2012

Web-based information/knowledge sharing system

Sai Laxmi Kotha

WEB-BASED INFORMATION/KNOWLEDGE SHARING SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Sai Laxmi Kotha

December 2012

WEB-BASED INFORMATION/KNOWLEDGE SHARING SYSTEM

A Project

Presented to the

Faculty of

California State University,

San Bernardino

by

Sai Laxmi Kotha

December 2012

Approved by:

[redacted]

Dr. Richard Botting, Advisor, School of Computer Science and Engineering

Dec 3rd 2012

Date

[redacted]

Dr. David Turner

[redacted]

Dr. Kerstin Voigt

ABSTRACT

Web-Based Information/Knowledge Sharing System(WBIKS) refers to a software system that can be used by an institution or an organization to manage knowledge across the system. This project can be used to accomplish goals like shared knowledge and higher performance through collaboration.

It is a web-based knowledge management system that can be used to manage, share, and collaborate with different users, teams or groups across the whole institution. The project will provide facilities for the users to share documents and also search for them, in addition it also has social features like "Testimonials".

The project is designed and built on N-tier Client-Server architecture to facilitate a modular design. The project has been implemented using .NET Framework with C# and ASP.NET along with SQL Server and IIS.

The project satisfies most of the objectives including document sharing and searching and social features such as testimonials and feedbacks. The project can be deployed for use by small institutions and/or organizations.

ACKNOWLEDGEMENTS

My sincere and heartfelt thanks to the following people who helped me during this project. Firstly, I would like to thank my project advisor Dr.Richard Botting, for his help, advice and tremendous support. He has always been very quick in his feedback about the implementation and design. I would also like to thank Dr. Voigt and Dr. Turner who were in my committee, for their support. Dr. Mendoza, my graduate advisor, for her advice throughout my study at Cal State. I would also like to thank Monica for being so helpful and patient with me for all the requests I have made. My most sincere thanks to all the faculty members of Computer Science and Engineering Department for giving me this opportunity to pursue my Master's Degree. Lastly I am very thankful to my parents and family for being supportive of my decisions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

The chapter gives a brief description about the background, scope and purpose of the project.

Background

'WBIKS' acronym for Web Based Information/Knowledge sharing system is a web based application, which refers to software system that can be used by an institution or an organization to manage knowledge sharing across the system. It enables efficient knowledge management across the organization. It is a web-based application that can accessed within the system. It has three modules, viz., Administrator, Users (Student/Staff)and Public users. All the modules have different level of accessibility to the system which leads to a better security, as the information they can access is limited to their accessibility level on the system. Users are provided facilities to share and search documents or other kind of knowledge. This application also allows public users (in this cases users

who are not registered in the system) to view public documents and information.

The need for such a system to exist in an organization arises because it can be used to share knowledge and improve performance through collaboration. It is also important that system be secure because the age of technology has also given rise to extensive exploitation of personal property and information by hackers. Being intranet based and limited accessibility ensures the system be highly secure.

Purpose

The purpose of this web application is to provide knowledge management for an institution and to build a private, custom tailored application which can be locally managed.

The system is user friendly with negligible risk of data mismanagement and a system with high level of security through different levels of authentication.

Scope

The scope of the project was to develop a web application that is reliable and scalable with minimum impact on cost and with high security. To create a system with integrated social features such as document feedbacks and user testimonials which help other users of system to efficiently share and search for content in the system. The application is ADA compliant and therefore accessible to users with disabilities. The integrated search feature allows users to search for content efficiently and locally managed means better control over the system.

Definitions, Acronyms, and Abbreviations

The following terms and definitions are used in the project.

SDK - Short for Software Development Kit, a programming package that enables a programmer to develop applications for a specific platform. Typically an SDK includes one or more APIs, programming tools and documentation. [1]

.NET Framework - A programming infrastructure created by Microsoft for building, deploying and running applications and services that use .NET technologies, such as desktop

applications and web services. It has two major parts CLR and Framework Class Library. [2]

Visual Studio.NET - It is a multi-language development environment with an integrated development environment (IDE) for Microsoft developed languages like Visual Basic, Visual C#, Visual C++. It is easily extended to have a complete set of tightly integrated tools to develop projects using Microsoft Technologies. It is a commercial software developed by Microsoft Corporation. [3]

ASP.NET - It is a Web application framework developed and marketed by Microsoft to allow programmers to build dynamic websites, web applications, and web services. It is a runtime host and an architecture which supports managed code. [4]

ADO.NET - It is a set of software components that programmers can use to access data and data services based on disconnected Datasets and XML. It is commonly used to access and modify data stored in a RDBMS. [5]

API - Short for Application Program Interface, is a set of routines, protocols and tool for building software applications. It provides an interface between different software components and provides interaction between different modules. [6]

HTTP - The Hypertext Transfer Protocol (HTTP) is a networking protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. [7]

UML - Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created, by the Object Management Group. [8]

XML - Extensible Markup Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. [9]

Dia - is free and open source general-purpose diagramming software, developed originally by Alexander Larsson. [10]

HTML - Hypertext Markup Language (HTML) is the predominant markup language for web pages. HTML elements are the basic building-blocks of web pages. [11]

CSS - Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. [12]

JavaScript - JavaScript is a prototype-based scripting language that is dynamic, weakly typed and has first-class functions. It is a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. [13]

IIS - is a web server application and set of feature extension modules created by Microsoft for use with Microsoft Windows and .NET Framework. [14]

CHAPTER TWO

ARCHITECTURE

Project Design

'WBIKS' implements a N-tier client-server architecture. It is implemented using ASP.NET and C# for frontend and server side needs and SQL Server for database. .NET framework is used to maintain the lifecycle of the pages served to the web browser. The hosting is done on IIS.

N-Tier Architecture

The N-tier architecture (often referred to as multi-tier architecture) is a client-server architecture in which presentation, application processing and data management functions are logically separated. The typical features of n-tier architecture may include security, availability and scalability, manageability and data abstraction.

The system can be designed using a 3-tier architecture and the overall functionality can be distributed into various tiers or layers, as follows:

- Presentation Layer - Also called as the client layer and comprises of components that are dedicated to present data to the user.
- Business Rules Layer - The layer encapsulates the business rules or the business logic. Changes in the business rules can be easily handled in this layer without modifying other layers as long as the interface between the layers remains the.
- Data Access Layer - This layer comprises of components that help in accessing the database system. This layer provides a level of abstraction for database structures. Simply put the changes made to the database, tables, etc.. do not affect the rest of the application. The different application layers send data requests to this layer and receive the response from this layer.

The following figure depicts the flow of N-tier architecture.

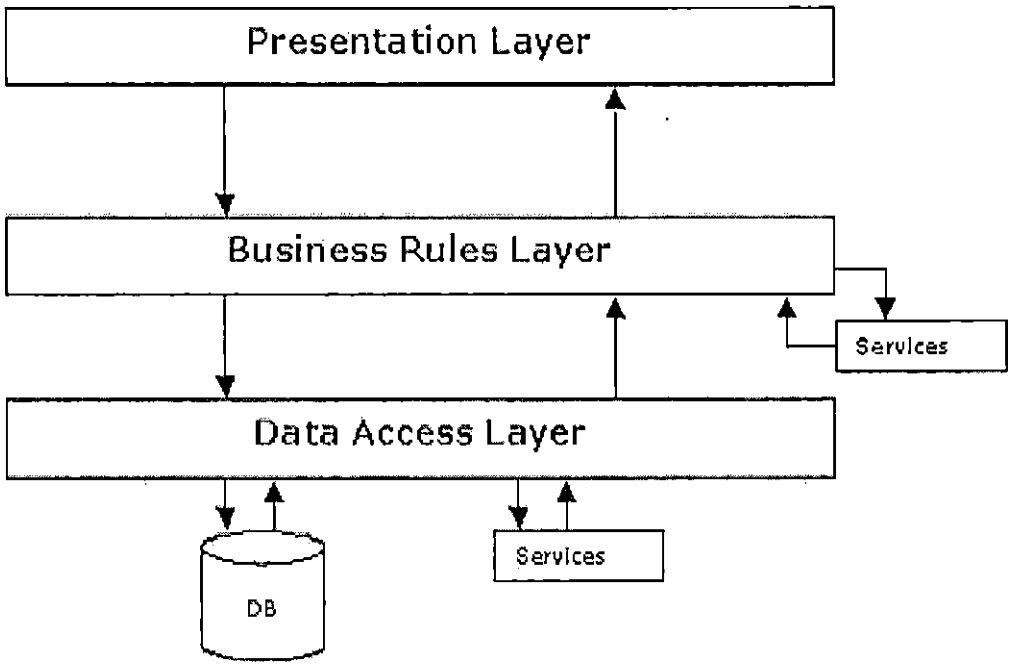

Figure 1. N-tier Client-Server Architecture

System Security

Security is an important for any web based application since database or code vulnerabilities could lead to information exploitation. Since many pages take input from the user, there is a risk of compromising user information

at a database level using SQL injection. The system provides adequate security against SQL injection using ADO.NET parameterized SQL. OOP structure of the system code allows encapsulation for code level security and standard HTTP(S) security provides protection against web based exploitation. Passwords in the database are stored as encrypted hash values and they are decrypted as user logs in and verified against.

Interface Design

This section elaborates about the different interfaces used in the application.

System Interface Design

'WBIKS' is 3-tier based architecture which is database driven, there are following three components:

1. A user interface which includes client machine with web browser.
2. A business interface with business logic and web server like IIS.
3. A data interface with database server like SQL Server used in this project.

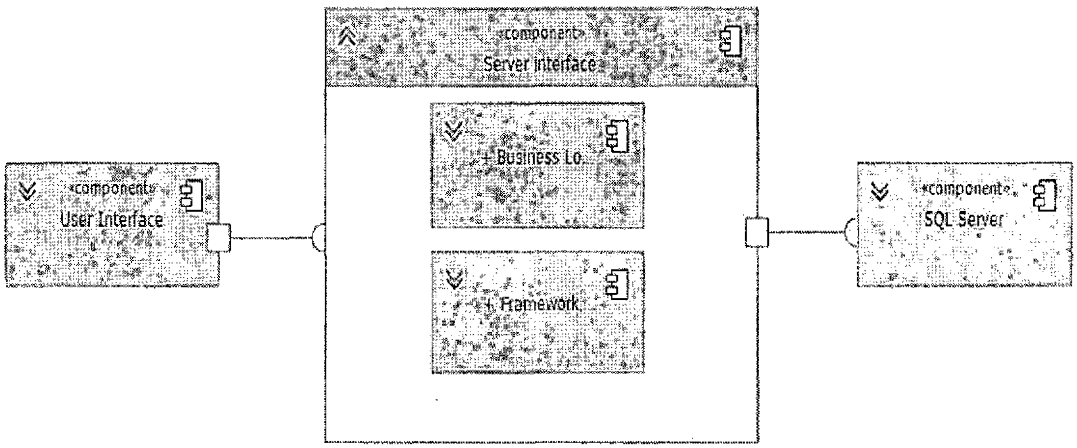

Figure 2. Component Diagram

User Interface

'WBIKS' user interface is can be categorized as below:

1. Admin Module
2. User Module
3. Public User Module

Admin and User module includes login page for user type validation and access level is granted based on this validation.

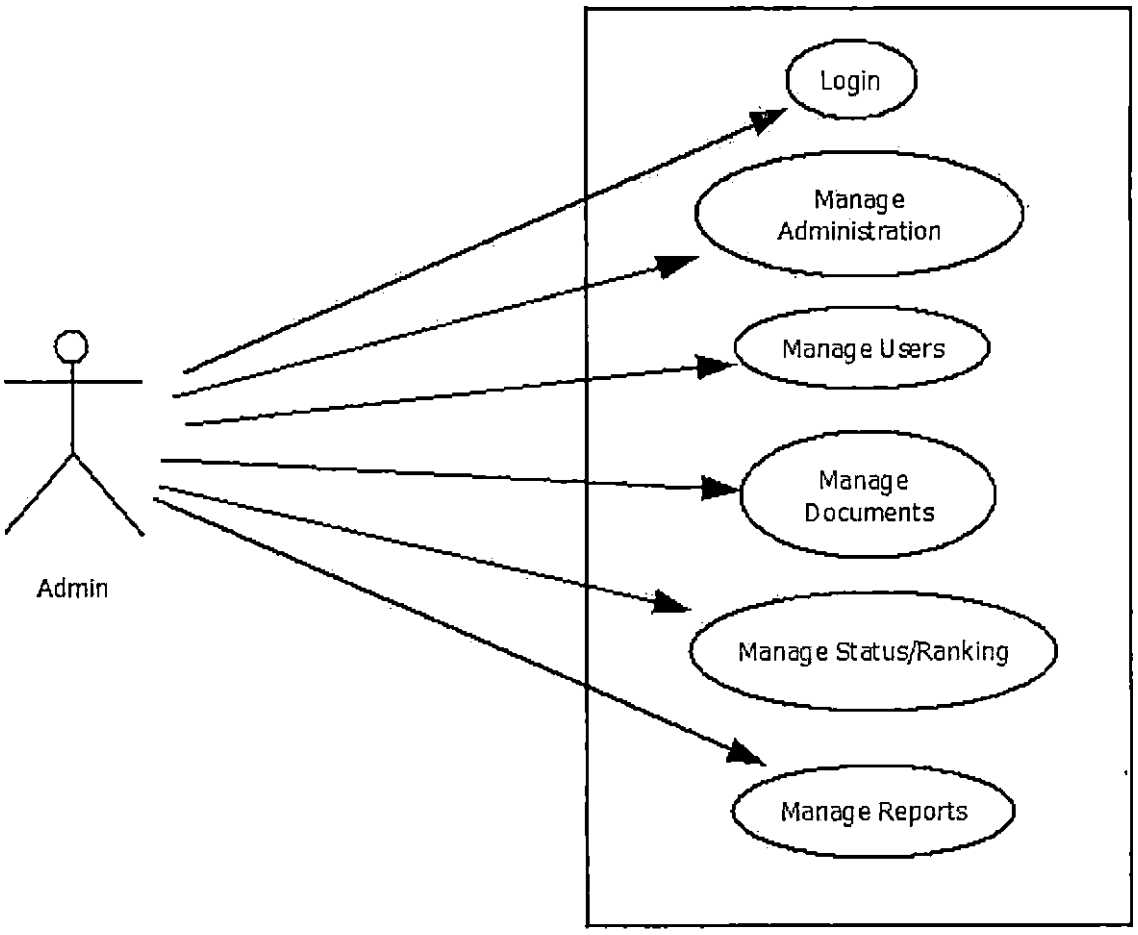

Figure 3. Admin - Use Case Diagram

Admin is the super user in the system. Admin can perform various responsibilities such as manage administration of the system, manage users, manage documents, manage status/rating/ranking, manage reports, etc... The Use case for admin module is shown above.

