

A Project Report
on

KNOWLEDGE MANAGEMENT SYSTEM

Submitted in partial fulfillment of the requirements for the Major Project II of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Deeksha Gupta
Priya Gupta

R780209011
R780209022

Under the guidance
Mr. G L Prakash
Assistant Professor



Department of Computer Science & Engineering
COLLEGE OF ENGINEERING STUDIES
UNIVERSITY OF PETROLEUM & ENERGY STUDIES
Dehradun- 248007
April 2013

DECLARATION

We, Deeksha Gupta and Priya Gupta bearing the Roll No: R780209011 and R780209022 respectively hereby declare that this Project work entitled "Knowledge Management System" was carried out by us under the guidance and supervision of Mr. G L Prakash. This Project work is submitted to University of Petroleum & Energy Studies in partial fulfilment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering during the Academic Semester Jan 2013 - April - 2013. We also declare that, we have not submitted this dissertation work to any other university for the award of either degree or diploma.

Place: Dehradun

Deeksha Gupta
Deeksha Gupta

Priya Gupta
Priya Gupta

Date: 26th April '13

ABSTRACT

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. Students/Staff logging in should be able to upload any kind of educational information. Students/staff logging in may also access/search any information put up by others. KMS should facilitate knowledge sharing from the grass root level like project teams to departments to the entire college.

ACKNOWLEDGEMENT

It is a pleasure to thank all those great many people who helped, supported and encouraged us during this project work.

Firstly we express our sincere gratitude to Mr. G L Prakash, the guide of the project who carefully and patiently lent his valuable time and effort to give directions as well as to correct various documents with attention and care.

It is a great honour to do this project in this esteemed institution, and we would extend our thanks to the HOD, Prof Manish Prateek and other faculty members who have shared their vast knowledge and experience during our stay.

We do also like to appreciate the consideration of the Project Coordinator, our Faculties and colleagues, which enabled us to balance our work along with this project. It was their attitude that inspired us to do such an efficient and apposite work.

We are indebted to those people across the globe who have shared their knowledge and perspectives in the form of online tutorials, forums and other resources which helped us to a great extent whenever we met with technical obstacles during this endeavour.

We wish to avail this opportunity to express a sense of gratitude and love to all our friends and our family for their unwavering support, strength, help and in short for everything they have done during the crucial times of the progress of our project.

Last but not the least we thank GOD ALMIGHTY for HIS blessings and guidance without which this dream project wouldn't have been reality.

Deeksha Gupta

Priya Gupta

CONTENTS

Declaration	I
Abstract	li
Acknowledgement	lii
Contents	lv
List of figures	Vii

1 Introduction	01 – 02
1.1 What is Knowledge Management System?	01
1.2 Why KMS?	02
2 Problem Definitions	03
3 Literature Survey	04 – 08
3.1 Introduction to .NET Framework	04
3.1.1 Common Language Runtime	05
3.1.2 .NET framework Class Library	06
3.2 Client Application Development	06
3.3 Server Application Development	07
3.4 ASP.NET	07
3.5 SQL Server	08
4 System Requirement Specification	09 – 14
4.1 Overall Description	09
4.1.1 Product Perspective	09
4.1.2 Product Features	09
4.1.3 Inputs & Outputs	10
4.1.4 Input Design	10
4.1.5 Output Design	10
4.1.6 SDLC Methodologies	11

Knowledge Management System

4.2	System Features	12
4.2.1	Administrative User Interface	12
4.2.2	Operational User Interface	12
4.3	Non-functional Requirements	13
4.3.1	Performance Requirements	13
4.3.2	Security Requirements	13
	(A) Client Side Validation	14
	(B) Server Side Validation	
5	System Design	
5.1	Detailed Design	
5.1.1	Administration Module	
5.1.2	User module	
5.1.3	Public User Module	
5.1.4	Document Module	
5.1.5	Reports	
5.1.6	Authentication Module	
6	Testing	23 – 25
6.1	Introduction	23
6.1.1	Strategic Approach to S/W Testing	23
6.2	Unit Testing	24
6.2.1	White Box Testing	24
6.2.2	Basic Path Testing	24
6.2.3	Conditional Testing	25
6.2.4	Data Flow Testing	25
6.2.5	Loop Testing	25
7	Conclusion	26
	Bibliography	27
	Appendix	28-32

LIST OF FIGURES

No.	Description	Page No.
FIGURES		
Fig: 4.1	Water Fall Model	11
Fig: 5.1	High level Design	15
Fig: 5.2	Use Case Diagram	16
Fig: 5.3	DFD for Administration Module	17
Fig: 5.4	DFD for Adding Staff	17
Fig: 5.5	DFD for Adding Students	18
Fig: 5.6	DFD for Student Process	19
Fig: 5.7	DFD for User Module	19

CHAPTER 1

INTRODUCTION

1.1 What is Knowledge Management System?

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. Students/Staff logging in should be able to upload any kind of educational information. Students/staff logging in may also access/search any information put up by others. KMS should facilitate knowledge sharing from the grass root level like project teams to departments to the entire college.

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.

1.2 Why KM?

Knowledge Management (KM) refers to a multi-disciplined approach to achieve organizational objectives by making the best use of knowledge. KM focuses on processes such as acquiring, creating and sharing knowledge and the cultural and technical foundations that support them.

Knowledge management draws from a wide range of disciplines and technologies:

- Cognitive science
- Computer-supported collaborative work (groupware)
- Library and information science
- Decision support systems
- Semantic networks
- Organizational science
- performance support systems

A number of management consulting firms had begun in-house knowledge management programs. Knowledge management was introduced in the popular press, the most widely read work to date is Ikujiro Nonaka's and Hirotaka Takeuchi's *The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation* (1995). Knowledge management has become big business for such major international consulting firms as *Ernst & Young, Arthur Andersen, and Booz-Allen & Hamilton.*

CHAPTER 2

PROBLEM DEFINITION

The future of the existing system has to be seen as one of continuous change, where the information that is stored is of increasing complexity and quantity. The existing system is required to provide a useable and well managed interface for student, academic and administrator users to view and manipulate the data for which it is responsible. For each it must allow the rapid formulation and resolution of queries related to the student information. There is also a requirement for the system to interact with other information sources as required, both as an information source and as a consumer of related information during the resolution of queries.

CHAPTER 3

LITERATURE SURVEY

3.1 Introduction to .NET framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. 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 guarantees 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 has two main components: 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 ensure 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, the other main component of the .NET Framework, 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.

Knowledge Management System

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.

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.

3.1.1 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 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.

3.1.2 .NET Framework Class Library

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.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.
- XML Web services.
- Windows services.

3.2 Client Application Development

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

The Windows Forms classes contained in the .NET Framework are designed to be used for GUI development. You can easily create command windows, buttons, menus, toolbars, and other screen elements with the flexibility necessary to accommodate shifting business needs.

For example, the .NET Framework provides simple properties to adjust visual attributes associated with forms. In some cases the underlying operating system does not support changing these attributes directly, and in these cases the .NET Framework automatically recreates the forms. This is one of many ways in which the .NET Framework integrates the developer interface, making coding simpler and more consistent.

3.3 Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and SQL Server can perform standard operations while your application logic executes through the managed code.

3.4 ASP.NET

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data

Knowledge Management System

format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- Enhanced Performance.
- Power and Flexibility
- Simplicity.
- Manageability.
- Scalability and Availability.
- Customizability and Extensibility
- Security.

3.5 SQL Server

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

CHAPTER 4 SYSTEM REQUIREMENT SPECIFICATION

4.1 Overall Description

4.1.1 Product Perspective

The Knowledge Mgmt System is a web based application that can be accessed throughout the organization or a specified group/Dept. This system can be used as a knowledge/information mgmt system for the college. Students/Staff logging should be able to upload any kind of technical information. Students/staffs logging in may also access/search any information put up by others. Knowledge Mgmt System should facilitate knowledge sharing from the grass root level like project teams to the entire college.

4.1.2 Product Features

The major functionalities of the proposed system are:

- There is well defined categories under which one can post information
- KMS supports posting information under these categories.
- KMS supports uploading of word / pdf / excel / ppt documents under the categories
- There is a Search feature and the results should be displayed in a very effective way.
- User friendliness is provided in the application with various controls.
- The system makes the overall project management much easier and flexible.
- It can be accessed over the Internet.
- Various classes have been used to provide file upload and mail features.
- There is no risk of data mismanagement at any level while the project development is under process.
- Report generation feature is provided using Crystal Reports to generate different kinds of reports like bar graphs, pie charts and table type charts etc.
- It provides high level of security using different protocols like https etc.

4.1.3 Inputs & Outputs

The main inputs, outputs and major functions of the system are as follows.

Inputs:

- Admin enters his or her user id and password.
- Users enter his or her user id and password.
- User requests the reports.
- User requests the search.
- Public user requests the search.
- Admin can edit the personal details and so on.

Outputs:

- Admin receives personal details.
- Users receive the personal details.
- Users can see the document details.
- Displays search result.

4.1.4 Input Design

Input design is a part of overall system design. The main objective during the input design is as given below:

- To produce a cost-effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and understood by the user.

