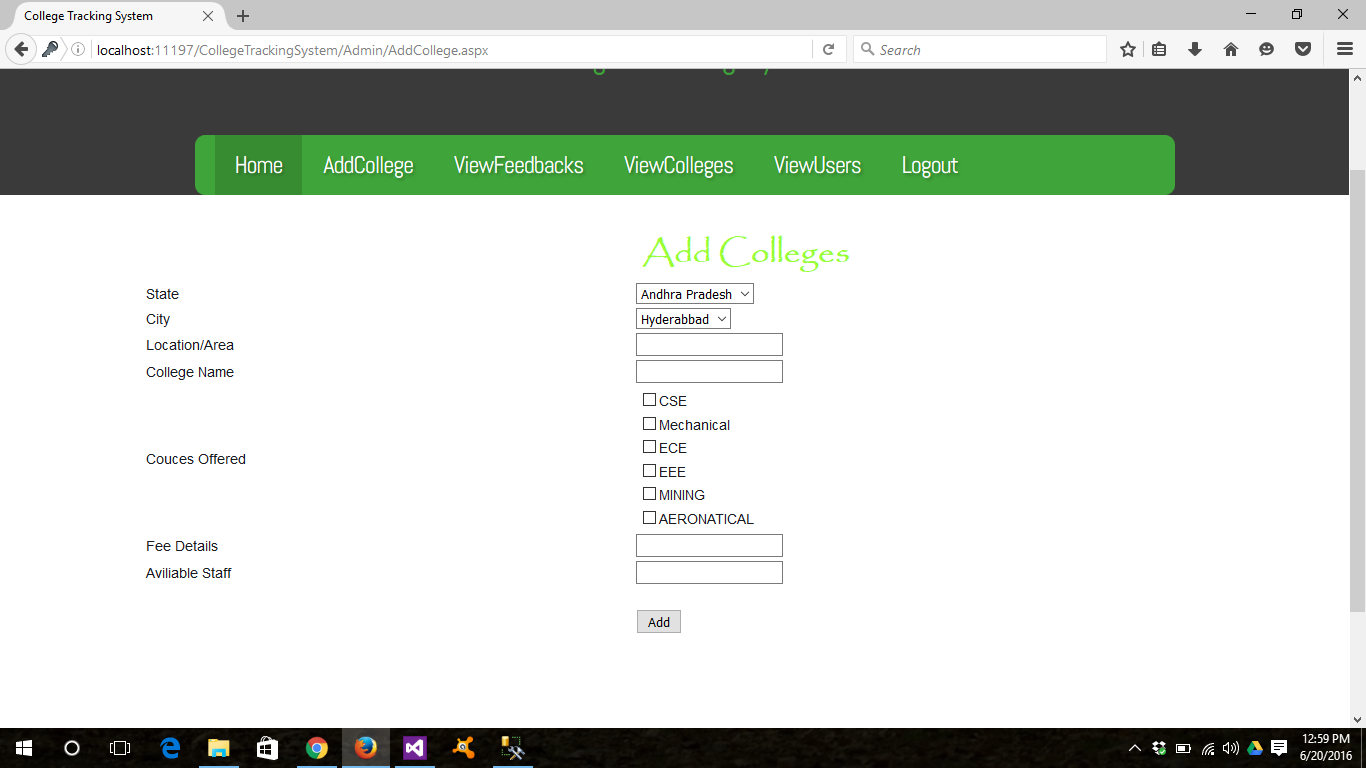


Fig:7.1.5 –Add Colleges Page

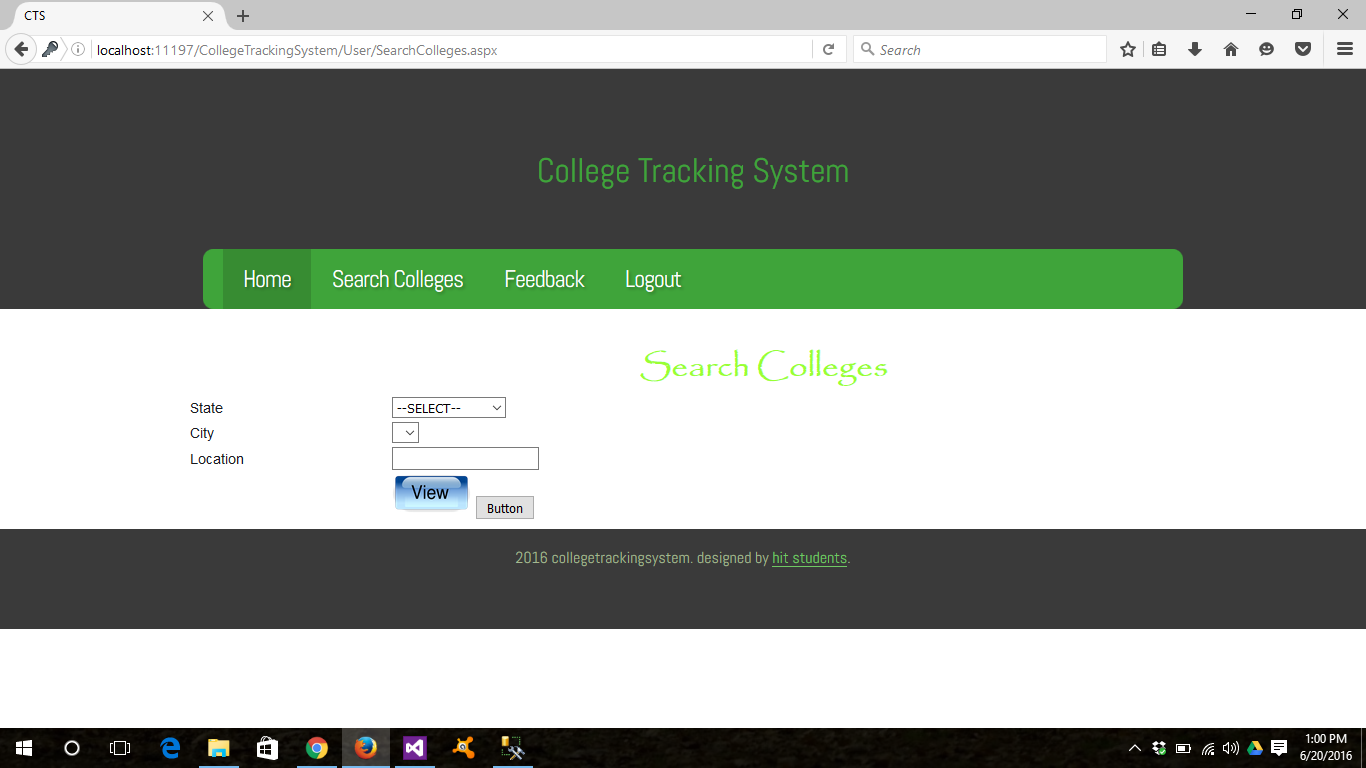


Fig:7.1.6 –Search Colleges Page

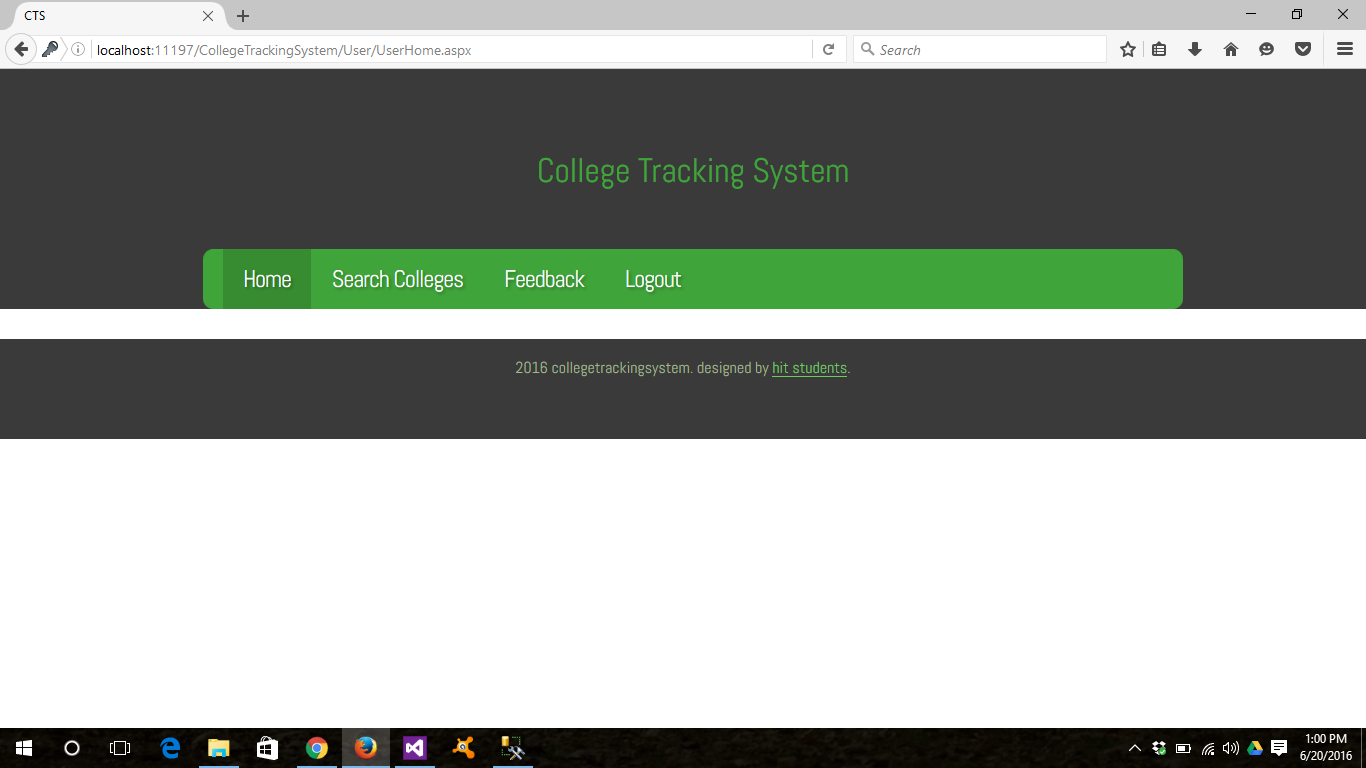


Fig:7.1.7 – User Home Page

**Chapter 8**

**Conclusion**

This project aims to automate all the processes happening in the educational intuitions. In the existing system all the details of the students are maintained manually in paper records. In our system everything is automated, thus reducing paper work and increasing the efficiency of administrative staff. It basically includes the information like, College’s information, Fee details, Staff information, and Placement details, List of courses offered, Details of previous academic year admissions. Here our project College Tracking System is made in visual studio by using c# language under .Net framework. The whole information is saved in the database which is created by SQL server2008r2. By accessing this website lot of burden is reduced for the people who are searching for different colleges. Each and every person’s login is secured by protecting them login names and passwords. Users can have day to day updates by using this.

Clarifying where you would like to go to college can help to narrow down your search using our advanced tool. For some students, remaining close to home and having the option to return from time to time is a significant