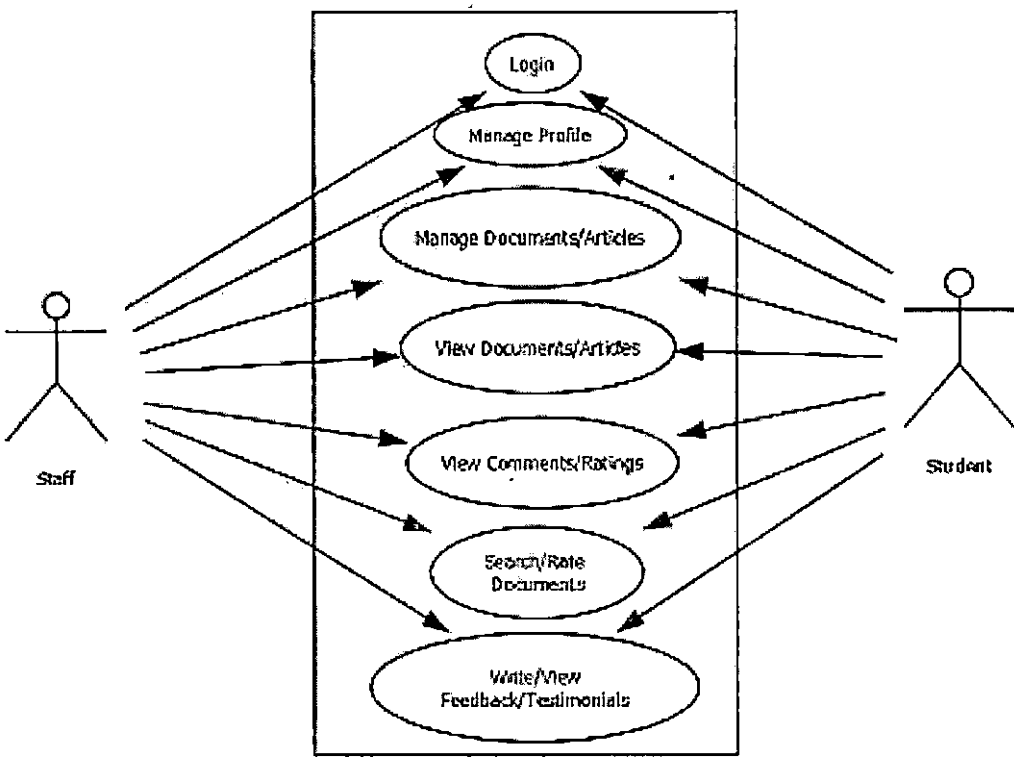

Figure 4. User (Staff/Student) - Use Case Diagram

A User can be Staff or Student. User can manage profile, manage documents, view documents, view ratings and comments, write comments/ratings/testimonials, search documents, etc… The use case diagram for Users is shown above.

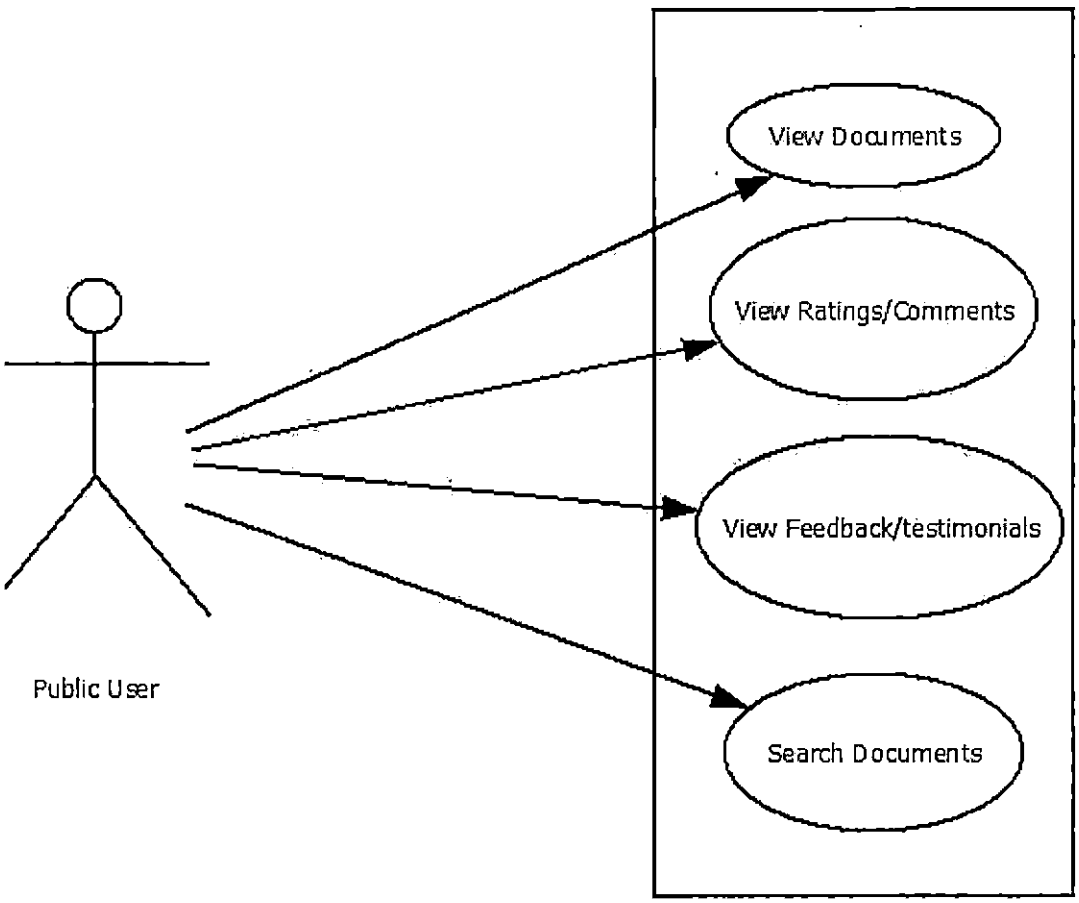

Figure 5. Public User - Use Case Diagram

Public user is a non-registered user of the system and can view documents, view ratings/comments, view testimonials and search for documents.

System Access Flowchart

A flowchart describing both admin and user processes and gives complete description of the system.

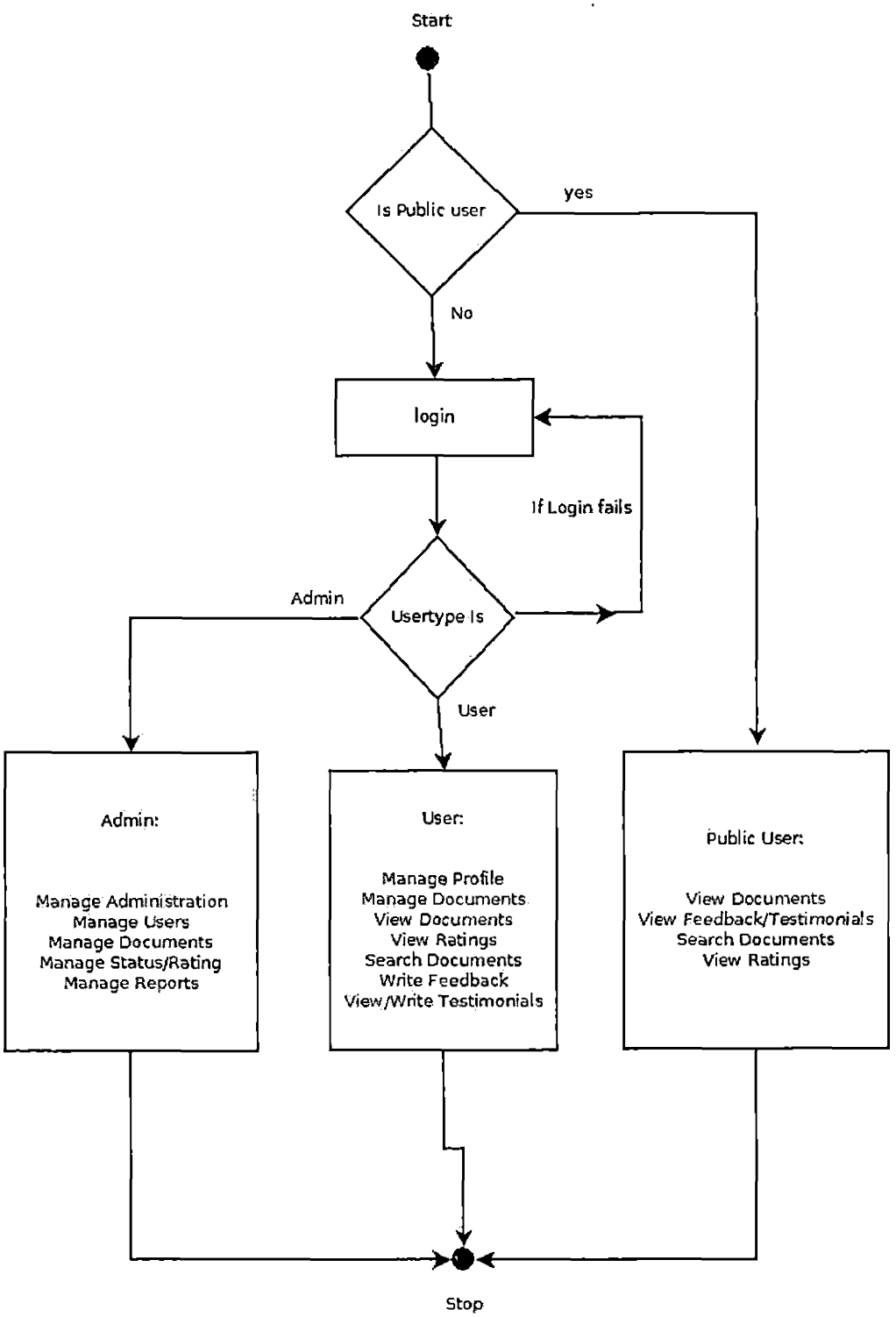

Figure 6. System Flowchart

CHAPTER THREE

DATABASE DESIGN

Data Analysis

'WBKIS' application is database driven, hence, database is a major system module and it's contributes to the overall system performance. The project SQL Server as the database server with ADO.NET and OLEDB being the database drivers.

OLEDB with ADO.NET acts as interface between server pages and database. .NET framework used in the project simplifies database read/writes using modules called 'Dataset' and 'Datareader'. This is useful to maintain database integrity and to increase database access performance and security.

Database Specifications

The database design schema is classified into two namely ER model and logical module. The latter includes constraints and structures. The ER model can be seen below

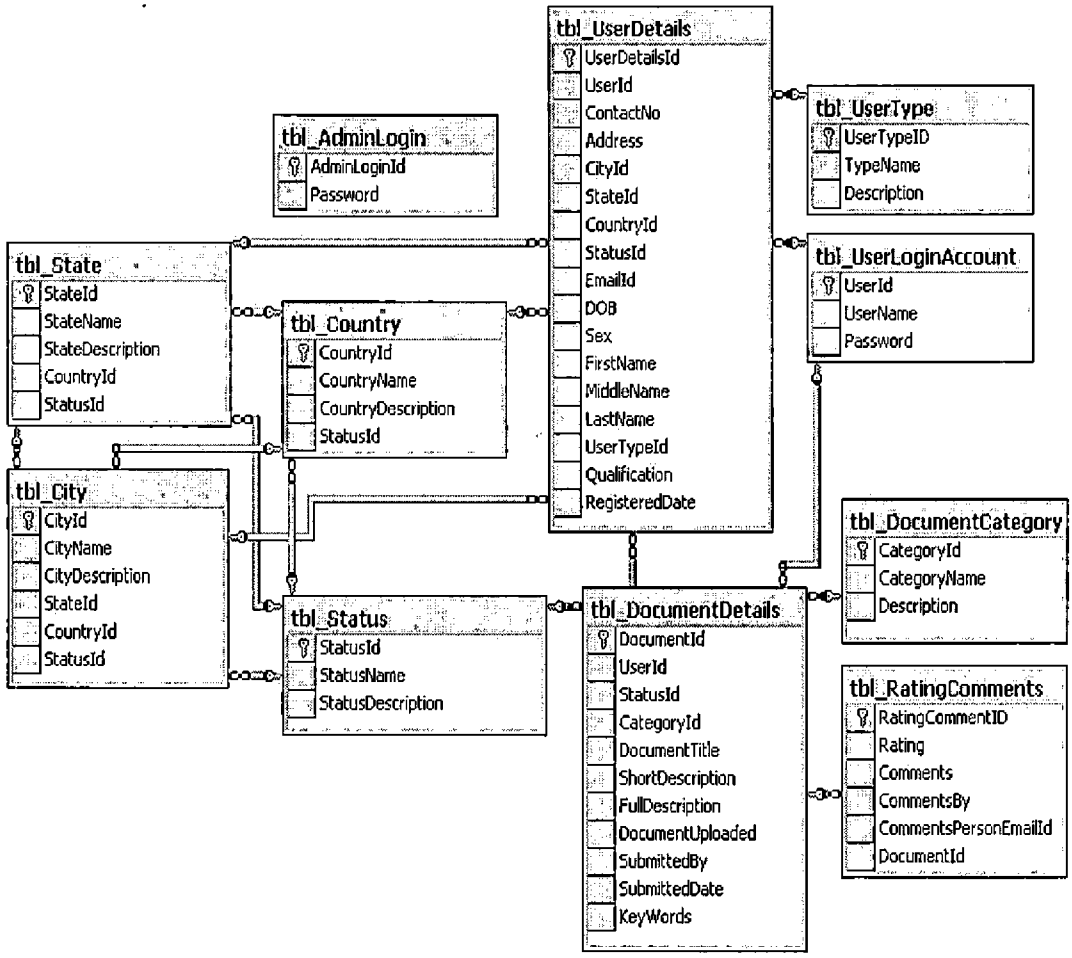

Figure 7. ER Diagram

Database Design

The following figures show the structures of all tables. The tables have all been normalized to avoid anomalies.