4.1.5 Output Design

Outputs from computer systems are required primarily to communicate the results of processing to users. They are also used to provide a permanent copy of the results for later consultation. The various types of outputs in general are:

- External Outputs, whose destination is outside the organization.
- Internal Outputs whose destination is within organization and they are the
 - User's main interface with the computer.
- Operational outputs whose use is purely within the computer department.
- Interface outputs

4.1.6 SDLC Methodologies

This document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

WATER FALL MODEL was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

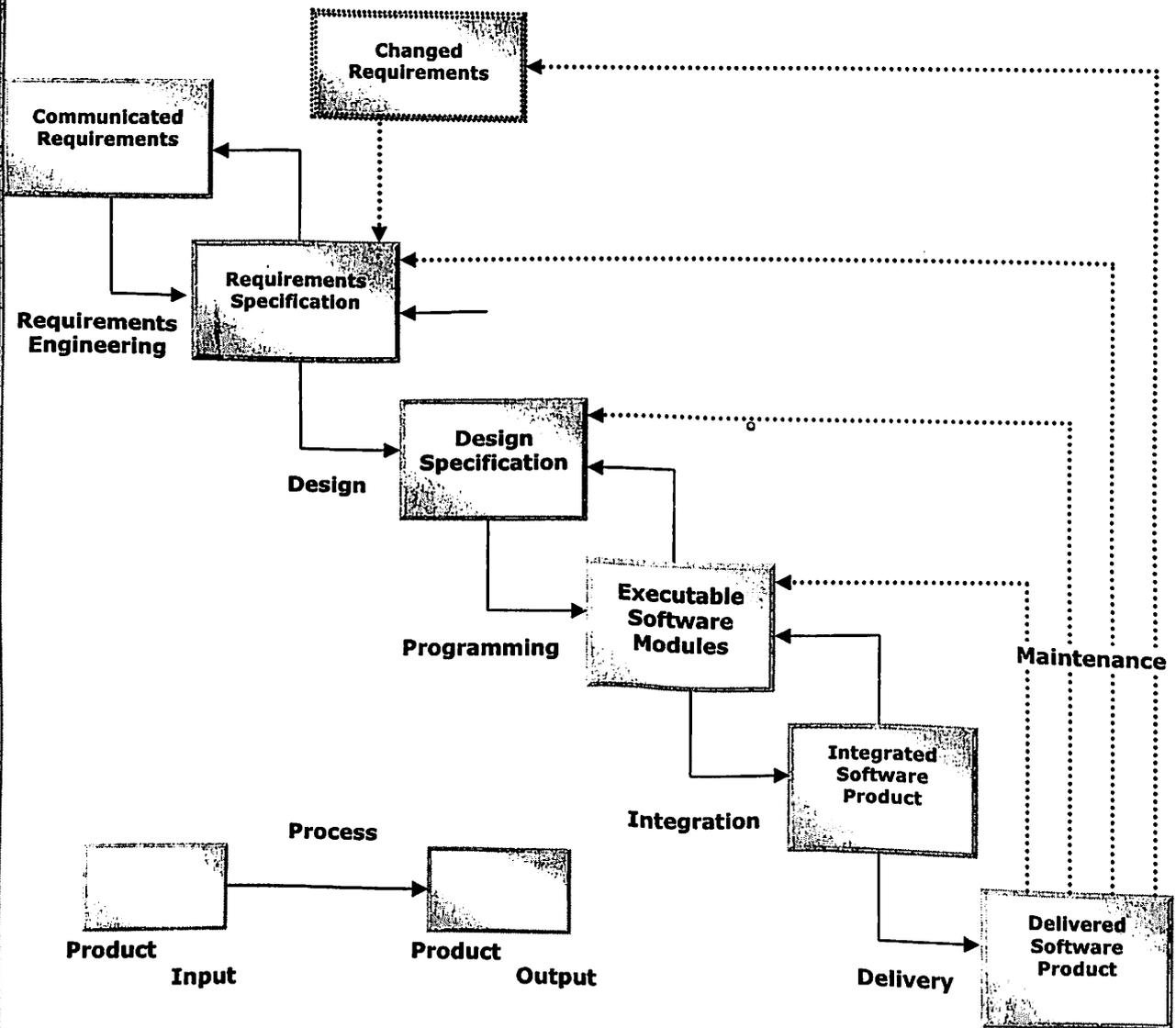


Fig 4.1: Water Fall Model

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
 - Maintaining the system for a period of one year after installation.

4.2 System Features

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a browser interface. The GUI'S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

4.2.1 Administrative User Interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

4.2.2 Operational User Interface

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

4.3 Non-functional Requirements

4.3.1 Performance Requirements

Performance is measured in terms of the output provided by the application.

Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely in the part of the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

4.3.2 Security Requirements

System security refers to various validations on data in form of checks and controls to avoid the system from failing. It is always important to ensure that only valid data is entered and only valid operations are performed on the system. The system employs two types of checks and controls:

(A) CLIENT SIDE VALIDATION

Knowledge Management System

Various client side validations are used to ensure on the client side that only valid data is entered. Client side validation saves server time and load to handle invalid data. Some checks imposed are:

- JavaScript is used to ensure those required fields are filled with suitable data only. Maximum lengths of the fields of the forms are appropriately defined.
- Forms cannot be submitted without filling up the mandatory data so that manual mistakes of submitting empty fields that are mandatory can be sorted out at the client side to save the server time and load.

(B) SERVER SIDE VALIDATION

Some checks cannot be applied at client side. Server side checks are necessary to save the system from failing and intimating the user that some invalid operation has been performed or the performed operation is restricted. Some of the server side checks imposed is:

- Server side constraint has been imposed to check for the validity of primary key and foreign key. A primary key value cannot be duplicated. Any attempt to duplicate the primary value results into a message intimating the user about those values through the forms using foreign key can be updated only of the existing foreign key values.
- User is intimating through appropriate messages about the successful operations or exceptions occurring at server side.
- Various Access Control Mechanisms have been built so that one user may not agitate upon another. Access permissions to various types of users are controlled according to the organizational structure. Only permitted users can log on to the system and can have access according to their category. User- name, passwords and permissions are controlled on the server side.
- Using server side validation, constraints on several restricted operations are imposed.

CHAPTER 5

SYSTEM DESIGN

The idea for implementing is to strengthen the functioning of Audit Status Monitoring and make them effective and better. The proposed software will cover the information needs with respect to each request of the user group viz. accepting the request, providing vulnerability document report and the current status of the audit.

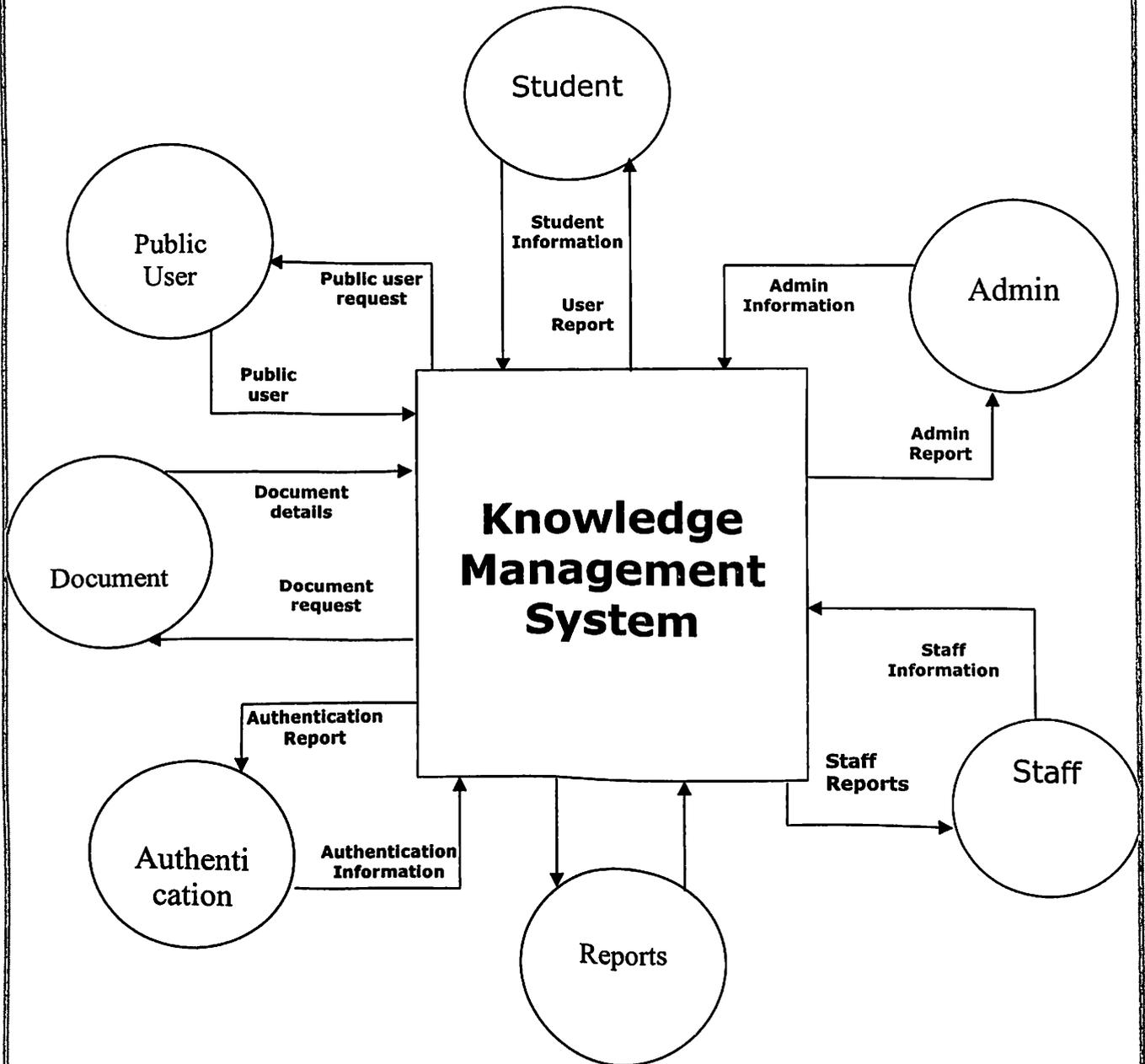


Fig: 5.1 - High-level Design

5.1 Detailed Design

The System is mainly having six modules:

- Administration
- Users(Students, Staff)
- Public Users
- Documents
- Reports
- Authentication

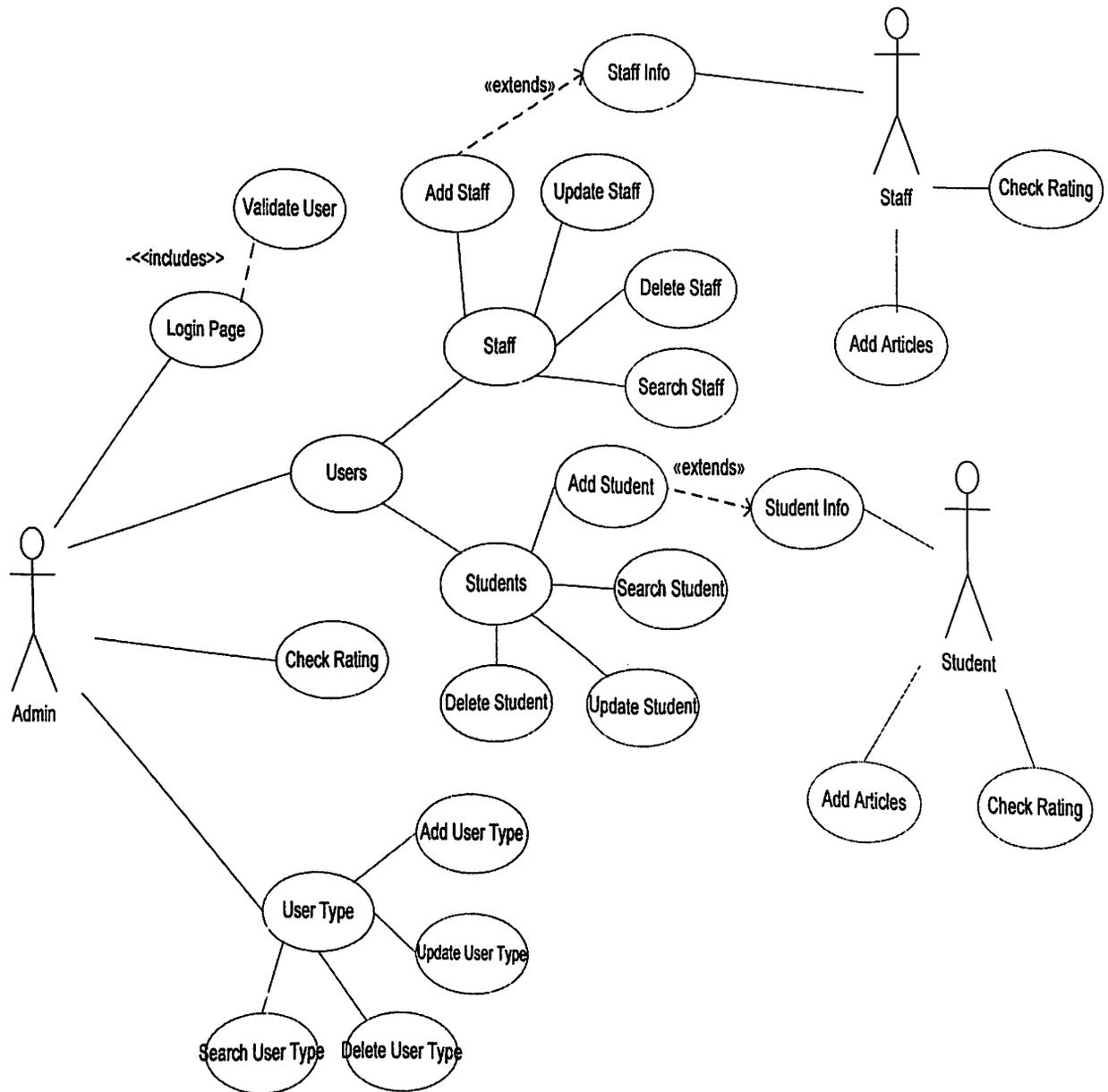


Fig: 5.2 – Use-Case Diagram

5.1.1 Administrator Module

In this module the Administrator has the privileges to add all the Users (Students of Staff), Document Type and delete any documents which was posted by user. He can search all the info about the Documents and Users. He can read as well as delete any irrelevant posts by the members.

