**Major Project for**

**6th Semester**

**Done by: SOURAV DEY**

**University Registration No: 121251010123**

**University Roll No.: 12501212050**

**College Roll No.: 12856**

**Department: Bachelor of Computer Application**

**College: Bengal College of Engineering and Technology, Durgapur**

**Year: 2015**

ONLINE DRIVE AND FILE MANAGEMENT SYSTEM

**[Project Title: CloudDrive]**

*ONLINE DRIVE AND FILE MANAGEMENT SYSTEM*

SELF-CERTIFICATE

This is to certify that the project report entitled **“ONLINE DRIVE AND FILE MANAGEMENT SYSTEM”** is done by me is an authenticate work carried out for partial fulfillment of the requirements for the award of the training of **ASP.NET using C#** under the guidance of **Miss. Mithu Ghosh** faculty member of **CMC, Durgapur**. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

--------------------------------------------

*Signature of the student*

*Sourav Dey*

*CMC, Durgapur*

Bengal College of Engineering and Technology

BIDHANNAGAR, DURGAPUR-12



*CERTIFICATE*

Certified that project work entitled “Online Drive And File Management System**”** is a confide work carried out by **Sourav Dey**

(Roll No: 12501212050) in partial fulfilment for the award of the degree of Bachelor of Computer Application from Bengal College of Engineering and Technology during the academic year 2012 – 2015.

-----------------------------------------

Praveen Kumar Singh

(HOD-Computer Application)

BCET

*ONLINE DRIVE AND FILE MANAGEMENT SYSTEM*

*ACKOWLEDGEMENT*

***“Coming together is beginning, keeping together is success”.***

The experience that we have gathered during the project has been unique and overwhelming. For this we are pleased to express our deepest sense gratitude and regards to our respected project guide Miss. Mithu Ghosh(faculty of CMC Durgapur) for consistence guidance and encouragement during developing the project and providing us with resources as and when required and allowing us to pursue the project in “Microsoft Visual Studio 2012 & MYSQL (As Database)”.

In spite of busy schedules, her frequent valuable advice and master suggestions at every stage lead to systematic approach in completing this project without which this project work could not take its shape.

I also want to thanks my partners for their co-operation and help.

Lastly, with all our heart, we express our deep gratitude to our parents.

***Sourav Dey***

*ONLINE DRIVE AND FILE*

*MANAGEMENT SYSTEM*

CloudDrive

* INDEX: PAGE NO

1. Introduction

* About ASP.NET 6

1. Objectives And Features of our projects 7
2. Project category 8
3. Tools/Platform

* Hardware configuration 8
* Software configuration 8

1. Problem Identification And Specification Requirement

* Problem definition 9
* System Requirement Specification 9

1. Project Planning And Scheduling

* Project Planning 10
* Project Estimates 11
* Effort Estimation Graph 11
* PERT chart/GANTT chart 12

1. System Analysis And Design

* Process logic of each module 13
* Data Flow Diagram 14,15,16
* Entity Relationship Diagram 17

1. Table Design 18
2. User Interface 19, 27
3. Coding 28-29
4. Conclusion & Future Enhancements 30
5. Bibliography

* Reference Books 31
* Websites 31
* Introduction:

1. What is ASP.Net?

* ASP.NET is an open source Server-side Web Application framework designed for Web development to produce dynamic Web pages.
* It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.
* ASP.NET is built on the Common Language Runtime(CLR), allowing programmers to write ASP.NET code using any supported .NET language.
* ASP.NET is in the process of being re-implemented as a modern and modular web framework, together with other frameworks like Entity Framework.
* ASP.NET Web pages, known officially as Web Forms, are the main building blocks for application development. Web forms are contained in files with a ".aspx" extension; these files typically contain static (X)HTML mark-up, as well as mark-up defining server-side Web Controls.
* In general, the ASP.NET directory structure can be determined by the developer's preferences.
  + App\_Code : This is the "raw code" directory.
  + App\_Data: The App\_Data ASP.NET Directory is the default directory for any database used by the ASP.NET Website.
  + App\_GlobalResources: Holds resx files with localized resources available to every page of the site.
  + App\_Themes: Adds a folder that holds files related to themes which is a new ASP.NET feature that helps ensure a consistent appearance throughout a Web site and makes it easier to change the Web site’s appearance when necessary.
  + App\_Web References: holds discovery files and WSDL files for references to Web services to be consumed in the site.
  + Bin: Contains compiled code (.dll files) for controls, components, or other code that you want to reference in your application. Any classes represented by code in the Bin folder are automatically referenced in your application.
* Objectives And Features Of Our Project:

1. Objectives:
2. Capable of handling target DATABASE mapped to the physical server files and directories in an efficient manner.
3. Capable of carrying online searching, downloading, viewing, uploading and managing data and files in an efficient way.
4. Gives privacy to sensitive user information from exposing.
5. Provides various application-level and protocol-layer based security to user data and files.
6. Admin can control all the activities going on in the website.
7. Admin can control all the data provided into the website along with the files uploaded.
8. Features:
9. This project has been hosted online to try out the working and live version of the services promised to be delivered by this site. Web address of the online-hosted version is **www.OSCorpCloudDrive.somee.com**.
10. SSL security layer and application-level security for secure file upload and download and protection of user information against unauthorized access and hacking.
11. Uploaded files gets stored in Server database in encrypted format, which provides security against misuse even if it falls in wrong hand, and in compressed form, which saves consumption of physical disk space of server. All this techniques are completely transparent to the user.
12. Various controls against unauthorized access are implemented and no unauthorized user can modify or even view other users' information or can gain admin privileges without a valid user/admin session.
13. Enhanced searching technique for quicker response of the search results.
14. Even unregistered users can search, view and download files category-wise.
15. Users can opt for an easy and free process of signing up to gain additional privileges from the websites and can manage their personal information using various tools provided.
16. In the signing up process maximum automation in case of data inputs are provided and error correction measures are taken to reduce the chances of wrong data entry.
17. Auto-logging out from user account after a certain period of inactivity is detected for security reasons.
18. Registered users can additionally upload, delete and share files category-wise and can also upload in a secure mode to have only private access to them.
19. While sharing files users can control their visibility and also can make use of the auto completion and suggestion facility to find other users in the website.
20. In case of reception of file sharing requests from others, user can either confirm or decline their requests and can also view the public information of the user sending the request.
21. Notification system for the users for each new unnoticed events occurred.
22. Easy retrieval process of passwords in case users' have forgotten theirs.
23. Site's Admin can check for the latest users signed up and latest files uploaded since last Admin logout.
24. Admin can also view all registered users' details (but not their sensitive information) and can send each of them messages in the form of notifications or even delete their account permanently.
25. Admin can search, view or download files irrespective of their categories.
26. Admin can also change his/her login credentials.
27. Measures are taken against uploading duplicate files by users and sharing files which have been already shared to another user.
28. A progress bar denotes the percentage and amount of space utilised and the remaining space required.

* Project Category:

RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

RDBMS approach is used for developing this project because scope of data redundancy.

**Moreover there are various advantages of using a RDBMS approach which are as follows:**

* Better technique for data storage.
* Less data retrieval time.
* Better space management.
* Scope of creating logical relationships among the entities.
* Application over a network such as intranet.
* No need to distribute and software on all the client computer accessing the system.
* Tools/Platform, Hardware and Software requirement Specification:

1. Minimum hardware configuration required to build the project:

|  |  |
| --- | --- |
| Hardware Parts | Configuration |
| 1. Motherboard | Intel / ASUS |
| 1. Processor | Intel Pentium Dual Core |
| 1. Hard Disk Drive | 10GB |
| 1. RAM | 1GB |
| 1. Monitor | 15.6” Color |
| 6. Keyboard | Multimedia |
| 1. Mouse | Optical mouse |

2. Software configuration:

* + - Front-end Tool:
* ASP.NET 4.5 (Active server page)
* CSS (Cascading Style Sheets)
* JQUERY
* AJAX Controls
* HTML
* JavaScript
  + - Back-end Tool:
      * + C# Language
        + Microsoft SQL Server 2012
    - Operating system:
* Microsoft Windows 7
* Problem Definition And Specification Requirement:

1. Problem Definition:

The first problem is that there are loads of hard copied document being generated. This brings us to the age-old discussion of keeping information in the form of database versus keeping the same on sheets of paper. Keeping the information in the form of hard-copied documents leads to the following problems:

* LACK OF SPACE: It becomes a problem in itself to find space to keep the sheets of paper being generated as a result of the ongoing discussion. The documents being generated are too important to be ill-treated.
* FILINGPOSES A PROBLEM: Filing the documents. Categorically is a time consuming and tedious exercise. Result processing is slow due to paper work and requirement of staff.

2. System Requirement Specification:

1. Functional system requirement:

This section gives a functional requirement that is applicable to the Online drive and file management system.

There are two sub modules in this phase:

* USER MODULE.
* ADMINISTRATOR MODULE.

The features that are available to the User module are:

* Can create account.
* Can login in his/her account.
* Can upload files, documents, and can view own uploads.
* Can delete own uploaded files and document?
* Can search any type of file and download directly.
* Can keep files in secure achieve mode for privacy.
* Can change user information and password, change or remove profile picture.

The features that are available to the Administrator module are:

* Can delete, view user account.
* Can search files without any category.
* Can check how many user have logging, and also check user activity.

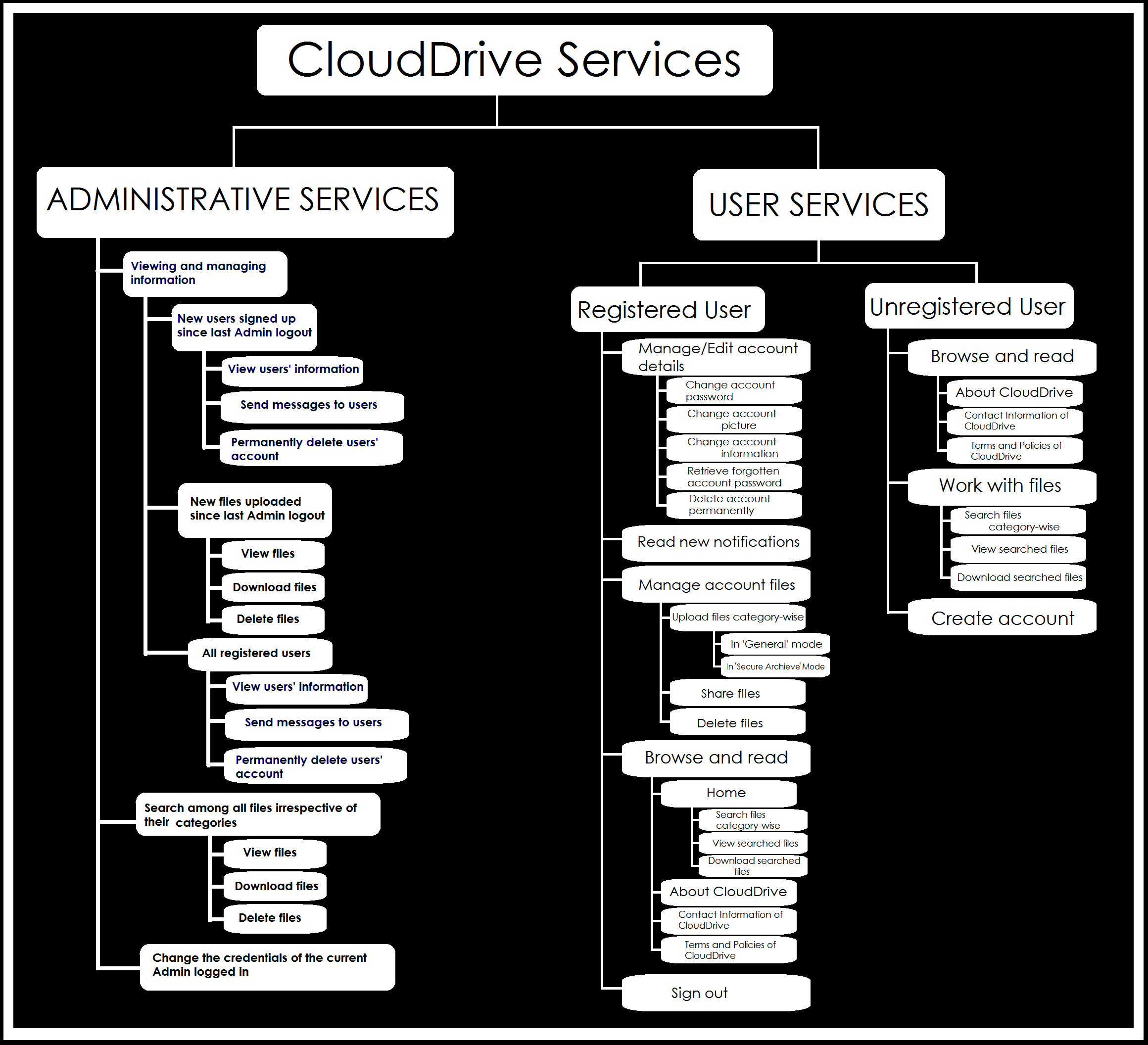
1. Non-Functional System Requirement:

* PERFORMANCE REQUIREMENTS:
* The software shall support use of multiple users at a time.
* There are no other specfic performance requirements that will affect development.
* SAFETY REQUIREMENTS:
* The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup.
* SECUIRTY REQUIREMENTS:
* Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction or disclosure.
* PROJECT PLANNING AND SCHEDULING:
* Project Planning:

The software project plan is produced at the culmination of the planning tasks. It provides the baseline cost and scheduling information that will be used throughout the software engineering process. The software project plan is a relatively brief document that is addressed to a diverse audience. It must:

* Communicate scope and resources to software management, technical staff and the customer.
* Define risks and suggest risk management techniques.
* Define cost and schedule for management review.
* Provide an overall approach to software development for all people associated with the project.
* Outline how quality will be ensured and changed will be managed.

Functional Decomposition Diagram:



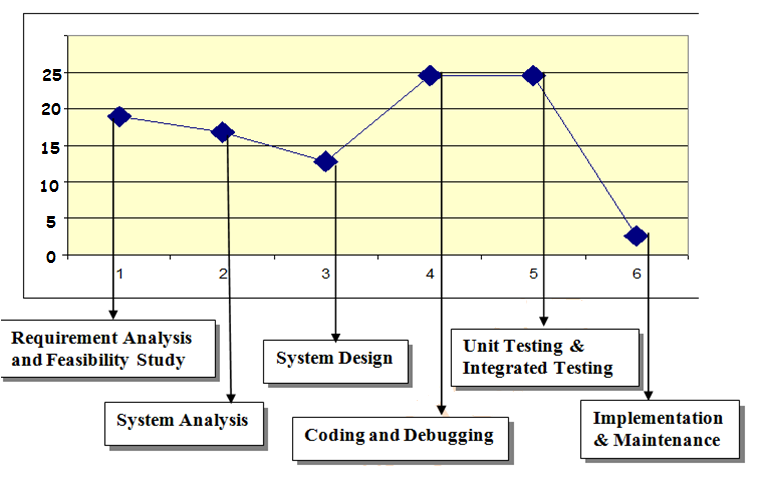
* Project Estimates:

ESTIMATION OF EFFORT (FOR THE PROJECT DEVELPOMENT)

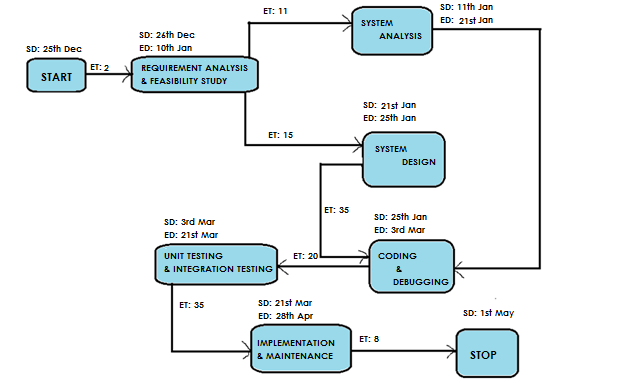
* No. of persons working to develop the project: 4 persons
* Time period estimated: 1month (6 hrs each day)
* Phases to cover: 80%

The various phases are to be covered under this project are:

* Requirement Analysis and Feasibility Study
* System Analysis
* System Design
* Coding and Debugging
* Unit Testing and integration Testing
* Implementation and Maintenance
* Effort Estimation Graph:



* PERT Chart:



Note:

SD- Start date.

ED- End date.

ET- Estimated time/Duration in days.

* GANTT Chart:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Task | 5 Days | 5 Days | 5 Days | 5 Days | 5 Days | 5 Days |
| RAF |  |  |  |  |  |  |
| SA |  |  |  |  |  |  |
| SD |  |  |  |  |  |  |
| CD |  |  |  |  |  |  |
| UI |  |  |  |  |  |  |
| IM |  |  |  |  |  |  |

Note:

* RAF: Requirement Analysis and Feasibility Study.
* SA: System Analysis.
* SD: System Design.
* CD: Coding and Debugging.
* UI: Unit testing and integration testing.
* IM: Implementation and Maintenance.
* SYSTEM ANALYSIS AND DESIGN:

Process Logic of each module:

* Signing up process:

Signing up is not mandatory to use certain basic and limited services offered by the site. But in order to avail the full services offered the user have to sign up in the prescribed format which also includes the agreement with the user policies of the site. After signing up user can upload his/her files to the sites database and can control their visibilities and can even share them with other users present in the site.

* Signing in process:

Signing in or logging in is not applicable if the user have previously signed up into the website and have a valid account. Both user and the site’s Admin have to sign in to the website to view their account. Signing in unlocks all the information controlling features provided by the site.

* Working with files:

Both registered and unregistered user can search, view and download the files which are publicly uploaded by the registered users. However registered users can additionally upload, share, control visibility and can even delete their files.

* Receiving notifications

Users are notified about new events by sending notifications to their accounts. The new and unread notifications are kept at the top and are notified specially. The number of new and unread notifications are displayed at the menu bar of the user account page.

* User controlling account information

Users are provided all the tools through which they can alter the information provided by them at the time of signing up. They can even change their passwords and account pictures easily via the tools and options provided.

* Administrative privileges:

The Site’s Administrator, or Admin as commonly referred, can view and control all the activities going on in the website. He can view, download and delete all the files uploaded and can see all the users and their public information and can send them messages and even delete their accounts permanently.

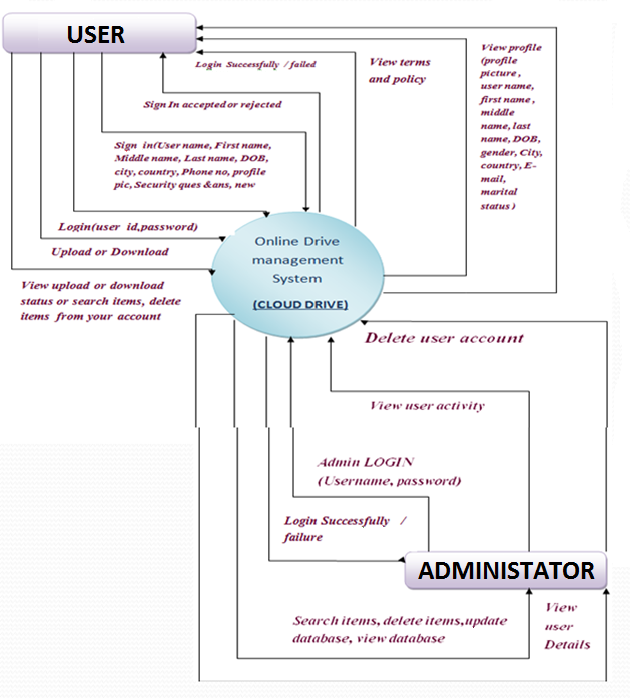
* Securities provided:

Various application-level securities and protocol-layer based securities are provided for the maximum security possible. Various controls against unauthorized access are implemented and no unauthorized user can modify or even view other users' information or can gain admin privileges without a valid user/admin session.

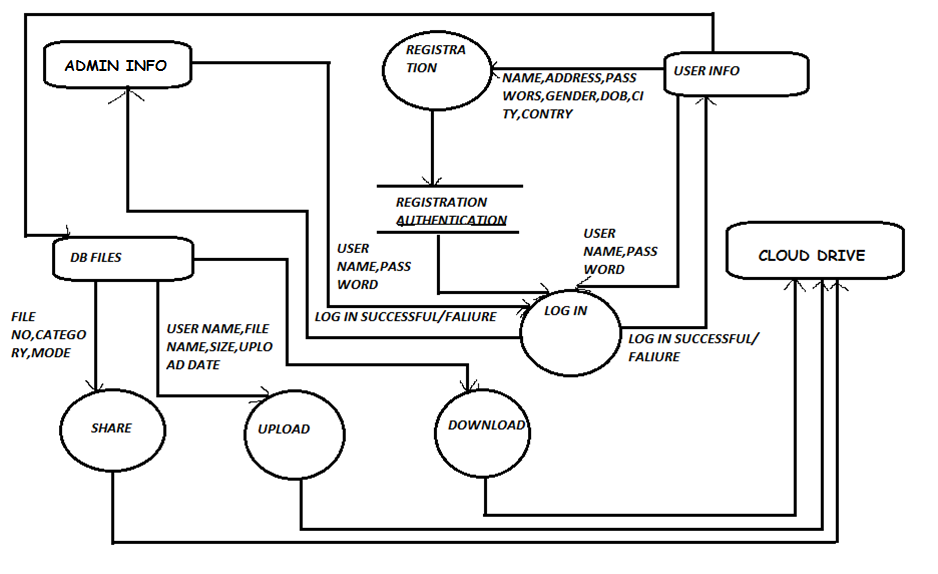
For example SSL security layer is added for secure file upload and download and protection from unauthorized access and hacking. Even auto-logging out from user account is also provided when a certain period of inactivity is detected for obvious security reasons.