	Column Name	Data Type	Length	Allow Nulls
▶🔑	AdminLoginId	varchar	50	
	Password	varchar	20	

Figure 8. Admin Login

	Column Name	Data Type	Length	Allow Nulls
▶🔑	CityId	int	4	
	CityName	varchar	50	
	CityDescription	varchar	255	✓
	StateId	int	4	
	CountryId	int	4	
	StatusId	int	4	

Figure 9. City Details

	Column Name	Data Type	Length	Allow Nulls
▶🔑	CountryId	int	4	
	CountryName	varchar	50	
	CountryDescription	varchar	255	✓
	StatusId	int	4	✓

Figure 10. Country Details

	Column Name	Data Type	Length	Allow Nulls
▶🔑	CategoryId	int	4	
	CategoryName	varchar	50	
	Description	varchar	250	✓

Figure 11. Document Category

	Column Name	Data Type	Length	Allow Nulls
PK	DocumentId	int	4	
	UserId	int	4	
	StatusId	int	4	
	CategoryId	int	4	
	DocumentTitle	varchar	100	
	ShortDescription	nvarchar	2000	✓
	FullDescription	ntext	16	✓
	DocumentUploaded	varchar	250	✓
	SubmittedBy	varchar	50	✓
	SubmittedDate	datetime	8	✓
	KeyWords	varchar	250	✓

Figure 12. Document Details

	Column Name	Data Type	Length	Allow Nulls
PK	RatingCommentID	int	4	
	Rating	int	4	
	Comments	nvarchar	2000	
	CommentsBy	varchar	250	
	CommentsPersonEmai	varchar	250	
	DocumentId	int	4	

Figure 13. Rating Comments

	Column Name	Data Type	Length	Allow Nulls
🔑	StateId	int	4	
	StateName	varchar	50	
	StateDescription	varchar	255	✓
	CountryId	int	4	
	StatusId	int	4	

Figure 14. State Details

	Column Name	Data Type	Length	Allow Nulls
🔑	StatusId	int	4	
	StatusName	varchar	20	
	StatusDescription	varchar	255	✓

Figure 15. Status

	Column Name	Data Type	Length	Allow Nulls
▶🔑	UserDetailsId	int	4	
	UserId	int	4	
	ContactNo	varchar	50	
	Address	varchar	150	
	CityId	int	4	
	StateId	int	4	
	CountryId	int	4	
	StatusId	int	4	
	EmailId	varchar	50	
	DOB	datetime	8	
	Sex	varchar	10	
	FirstName	varchar	50	
	MiddleName	varchar	50	✓
	LastName	varchar	50	✓
	UserTypeId	int	4	
	Qualification	varchar	50	
	RegisteredDate	datetime	8	✓

Figure 16. User Details

	Column Name	Data Type	Length	Allow Nulls
▶🔑	UserId	int	4	
	UserName	varchar	30	
	Password	varchar	30	

Figure 17. User Login Account

	Column Name	Data Type	Length	Allow Nulls
🔑	UserTypeID	int	4	
	TypeName	varchar	100	✓
	Description	varchar	255	✓

Figure 18. User Type

CHAPTER FOUR

PROJECT IMPLEMENTATION

User Interface and Design

'WBIKS' being a web based application requires an intuitive user interface and compatible with all browsers and OS independent. Therefore the user interface is designed in such a way that it supports all browsers. Such as Microsoft Internet Explorer, Mozilla Firefox, Google Chrome. All pages are generated dynamically on server side based on the user access level.

Static Pages

These pages are the static pages of the application web site. Can be viewed by general public.

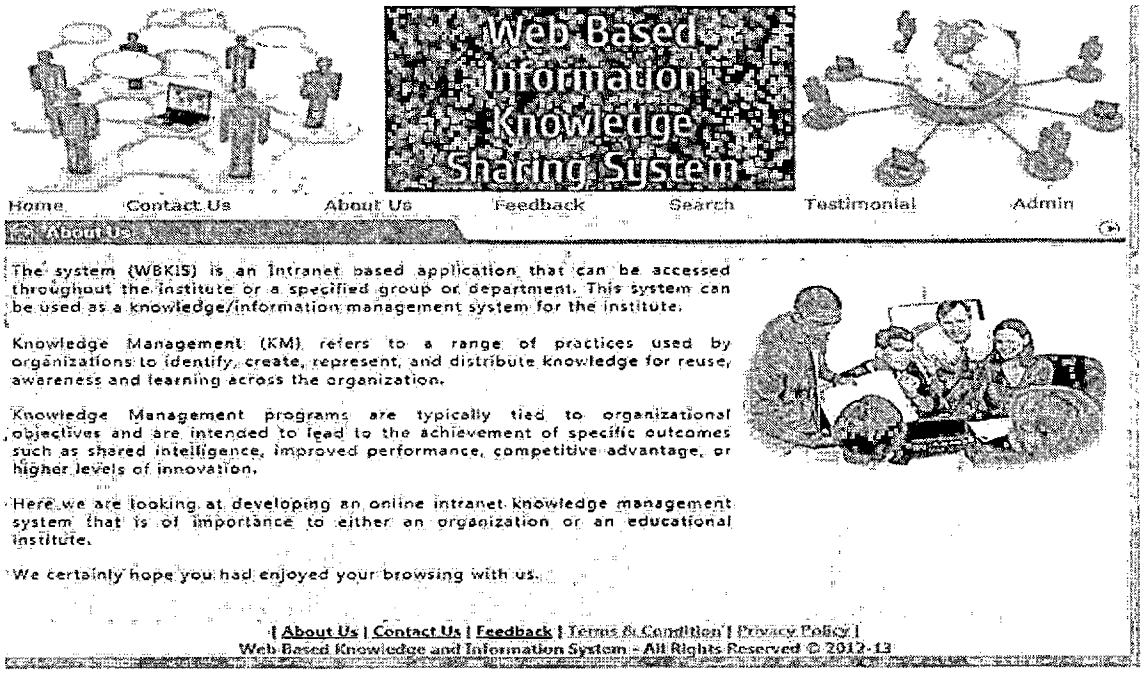

Figure 19. About Us Page

The page briefly describes the web application and services offered.

Login Page

The Login page is used to login into the application. There are two login pages for the application. One for User login and other for Admin login.

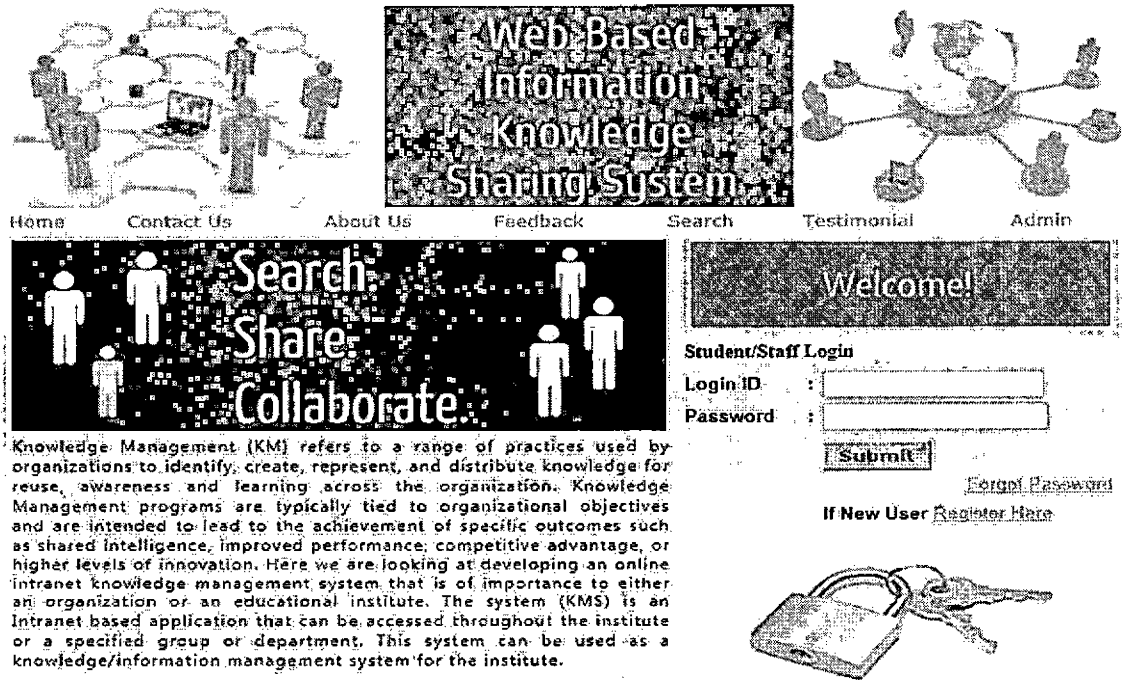

Figure 20. User Login Page

This page is also the homepage of the website and has links to other pages and to Admin login page.

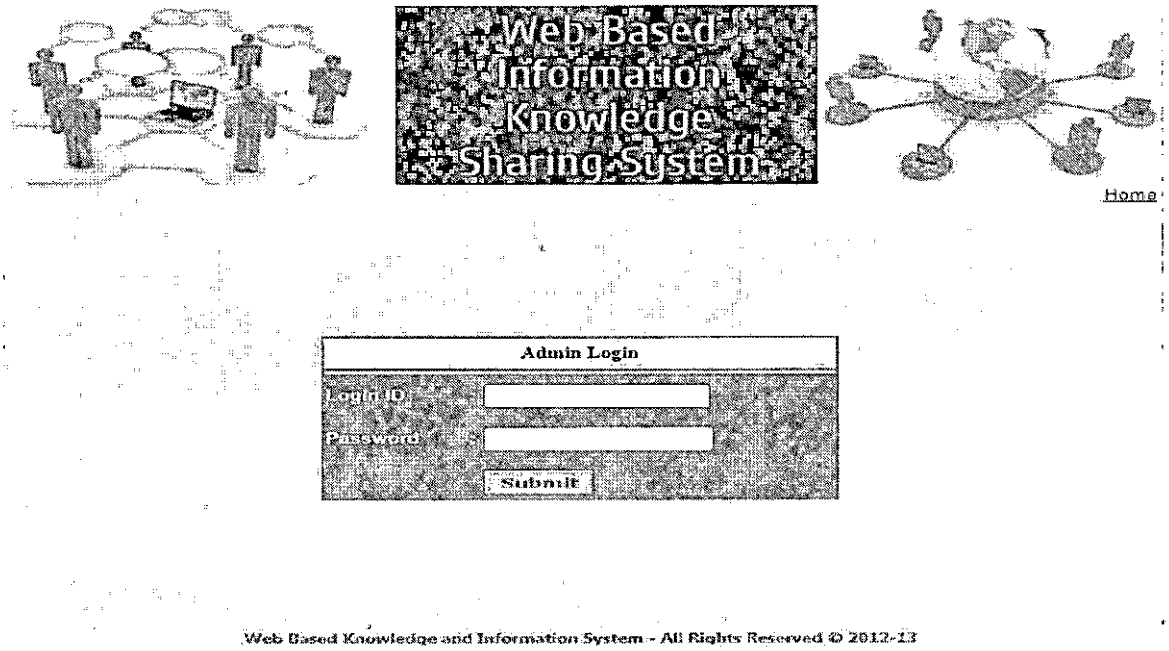

Figure 21. Admin Login Page

This page is for admin login and only admins can login into the web application from this page.

Search Page -

The search page is used to search for users and documents on the system and can be accessed by public and registered users.

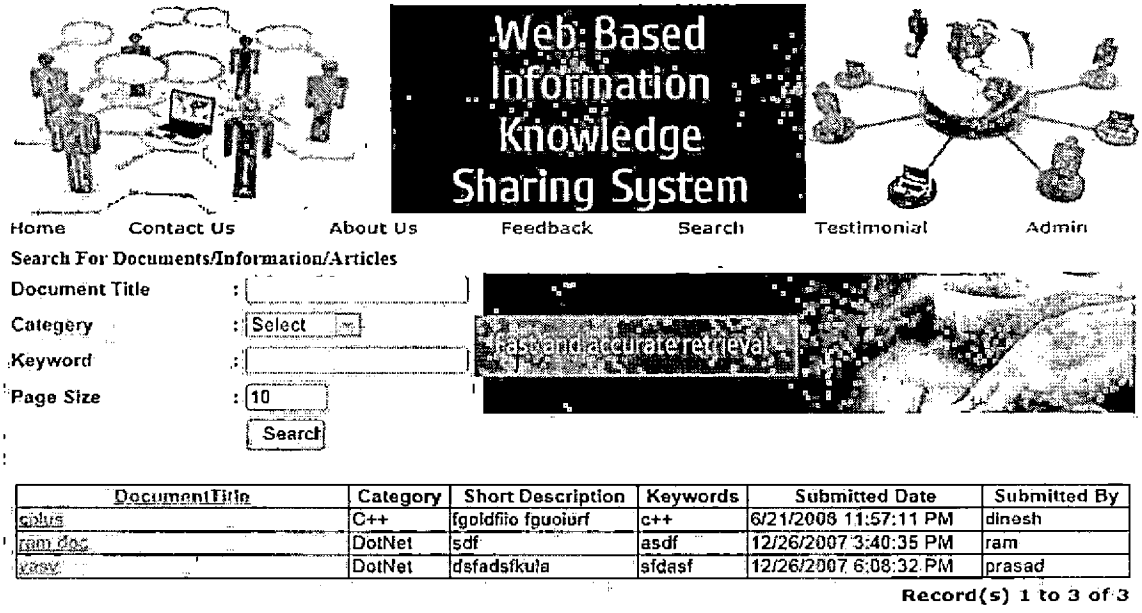

Figure 22. Search Page

Contact Us Page -

This page allows users of the system to contact admin about various requests/inquiries/information.