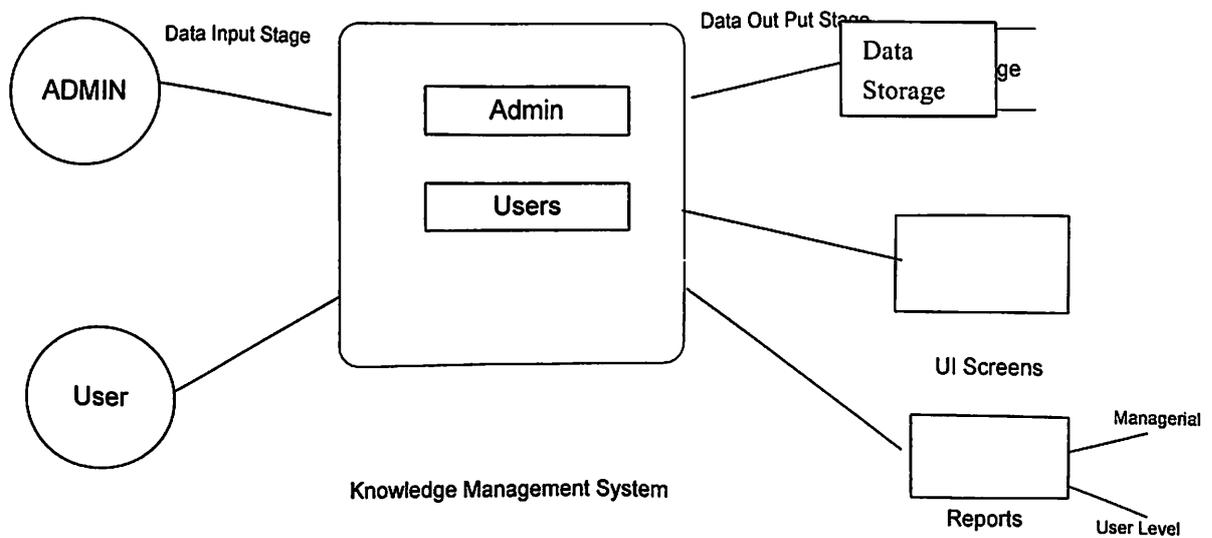


Fig: 5.3 - DFD for Administrator Module

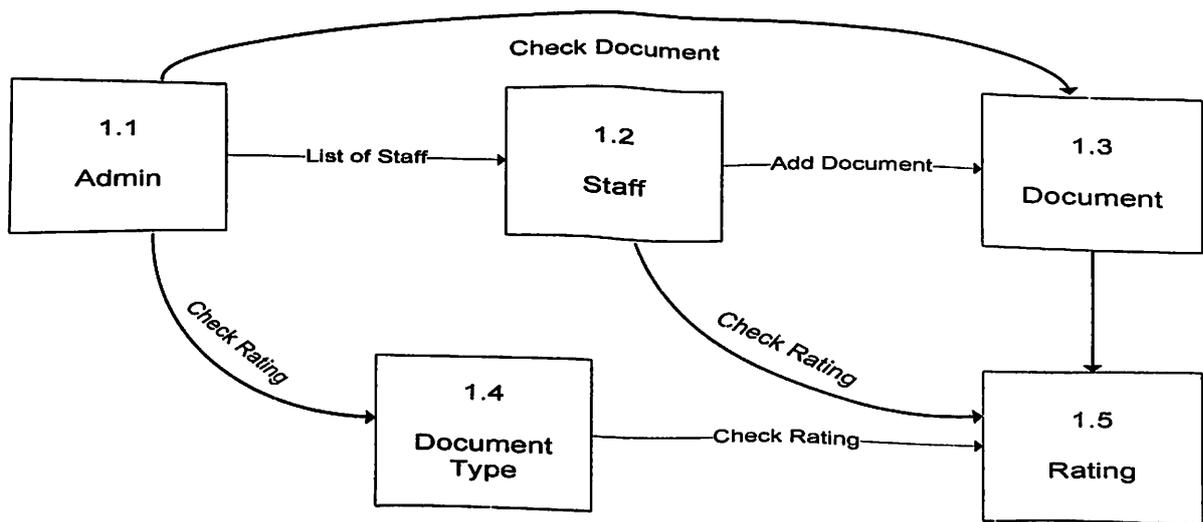


Fig: 5.4 - DFD for Adding Staff

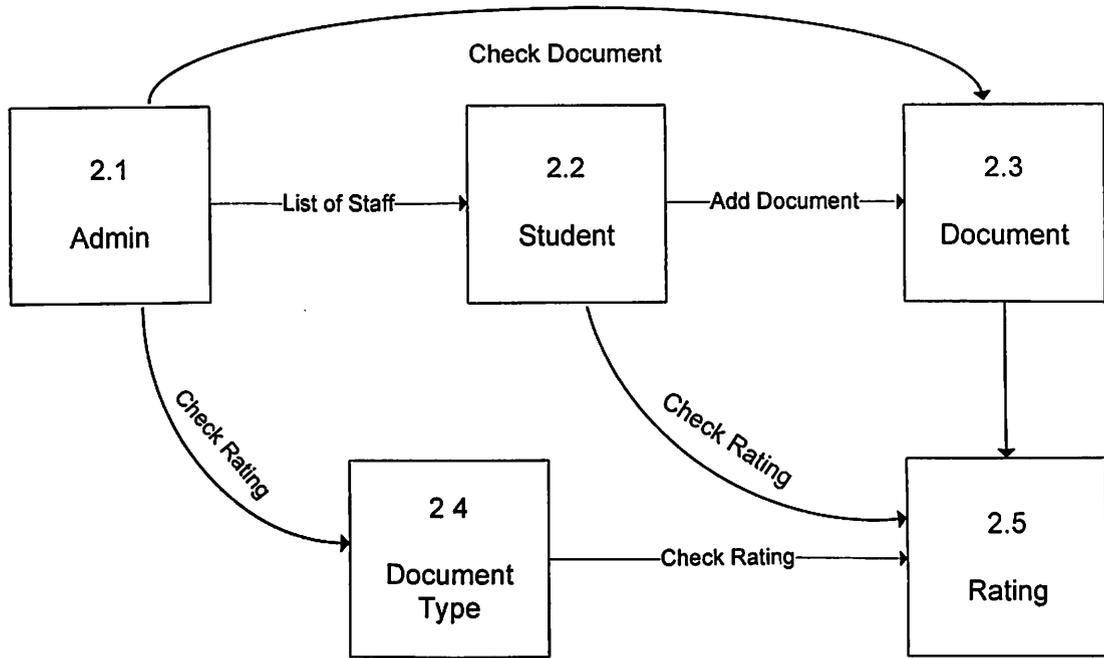


Fig: 5.5 - DFD for Adding student

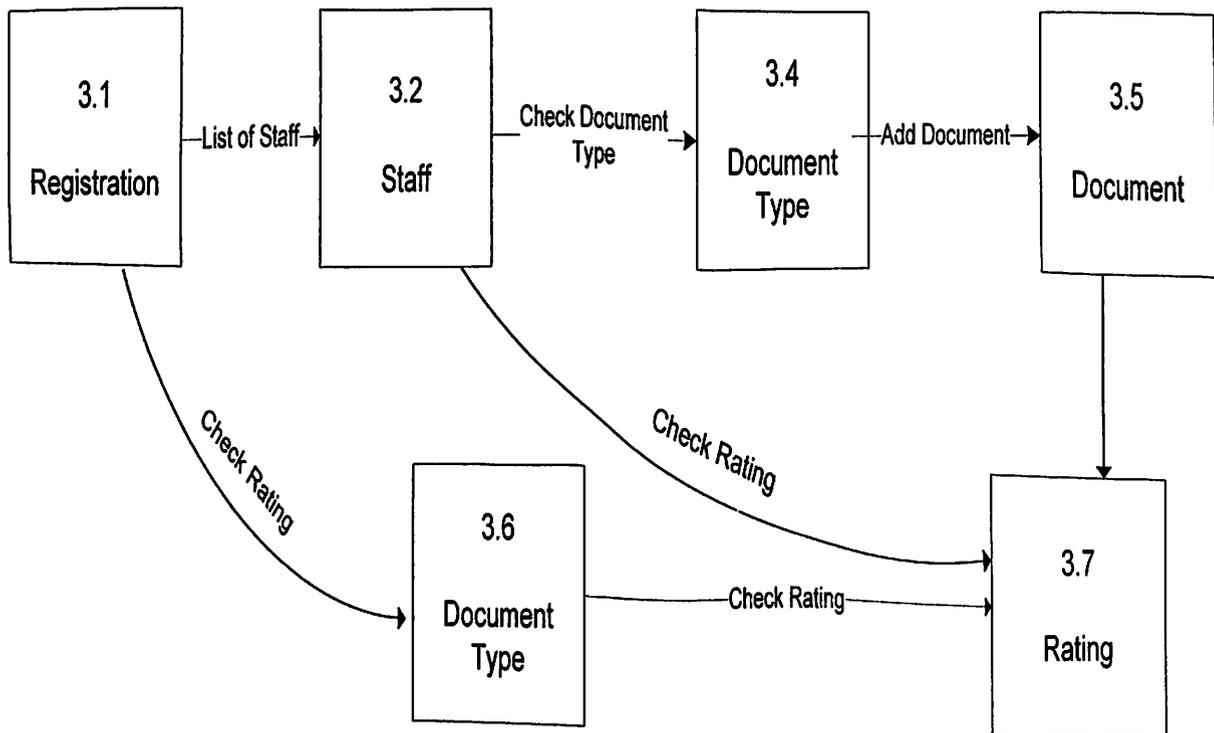


Fig: 5.6 - DFD for staff process

Knowledge Management System

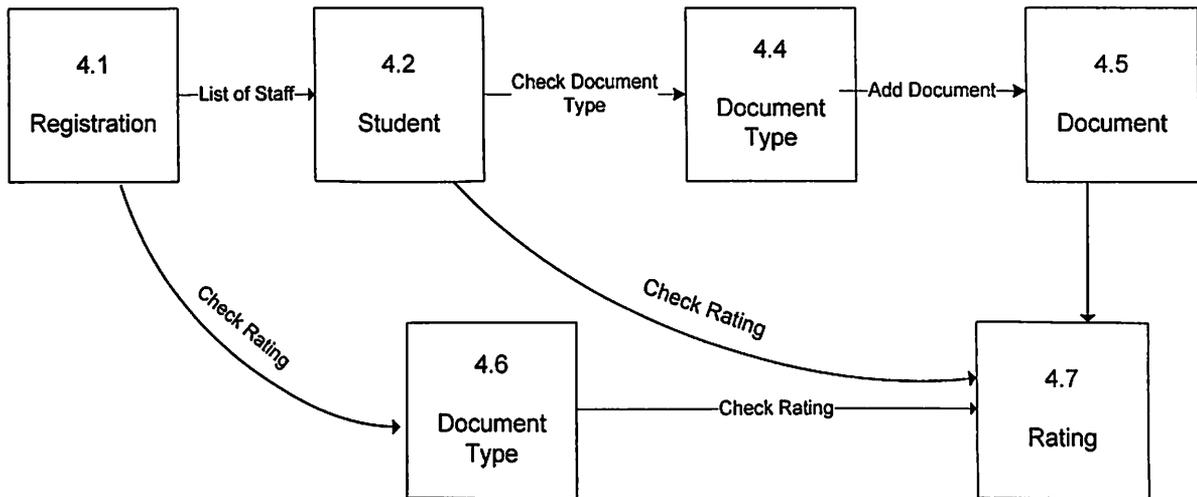


Fig: 5.7 - DFD for Student Process

5.1.2 User Module

A user should be able to register giving basic information. Login to the system and change the password (if required) after logging in. Post documents/information and edit/delete his contributions. Access/ Search documents/information posted by others (only read access). Search information by different parameters - rating/key words/relevance/category In this module User can add and update his own documents. User means either student or staff. To upload any document user has to be registered.

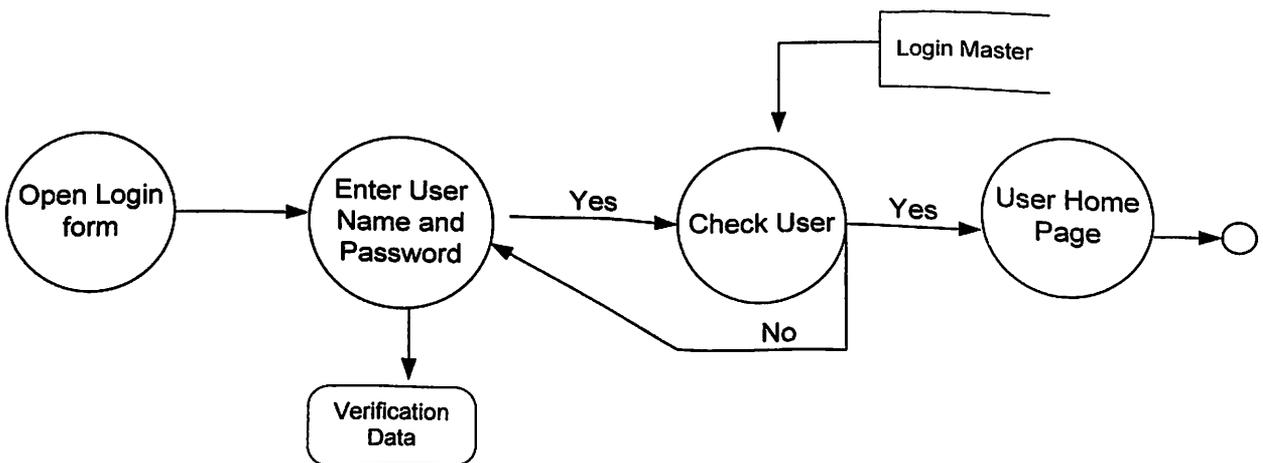


Fig: 5.8 - DFD for User module

5.1.3 Public User Module

In this module public user can search the documents and download any document that he wants without any authentication. He does not have any privileges to modify or delete any document.

5.1.4 Documents Module

In this module all the documents which are added by any user will be inserted.