1. DATA FLOW DIAGRAM (DFD):

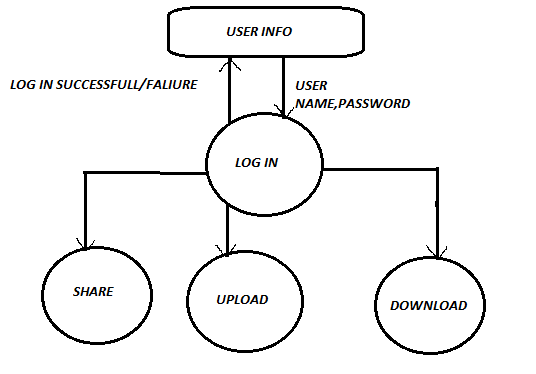
* Context Level Data Flow Diagram



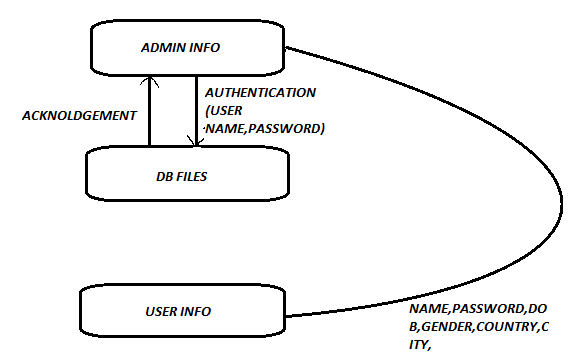
* First Level Data Flow Diagram



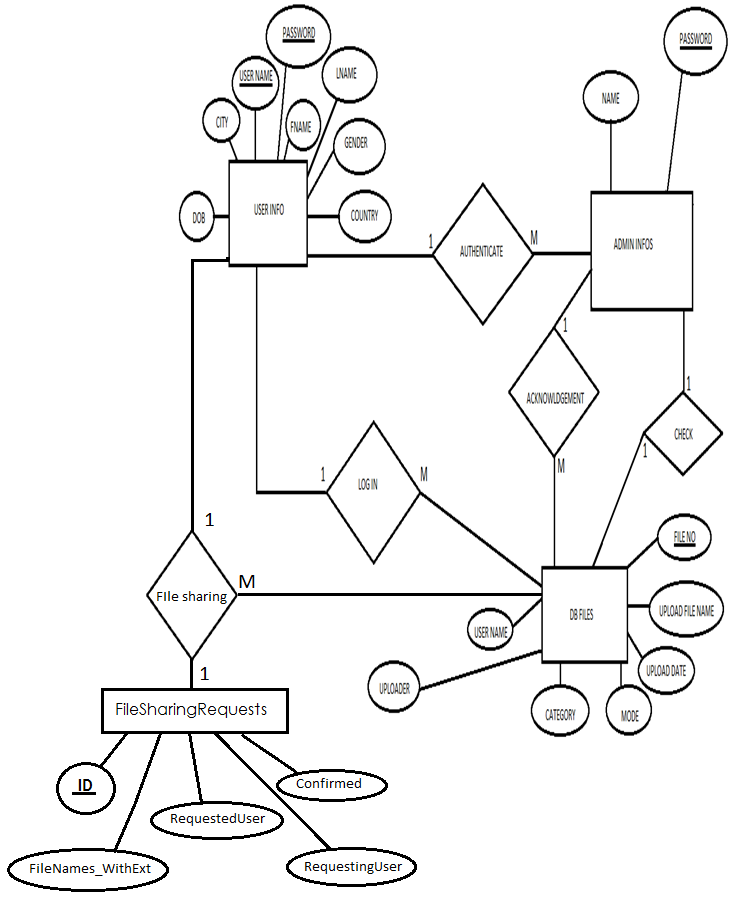
* Second level Data Flow Diagram for LOGIN PAGE



* Second Level Data Flow Diagram for ADMIN PAGE



3. ENTITY-RELATIONSHIP DIAGRAM (ERD)



* TABLE DESIGN:

A complete Database and tables details with primary key and foreign key and proper constraints in various fields are detailed below:

* Table: UserInfos

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Key Identification** | **Field Description** |
| Username | VARCHAR (50) | Primary Key | Unique ID of the user |
| UserPwd | VARCHAR (100) |  | Password of the user |
| UserFirstName | VARCHAR (100) |  | First name of the user |
| UserMiddleName | VARCHAR (100) |  | Middle name of the user |
| UserLastName | VARCHAR (100) |  | Last name of the user |
| UserGender | VARCHAR (50) |  | Gender of the user |
| UserDOB | VARCHAR (50) |  | Date of Birth of the user |
| UserCountry | VARCHAR (100) |  | Country of the user |
| UserCity | VARCHAR (100) |  | City of the user |
| UserPhoneNumber | VARCHAR (50) |  | Phone number of the user |
| UserMartialStatus | VARCHAR (1) |  | Marital status of the user |
| UserEMail | VARCHAR (100) |  | E-Mail address of the user |
| UserSecurityQuestion | VARCHAR (1000) |  | Security question of the user |
| UserSecurityAnswer | VARCHAR (1000) |  | Security answer of the user |
| UserProfilePicFile | VARCHAR (1000) |  | Profile picture of the user |
| JoiningDateTime | DATETIME |  | Date and time of joining of the user |
| Notifications | VARCHAR (MAX) |  | Notifications received by the user |

* Table: FileSharingRequests

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Key Identification** | **Field Description** |
| ID | INT | Primary Key | File sharing request ID |
| RequestingUser | VARCHAR (50) |  | The user accepting the file sharing request |
| RequestedUser | VARCHAR (50) |  | The user sending the file sharing request |
| FileNames\_WithExt | VARCHAR (MAX) |  | The full-names of the files shared |
| Confirmed | CHAR (1) |  | Flag symbolizing the confirmation status of the request |

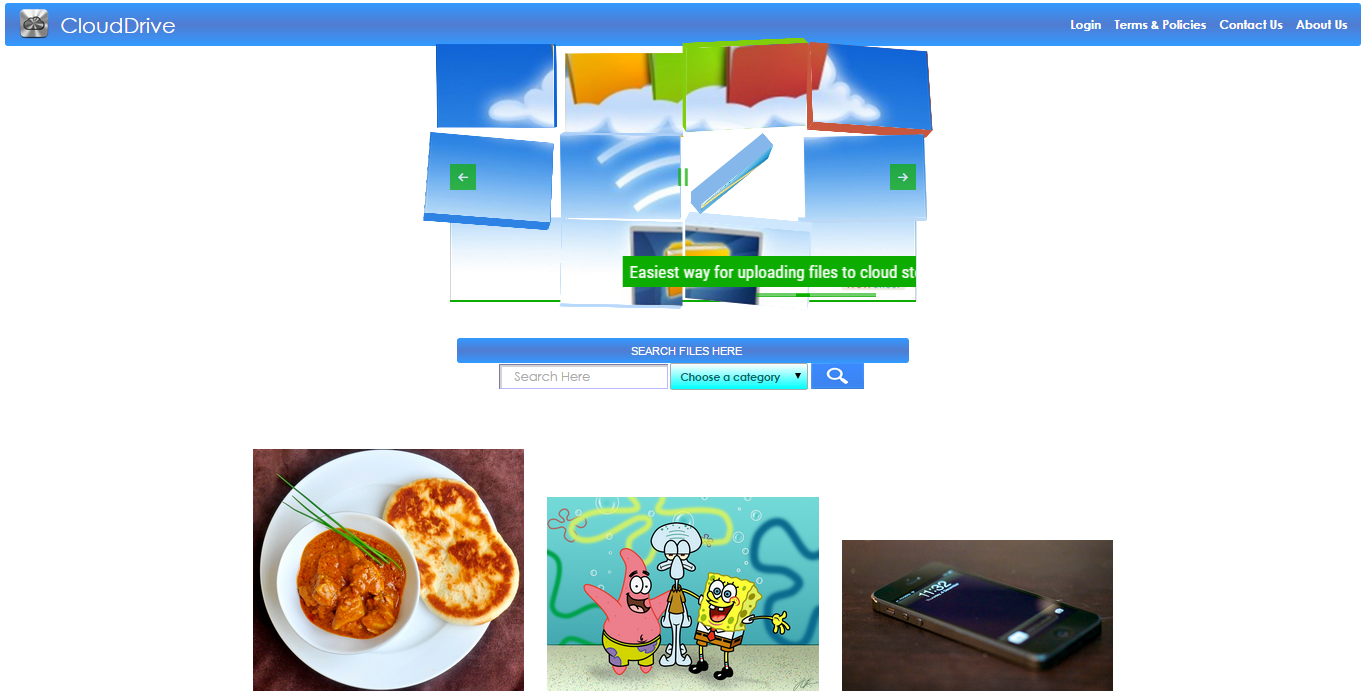
* Table: DBFiles

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Key Identification** | **Field Description** |
| FileNo | INT | Primary Key | File ID |
| UploaderUserName | VARCHAR (50) |  | Username of the uploader |
| Category | VARCHAR (100) |  | File Category |
| Mode | VARCHAR (50) |  | Upload mode of the file |
| UploadedFileName | VARCHAR (2000) |  | File name of the file |
| UploadedFileName\_WithExt | VARCHAR (2000) |  | File name of the file with format extension |
| UploadedFileSize | VARCHAR (50) |  | Size of the file |
| UploadDateTime | DATETIME |  | Date and time of upload of the file |
| SharedWith | VARCHAR (800) |  | Another users with whom the file has been shared |

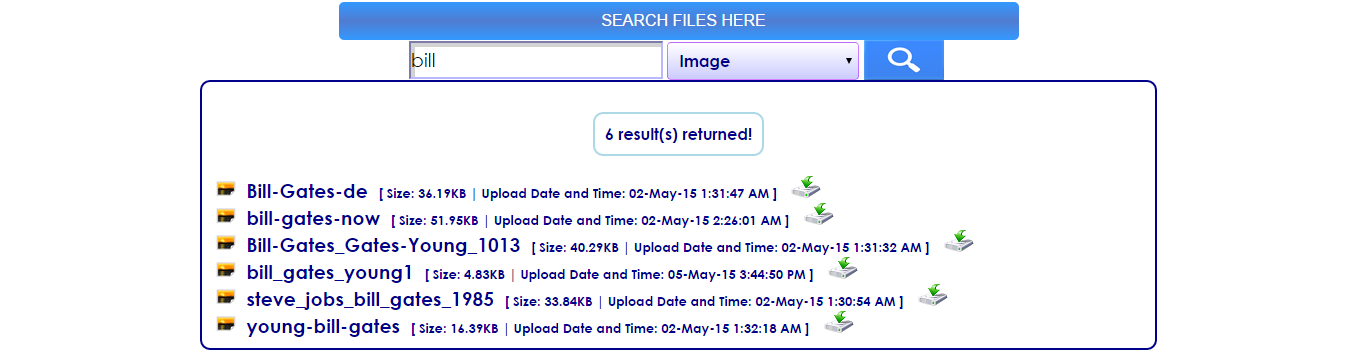
* Table: AdminInfo

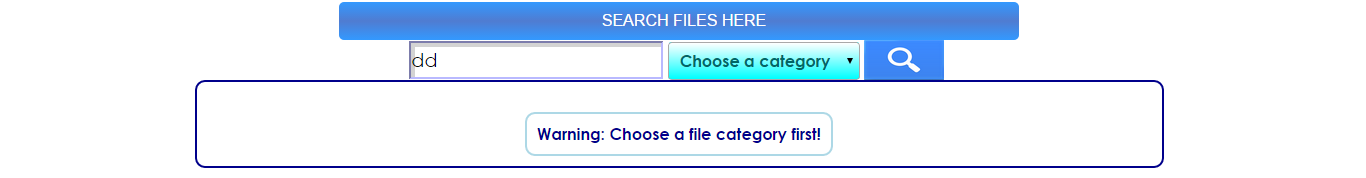
|  |  |  |  |
| --- | --- | --- | --- |
| **Data Name** | **Data Type** | **Key Identification** | **Field Description** |
| AdminName | VARCHAR (50) | Primary Key | Unique ID of the Admin |
| AdminPwd | VARCHAR (100) |  | Password of the Admin |
| LastLogoutDateTime | DATETIME |  | Latest date and time of Admin logout |

* User Interface:
* Homepage:

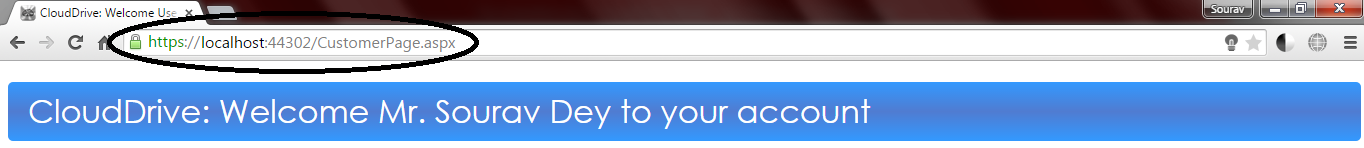


* Searching files category-wise by user in the Homepage and its various error checkings:

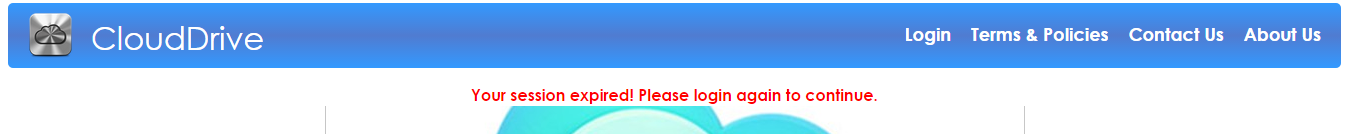




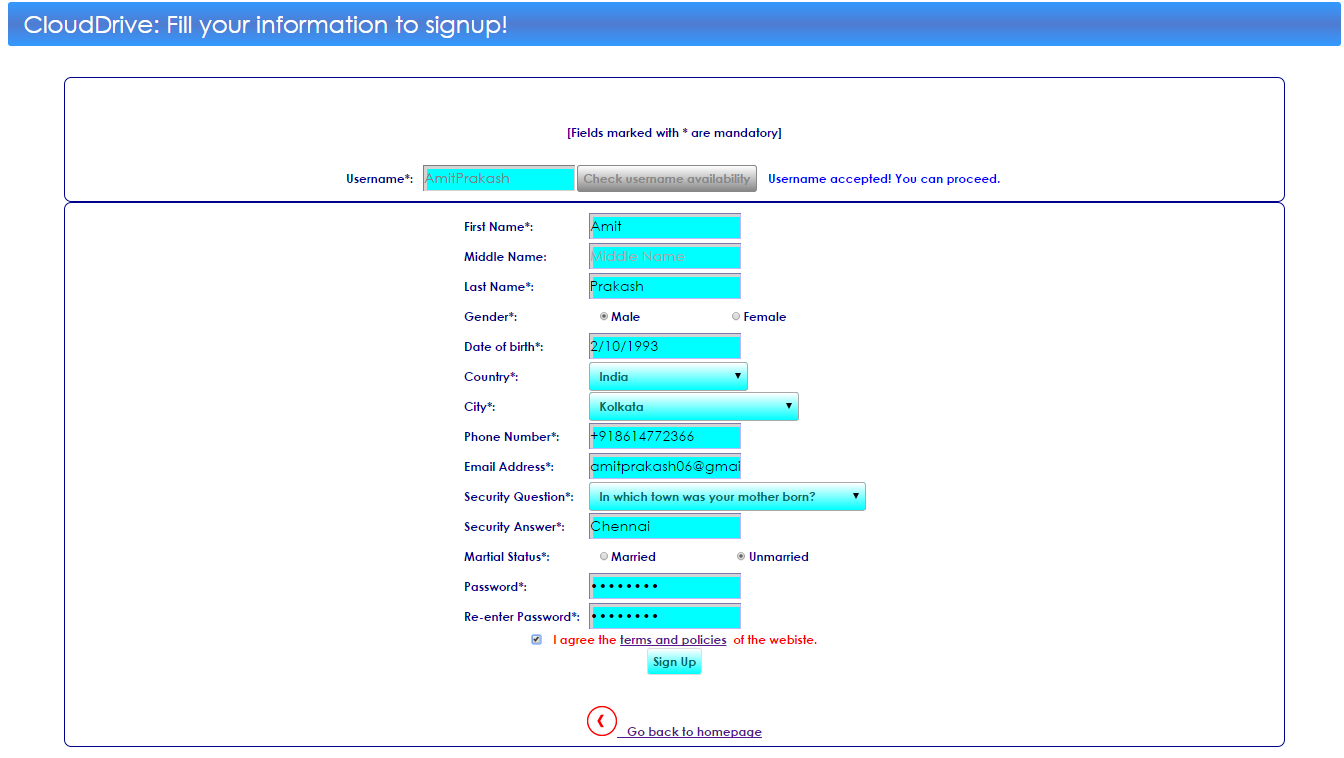
* SSL security layer provided for additional protocol-layer based security:



* Auto session expiry after a long period of inactivity:

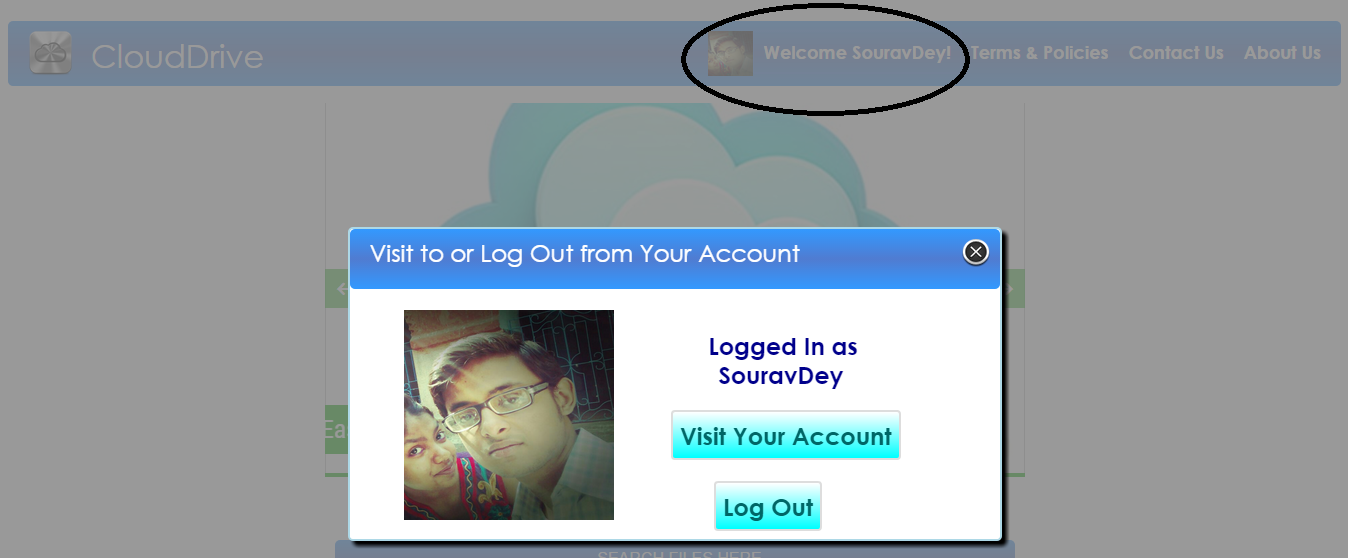


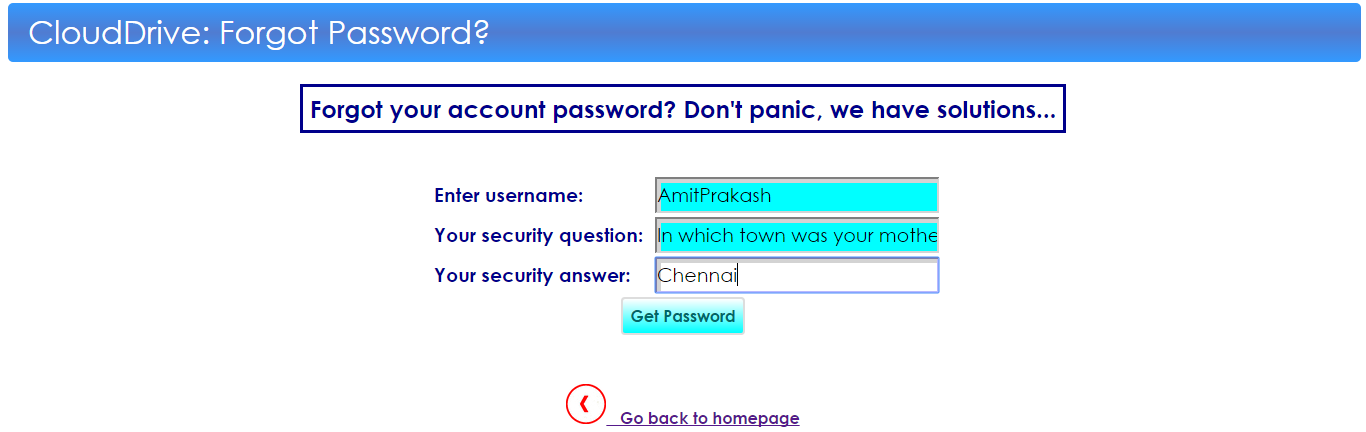
* User signing up into the website:

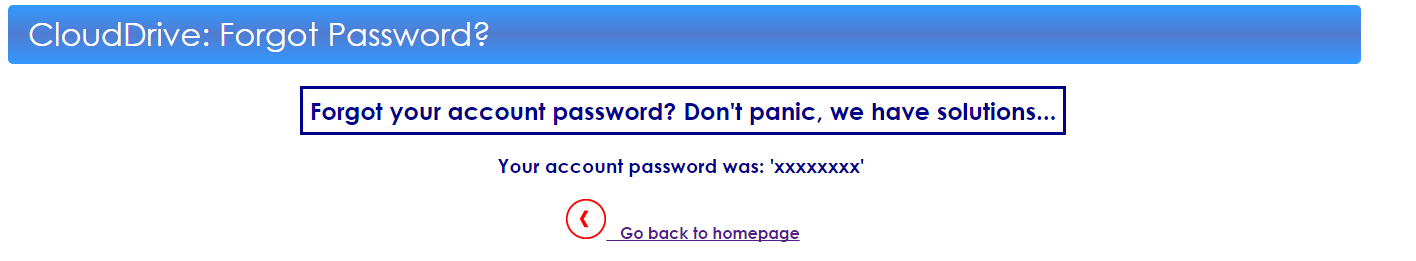


* Logging in from Homepage:

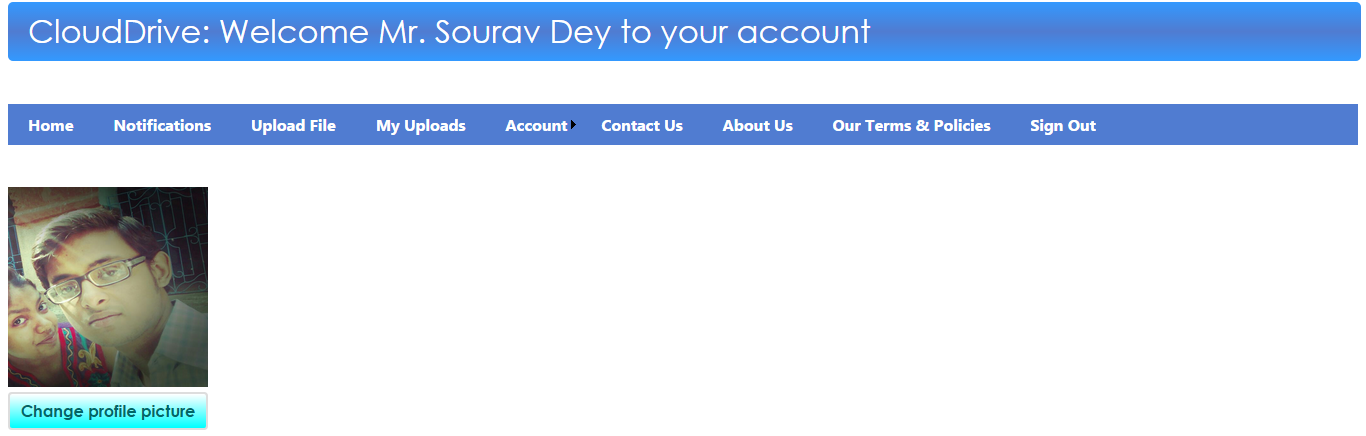


* On successful user logging in in the homepage:
* Retrieving forgotten password by user:

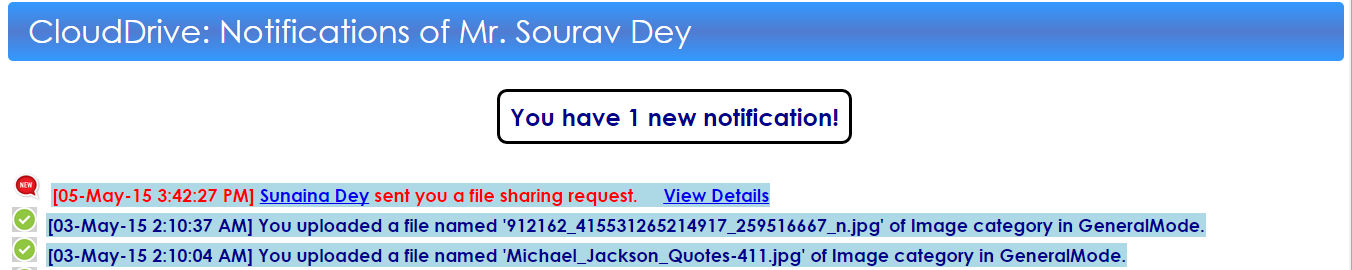




* User account page:



* Page showing new notifications to user:

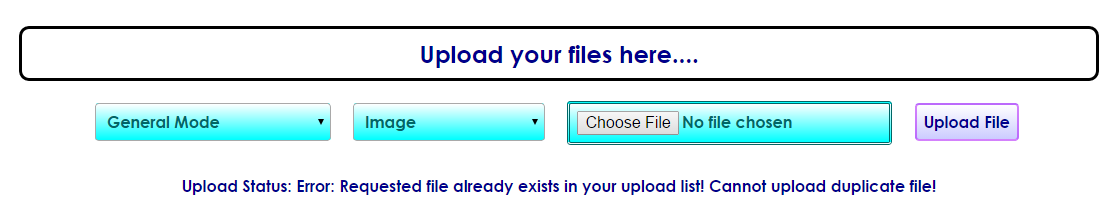


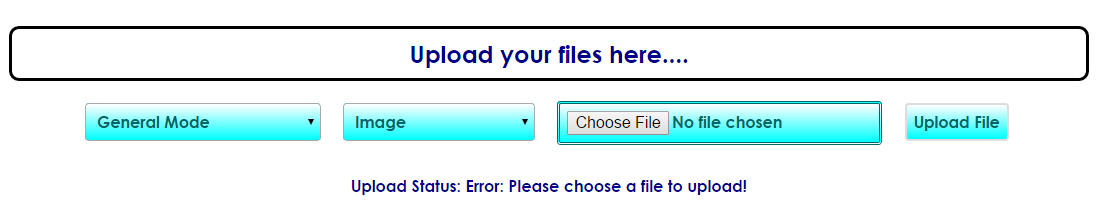
* Menu bar showing number of new notifications to user:

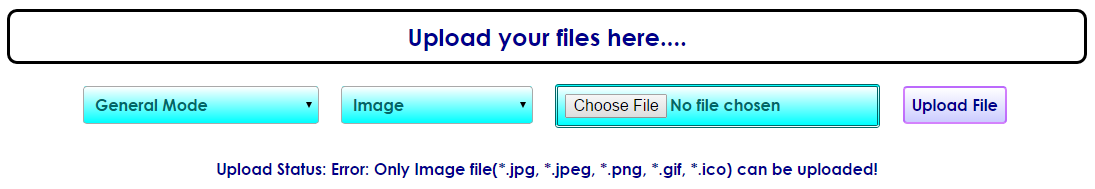
C:\Users\Sourav\Desktop\Docs_Temp\Screenshots\Choosed\ntf bar.png

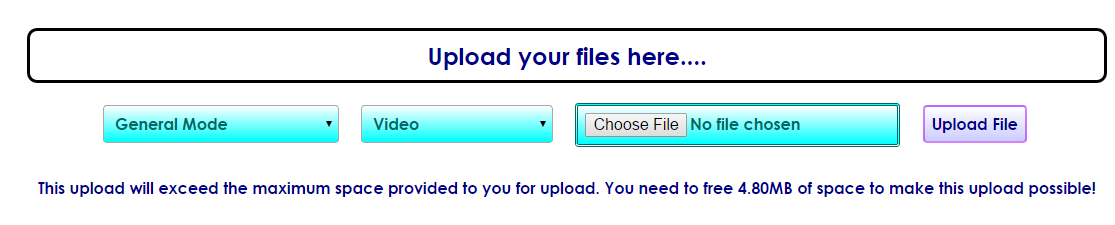
* Uploading file by user and its various error checkings:



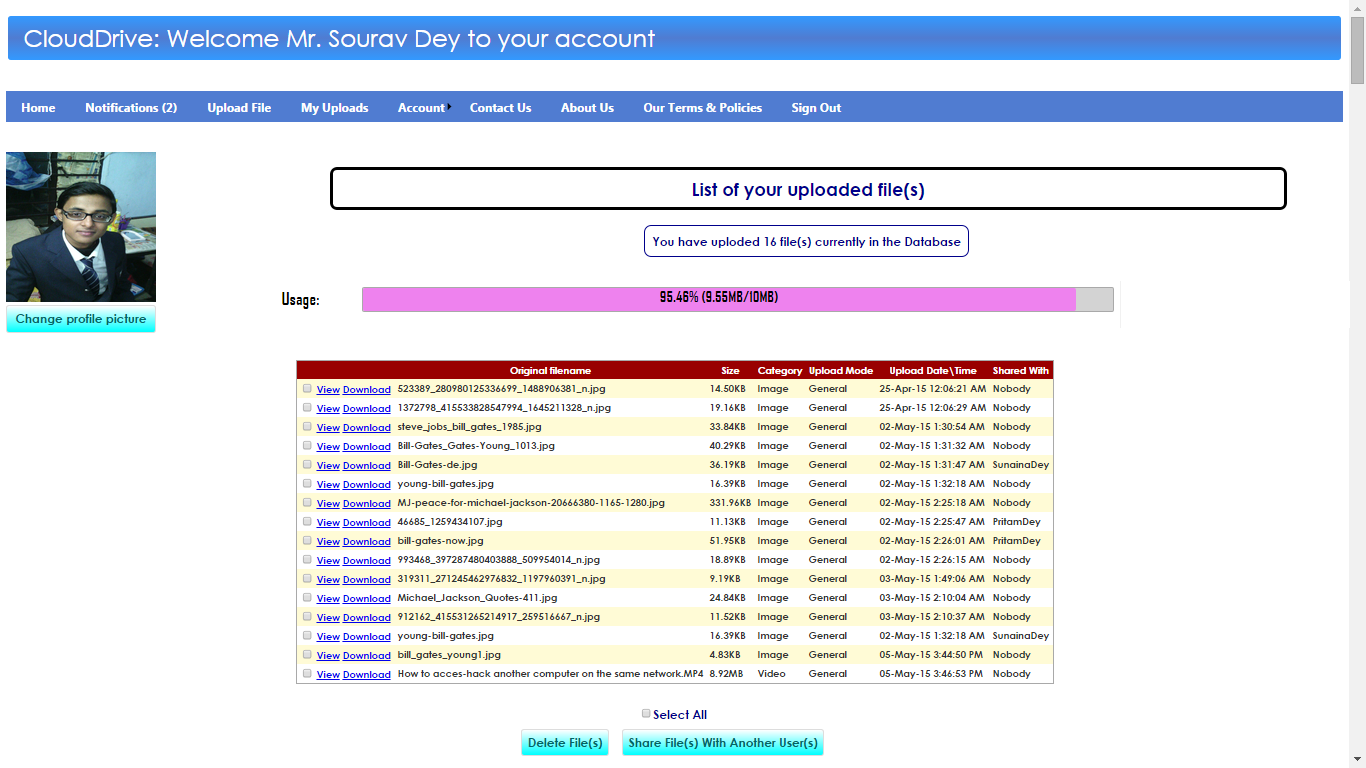




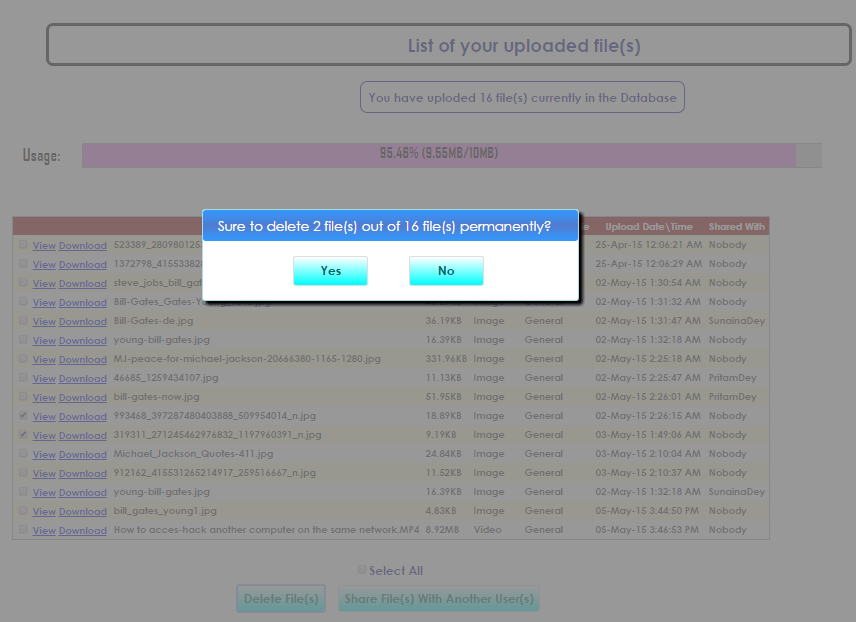




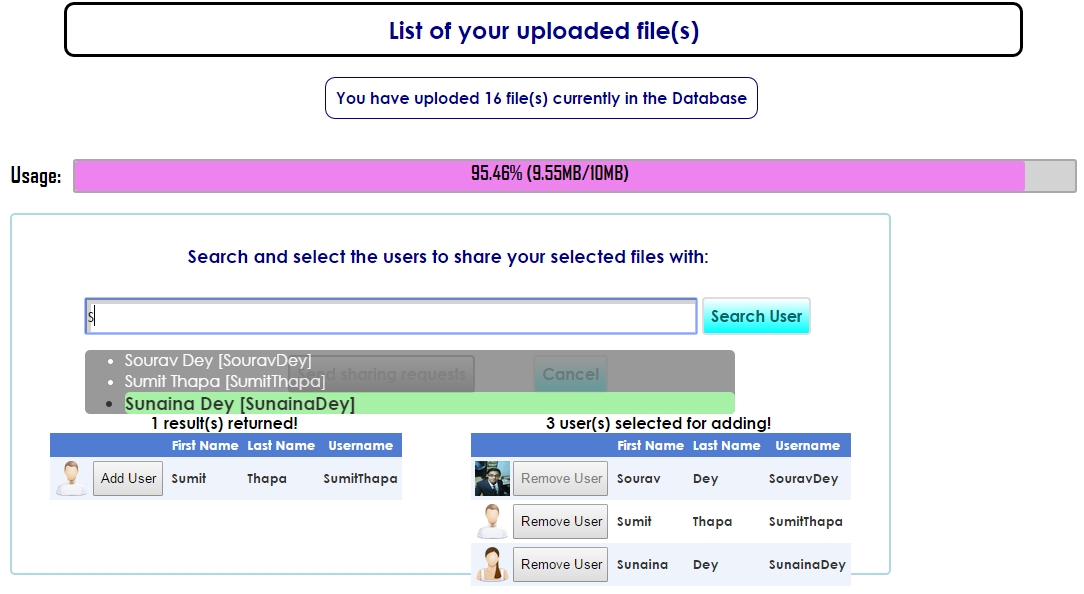
* User file upload list:



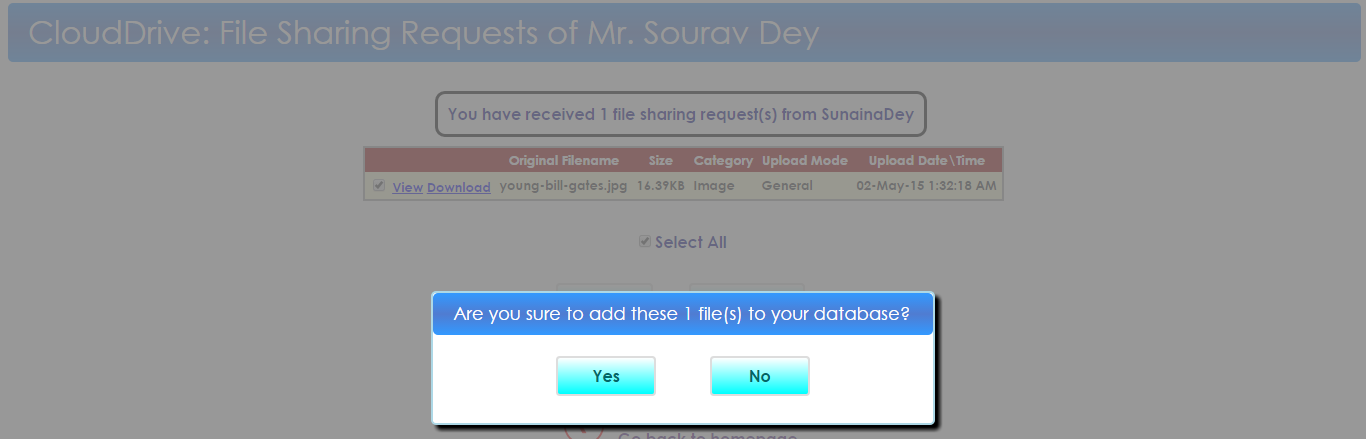
* Deleting uploaded files by user:



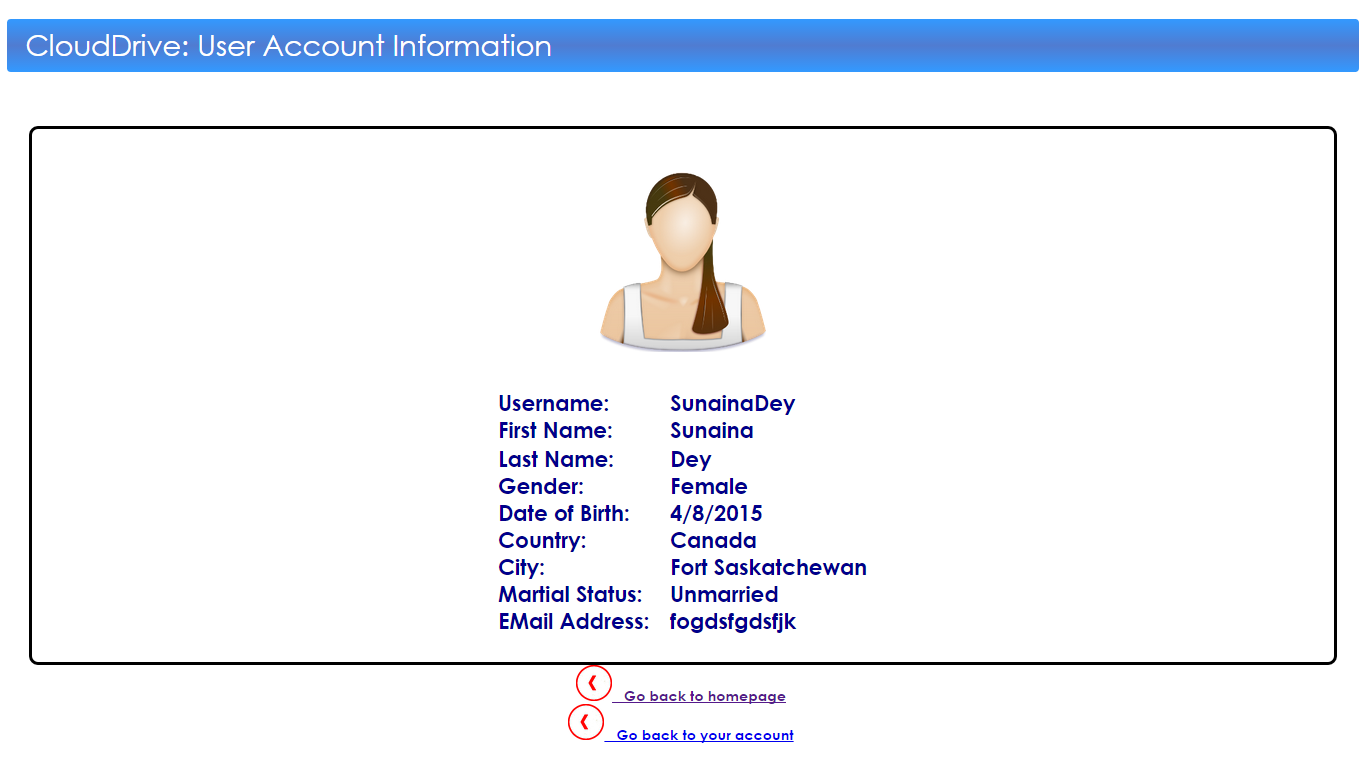
* Sharing files to other users:



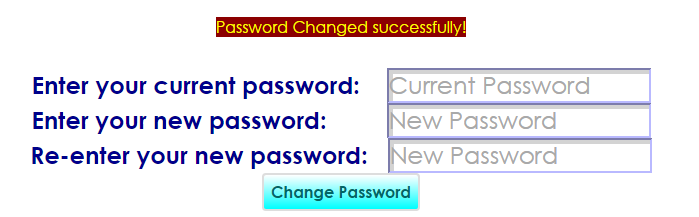
* Receiving and accepting new file sharing request by user:



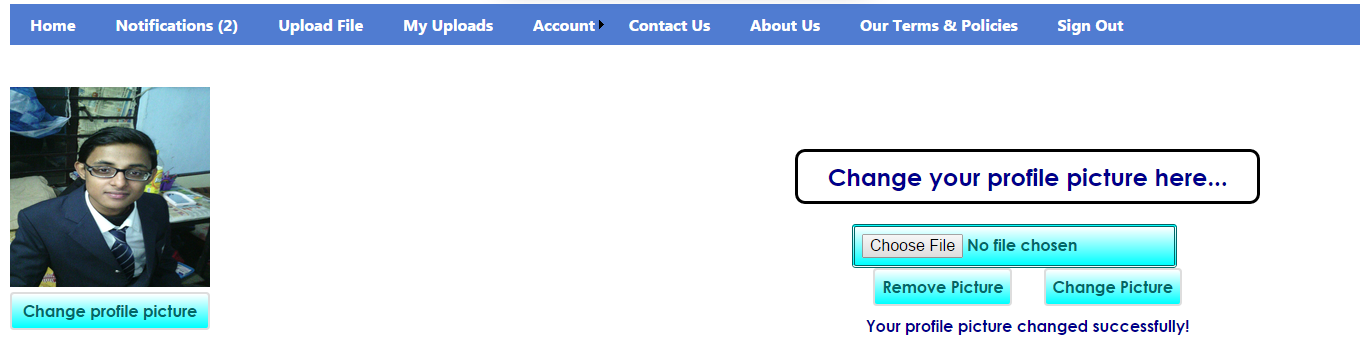
* User viewing account information of another user who sent file sharing request:

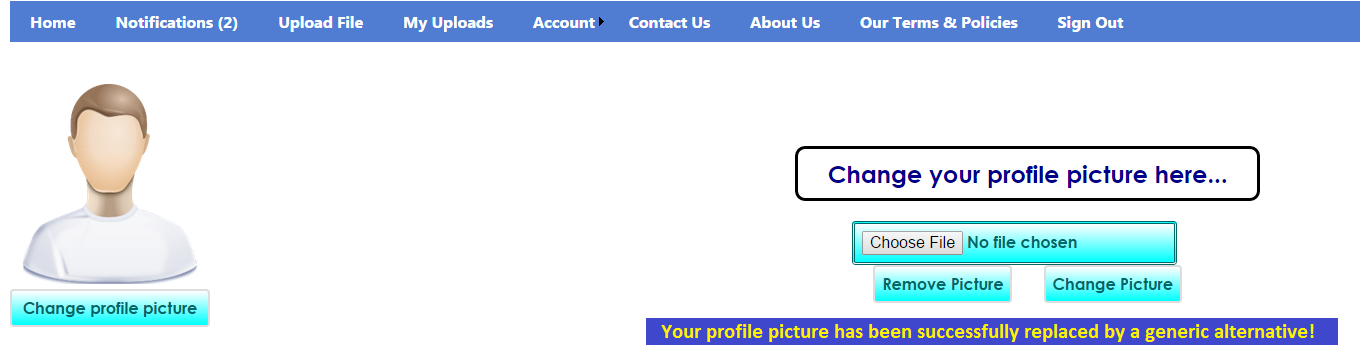


* User changing account password by user:

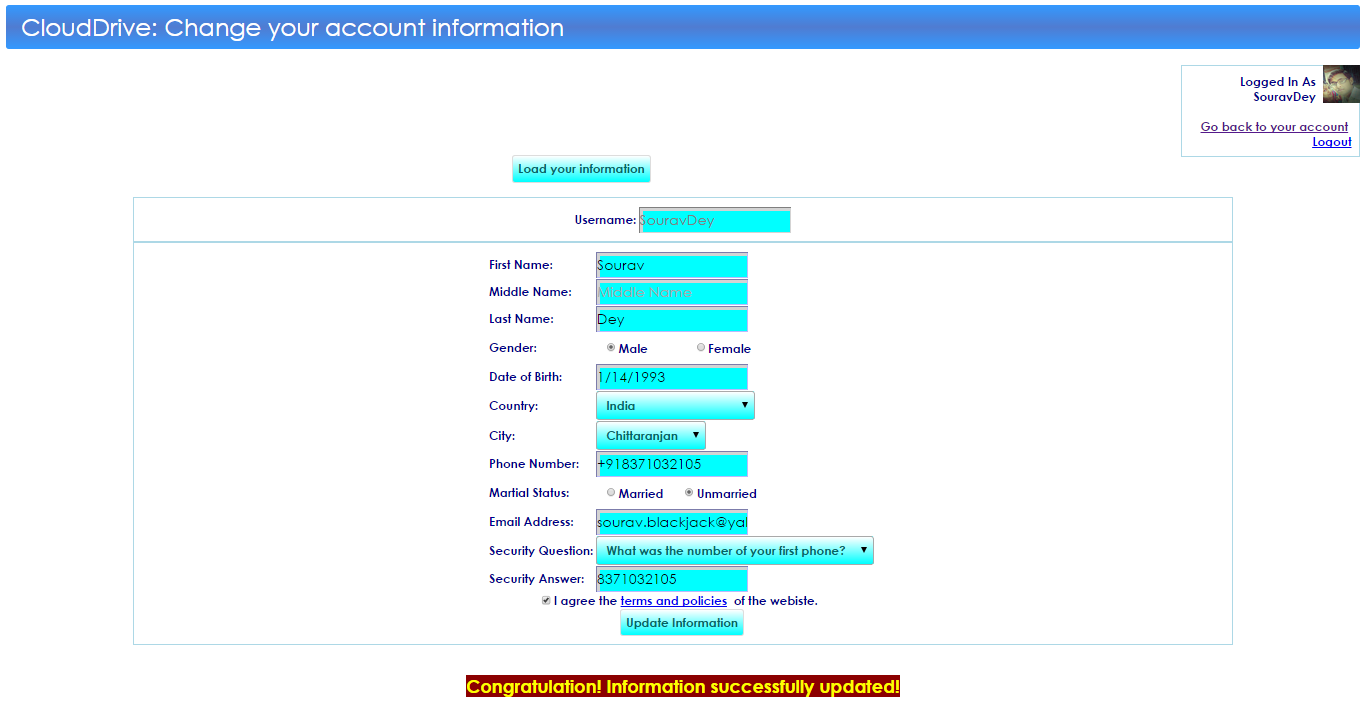


* Changing/removing account picture:

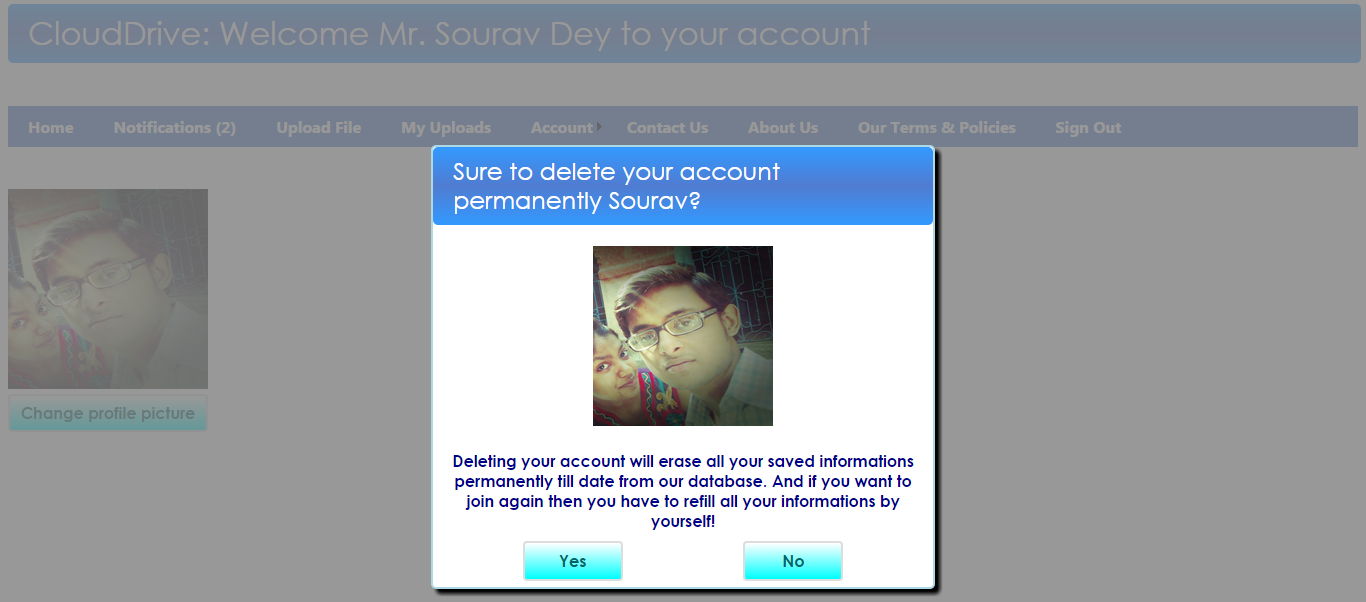




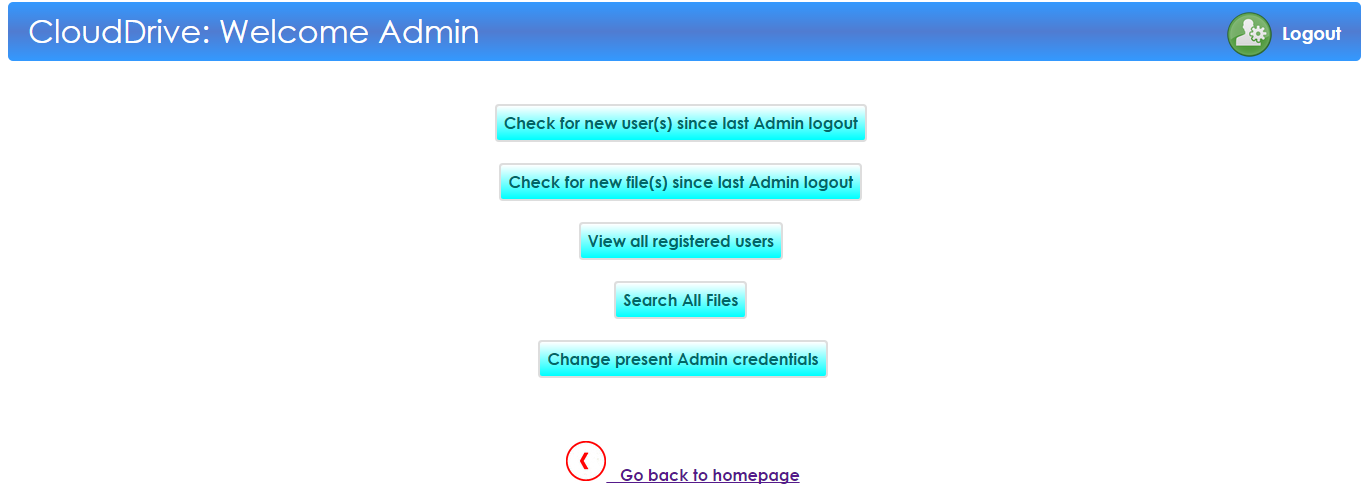
* Changing user account information:



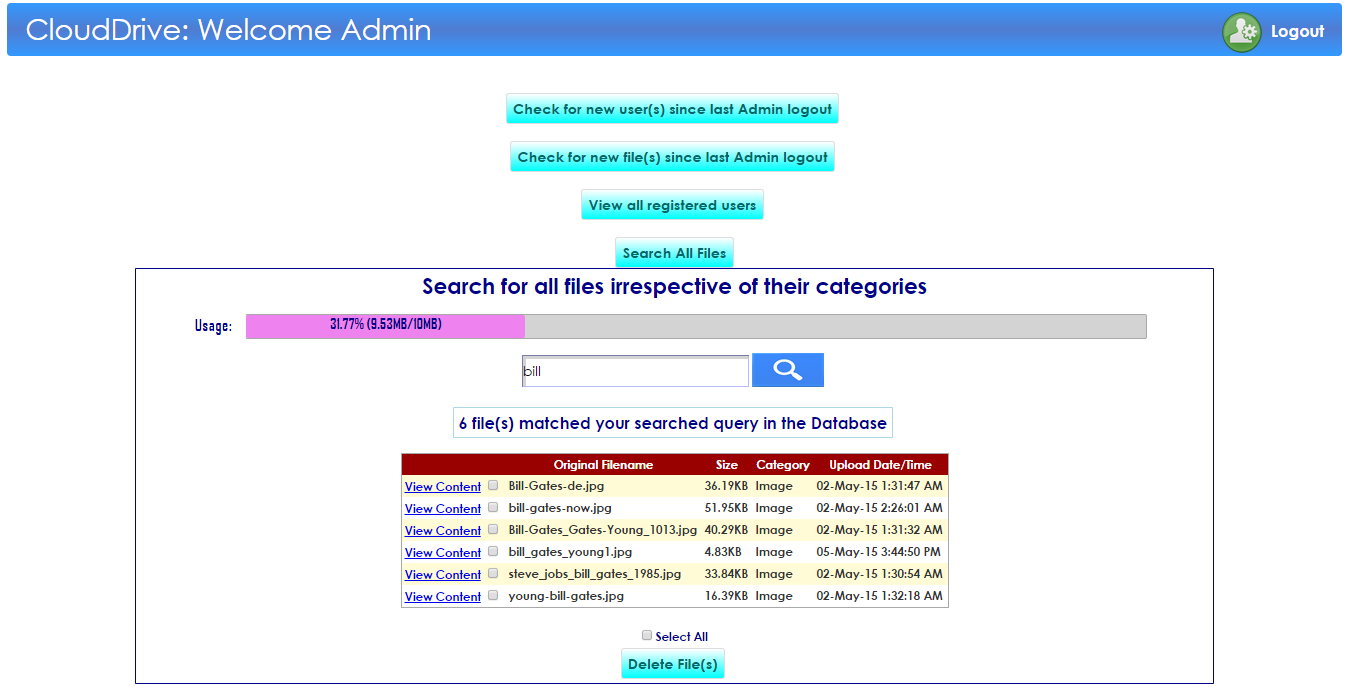
* Deleting user account:



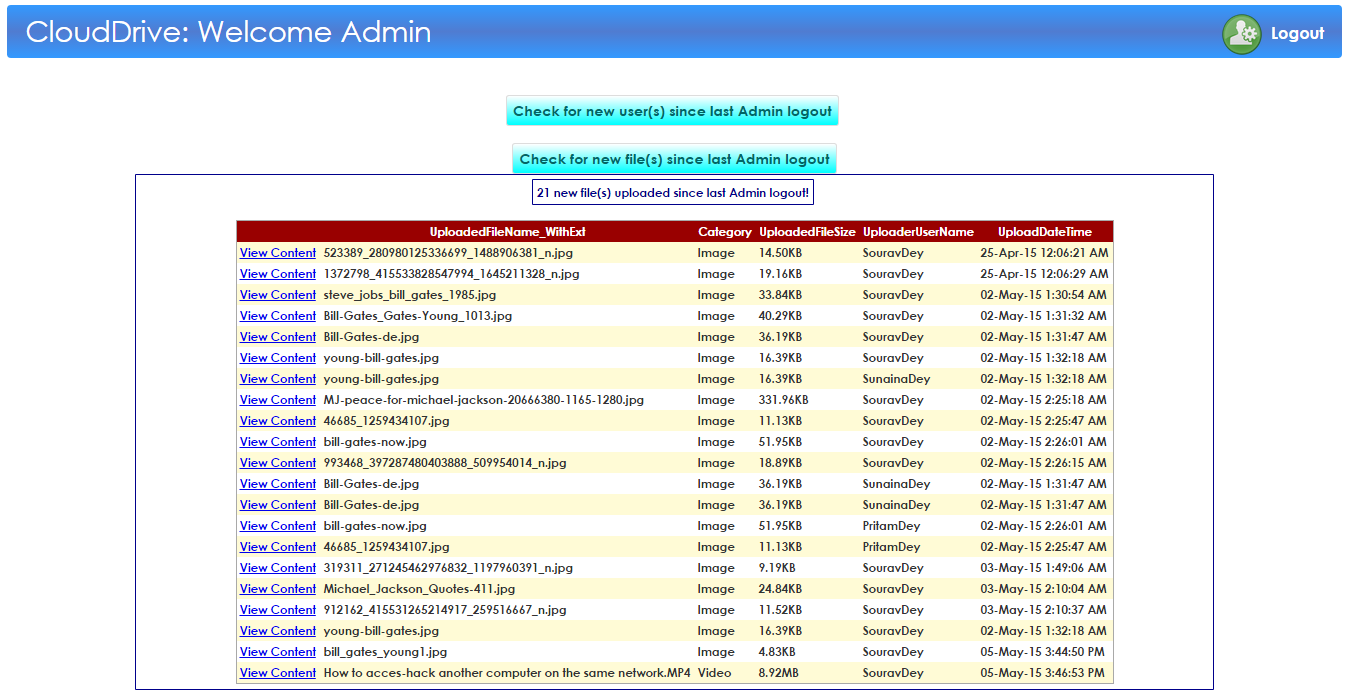
* Admin page:



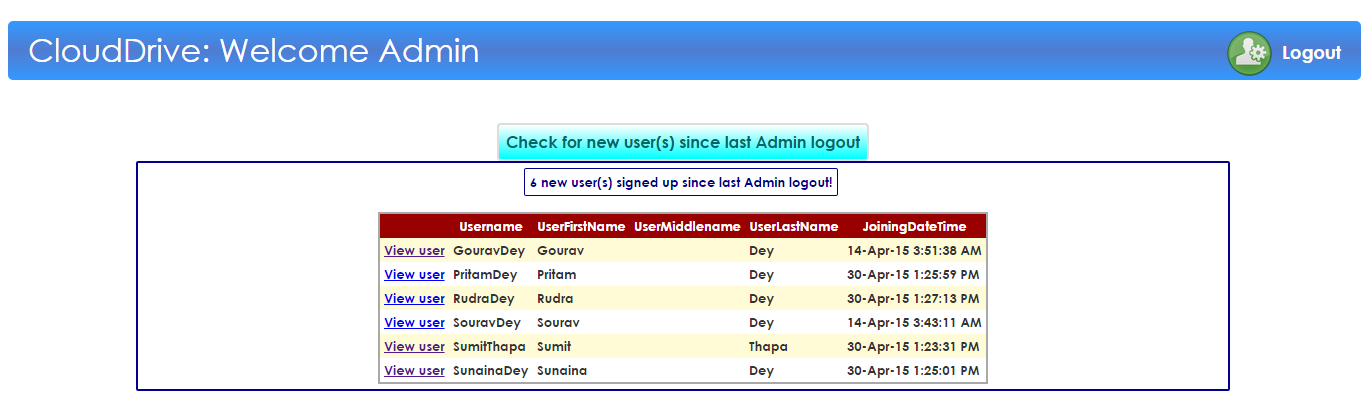
* File searching by Admin irrespective of their category:



* Viewing latest file uploads since last Admin logout:



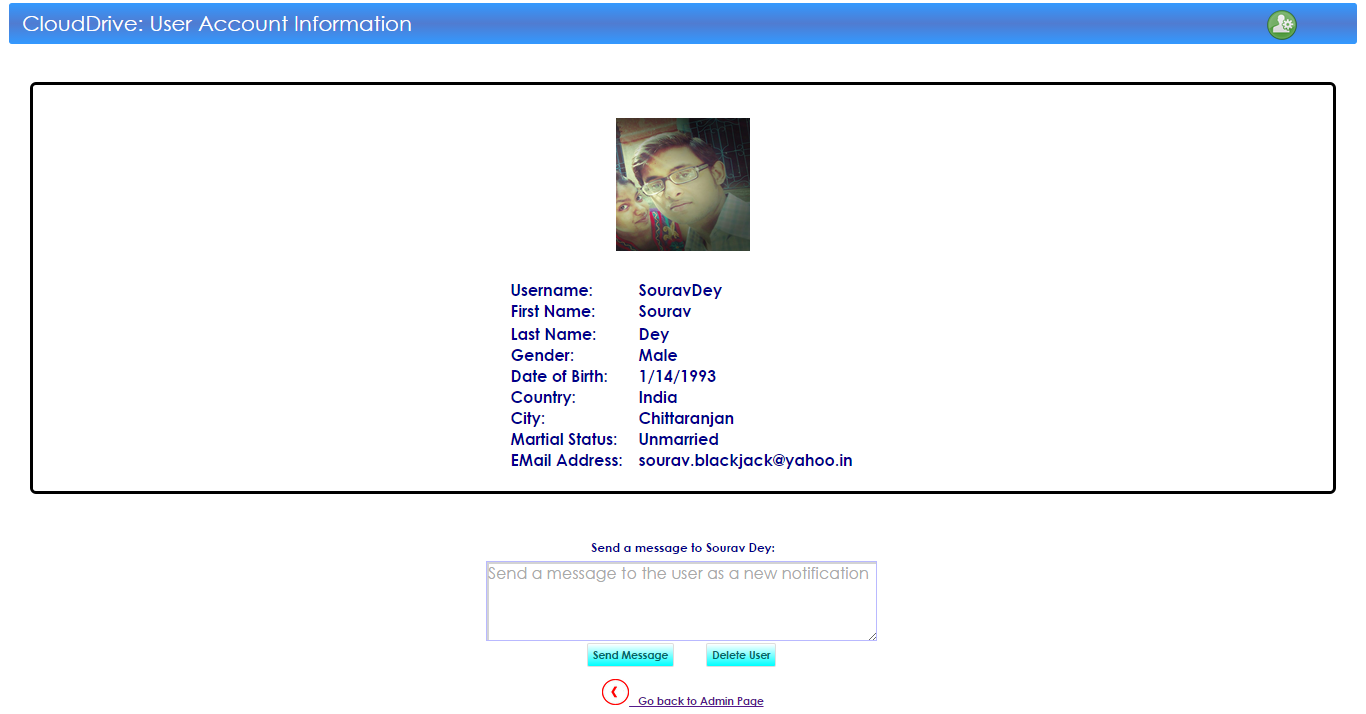
* Viewing latest users signed up since last Admin logout:



* Viewing all registered users by Admin:



* Viewing user information by Admin and additional facilities to send them messages and even delete their account permanently:



* CODING:

CODE SNIPLETS OF HOME PAGE IN C# LANGUAGE

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

{

stringconstr=System.Configuration.ConfigurationManager.ConnectionStrings["DatabaseConnectionString"].

ConnectionString;

void Wait()

{

System.Threading.Thread.Sleep(1000);

}

void Show\_LoginLogoutOption()

{

if (Session["user"] != null)

{

btnLogout.Visible=true; btnLogin.Visible=false;

btnLogout.Text = "Welcome " + Session["user"].ToString() +"!";

ImageProPic.Visible = true;

if (Session["user"].ToString() == "Admin")

{

imgPopupUserPic.ImageUrl = "~/Images/NotificationImages/AdminPic.png"; return;

}

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var res = from v in DB.GetTable<UserInfo>() where v.Username == Session["user"].ToString() select v;

foreach (var r in res)

{

ImageProPic.ImageUrl = r.UserProfilePicFile;

if (ImageProPic.ImageUrl == "")

{

if (r.UserGender == "Male")

ImageProPic.ImageUrl = "~/UploadedFiles/UserProfilePictures/NoProfilePicMale.png";

else

ImageProPic.ImageUrl = "~/UploadedFiles/UserProfilePictures/NoProfilePicFemale.png";

}

}

btnPopupVisitUserAcc.PostBackUrl = "CustomerPage.aspx?CustName=" + Session["user"].ToString();

}

else

{

btnLogout.Visible = false;

btnLogout.Text = "";

btnLogin.Visible = true;

ImageProPic.Visible = false;

ImageProPic.ImageUrl = "";

}

}

}

CODE SNIPLETS OF ADMIN PAGE IN C# LANGUAGE

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

{

void LoadAdminCredentials()

{

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var res = from v in DB.GetTable<AdminInfo>() select v;

foreach(var r in res)

{

txtAdminUsername.Text = r.AdminName;

txtAdminPassword.Text = r.AdminPwd;

}

}

void ChangeAdminCredentials()

{

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var res = from v in DB.GetTable<AdminInfo>() select v;

foreach (var r in res)

{

r.AdminName = txtAdminUsername.Text;

r.AdminPwd = txtAdminPassword.Text;

}

DB.SubmitChanges();

lblConfirmAdminCredentialsChange.Text = "Admin credentials changed successfully!";

}

void DeleteThreats()

{

string filepath = null;

string filename = null;

int TotalRows = GridViewThreatsList.Rows.Count;

int DeletedRows = 0;

for (int i = 0; i < TotalRows; i++)

{

CheckBox chkbox = new CheckBox();

chkbox = (CheckBox)GridViewThreatsList.Rows[i].FindControl("chk");

if (chkbox.Checked)

{

filename = GridViewThreatsList.Rows[i].Cells[5].Text;

//Response.Write("filename: " + filename);

filepath = GridViewThreatsList.Rows[i].Cells[8].Text;

//Response.Write("filepath: " + filepath);

filepath = filepath.Substring(0, (filepath.Length - 4));

filepath = filepath.Substring(0, (filepath.Length - filename.Length));

DirectoryInfo mydir = new DirectoryInfo(@Server.MapPath(filepath));

FileInfo[] f = mydir.GetFiles();

foreach (FileInfo file in f)

{

int no = (file.Name).Length;

string temp = file.Name.Substring(0, (no - 4));

if (temp == filename)

{

lblDeleteThreatsConfirmation.Text = "File '" + file.Name + "' deleted successfully!";

//Response.Write("File to be deleted: '" + file.Name + "'!");

file.Delete();

}

}

//GridViewThreatsList.DeleteRow(i);

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var del\_rec = from v in DB.GetTable<DBFile>() where v.FileAddressInServerDB == GridViewThreatsList.Rows[i].Cells[8].Text select v;

foreach (var r in del\_rec)

DB.DBFiles.DeleteOnSubmit(r);

DB.SubmitChanges();

DeletedRows++;

}

}

}

CODE SNIPLETS OF ADMIN CONTROL PAGE IN C# LANGUAGE

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

{

void DisplayUserInfo()

{

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var res = from v in DB.GetTable<UserInfo>() where v.Username == Request.QueryString["User"].ToString() select v;

foreach (var r in res)

{

lblUsername.Text = r.Username;

lblFirstName.Text = r.UserFirstName;

lblMiddleName.Text = r.UserMiddleName;

lblLastName.Text = r.UserLastName;

lblGender.Text = r.UserGender;

lblDOB.Text = r.UserDOB;

lblCountry.Text = r.UserCountry;

lblCity.Text = r.UserCity;

lblMartialStatus.Text = r.UserMartialStatus;

lblEMail.Text = r.UserEMail;

ImageUserProfilePicture.ImageUrl = r.UserProfilePicFile;

}

}

void DeleteUserAccount()

{

FunZoneDatabaseDataContext DB = new FunZoneDatabaseDataContext();

var res = from v in DB.GetTable<UserInfo>() where v.Username == Request.QueryString["User"].ToString() select v;

foreach (var r in res)

DB.UserInfos.DeleteOnSubmit(r);

DB.SubmitChanges();

Response.Redirect("AdminPage.aspx");

}

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

{

DisplayUserInfo();

}

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

{

DeleteUserAccount();

}

}

CODE SNIPLETS OF SIGN UP PAGE IN C# LANGUAGE

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

{

bool CheckForAnyErrors()

{

if (RadioButtonMale.Checked == false && RadioButtonFemale.Checked == false)

{

lblWarning.Text = "Please choose a gender!"; return false;

}

if (DropDownListCountry.SelectedItem.Text == "Choose your country")

{

lblWarning.Text = "Please choose a country!"; return false;

}

if (DropDownListCities.SelectedItem.Text == "Choose your city")

{

lblWarning.Text = "Please choose a city!"; return false;

}

if (txtPhNo.Text.Length < 5)

{

lblWarning.Text = "Please enter your phone number!"; return false;

}

if (RadioButtonMarried.Checked == false && RadioButtonUnmarried.Checked == false)

{

lblWarning.Text = "Please choose a martial status!"; return false;

}

return true;

}

void LoadCountries()

{

DropDownListCountry.Items.Clear();

string[] countries= {"Choose your country", "Afghanistan", "Algeria",

... "Vietnam", "Zimbabwe"};

foreach(string c in countries)

DropDownListCountry.Items.Add(c);

}

* CONCLUSION & FUTURE ENHANCEMENTS:

This project was developed to provide certain facilities to the user that may be useful to them in terms of cloud computing; however there are lots of scope to improve the performance of the online file and drive management system in the area of user interface, database performance, and query processing time etc. So there are many things for future enhancement of this project. Along with the current development on the projects there are a numerous features yet to be added. The future enhancements that are possible in the project are as follows:

1. Make the interactions of the users to the website more easy and intuitive by providing tools which are easy to control.
2. Upgrade the file sharing process of the registered users to more intuitive and efficient by adding the facilities to create new groups and adding members to it and even control the visibility of the files to the group members selectively if desired by the user.
3. Provide a way to scan each file while uploading to the server database for malwares and viruses and also the same while downloading the file by other users.
4. Enhance the duplicity-free storage of files while a user is uploading his/her files to the server database. This can be done by searching if the file contents uploaded by the user is already been uploaded by any other user before, if this condition seems to be true then a “file upload successful” message will be displayed to the user immediately and the duplicate file will not be uploaded to the server database but a link to the same file will be made in the database of the user uploading the duplicate file, with a different filename.
5. Provide greater and more efficient and precise regulatory tools to the Administrator for better site management.

* BIBLIOGRAPHY:

Reference Books:

1. Fundamentals of software engineering - Rajib Mall
2. ASP.NET 3.5 using C#- CMC Academy PRO.
3. Database system concepts - Korth and Siberschatz.

Websites:

* [*www.codeproject.com*](http://www.codeproject.com)
* *www.tutorialspoint.com*
* [*www.aspdotnetcodes.com*](http://www.aspdotnetcodes.com)
* [*www.*](http://www.onlinebuff.com)*stackoverflow.com*
* *www.social.msdn.microsoft.com/Forums*