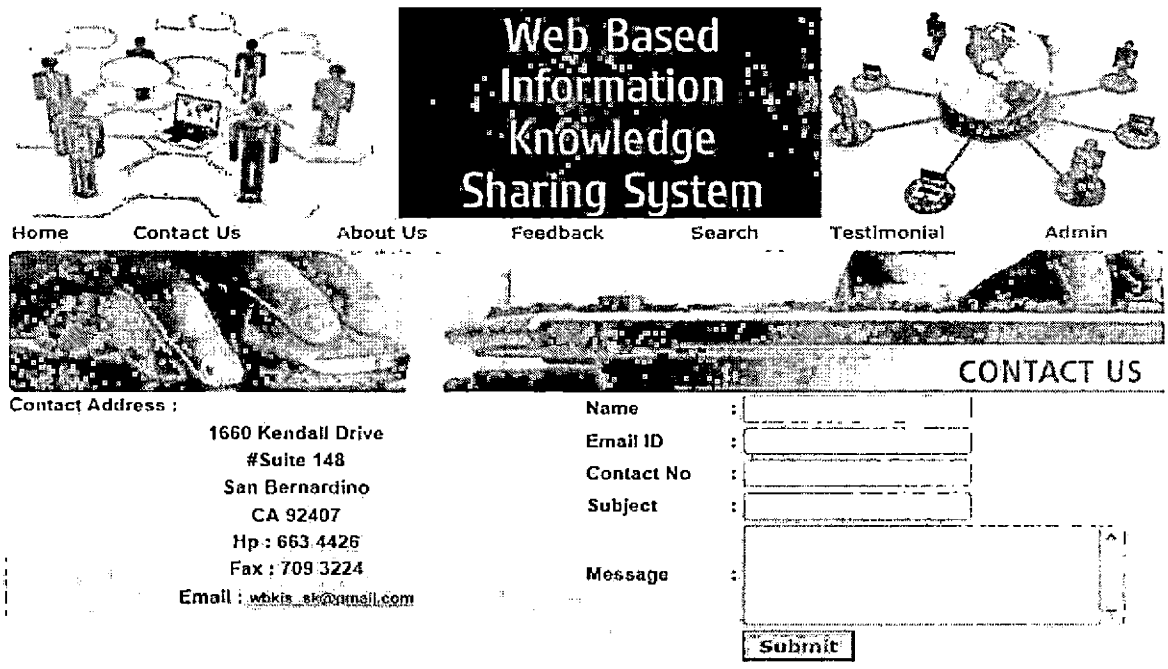

Figure 23. Contact Us

Feedback page -

This page is used to write feedback.

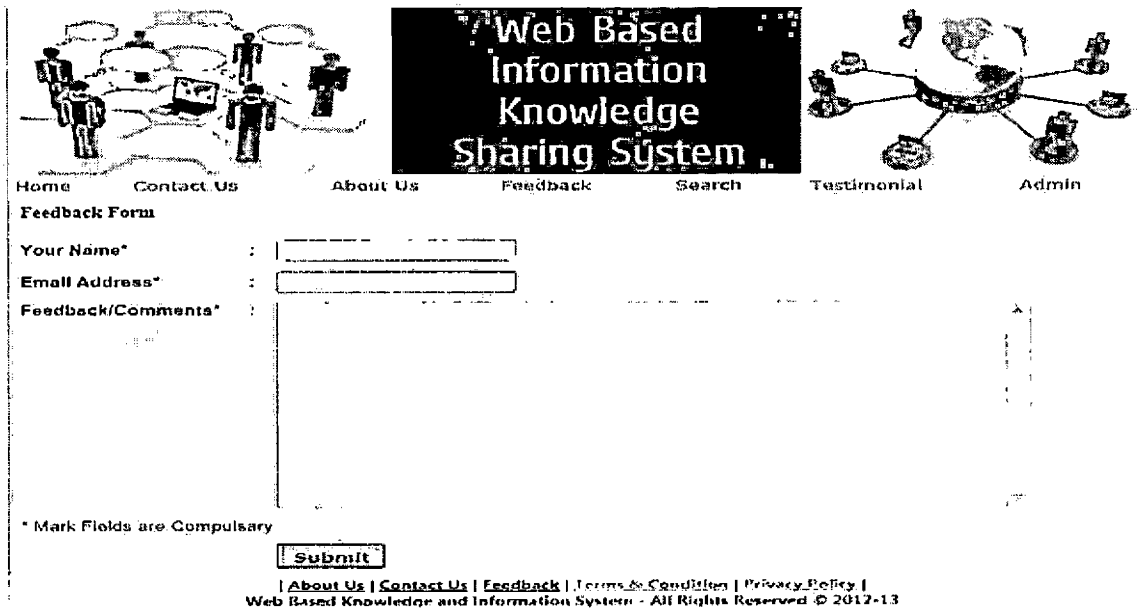

Figure 24. Feedback Page

Admin Interface Pages

Admin Home Page -

This page is the home page of the admin and menu is dynamically generated for the admin. Menu offers wide range of feature and function to the admin.

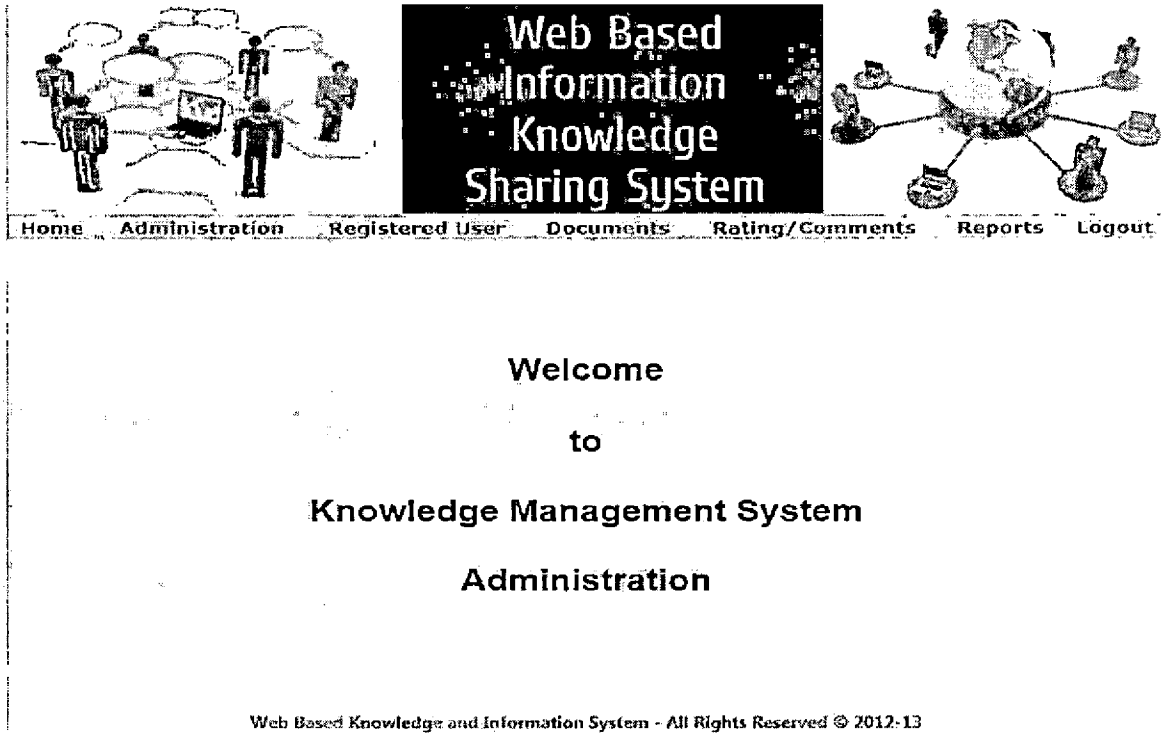

Figure 25. Admin Home Page

Admin Manage Document Page -

This particular page is used by the admin to manage documents uploaded by the users.

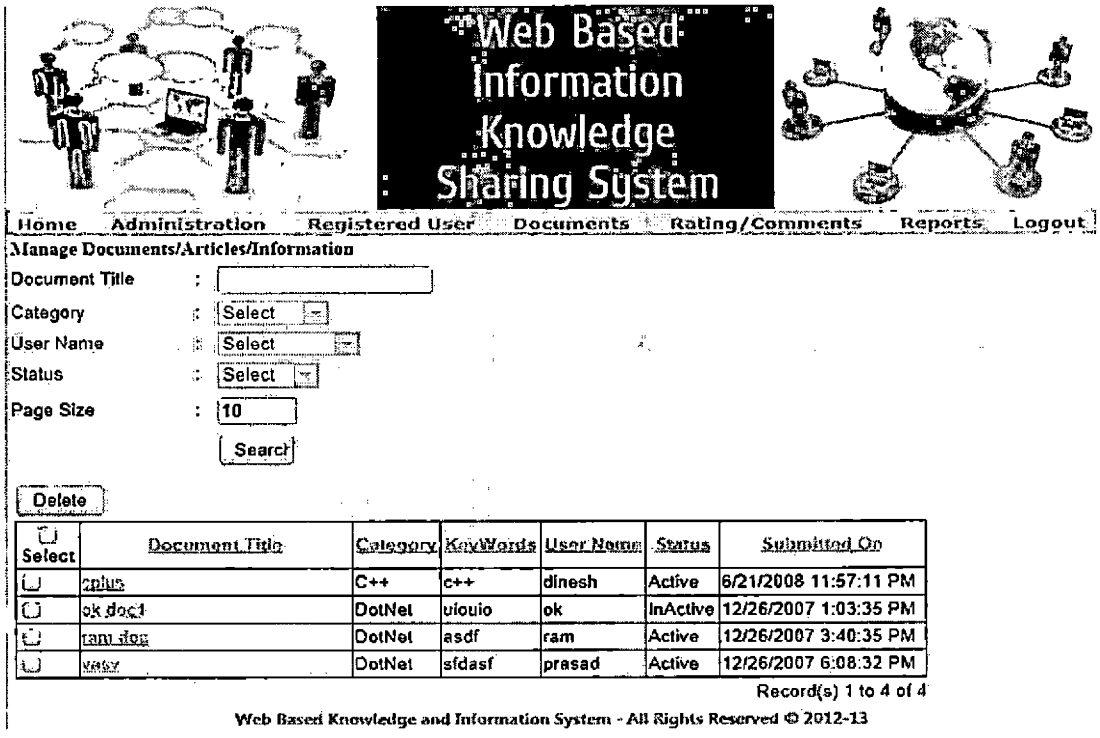

Figure 26. Admin Manage Document Page

Admin Manage Users Page -

This page allows admin to manage users and user types.

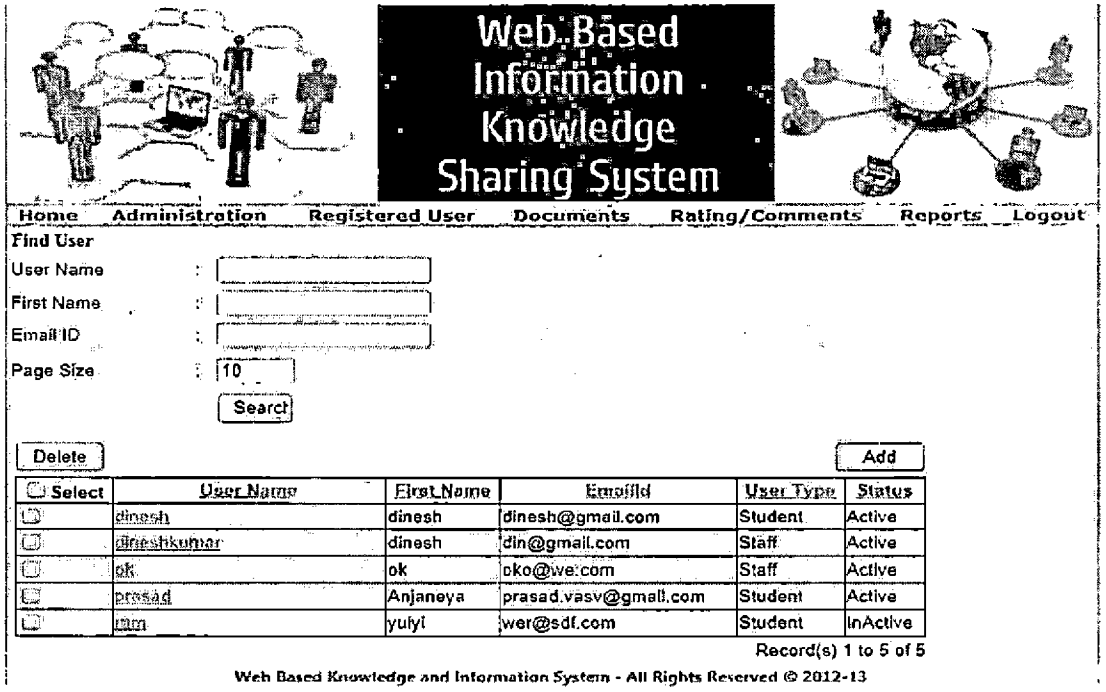

Figure 27. Admin Manage User Page

Admin Reports Page -

This page allows admin to generate reports. It allows to generate user reports and documents report.

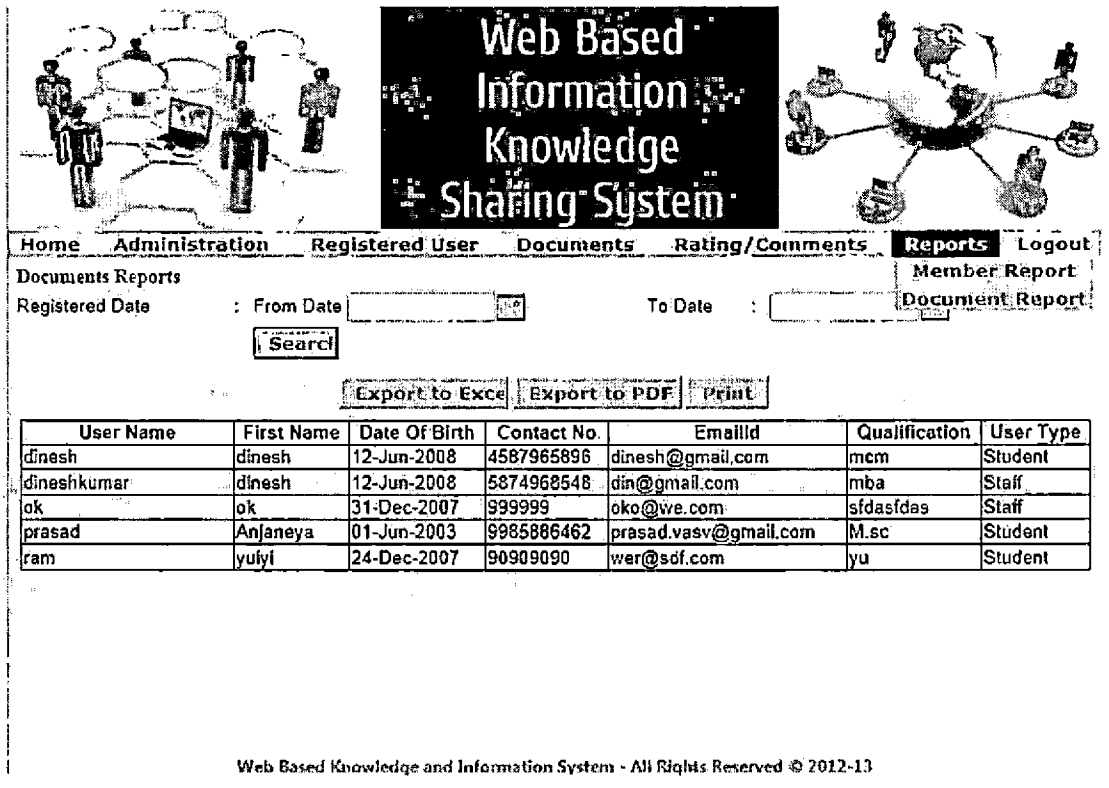

Figure 28. Admin Reports Page . .