Activity Diagram for Adding Doc Type

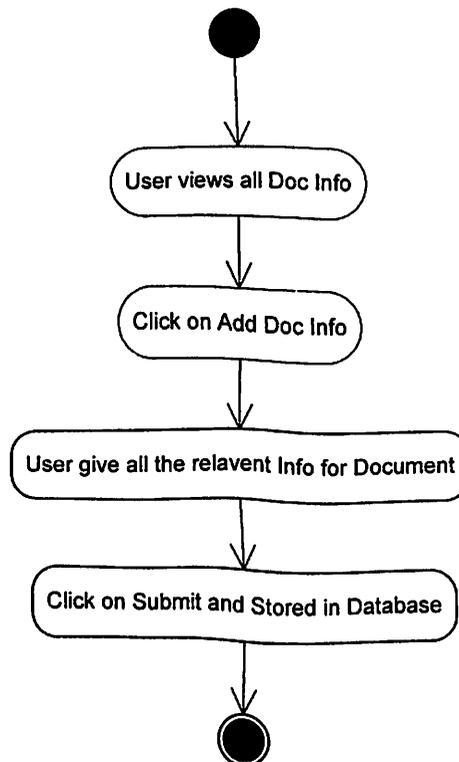


Fig: 5.9 – Activity Diagram for adding Doc type

Activity Diagram for Updating Doc Type

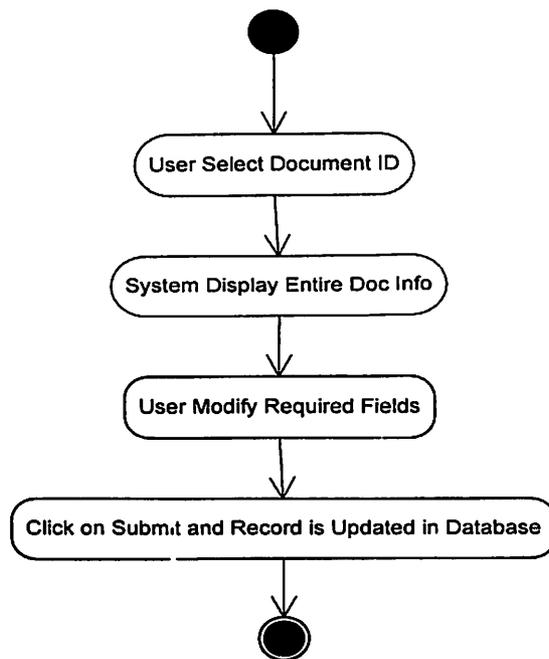


Fig: 5.10 – Activity Diagram for updating Doc type

Activity Diagram for Deleting Doc Type

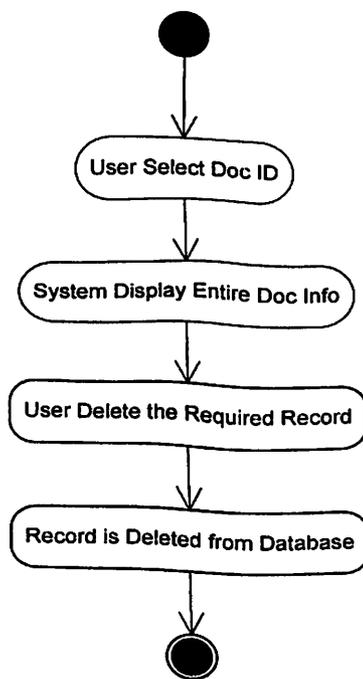


Fig: 5.11 – Activity Diagram for deleting Doc type

5.1.5 Report Module

The following reports can be generated. You can implement more reports which you think can be useful.

- List of members in the system and the number of contributions made by them.
- Number of Contributions by category and type

5.1.6 Authentication Module

This module contains all the information about the authenticated user. User without his username and password can't enter into the login, if he is only the authenticated user then he can enter to his login and he can see the his details and his document details.

CHAPTER 6

TESTING

6.1 Introduction

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

6.1.1 Strategic Approach to Software Testing

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

6.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

6.2.1 White Box Testing

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

6.2.2 Basic Path Testing

Established technique of flow graph with Cyclomatic complexity was used to derive test cases for all the functions. The main steps in deriving test cases were:

Use the design of the code and draw correspondent flow graph.

Determine the Cyclomatic complexity of resultant flow graph, using formula:

$$V(G)=E-N+2 \text{ or}$$

$$V(G)=P+1 \text{ or}$$

$$V(G)=\text{Number Of Regions}$$

Where $V(G)$ is Cyclomatic complexity,

E is the number of edges,

N is the number of flow graph nodes,

P is the number of predicate nodes.

Determine the basis of set of linearly independent paths.

6.2.3 Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

6.2.4 Data Flow Testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The *definition-use chain* method was used in this type of testing. These were particularly useful in nested statements.

6.2.5 Loop Testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.
- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated

CHAPTER 7

CONCLUSION

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in ASP.NET and VB.NET web based application and some extent Windows Application and SQL Server, but also about all handling procedure related with "Knowledge Management System". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

BENEFITS:

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

- It's a web-enabled project.
- This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
- The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
- Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
- User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
- From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can sat that the project is user friendly which is one of the primary concerns of any good project.
- Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.

Knowledge Management System

- Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time than manual system.
- Allocating of sample results becomes much faster because at a time the user can see the records of last years.
- Easier and faster data transfer through latest technology associated with the computer and communication.
- Through these features it will increase the efficiency, accuracy and transparency,

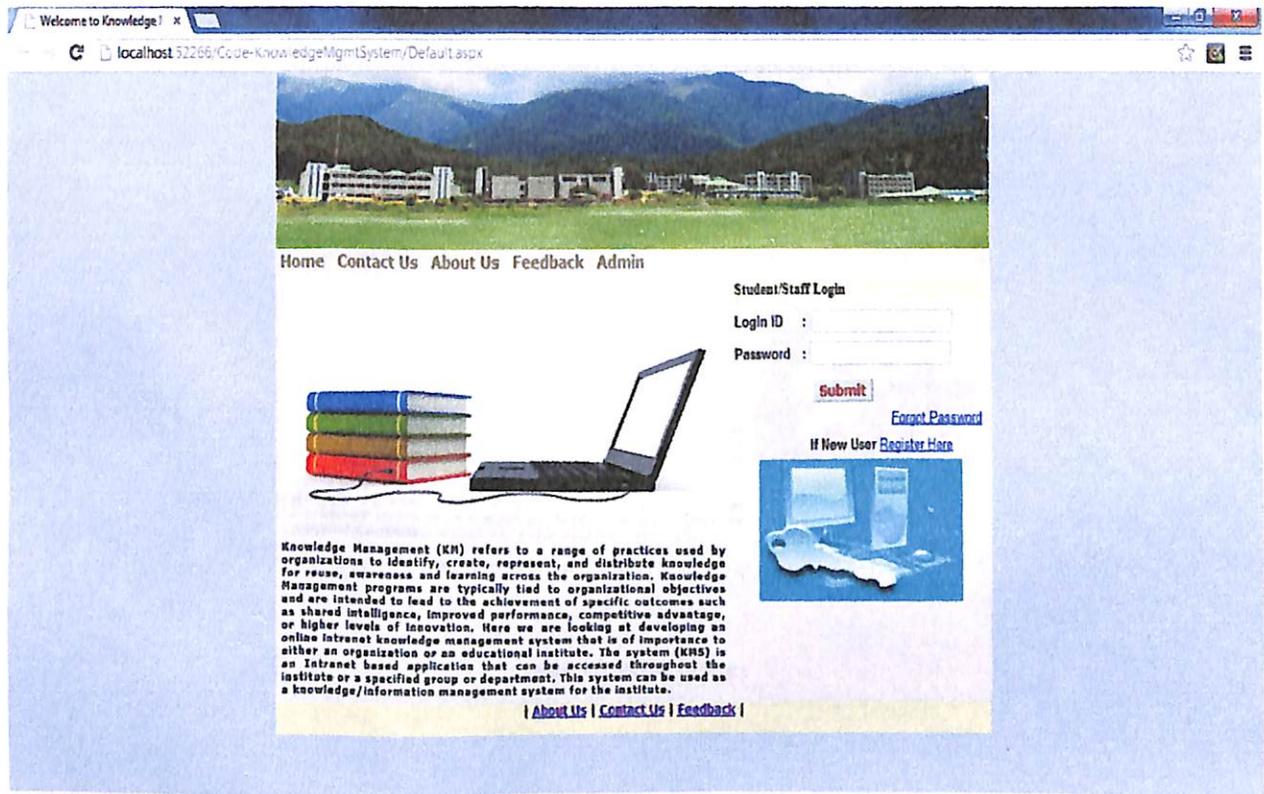
LIMITATIONS:

- The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
- Training for simple computer operations is necessary for the users working on the system.

