

# A Portable Content Management System

Srijita Mukherjee, Tarif Mohammad

Supreme Knowledge Foundation Group of Institutions, Academy of Technology ESL Lab, Chandannagar

---

**Abstract:** This report describes the design, implementation and usage of an efficient Content Management System. The aim of content management systems is to store and organize files, and provide version-controlled and user-controlled access to data. This project also includes Web-based publishing, search and retrieval of scientific data. In this project, every user can upload and view his/her own uploaded file. The uploaded files are organized and stored in the specific folders. They can send messages to each other. The sender's name along with all the previous messages is available with the new messages. A public area "Forum" is provided for all the users where they can choose a topic or subject they like to and view the related posts and can post their own queries.

---

## I. Introduction

A Content Management System (CMS) is computer program or a system administrator tool that allows publishing, editing and modifying content as well as maintenance from a central interface. Such systems of content management provide procedures to manage work flow in a collaborative environment. The core function and use of content management systems is to store and organize files, and provide control to access their data.

The content management system (CMS) has two elements:

- Content management application (CMA) is the front-end user interface that allows a user to add, modify and remove content from a system.
- Content delivery application (CDA) compiles that information and updates the system.

Content Management System serves as a central interface containing documents, movies, pictures, personal details and scientific data. CMS can be used for storing, controlling, modifying and publishing documentation. This content management application is simple to use and understandable to your individual needs. No programming knowledge is required to manage and control your content. With the ability to define multiple users and roles, we can control the privileges assigned to our users, allowing them, a complete control to their individual contents. It provides full management of your comprehensive document, image and media libraries including messages and organize, store and display all of your contents with ease.

## II. Motivation

The emerging data or information along with the advancement of technology has caused a need to have an efficient content management system. The CMS which are available in the market lacks features like organizing files based on their type and they don't deal with the permissions of the owners.

We aimed to develop a CMS which will have an efficient file organizing mechanism as well as features like forum, messaging to other users etc. We hope that this system will aid the users and the design concept described in this paper will help the developers' community in near future.

## III. Methodology

### Step Algorithm:

1. Ask the user to choose one of the following options:
  - Login
  - Register
2. When an existing user logs in he/she views the Home Page where he/she gets 4 options-
  - Upload
  - View
  - Message
  - Forum.
  - Logout
3. New user can register and start accessing
4. From Home Page, if the user chooses
  - 'Upload' then he/she can upload new files
  - 'View' option lets the user view only the files that particular user has uploaded

- ‘Message’ button has two options-
    - View Messages
    - Sent Messages
  - ‘Forum’ button too has two options-
    - Topic I
    - Topic II
5. Message button will lead the user to-  
View Message option-which will let the user view all the new as well as older messages from different users.  
Sent Message option-which executes when proper user-id of the receiver is given along with the message.
  6. In Topic I in Forum button, the user can view the details of that topic and can post queries  
Similar is for button Topic II.
  7. The logout button takes the user back to the first page(Login Page).

**Explaining the Message button-**

The main feature of messaging is to send messages to another user and view messages from other user. The user have to provide the receiver’s id. The application will append the receiver’s id and message to the sender’s dictionary. To view messages, user will select the view message option in which the entire list of the viewer will be shown. The dictionary stores the sender’s id first then the receiver’s id. To every receiver-id we have a list where the messages are appended. The dictionary actually looks like:  
{‘Sender’s-id’: { ‘receiver1’s-id’: [messages] , {receiver2’s-id : [messages]}}

**Explaining the Upload button-**

This button actually upload new files into the applications. The uploaded files are hidden from other users. When the user clicks on the upload button, a window pops-out which contains the files which are to be upload. We have a separate dictionary which contains the path of the uploaded file and the id of the user who uploads the file. The dictionary holds value in the following manner:  
{ ‘user1-id’ : [file1’s path, file2’s path]} , {‘user2’s-id’ : [file1’s path, file2’s path]}

**Explaining the View button-**

The View button, when clicked gives all the path of the uploaded files that the logged-in user has uploaded till date. When a user logs in, its id get stored into a variable that is matched with the user name present in the dictionary and thus all the uploaded files path are displayed.

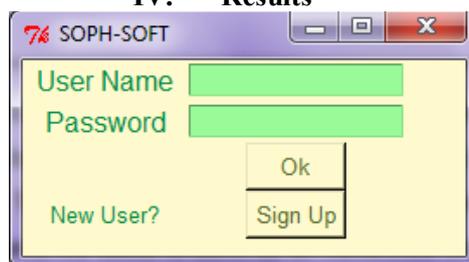
**Explaining the Forum button-**

In the forum, we have presented several topics in the form of buttons. Once the user selects a particular topic by clicking on the button with the required topic name, all the relevant posts on that particular topic are shown. We can also post queries or give answers to the queries of other user. We have a separates dictionaries for different topics that stores the posts of every individual user. The values in the dictionary are stored in the following manner:  
{ ‘user-id1’ : [‘post1’,’post2’],’user-id2’: [‘post1’,’post2’,’post3’]}

**Explaining the Logoutbutton-**

This button takes the user to the first page i.e. the Login Page.

**IV. Results**



**Figure 1: Screenshot of Login Page**



Figure 2: Screenshot of Home Page



Figure 3: Screenshot of View Button from Home Page

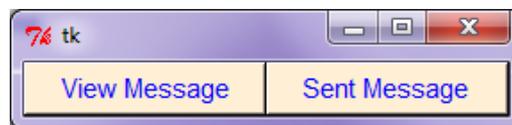


Figure 4: Screenshot of Message Button



Figure 5: Screenshot of Send Message Button

## V. Conclusion & Future Work

The implementation and design concept described in the methodology section of the report. The screenshots are showing the output of the system. The system works as per our expectation and produces the expected outcome. We hope that this system will be of great useful to the users and the developers' community will use the implementation ideas in near future.

The system lacks the security features. We will import the security features in order to make it more secure from the outside attacks.