User Interface Pages

User Login page –

This page is the login page for the users and also acts as a homepage of the website and has dynamic menu for other links.

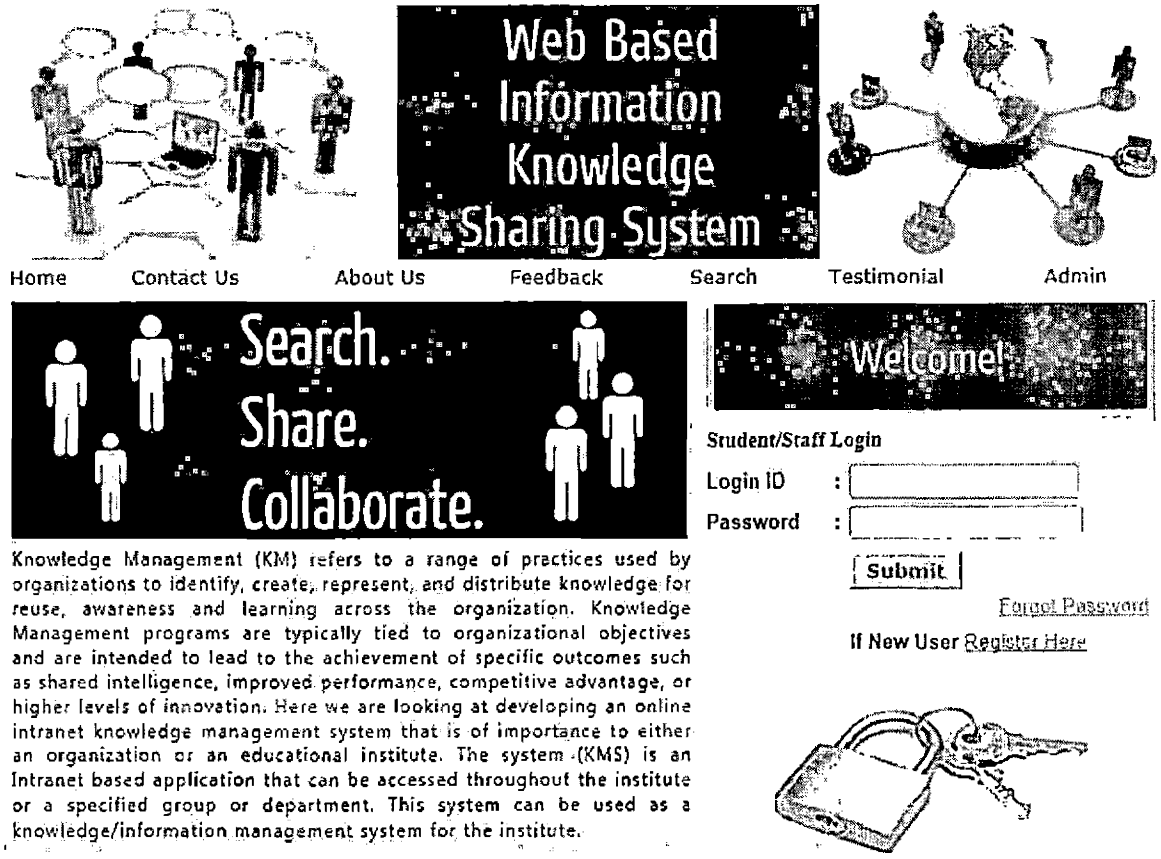

Figure 29. User Login Page

User Profile page –

This page allows the user to view and edit his/her profile on the system and has dynamically generated menu links to other pages.

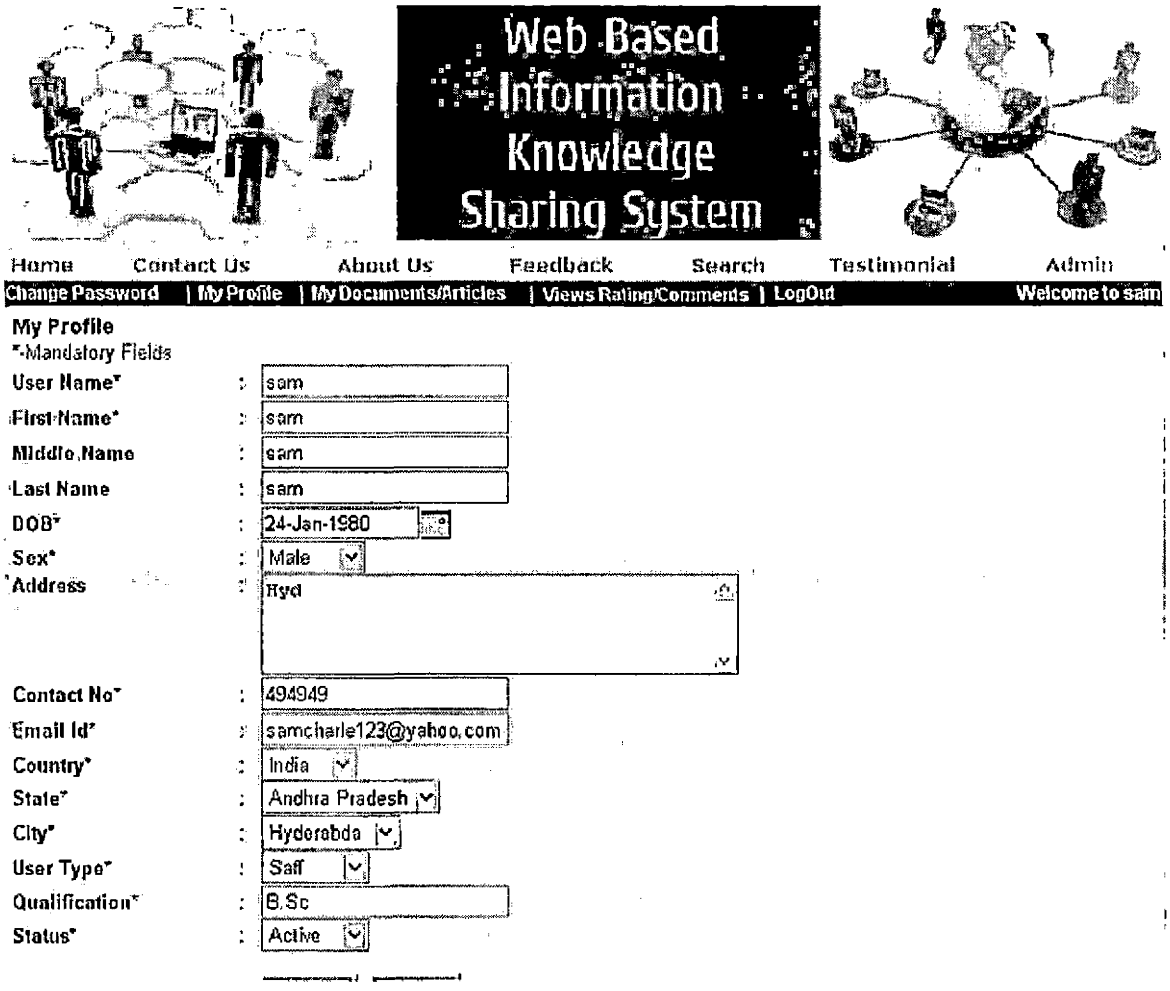

Figure 30. User Profile Page

User Documents Page –

This page allows user to view, upload or search for documents on the system.

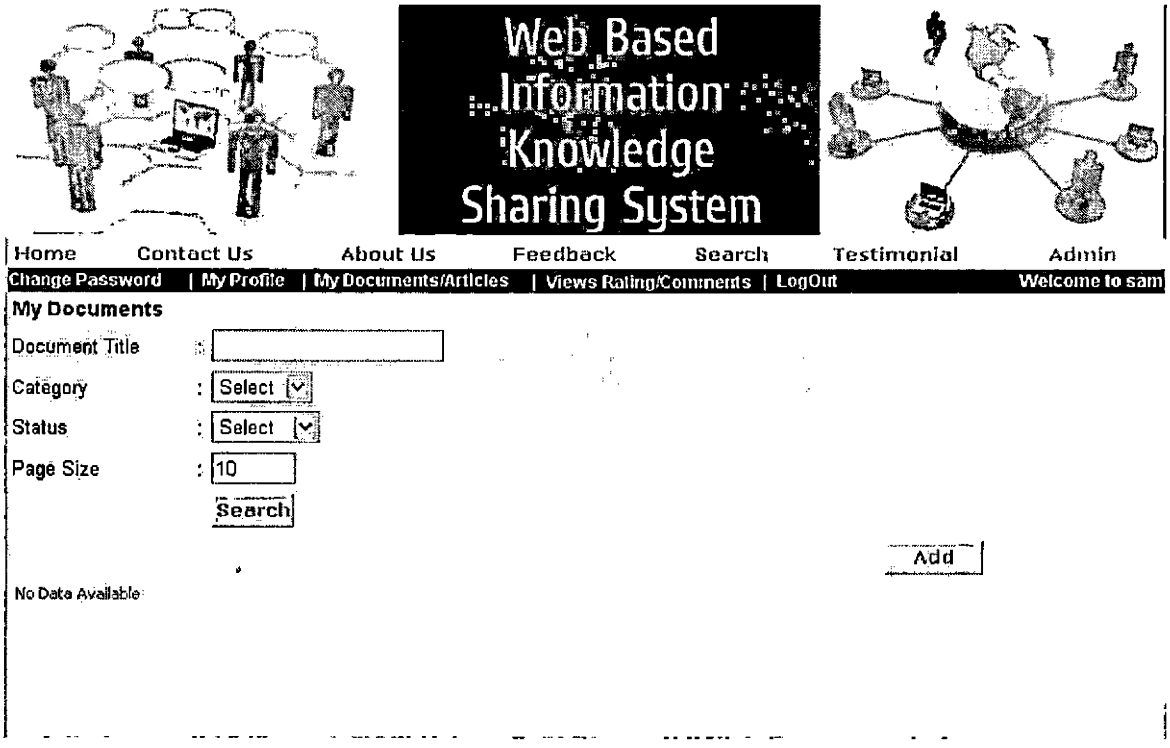

Figure 31. User Document Page

New User Registration Page -

This page allows new users to register into the system. The page asks for user details and once submitted will be sent to admin for review.

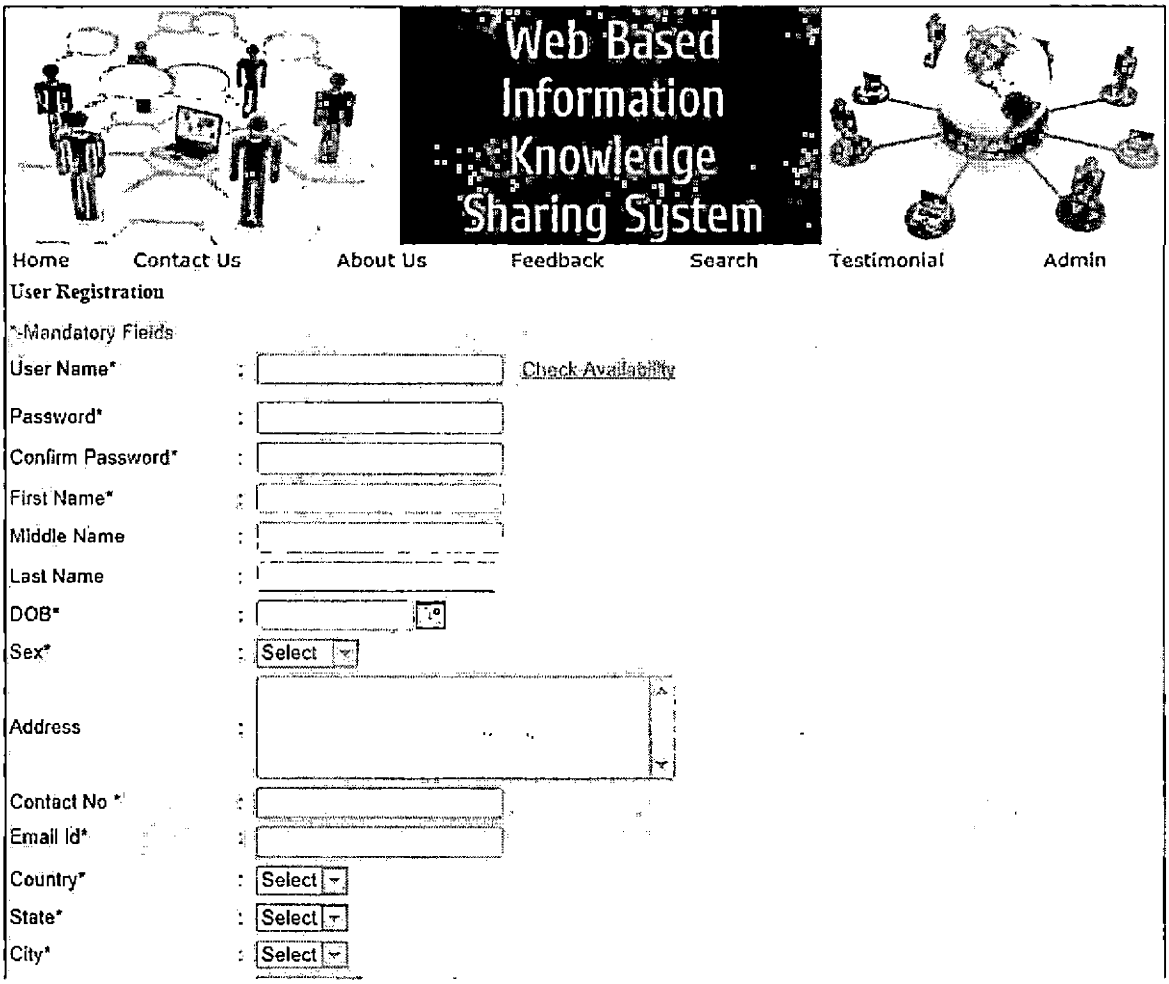

Figure 32. New User Registration Page

CHAPTER FIVE

SYSTEM VALIDATION

This section documents the performance and capabilities of the system, in this case 'WBIKS' System. There is need for validation because it is necessary to check if the application has met the SRS and it meets required results.