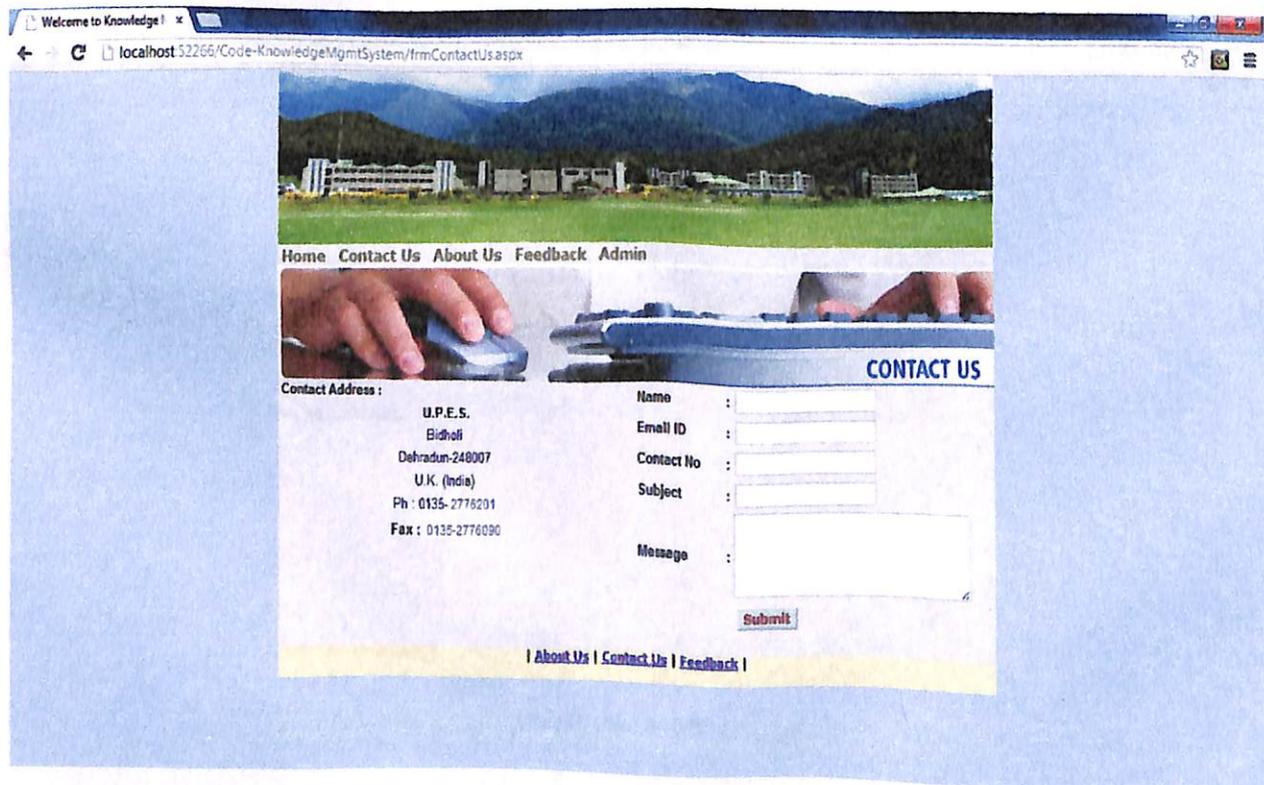
BIBLIOGRAPHY

1. **.NET Installation**, www.support.mircosoft.com
2. **Deployment and packing on server**, www.developer.com, www.15seconds.com
3. **SQL**, www.msdn.microsoft.com
4. **ASP.NET**, www.msdn.microsoft.com/net/quickstart/asplus/default.com