factor. For others, having a short trip home for an occasional visit is not as important, and some may even prefer to attend school out-of-state. Using our college search tool, you can create a list of the schools in your desired area.

**CHAPTER 09**

**FUTURE ENHANCEMENT:**

This study investigated the effectiveness, in terms of the attainment of revelant learning outcomes, of the types of learning promoted by educational features commonly incorporated in course management systems. Twenty-one courses with significant use of the internet, but with face- to-face teaching as the predominant instructional mode, were investigated. Five hundred and ninety-five students taking these 21 courses completed a questionnaire which gave feedback on the extent of use of and quality of implementation of internet features, as well as their perception of the attainment of outcomes relating to approaches to learning, communication skills and understanding of content. A conformatery factor analysis of scales pertinent to information-presentation and constructive- dialog features showed a very poor fit to the data, indicating that the to types of function did not act in concert. Structural equation of modelling walls used to test instructional models in presage-process-product formats for information and dialog features. The information one showed a marginal fit to the data, but the dialog one a very good fit. This shows that using the internet for presenting information in a blended environment does not seem to effectively help students achieve learning outcomes. Using features which promote constructive dialog and interactive learning activities encourages a deep approach to learning, the development of communication skills and enhanced understanding of content.

**CHAPTER 10**

**BIBLOGRAPHY:**

**For .NET installation**

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

**For deployment and packing on server** [www.developer.com](http://www.developer.com)

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

**For ASP.NET** <https://msdn.microsoft.com/en-IN/library/z1zx9t92.aspx>

<https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>

**Internet Drafts and RFCs**

* DARPA RFC791, IP (Internet Protocol) http://www.ietf.org/rfc/rfc791.txt September 1981
* Simpson, W. RFC 1661 The Point-to-Point Protocol (PPP) http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1661.html July 1994

http://www.ietf.org/rfc/rfc2433.txt October 1998

* Pall, G. for Microsoft and Zorn, G. for Cisco RFC 3078 MPPE (Microsoft Point to Point Encryption protocol) http://www.faqs.org/rfcs/rfc3078.html March 2001
* Potter, D. &Zamick, J. for Cisco Systems PPP EAP MS-CHAP-V2 Authentication Protocol

http://www.ietf.org/internet-drafts/draft-dpotter-pppext-eap-mschap-01.txt January 2002