Following steps are involved in this validation and are together called software testing.

Unit Testing

Unit testing is a basic and first step in software testing and SLDC. In this testing, all individual units are tested and ensured that they work as intended. Extensive unit testing has been done on the application and results are shown below.

Table 1. Unit Testing Results Table

Page	Test Performed	Results
Static Pages	Check if all the style sheets are loaded when the site is loaded.	Pass
User Login page	1. Check for null input in the text field for username and password field. 2. Check for proper page redirect and menu for appropriate user after logging in. 3. Show error message if credentials are wrong.	Pass
Search page	1. Check to see if the page shows you the expected result. 2. Check for appropriate messages on search criteria.	Pass
Admin Home page	1. Check if all the options are displayed on the admin home page as intended. 2. Verify all the links are live.	Pass

	3. Check for duplicate links.	
Admin Login page	1. Check for proper page redirection. 2. Check to see that page shows error for using wrong credentials. 3. After login admin will be directed to admin home page with menu.	Pass
New User Registration page	1. Check if the page displayed with all the text boxes. 2. Check to see if data is successfully saved into database. 3. Redirect after successful registration.	Pass
Public pages	1. Check to see if public pages are accessible to all. 2. Check to see if content is displayed properly and search is working fine.	Pass
Contact Us Page	1. Check to see if all text boxes are displayed correctly 2. Check to see contact details are shown on the page.	Pass
Forgot Password	1. Check to see password is emailed for that user. 2. Check if it shows error message if	

	user is not found. 3. Check for accuracy of password retrieval.	Pass

Integration Testing

Integration testing also known as System integration testing is done after unit testing and is done to validate proper system integration and compliance with requirements. The following table shows integration testing for the 'WBIKS' system.

Table 2. System Integration Test Results Table

System / Module	Tests Performed	Results
Admin	1. Verify only authenticated users, in this case Admins, can enter into the admin module. 2. Verify that all page functions are properly functioning. 3. Check to see if admin can manage	Pass

	users/documents.	
Users	1. Verify only authenticated users can enter into user module. 2. Verify all the links on the user module are generated accordingly and are working. 3. Check to see if users can edit/view profile and view/upload documents.	Pass
Public User	1. Check to see if public users can access all public pages. 2. Verify all the links an menu are generated properly and work accordingly.	Pass

System Testing

System testing is a form of software testing to check if all the software modules used to develop the application are working properly. Below are the results

Table 3. System Test Results Table

System	Tests	Results
SQL Server	Server instances are working and are accessible. Server connections are maintained after server restart	Pass
IIS	IIS is always up and running and application is hosted properly and connection to the application is always on	Pass
Browser	Check for proper browser functionality	Pass
Data test	Test the system for errors with data and ensure system is working without errors.	Pass

CHAPTER SIX

CONCLUSION

'WBIKS' is an practical, useful web based application that can be tailored to specific requirements of schools and organization that need a low cost knowledge sharing system. The system is built using state of the art technologies and frameworks, therefore very reliable and the use of modular design means greater scalability. It is a feature rich web based system. The system has many merits some them are, The project offers user to enter data through simple and interactive forms. The user is mainly concerned with the validity of data he enters, the project offers conditional checks at every stage which enables data validation. The system offers the user extensive data manipulation capabilities but with restriction over key primary data which helps to maintain validity of data to longer extent.

On every aspect of the application, the user is provided with links through framing so that he can jump one option of the project to other without hassle and therefore

is very user friendly. The data storage and retrieval is enhanced because of the use of a separate Database tier during the design of the system. Being Web based, it is platform independent and is truly cross platform compatible. It enhances decision making process because of faster information processing. The system makes for a paperless knowledge sharing system across wide variety of systems and users.

There are certain limitations to the system. As the application is database driven, as number of users and activity grows the size of the databases grown and large databases are difficult to maintain and performance issues pop up. For proper use of the system, the users of the system have to be trained, which is cost involved.

Future Directions

The project development is directed towards building a custom tailored application for specific organizations and therefore have many limitations for general use. But future developments can provide better features and social integration and true multi-purpose application for general use at low costs.

APPENDIX A

Maintenance Manual

Maintenance is an essential part of software lifecycle and is very important to troubleshoot any problems that may arise during the continued usage of the software system. The following sections will explain the installation and configuration of all software modules used for developing and deploying this application.

Install .NET Framework

.NET framework is a Microsoft technology that enables seamless development of software using Microsoft technologies. .NET Framework is pre-installed on all Windows based systems. Updates to the Framework are released by Microsoft and they have to be downloaded from their site and updates can be installed using the downloaded executable. Current revision is .NET Framework 4.5.

Install Visual Studio.NET

Visual Studio.NET is an IDE developed by Microsoft and is a commercial software that can be purchased at Microsoft

site. Visual Studio can be installed with minimal user effort using the executable and it provides a pre-configured development scenarios that can used for software development using any .NET languages. Extensive help and support is provided for using the IDE. Current version of the IDE is Visual Studio.NET 2010.

Install IIS

IIS is a web server application developed by Microsoft and comes pre-installed with Windows Systems. It is necessary for hosting the application over the web. It is not enable by default on all Window machines, instructions on how to enable IIS on a system can be found on the Microsoft website. Current version of IIS is IIS7 and IIS7.

Install SQL Server

SQL Server is a database application developed by Microsoft and is an essential part of the project. It is a commercial software that can be purchased from Microsoft. Installation instructions can be found on installation media and on Microsoft's website. Configuring SQL server is needed to setup a database. SQL server is configured graphically and on screen instructions are provided when

configuring the server. Current version of SQL Server is SQL Server 2012.

Backup

It is necessary for all software systems using database, to make periodic backups of the database in order to prevent data loss or corruption. The backup can be made using in built menu of the SQL Server. Backups can performed in Full, Partial or specific file modes. It is also necessary to maintain backups of code, which can be done using Visual Studio backup menu.

APPENDIX B

This section contains code snippets from the project and for different modules.

The code contains both C# and ASP.NET code bits.

The following code is for default.aspx page, it is an aspx and C# code segment:

Apsx code snippet:

```
<%@ Page Language="C#" AutoEventWireup="true"
MasterPageFile="~/SimplePublicMaster.master"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
runat="Server">
    <table border="0" cellpadding="0" cellspacing="0" width="100%"
height="375">
        <tr>
            <td style="width: 100%" valign="top" align="left">
                <table width="100%" height="375">
                    <tr>
                        <td style="width: 500px" valign="top">
                            <table width="100%" border="0">
                                <tr>
                                    <td width="100%">
                                        <img src="Images/Banner1.jpg"
alt="" width="455px" />
                                    </td>
                                </tr>
                                <tr>
                                    <td align="justify">
                                        <div align="justify">
                                            <span style="font-family:
'Segoe UI'; font-size: 12px; font-weight: normal; letter-spacing: 1px">
                                                Knowledge Management
(KM) refers to a range of practices used by organizations to
                                                identify, create,
represent, and distribute knowledge for reuse, awareness and learning
                                                across the
organization. Knowledge Management programs are typically tied to
organizational
                                                objectives and are
intended to lead to the achievement of specific outcomes such
                                                as shared intelligence,
improved performance, competitive advantage, or higher levels
                                                of innovation. Here we
are looking at developing an online intranet knowledge management
```

```
                                                    system that is of
importance to either an organization or an educational institute.
                                                    The system (KMS) is an
Intranet based application that can be accessed throughout
                                                    the institute or a
specified group or department. This system can be used as a
knowledge/information
                                                    management system for
the institute. </span>
                                        </div>
                                    </td>
                                </tr>
                            </table>
                        </td>
                        <td valign="top" width="280">
                            <table width="100%">
                                <tr>
                                    <td colspan="3">
                                        <img src="Images\logo3.jpg"
alt="" width="300" />
                                    </td>
                                </tr>
                                <tr>
                                    <td colspan="3" class="LoginTitle">
                                        Student/Staff Login
                                    </td>
                                </tr>
                                <tr>
                                    <td colspan="3" align="center">
                                        <asp:Label ID="lblError"
runat="server" Text="Invalid Login Id/Password" CssClass="lblerror"

Visible="False"></asp:Label>
                                    </td>
                                </tr>
                                <tr>
                                    <td align="left">
                                        <b>Login ID</b></td>
                                    <td>
                                        <b>:</b></td>
                                    <td align="left">
                                        <asp:TextBox ID="txtLoginId"
runat="server"></asp:TextBox>
                                        <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server"
ControlToValidate="txtLoginId"

ErrorMessage="*"></asp:RequiredFieldValidator></td>
                                </tr>
                                <tr>
                                    <td align="left">
                                        <b>Password</b></td>
                                    <td>
                                        <b>:</b></td>
```

```
                                    <td align="left">
                                        <asp:TextBox ID="txtPassword"
runat="server" TextMode="Password" Width="150px"></asp:TextBox>
                                        <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server"
ControlToValidate="txtPassword"

ErrorMessage="*"></asp:RequiredFieldValidator></td>
                                </tr>
                                <tr>
                                    <td>
                                    </td>
                                    <td>
                                    </td>
                                    <td>
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                    </td>
                                    <td>
                                    </td>
                                    <td align="left">
                                        <asp:Button ID="btnSubmit"
runat="server" Text="Submit" OnClick="btnSubmit_Click"
                                                CssClass="btnstyle" />
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                         </td>
                                    <td>
                                    </td>
                                    <td align="right">
                                        <asp:LinkButton
ID="lnkBtnForgotPassword" runat="server" CausesValidation="False"

OnClick="lnkBtnForgotPassword_Click">Forgot
Password</asp:LinkButton> 
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                    </td>
                                    <td>
                                    </td>
                                    <td align="left">
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                    </td>
                                    <td>
                                    </td>
```

```
                                        <td align="left">
                                            <b>If New User</b>
                                            <asp:LinkButton
ID="lnkBtnRegistration" runat="server" CausesValidation="False"
OnClick="lnkBtnRegistration_Click">Register Here</asp:LinkButton></td>
                                    </tr>
                                    <tr>
                                        <td align="center" colspan="3">
                                            <img height="130"
src="Images/login.jpg" width="220" /></td>
                                    </tr>
                                </table>
                            </td>
                        </tr>
                    </table>
                </td>
            </tr>
        </table>
</asp:Content>
```

C# code snippet:

```
using System;
using System.Data;
using System.Configuration;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class _Default : System.Web.UI.Page
{
    clsUser_Logic objUser = new clsUser_Logic();
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        try
        {
            objUser.UserName = txtLoginId.Text.Trim();
            objUser.Password = txtPassword.Text.Trim();
            DataSet dsUserLoginDetail = objUser.GetUserLoginDetails();
            DataRowCollection drc = dsUserLoginDetail.Tables[0].Rows;
            if (drc.Count >= 1)
            {
                DataRow dr = drc[0];
                Session["UserName"] = dr["UserName"].ToString();
```

```
                Session["UserId"] = dr["UserId"].ToString();
                Response.Redirect("~/Users/frmMyProfile.aspx");
            }
            else
            {
                lblError.Visible = true;
            }
        }
        catch (Exception ex)
        {

            lblError.Text = ex.ToString();
        }
    }
    protected void lnkBtnRegistration_Click(object sender, EventArgs e)
    {
        Response.Redirect("~/Users/frmUserRegistration.aspx");
    }

    protected void lnkBtnForgotPassword_Click(object sender, EventArgs
e)
    {
        Response.Redirect("~/frmForgotPassword.aspx");
    }
}
```

The following code is for search.aspx page, which implements search functionality. The code is in C#:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class frmSearch : System.Web.UI.Page
{
    clsDocuments_Logic objDocument = new clsDocuments_Logic();
    clsCategory_Logic objCategory = new clsCategory_Logic();
    private string strError = "No Data Available";
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!Page.IsPostBack)
        {
            this.txtPageSize.Text = "10";
            BindCategory();
            BindData();
```

```
        }
    }

    public void BindCategory()
    {
        DataSet dsCategory = objCategory.GetCategory();
        ddlCategory.DataSource = dsCategory.Tables[0];
        ddlCategory.DataTextField = "CategoryName";
        ddlCategory.DataValueField = "CategoryId";
        ddlCategory.DataBind();
        ListItem li = new ListItem("Select", "0");
        ddlCategory.Items.Insert(0, li);
    }

    void BindData()
    {
        objDocument.DocumentTitle = txtDocumentTitle.Text.Trim();
        if(ddlCategory.SelectedItem.Text!="Select")
        objDocument.CategoryId =
Convert.ToInt32(ddlCategory.SelectedItem.Value);
        objDocument.StatusId = 3;
        objDocument.Sort_On = "";
        if (ViewState["Sort_On"] != null)
            objDocument.Sort_On = ViewState["Sort_On"].ToString() + " "
+ ViewState["Sort_By"].ToString();
        lblError.Visible = false;
        DataSet dsTemp = objDocument.GetDocument();
        DataTable dtTemp = dsTemp.Tables[0];
        if (dtTemp.Rows.Count > 0)
        {
            lblError.Visible = false;
        }
        else
        {
            lblError.Visible = true;
            lblError.Text = strError;
        }
        if (this.txtPageSize.Text != "")
        {
            if (System.Convert.ToInt32(this.txtPageSize.Text) > 0)
            {
                this.gvDocument.PageSize =
System.Convert.ToInt32(this.txtPageSize.Text);
            }
        }
        gvDocument.DataSource = dtTemp;
        gvDocument.DataBind();
        if (dtTemp.Rows.Count == 0)
        {
            this.Lbl_Pageinfo.Visible = false;
        }
        else
        {
            Int16 intTo;
```