APPENDIX – SAMPLE SCREEN PRINTS

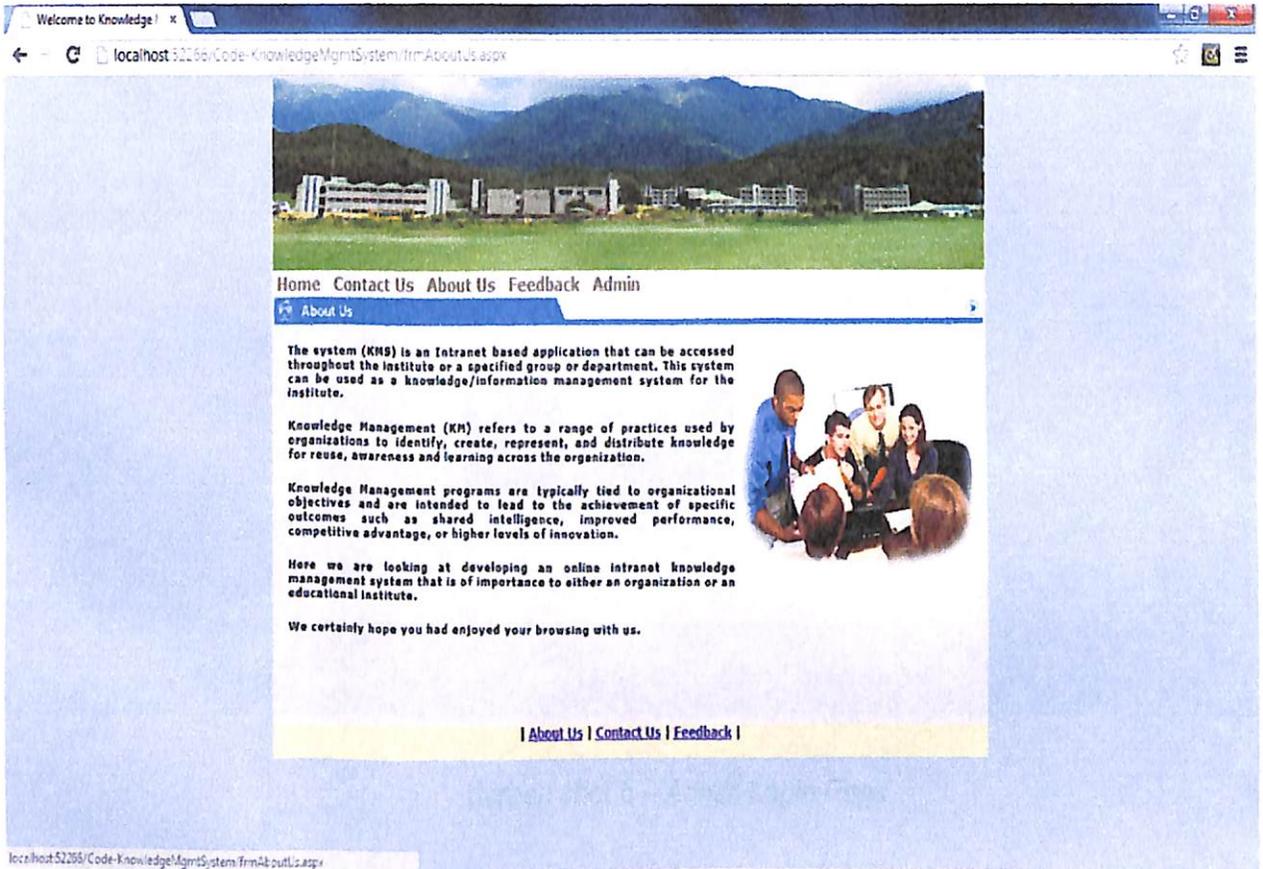


Screen shot 1 – Master Page

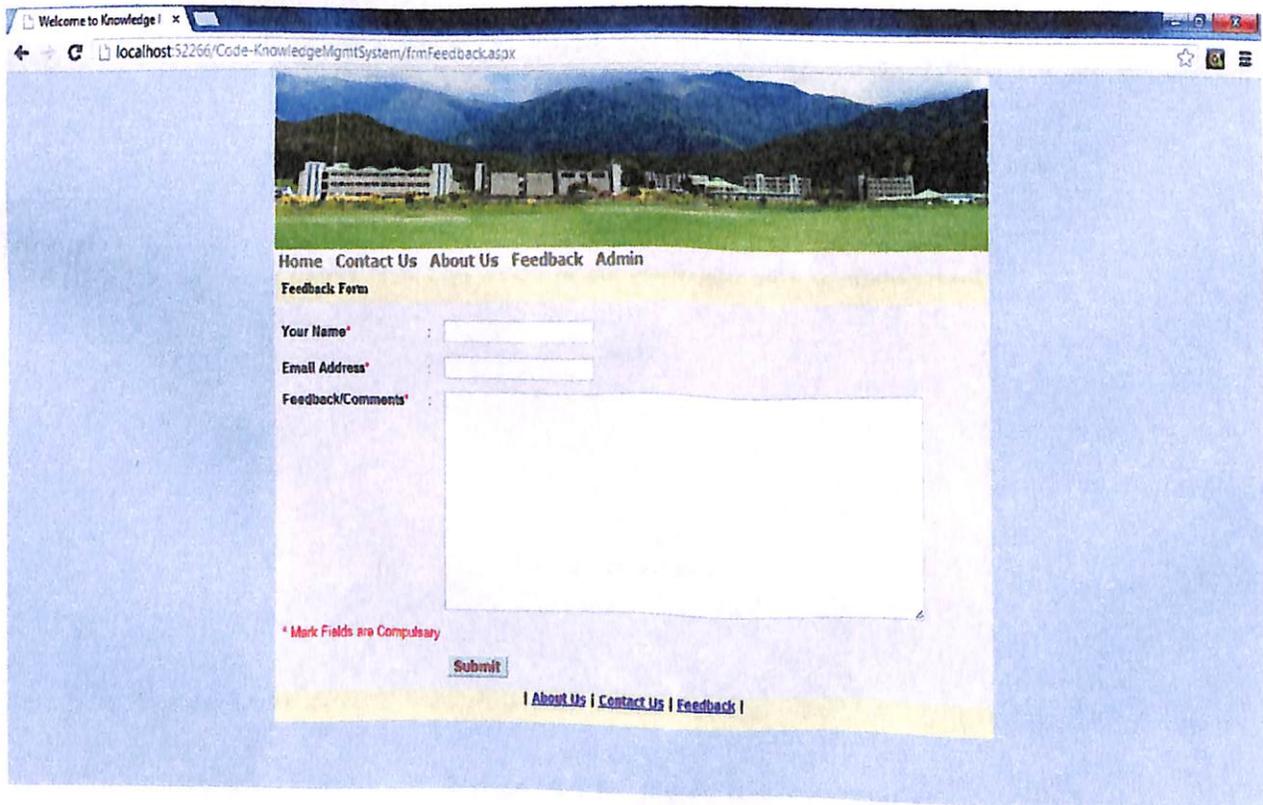


Screen shot 2 – Contact Us Page

Knowledge Management System

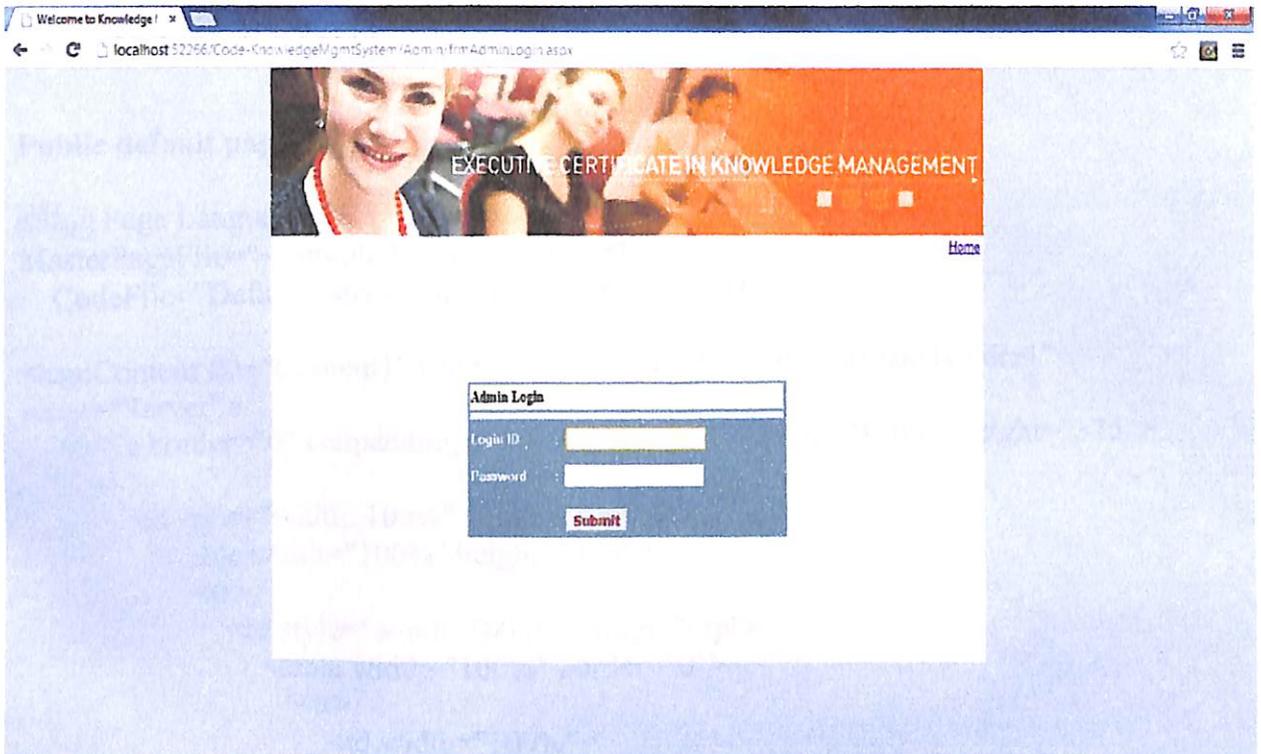


Screen shot 3–About Us Page

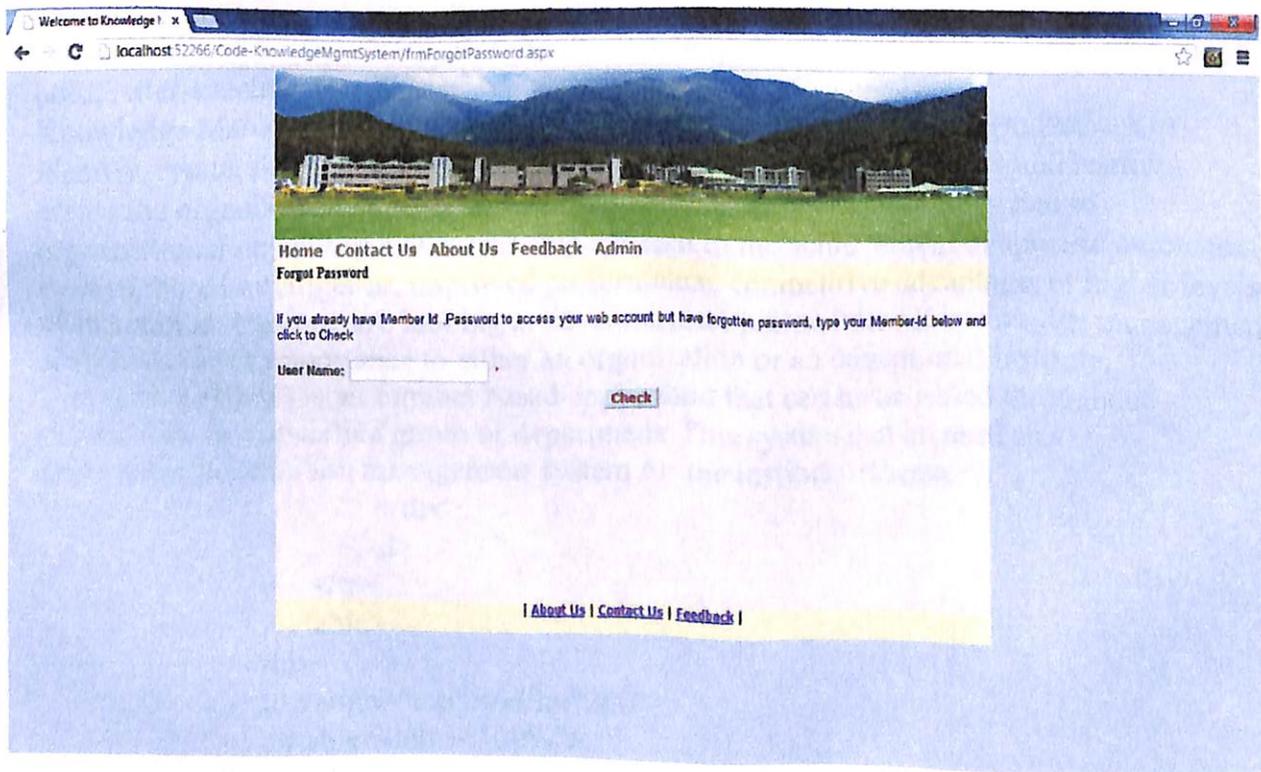


Screen shot 4 – FeedBack Form

Knowledge Management System



Screen shot 5 – Admin Login Page



Screen shot 6 – Forgot Password Page

SAMPLE CODE

Public default page.aspx

```

<%@ 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%">
                                        
                                    </td>
                                </tr>
                                <tr>
                                    <td align="justify">
                                        <div align="justify">
                                            <span style="font-family: Verdana; font-size: 10px; font-weight:
bold; 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>
                    </tr>
                </table>
            </td>
            <td valign="top" width="280">
                <table width="100%">
                    <tr>
                    </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>
</tr>
<tr>
    <td>
</td>
</tr>
<tr>
    <td align="left">
        <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click"
        CssClass="btnstyle" />

```


Public default page.cs

```
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 > 0)
            {
                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");
    }
}
```

Knowledge Management System

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

Knowledge Management System

Public master.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="SimplePublicMaster.maste .cs" Inherits="SimplePublicMaster" %>
<%@ Register Src="Controls/ucPublicFooterMenu.ascx" TagName="ucPublicFooterMenu"
TagPrefix="uc3" %>
<%@ Register Src="Admin/Controls/ucAdminFooter.ascx" TagName="ucAdminFooter"
TagPrefix="uc1" %>
<%@ Register Src="Controls/ucPublicHeaderMenu.ascx"
TagName="ucPublicHeaderMenu"
TagPrefix="uc2" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
<title>Welcome to Knowledge Management System</title>
<link href="App_Themes/Theme/black_world.css" rel="stylesheet" type="text/css" />
<script language="javascript" type="text/javascript">
```

```
function TABLE1_onclick()
```

```
{
}
```

```
function onlyNumbers(evt)
```

```
{
var e = event || evt;
var charCode = e.which || e.keyCode;
if ((charCode > 47 && charCode < 58))
return true;
else
return false ;
}
```

```
function OnlyChars(evt)
```

```
{
var e = event || evt;
var charCode = e.which || e.keyCode;
if ((charCode > 64 && charCode < 91)||charCode > 96 && charCode < 123) ||
charCode==32)
return true;
else
return false ;
}
```

```
function onlyNumbersdot(evt)
```

```
{
var e = event || evt;
var charCode = e.which || e.keyCode;
if (charCode > 47 && charCode < 58 || charCode==46)
return true;
```

```

else
return false ;
}
function onlyNumbersHifen(evt)
{
var e = event || evt;
var charCode = e.which || e.keyCode;
if ((charCode > 47 && charCode < 58 || charCode==45))
return true;
else
return false ;
}
function onlyNumbersUnderScore(evt)
{
var e = event || evt;
var charCode = e.which || e.keyCode;
if ((charCode > 47 && charCode < 58 || charCode==95))
return true;
else
return false ;
}

```

```

</script>
</head>
<body>
<form id="form1" runat="server">
<asp:ScriptManager ID="ScriptManager1" runat="server">
</asp:ScriptManager>
<div>
<table width="780" border="0" align="center" cellpadding="0" cellspacing="0"
class="bordermain" style="background-color:#F2EAF8">
<tr>
<td height="130" valign="top" style="background-image:url(..Images/bg.jpg);
background-repeat:repeat-x;">
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td><table border="0" cellspacing="0" cellpadding="0">
<tr>
<td >

</td>
<td></td>
</tr>
</tr>
</table></td>
</tr>
</table>
<uc2:ucPublicHeaderMenu ID="UcPublicHeaderMenu1" runat="server" />
<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
</asp:ContentPlaceHolder>
</td>

```

Knowledge Management System