```
            Int16 intFrom;
            if (gvDocument.PageSize * (gvDocument.PageIndex + 1) <
dtTemp.Rows.Count)
            {
                intTo = System.Convert.ToInt16(gvDocument.PageSize *
(gvDocument.PageIndex + 1));
            }
            else
            {
                intTo = System.Convert.ToInt16(dtTemp.Rows.Count);
            }
            intFrom = System.Convert.ToInt16((gvDocument.PageSize *
gvDocument.PageIndex) + 1);
            this.Lbl_Pageinfo.Text = "Record(s) " + intFrom + " to " +
intTo + " of " + dtTemp.Rows.Count;
            this.Lbl_Pageinfo.Visible = true;
        }
    }
    protected void gvDocument_PageIndexChanging(object sender,
GridViewPageEventArgs e)
    {
        if (ViewState["Sort_On"] != null)
            objDocument.Sort_On = ViewState["Sort_On"].ToString();
        else
            objDocument.Sort_On = "";
        gvDocument.PageIndex = e.NewPageIndex;
        BindData();
    }
    protected void gvDocument_Sorting(object sender,
GridViewSortEventArgs e)
    {
        objDocument.Sort_On = e.SortExpression;
        ViewState["Sort_On"] = objDocument.Sort_On;
        if (ViewState["Sort_By"] == null)
            ViewState["Sort_By"] = "Asc";
        if (ViewState["Sort_By"].ToString() == "Asc")
        {
            ViewState["Sort_By"] = "Desc";
        }
        else
        {
            ViewState["Sort_By"] = "Asc";
        }
        BindData();
    }
    protected void gvDocument_RowCommand(object sender,
GridViewCommandEventArgs e)
    {
        if (e.CommandName.ToUpper() == "UPDATE")
        {
            Response.Redirect("frmDocumentDetails.aspx?Id=" +
e.CommandArgument.ToString());
        }
    }
```

```
    protected void btnSearch_Click(object sender, EventArgs e)
    {
        strError = "No data matching with your searching criteria";
        gvDocument.PageIndex = 0;
        BindData();
    }
}
```

The following code is for forgotpassword.aspx and it implements a password recovery function. The code is in c#:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class frmForgotPassword : System.Web.UI.Page
{
    clsUser_Logic objUser = new clsUser_Logic();
    protected void Page_Load(object sender, EventArgs e)
    {
        lblError.Visible = false;
        Page.Validate();
    }
    protected void btnCheck_Click(object sender, EventArgs e)
    {
        Page.Validate();
        if (Page.IsValid)
        {
            lblError.Visible = false;
            objUser.UserName = txtMemberId.Text.Trim();
            if (objUser.checkUserIdForGetPwd())
            {
                if (!String.IsNullOrEmpty(objUser.EmailId))
                {
                    sendMAIL();

Response.Redirect("frmForgotPassword_Success.aspx");
                }
                else
                {
                    lblError.Text = "User doesn't have email Id,
Password can't be sent. Please Contact admin@kms.com";
                    lblError.Visible = true;
```

```
                    return;

                }
            }
            else
            {
                lblError.Text = "User Name doesn't exists. Please Try
again.";
                lblError.Visible = true;
                return;
            }
        }
    }
    private void sendMAIL()
    {
        string From, Subject, Salutation, username, Hearder,
MainContent, MainContent1, signature, footer, adminUser, adminRole,
strbody = "";
        username = "";
        Hearder = "";
        footer = "";
        adminUser = "";
        adminRole = "";

        string[] To ={ objUser.EmailId };
        string[] ccList = { string.Empty };
        From =
ConfigurationManager.AppSettings["AdminMail"].ToString();
        signature = "<b>Thank You</b><br><br>Knowledge Management
System";
        Subject = "Forgot Password";
        Salutation = "Dear " + objUser.FirstName;
        strbody = strbody + "<tr><td colspan=2> <font face=Verdana
size=2 color=black> Your User Name is :<B>" + objUser.UserName +
"</B></font></tr></td>";
        strbody = strbody + " <tr><td colspan=2><font face=Verdana
size=2 color=black> Your Password is :<B>" + objUser.Password +
"</B></font></tr></td>";
        MainContent = strbody;
        MainContent1 = "";
        Send_email(To, From, Subject, Salutation, username, Hearder,
MainContent, MainContent1, signature, footer, adminUser, adminRole,
ccList);
    }

    public void Send_email(string[] mailto, string mailFrom, string
strSubject, string strSalutation, string username, string strHearder,
string strMainContent, string strMainContent1, string strsignature,
string strfooter, string adminUser, string adminRole, string[] ccList)
    {
        string strbody = "";
        strbody = strbody +
"<html><HEAD><title>eMailTemplate</title><style>.formLabel";
```

```
        strbody = strbody + "{color:white;FONT-
FAMILY:Verdana,Arial,haettenschweiler; ";
        strbody = strbody + "font-size:10pt;background-
color:#737164;font-weight:normal;}";
        strbody = strbody + "</style></HEAD><body
MS_POSITIONING='GridLayout'>";
        strbody = strbody + "<TABLE id='Table1' cellSpacing='0'
align=center cellPadding='2' width='99%' border='1'>";
        strbody = strbody + "<TR><TD colspan=2 bgColor='#E7E7EF'><table
width='100%'><tr bgcolor=#FFFFFF height=28><td width='16%'
valign=middle><img alt='Logo' ";
        strbody = strbody + "src='" +
ConfigurationManager.AppSettings["InternalURL"] +
"images/logo1.jpg'></TD>";
        strbody = strbody + "</tr></table>";
        strbody = strbody + "<table border=0 width=99% valign=top
bgColor='#F3F3F3'>";
        strbody = strbody + "<tr><td colspan=2><font face=Verdana
size=2 color=black>" + strSalutation + ",</font></td></tr>";
        strbody = strbody + "<tr><td colspan=2 height=7></td></tr>";
        strbody = strbody + "<tr><td colspan=2><font face=Verdana
size=2 color=black>" + strHearder + "</font></td></tr>";
        strbody = strbody + strMainContent;
        strbody = strbody + strMainContent1;
        strbody = strbody + "<tr><td colspan=2><font face=Verdana
size=2 color=Black><br> " + strsignature + "<br> " + mailFrom
+ ".</font></td></tr>";
        strbody = strbody + "<tr><td colspan=2>" + "<font face=Verdana
size=2 color=red>" + strfooter + "</font></td></tr>";
        strbody = strbody + "</table>";
        strbody = strbody + "</body>";
        strbody = strbody + "</html>";
        clsCommon_Logic.SendMail(mailto, mailFrom, strbody, strSubject,
ccList);
        strbody = "";
    }
}
```

The following code is for documentdetails.aspx and is needed to generate document details while retrieving a document and uploading one. The code is in C#:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
public partial class Admin_frmDocumentDetails : System.Web.UI.Page
{
    string docName;
    clsDocuments_Logic objDocument = new clsDocuments_Logic();
    clsCommon_Logic objCommon = new clsCommon_Logic();
    clsCategory_Logic objCategory = new clsCategory_Logic();
    clsRatingComments_Logic objRatingComments = new
clsRatingComments_Logic();
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            if (Request["Id"] != null)
            {
                ViewState["Id"] = Request["Id"].ToString();
            }
            else
            {
                Response.Redirect("frmSearch.aspx");
            }
            BindData();
            BindRating();
        }
    }
    void BindRating()
    {
        DataTable dtTemp;
        dtTemp =
objRatingComments.GetAllRatings(Convert.ToInt32(ViewState["Id"].ToStrin
g())).Tables[0];
        int CountPerson = 0;
        ViewState["Rating"] = 0;
        if (dtTemp.Rows.Count > 0)
        {
            CountPerson = dtTemp.Rows.Count;
            DataRowCollection drc = dtTemp.Rows;
            foreach (DataRow dr in drc)
            {
                ViewState["Rating"] =
Convert.ToInt32(ViewState["Rating"].ToString()) +
Convert.ToInt32(dr["Rating"].ToString());
            }
            decimal RatingAvg =
Convert.ToInt32(ViewState["Rating"].ToString()) / CountPerson;
            lblRatings.Text = RatingAvg.ToString() + " Out Of 5";
        }
        else
        {
            lblRatings.Text = "No Rating has given";
        }

    }
```

```
    protected void BindData()
    {
        objDocument.DecumentId =
Convert.ToInt32(ViewState["Id"].ToString());
        objDocument.GetDocumentDetails();
        lblDocumentTitle.Text = objDocument.DocumentTitle;
        ViewState["doc"] = objDocument.DocumentUpload;
        docName = ViewState["doc"].ToString();
        lblShortDescription.Text =
objDocument.ShortDesc.ToString().Replace("\n","<br>");
        lblFullDescription.Text =
objDocument.FullDesc.ToString().Replace("\n", "<br>");
        lblKeyword.Text = objDocument.KeyWords;
        lblStatus.Text = objDocument.StatusName;
        lblCategory.Text = objDocument.CategoryName;

        string file = ViewState["doc"].ToString();
        if (file != "" && file != null)
        {
            string filePath = Server.MapPath("UploadedDocuments/" +
file);
            if(System.IO.File.Exists(filePath))
            {
                hlnkFileUpload.Text = ViewState["doc"].ToString();
                hlnkFileUpload.NavigateUrl = "~/UploadedDocuments/" +
ViewState["doc"].ToString();
                hlnkFileUpload.Target = "_blank";

            }
            else
            {
                hlnkFileUpload.Text = "No File available for Download";

            }
        }
    }

    protected void btnBack_Click(object sender, EventArgs e)
    {
        Response.Redirect("frmSearch.aspx");
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        ViewState["select"] = "";
        lblError.Visible = false;
        foreach (ListItem li in rdblRatings.Items)
        {
            if (li.Selected == true)
            {
                ViewState["select"] = "selected";
                break;
            }
            else
            {
```

```
                ViewState["select"] = "";
            }
        }
        if (ViewState["select"].ToString() != "selected")
        {
            lblError.Text = "Please fill all * mark fields";
            lblError.Visible = true;
            return;
        }
        if (txtWriteComments.Text.Trim() == "" ||
txtYourName.Text.Trim() == "" || txtEmailID.Text.Trim() == "")
        {
            lblError.Text = "Please fill all * mark fields";
            lblError.Visible = true;
            return;
        }

        foreach (ListItem li in rdblRatings.Items)
        {
            if (li.Selected == true)
            {
                objRatingComments.Pro_Rating
=Convert.ToInt32(li.Value.ToString());
            }
        }
        objRatingComments.Pro_Comments = txtWriteComments.Text.Trim();
        objRatingComments.Pro_CommentsBy = txtYourName.Text.Trim();
        objRatingComments.Pro_CommentsPersonEmailId =
txtEmailID.Text.Trim();
        try
        {
            if
(objRatingComments.GetAlreadyGivenRatingByPerson(Convert.ToInt32(ViewSt
ate["Id"].ToString())) == 0)
            {

objRatingComments.AddRatingComments(Convert.ToInt32(ViewState["Id"].ToS
tring()));
                Response.Write("<script>alert('Comments Added
Successfully')</script>");
            }
            else
            {
                lblError.Visible = true;
                lblError.Text = "You Have Already Given Rating for This
Documents/Articles/Information";
                return;
            }
        }
        catch (Exception ex)
        {
            lblError.Visible = true;
            lblError.Text = ex.Message;
        }
```

```
    }
}
```

The following code is needed to implement admin login page and code is in c#:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class Admin_frmAdminLogin : System.Web.UI.Page
{
    clsAdminLogin_Logic objAmin = new clsAdminLogin_Logic();
    protected void Page_Load(object sender, EventArgs e)
    {
        txtLoginId.Focus();
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        try
        {
            objAmin.AdminLoginId = txtLoginId.Text.Trim();
            objAmin.Password = txtPassword.Text.Trim();
            DataSet dsAdminLoginDetail =
objAmin.GetAdminLoginDetails();
            DataRowCollection drc = dsAdminLoginDetail.Tables[0].Rows;
            if (drc.Count > 0)
            {
                lblError.Visible = false;
                DataRow dr = drc[0];
                Session["AdminId"] = dr["AdminLoginId"].ToString();
                Response.Redirect("frmAdminHome.aspx");
            }
            else
            {
                lblError.Visible = true;
                lblError.Text = "Invalid Login ID/Password";
            }
        }
        catch (Exception ex)
        {

            lblError.Text = ex.Message.ToString();
```

```
        }
    }
}
```

The following code is for different admin functions like managing users, usertypes, state, country, document. The code is in both apsx and c#:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Drawing;

public partial class Admin_frmAddUser : System.Web.UI.Page
{
    clsCommon_Logic objCommon = new clsCommon_Logic();
    clsUser_Logic objUser = new clsUser_Logic();
    clsUserType_Logic objUserType = new clsUserType_Logic();
    protected void Page_Load(object sender, EventArgs e)
    {
        GMDatePkDOB.MaxDate = System.DateTime.Now;
        GMDatePkDOB.MinDate = System.DateTime.Now.AddYears(-100);
        if (!IsPostBack)
        {
            BindCountry();
            BindStatus();
            BindUserTypes();
        }
        GMDatePkDOB.Attributes.Add("readonly", "readonly()");

    }

    public void BindCountry()
    {
        DataSet dsCountry = objCommon.GetCountryName();
        ddlCountry.DataSource = dsCountry.Tables[0];
        ddlCountry.DataTextField = "CountryName";
        ddlCountry.DataValueField = "CountryId";
        ddlCountry.DataBind();
        ddlCountry.Items.Insert(0, "Select");
    }
```

```
    public void BindState()
    {
        DataSet dsState = objCommon.GetStateName();
        ddlState.DataSource = dsState.Tables[0];
        ddlState.DataTextField = "StateName";
        ddlState.DataValueField = "StateId";
        ddlState.DataBind();
        ddlState.Items.Insert(0, "Select");
    }
    public void BindCity()
    {
        DataSet dsCity = objCommon.GetDistrictName();
        ddlCity.DataSource = dsCity.Tables[0];
        ddlCity.DataTextField = "CityName";
        ddlCity.DataValueField = "CityId";
        ddlCity.DataBind();
        ddlCity.Items.Insert(0, "Select");
    }

    public void BindStatus()
    {
        DataSet dsStatus = objCommon.GetStatusName();
        ddlStatus.DataSource = dsStatus.Tables[0];
        ddlStatus.DataTextField = "StatusName";
        ddlStatus.DataValueField = "StatusId";
        ddlStatus.DataBind();
        ddlStatus.Items.Insert(0, "Select");
    }

    public void BindUserTypes()
    {
        DataSet dsUserType = objUserType.GetUserType();
        ddlUserType.DataSource = dsUserType.Tables[0];
        ddlUserType.DataTextField = "Typename";
        ddlUserType.DataValueField = "UserTypeId";
        ddlUserType.DataBind();
        ddlUserType.Items.Insert(0, "Select");
    }
    protected void lnkbtnCheckAvailability_Click(object sender,
EventArgs e)
    {
        if (txtUserName.Text.Trim() != "")
        {
            objUser.UserName = txtUserName.Text.Trim();
            lblExistsMemberId.Visible = true;
            if (objUser.CheckUserName())
            {
                lblExistsMemberId.ForeColor = Color.Red;
                lblExistsMemberId.Text = "Sorry,Already a User is
existing with this Name";
            }
            else
            {
                lblExistsMemberId.ForeColor = Color.Green;
```

```
                lblExistsMemberId.Text = "No User is existing with this
Name.So You can use with this";
            }
        }
        else
        {
            lblExistsMemberId.Visible = true;
            lblExistsMemberId.ForeColor = Color.Red;
            lblExistsMemberId.Text = "Please enter the User Name";
        }
    }
    protected void ddlCountry_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (ddlCountry.SelectedItem.Text != "Select")
        {
            objCommon.pro_CountryId =
Convert.ToInt32(ddlCountry.SelectedItem.Value);
            BindState();
        }
    }
    protected void ddlState_SelectedIndexChanged(object sender,
EventArgs e)
    {
        if (ddlState.SelectedItem.Text != "Select")
        {
            objCommon.pro_StateId =
Convert.ToInt32(ddlState.SelectedItem.Value);
            BindCity();
        }
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        objUser.UserName = txtUserName.Text.Trim();
        objUser.Password = txtPassword.Text.Trim();
        objUser.ContactNo = txtContactNo.Text.Trim();
        objUser.Address = txtAddress.Text.Trim();
        objUser.EmailId = txtEmailid.Text.Trim();
        objUser.CityId = Convert.ToInt32(ddlCity.SelectedItem.Value);
        objUser.CountryId =
Convert.ToInt32(ddlCountry.SelectedItem.Value);
        objUser.StateId = Convert.ToInt32(ddlState.SelectedItem.Value);
        objUser.DOB = Convert.ToDateTime(GMDatePkDOB.Date);
        objUser.Sex = ddlSex.SelectedItem.Text;
        objUser.FirstName = txtFirstName.Text.Trim();
        objUser.MiddleName = txtMiidleName.Text.Trim();
        objUser.LastName = txtLastName.Text.Trim();
        objUser.UserTypeId =
Convert.ToInt32(ddlUserType.SelectedItem.Value);
        objUser.Qualification = txtQualification.Text.Trim();
        objUser.StatusId =
Convert.ToInt32(ddlStatus.SelectedItem.Value);

        int i = objUser.AddUser();
```

```
        if (i == 1)
        {
            Response.Redirect("frmManageUser.aspx");
        }
        if (i == -2)
        {
            lblerror.Text = "Sorry This user name is already existing
please choose another username";
        }
    }
    protected void btnBack_Click(object sender, EventArgs e)
    {
        Response.Redirect("frmManageUser.aspx");
    }
}
```

In aspx:

```
<%-- Add User, Admin accesible only page/form, can be modified to
include more advanced users --%>
<%@ Page Language="C#" MasterPageFile="~/Admin/AdminMasterMenu.master"
AutoEventWireup="true" CodeFile="frmAddUser.aspx.cs"
Inherits="Admin_frmAddUser"  %>

<%@ Register Assembly="GMDatePicker" Namespace="GrayMatterSoft"
TagPrefix="cc1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder1"
Runat="Server">
<table border="0" cellpadding="0" cellspacing="0" width="100%">
        <tr>
            <td style="width: 100%" valign="top" align="left">
                <table width="100%" border="0" cellpadding="3"
cellspacing="0">
                    <tr>
                        <td colspan="4" class="LoginTitle"
style="height: 18px">
                            User Registration</td>
                    </tr>
                    <tr>
                        <td align="center" colspan="4">
                            <asp:Label ID="lblerror" runat="server"
CssClass="redtext" Style="position: relative" Font-Bold="True" Font-
Names="Verdana" Font-Size="9pt" ForeColor="Red"></asp:Label>
                        </td>
                    </tr>
                    <tr>
                        <td align="left" colspan="4" >
                            <span style="color:red">*-Mandatory
Fields</span>
                        </td>
                    </tr>
```

```
                    <tr>
                        <td width="25%">
                            User Name<span
class="redtext">*</span></td>
                        <td width="2%">
                            :</td>
                        <td width="40%">
                            <asp:TextBox ID="txtUserName"
runat="server" Width="160px"></asp:TextBox>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator2" runat="server"
ControlToValidate="txtUserName"

ErrorMessage="*"></asp:RequiredFieldValidator> 
                            <asp:LinkButton
ID="lnkbtnCheckAvailability" runat="server" CssClass="linkbtn"
CausesValidation="False"

OnClick="lnkbtnCheckAvailability_Click">Check
Availability</asp:LinkButton></td>
                        <td>
                             </td>
                    </tr>
                    <tr>
                        <td>
                        </td>
                        <td>
                        </td>
                        <td align="left" colspan="2">
                            <asp:Label ID="lblExistsMemberId"
runat="server" Text="" Visible="false"></asp:Label></td>

                    </tr>

                     <tr>
                        <td>
                            Password<span class="redtext">*</span></td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtPassword"
runat="server" CssClass="txt" Width="160px" TextMode="Password"/>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator4" runat="server"
ControlToValidate="txtPassword"

ErrorMessage="*"></asp:RequiredFieldValidator>
                        </td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Confirm Password*</td>
```

```
                        <td>
                            :</td>
                        <td align="left" colspan="2">
                            <asp:TextBox ID="txtConfirmPassword"
runat="server" CssClass="txt" Width="160px" TextMode="Password"/>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator7" runat="server"
ControlToValidate="txtPassword"

ErrorMessage="*"></asp:RequiredFieldValidator> 
                            <asp:CompareValidator
ID="CompareValidator1" runat="server" ControlToCompare="txtPassword"
                                ControlToValidate="txtConfirmPassword"
ErrorMessage="Conform Password should be same as
Password"></asp:CompareValidator></td>

                    </tr>
                    <tr>
                        <td>
                            First Name<span
class="redtext">*</span></td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtFirstName"
runat="server" CssClass="txt" Width="160px" />
                            <asp:RequiredFieldValidator
ID="rfvFirstName" runat="server" ControlToValidate="txtFirstName"

ErrorMessage="*"></asp:RequiredFieldValidator></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Middle Name</td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtMiidleName"
runat="server" CssClass="txt" Width="160px" /></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Last Name</td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtLastName"
runat="server" CssClass="txt" Width="160px" /></td>
                        <td>
                        </td>
```

```
                    </tr>
                    <tr>
                        <td>
                            DOB*</td>
                        <td>
                            :</td>
                        <td align="left">

                         <cc1:GMDatePicker ID="GMDatePkDOB"
runat="server" YearDropDownRange="200" TextBoxWidth="100"

MaxDate="9999-12-31" InitialValueMode="Null"

DateFormat="dd-MMM-yyyy" CalendarWidth="188px" CalendarTheme="Blue"
NoneButtonText="Clear" EnableTheming="True">

</cc1:GMDatePicker>



                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator1" runat="server" ErrorMessage="*"

ControlToValidate="GMDatePkDOB"></asp:RequiredFieldValidator></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Sex*</td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:DropDownList ID="ddlSex"
runat="server">
                                <asp:ListItem>Select</asp:ListItem>
                                <asp:ListItem>Male</asp:ListItem>
                                <asp:ListItem>Female</asp:ListItem>
                            </asp:DropDownList>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator5" runat="server" ControlToValidate="ddlSex"
                                ErrorMessage="*"
InitialValue="Select"></asp:RequiredFieldValidator></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Address</td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtAddress" runat="server"
TextMode="MultiLine" Width="276px" Height="65px"></asp:TextBox></td>
```

```
                        <td>
                        </td>
                    </tr>

                    <tr>
                        <td>
                            Contact No <span
class="redtext">*</span></td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtContactNo"
OnKeypress="return onlyNumbershifen(event)" runat="server"
CssClass="txt" Width="160px"/>
                            <asp:RequiredFieldValidator
ID="rfvContactNo" runat="server" ControlToValidate="txtContactNo"

ErrorMessage="*"></asp:RequiredFieldValidator>
                        </td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td valign="top">
                            Email Id<span class="redtext">*</span></td>
                        <td valign="top">
                            :</td>
                        <td align="left" valign="middle">
                            <asp:TextBox ID="txtEmailid" runat="server"
CssClass="txt" Width="160px" />
                            <asp:RequiredFieldValidator ID="rfvEmailI"
runat="server" ControlToValidate="txtEmailid"

ErrorMessage="*"></asp:RequiredFieldValidator></td>
                        <td></td>
                    </tr>
                    <tr>
                        <td>
                            Country<span class="redtext">*</span></td>
                        <td >
                            :</td>
                        <td align="left">
                            <asp:DropDownList ID="ddlCountry"
runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlCountry_SelectedIndexChanged">
                                <asp:ListItem>Select</asp:ListItem>
                            </asp:DropDownList>
                            <asp:RequiredFieldValidator ID="rfvCountry"
runat="server" ControlToValidate="ddlCountry"
                                ErrorMessage="*"
InitialValue="Select"></asp:RequiredFieldValidator>
                        </td>

                        <td align="left">
```

```
                    </td>

                </tr>
                <tr>
                    <td>
                        State<span class="redtext">*</span></td>
                    <td>
                        :</td>
                    <td align="left">
                        <asp:DropDownList ID="ddlState"
runat="server" AutoPostBack="True"
OnSelectedIndexChanged="ddlState_SelectedIndexChanged">
                            <asp:ListItem>Select</asp:ListItem>
                        </asp:DropDownList>
                        <asp:RequiredFieldValidator ID="rfvState"
runat="server" ControlToValidate="ddlState"
                            ErrorMessage="*"
InitialValue="Select"></asp:RequiredFieldValidator>
                    </td>
                    <td></td>
                </tr>

                <tr>
                    <td>
                        City<span class="redtext">*</span>
                    </td>
                    <td>
                        :</td>
                    <td align="left">
                        <asp:DropDownList ID="ddlCity"
runat="server">
                            <asp:ListItem>Select</asp:ListItem>
                        </asp:DropDownList>
                        <asp:RequiredFieldValidator ID="rfvCity"
runat="server" ControlToValidate="ddlCity"
                            ErrorMessage="*"
InitialValue="Select"></asp:RequiredFieldValidator>
                    </td>
                    <td></td>
                </tr>
                <tr>
                    <td>
                        User Type</td>
                    <td>
                        :</td>
                    <td align="left">
                        <asp:DropDownList ID="ddlUserType"
runat="server">
                            <asp:ListItem>Select</asp:ListItem>
                        </asp:DropDownList>
                        <asp:RequiredFieldValidator
ID="RequiredFieldValidator3" runat="server"
ControlToValidate="ddlUserType" InitialValue="Select"
```

```
ErrorMessage="*"></asp:RequiredFieldValidator></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td>
                            Qualification</td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:TextBox ID="txtQualification"
runat="server" Width="160px"></asp:TextBox>
                            <asp:RequiredFieldValidator
ID="rfvQualification" runat="server"
ControlToValidate="txtQualification"

ErrorMessage="*"></asp:RequiredFieldValidator></td>
                        <td>
                        </td>
                    </tr>


                     <tr>
                        <td>
                            Status<span class="redtext">*</span>
                        </td>
                        <td>
                            :</td>
                        <td align="left">
                            <asp:DropDownList ID="ddlStatus"
runat="server">
                                <asp:ListItem>Select</asp:ListItem>
                            </asp:DropDownList>
                            <asp:RequiredFieldValidator
ID="RequiredFieldValidator6" runat="server"
ControlToValidate="ddlStatus"
                                ErrorMessage="*"
InitialValue="Select"></asp:RequiredFieldValidator>
                        </td>
                        <td></td>
                    </tr>
                    <tr>
                        <td>
                        </td>
                        <td>
                        </td>
                        <td align="left">
                        </td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td align="right">
```

```
                        </td>
                        <td>
                        </td>
                        <td align="left">
                            <asp:Button ID="btnSubmit" runat="server"
Text="Submit" CssClass="btnstyle" OnClick="btnSubmit_Click"
                                Width="65px" />  <asp:Button
ID="btnBack" runat="server" Text="Back" CssClass="btnstyle"
OnClick="btnBack_Click"
                                Width="65px" CausesValidation="False"
/></td>
                        <td>
                        </td>
                    </tr>
                    <tr>
                        <td align="right">
                             </td>
                        <td>
                        </td>
                        <td align="left">
                             </td>
                        <td>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>


</asp:Content>
```

REFERENCES

[1] SDK [Accessed: 22 October 2012]

http://www.webopedia.com/TERM/S/SDK.html

[2] .NET Framework [Accessed: 30 September 2011]

http://www.webopedia.com/TERM/D/dot_NET_Framework.html

[3] Visual Studio.NET [Accessed: 22 October 2012]

http://www.webopedia.com/TERM/V/VSIP.html

[4] ASP.NET [Accessed: 22 October 2012]

http://en.wikipedia.org/wiki/ASP.NET

[5] ADO.NET [Accessed: 22 October 2012]

http://en.wikipedia.org/wiki/ADO.NET

[6] API [Accessed: 22 October 2012]

http://www.webopedia.com/TERM/A/API.html

[7] HTTP [Accessed: 22 October 2012]

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

[8] UML [Accessed: 22 October 2012]

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

[9] XML [Accessed: 22 October 2012]

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

[10] Dia UML tool [Accessed: 22 October 2012]

http://en.wikipedia.org/wiki/Dia_(software)

[11] HTML [Accessed: 22 October 2012]

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

[12] CSS [Accessed: 22 October 2012]

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

[13] JavaScript [Accessed: 22 October 2012]

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

[14] IIS [Accessed: 22 October 2012]

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

[15] N-Tier Architecture

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

[16] IIS Server Configuration

http://support.microsoft.com/kb/323972

[17] SQL Server Setup

http://technet.microsoft.com/en-us/sqlserver/install.aspx

[18] XML ASP.NET configuration

http://stackoverflow.com/questions/4277286/using-xml-config-in-asp-net-application

[19] Site templates

http://www.microsoft.com/web/post/how-to-use-the-starter-site-template-for-aspnet-web-pages