```
</tr>
  <tr>
    <td align="center" bgcolor="#96c942" class="content" style="background-color:
#ffffcc;
    text-align: center">
      <uc3:ucPublicFooterMenu ID="UcPublicFooterMenu1" runat="server" />
    </td>
  </tr>
</tr>
<tr>
  <td bgcolor="#96C942" class="content" align="center" style="text-align:
center;background-color: #ffffcc;" >
    &nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</div>
</form>
</body></html>
```

Admin header.ascx

```

<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="ucAdminHeaderMenu.ascx.cs" Inherits="Admin_Controls_ucHeaderMenu" %>
<table border="0" cellpadding="0" cellspacing="0" width="100%" align="center" >
<tr><td align="center">
    <asp:Menu ID="Menu1" runat="server" BackColor="#C4CCCF"
DynamicHorizontalOffset="2" Font-Names="Verdana" Font-Size="10pt"
ForeColor="#284E98" Orientation="Horizontal" StaticSubMenuIndent="10px"
Width="100%" Font-Bold="True" StaticEnableDefaultPopOutImage="False">
    <StaticMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
    <DynamicHoverStyle BackColor="#284E98" ForeColor="White" />
    <DynamicMenuStyle BackColor="#B5C7DE" />
    <StaticSelectedStyle BackColor="#507CD1" />
    <DynamicSelectedStyle BackColor="#507CD1" />
    <DynamicMenuItemStyle HorizontalPadding="5px" VerticalPadding="2px" />
    <Items>
        <asp:MenuItem NavigateUrl="~/Default.aspx" Text="Home"
Value="Home"></asp:MenuItem>
        <asp:MenuItem Text="Administration" Value="Administration">
            <asp:MenuItem Text="Change Password" Value="Change Password"
NavigateUrl="~/frmChangePassword.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage Document Category" Value="Manage Document
Category" NavigateUrl="~/Admin/frmManageCategory.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage User Type" Value="Manage User Type"
NavigateUrl="~/Admin/frmManageUserType.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage Country" Value="Manage Country"
NavigateUrl="~/frmManageCountry.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage State" Value="Manage State"
NavigateUrl="~/frmManageState.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage District" Value="Manage City"
NavigateUrl="~/frmManageDistrict.aspx"></asp:MenuItem>
            <asp:MenuItem Text="Manage Status" Value="Manage Status"
NavigateUrl="~/frmManageStatus.aspx"></asp:MenuItem>
            </asp:MenuItem>
            <asp:MenuItem Text="Registered User" Value="Manage User">
                <asp:MenuItem Text="Manage Registered User" Value="Manage User"
NavigateUrl="~/Admin/frmManageUser.aspx"></asp:MenuItem>
            </asp:MenuItem>
            <asp:MenuItem Text="Documents" Value="Documents">
                <asp:MenuItem Text="Manage Documents" Value="Manage Documents"
NavigateUrl="~/Admin/frmManageDocument.aspx"></asp:MenuItem>
            </asp:MenuItem>
            <asp:MenuItem Text="Rating/Comments" Value="Rating/Comments">
                <asp:MenuItem Text="Manage Rating/Comments" Value="Manage
Rating/Comments"
NavigateUrl="~/Admin/frmViewRatingComments.aspx"></asp:MenuItem>
            </asp:MenuItem>
            <asp:MenuItem Text="Reports" Value="Reports">

```

Knowledge Management System

```
<asp:MenuItem Text="Member Report" Value="Member Report"
NavigateUrl="~/Admin/frmMemberReport.aspx"></asp:MenuItem>
  <asp:MenuItem Text="Document Report" Value="Document Report"
NavigateUrl="~/Admin/frmDocumentsReport.aspx"></asp:MenuItem>

</asp:MenuItem>
  <asp:MenuItem Text="Logout" Value="Logout"
NavigateUrl="~/Admin/frmSingOut.aspx"></asp:MenuItem>
</Items>
  <StaticHoverStyle BackColor="#284E98" ForeColor="White" />
</asp:Menu>
</td></tr>
</table>
```

Admin_master_menu.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="AdminMasterMenu.master.cs" Inherits="Admin_AdminMasterMenu" %>
<%@ Register Src="Controls/ucAdminFooter.ascx" TagName="ucAdminFooter"
TagPrefix="uc2" %>
<%@ Register Src="Controls/ucAdminHeaderMenu.ascx"
TagName="ucAdminHeaderMenu" TagPrefix="uc1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Welcome to Knowledge Management System</title>
  <link href="../App_Themes/Theme/black_world.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" language="javascript">

function toggleAllCheckboexs(toggle)
{
    n = document.forms[0].length;
    var frm = document.forms[0];
    for(i=0;i<frm.length;i++)

        if(frm.elements[i].type=="checkbox")
            if (frm.elements[i].name.indexOf('Cbx')==0)
                frm.elements[i].checked=toggle;
    }

function onlyNumbers(evt)
{
    var e = event || evt;
    var charCode = e.which || e.keyCode;
    if ((charCode > 47 && charCode < 58))
        return true;
    else
        return false ;
}
function OnlyChars(evt)
{
    var e = event || evt;
    var charCode = e.which || e.keyCode;
    if ((charCode > 64 && charCode < 91)|| (charCode > 96 && charCode < 123) ||
charCode==32)
        return true;
    else
        return false ;
}
function onlyNumbersdot(evt)
```

Knowledge Management System

```
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if (charCode > 47 && charCode < 58 || charCode==46)
  return true;
  else
  return false ;
}
function onlyNumbersshifen(evt)
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if ((charCode > 47 && charCode < 58 || charCode==45))
  return true;
  else
  return false ;
}
function onlyNumbersUnderScore(evt)
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if ((charCode > 47 && charCode < 58 || charCode==95))
  return true;
  else
  return false ;
}
}

</script>

</head>
<body>
  <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager1" runat="server">
    </asp:ScriptManager>
  </div>
  <table width="780" height="380px" border="0" align="center" cellpadding="0"
  cellspacing="0" class="bordermain" style="background-color:#F2EAFA" >
  <tr>
  <td valign="top">
  <table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr>
  <td valign="top">
  
  </td>
  </tr>
  <tr>
  <td>
  <uc1:ucAdminHeaderMenu ID="UcAdminHeaderMenu1" runat="server" />
  </td>
  </tr>
  </tr>
  </table>
  </td>
  </tr>
  </table>
  </body>
  </html>
```

Knowledge Management System

```
</table></td>
</tr>
<tr>
  <td class="content">
    <asp:contentplaceholder id="ContentPlaceHolder1" runat="server">
      </asp:contentplaceholder>
    </td>
  </tr>
</table>
</div>
</form>
</body>
</html>
```

Knowledge Management System

User_master_page.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="UserMasterPage.master.cs" Inherits="Users_UserMasterPage" %>
<%@ Register Src="~/Controls/FooteMenuForUser.ascx" TagName="FooteMenuForUser"
TagPrefix="uc3" %>

<%@ Register Src="~/Admin/Controls/ucAdminFooter.ascx" TagName="ucAdminFooter"
TagPrefix="uc1" %>
<%@ Register Src="~/Controls/ucPublicHeaderMenu.ascx"
TagName="ucPublicHeaderMenu"
TagPrefix="uc2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" >
<head id="Head1" runat="server">
  <title>Welcome to Knowledge Management System</title>
  <link href="~/App_Themes/Theme/black_world.css" rel="stylesheet" type="text/css" />
  <script language="javascript" type="text/javascript">

function onlyNumbers(evt)
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if ((charCode > 47 && charCode < 58))
    return true;
  else
    return false ;
}
function OnlyChars(evt)
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if ((charCode > 64 && charCode < 91)|| (charCode > 96 && charCode < 123) ||
charCode==32)
    return true;
  else
    return false ;
}
function onlyNumbersdot(evt)
{
  var e = event || evt;
  var charCode = e.which || e.keyCode;
  if (charCode > 47 && charCode < 58 || charCode==46)
    return true;
  else
    return false ;
}
}
```

Knowledge Management System

```
function onlyNumbersshifen(evt)
{
    var e = event || evt;
    var charCode = e.which || e.keyCode;
    if ((charCode > 47 && charCode < 58 || charCode==45))
        return true;
    else
        return false ;
}
function onlyNumbersUnderScore(evt)
{
    var e = event || evt;
    var charCode = e.which || e.keyCode;
    if ((charCode > 47 && charCode < 58 || charCode==95))
        return true;
    else
        return false ;
}
</script>
</head>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
        <div>
            <table width="780" border="1" align="center" cellpadding="0" cellspacing="0"
class="bordermain" style="background-color:#F2EAFA">
                <tr>
                    <td height="130" valign="top" style="background-image:url(..Images/bg.jpg);
background-repeat:repeat-x;">
                        <table width="100%" border="0" cellspacing="0" cellpadding="0">
                            <tr>
                                <td><table border="0" cellspacing="0" cellpadding="0">
                                    <tr>
                                        <td >
                                            
                                        </td>
                                        <td></td>
                                    </tr>
                                </tr>
                            </table></td>
                        </tr>
                    </table>
                </tr>
            </table>
            <uc2:ucPublicHeaderMenu ID="UcPublicHeaderMenu1" runat="server" />
            <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

                </asp:ContentPlaceHolder>
            </td>
        </tr>
    </tr>
</body>
```

Knowledge Management System

```
<td align="center" bgcolor="#96c942" class="content" style="background-color:
#ffffcc;
text-align: center">
<uc3:FooteMenuForUser ID="FooteMenuForUser1" runat="server" />
&nbsp;</td>
</tr>
<tr>
<td bgcolor="#96C942" class="content" align="center" style="text-align:
center;background-color: #ffffcc;" >
<uc1:ucAdminFooter ID="UcAdminFooter1" runat="server" />

</td>
</tr>
</table>
</div>
</form>
</body>
</html>
```