A PROJECT REPORT

ON

CONTACT MANAGEMENT SYSTEM

**By**

**CHAUHAN RIKIN DIPAKKUMAR (CE-022) (19CEUOS106)**

**GOPANI KENIL GHANSHYAMBHAI (CE-049) (19CEUOS080)**

**B.Tech CE Semester-VI**

**Subject: Service Oriented Computing (SOC) (CE - 619)**

**Guided by:**

Prof. Prashant M. Jadhav Associate Professor Prof. Ankit P. Vaishnav Assistant Professor Dept. of Computer Engg.



**Faculty of Technology Department of Computer Engineering**

**Dharmsinh Desai University**



**Faculty of Technology Department of Computer Engineering**

**Dharmsinh Desai University**

**CERTIFICATE**

This is to certify that the practical / term work carried out in the subject of

**Service Oriented Computing (CE - 619)** and recorded in this journal is the

bonafide work of

**CHAUHAN RIKIN DIPAKKUMAR (CE-022) (19CEUOS106)**

**GOPANI KENIL GHANSHYAMBHAI (CE-049) (19CEUOS080)**

of B.Tech Semester **VI** in the branch of **Computer Engineering**

during the academic year **2021-2022**.

**Project Guide:**

Prof. Prashant M. Jadhav Prof. Ankit P. Vaishnav

Associate Professor, Assistant Professor,

Dept. of Computer Engg., Dept. of Computer Engg.,

Faculty of Technology Faculty of Technology

Dharmsinh Desai University, Nadiad Dharmsinh Desai University, Nadiad

**Head of the Department**

Dr. C. K. Bhensdadia,

Professor & Head,

Dept. of Computer Engg.,

Faculty of Technology

Dharmsinh Desai University, Nadiad

# Table of Contents

[TABLE OF CONTENTS 3](#_bookmark0)

1. [ABSTRACT 4](#_bookmark1)
2. [INTRODUCTION 5](#_bookmark2)
   1. [BRIEF INTRODUCTION 5](#_bookmark3)
   2. [TOOLS/TECHNOLOGIES USED 6](#_Tools/Technologies_Used)
3. [SOFTWARE REQUIREMENT SPECIFICATIONS 7](#_bookmark5)
4. [DESIGN 1](#_bookmark6)2
   1. [USE CASE DIAGRAM 1](#_bookmark7)2
   2. [CLASS DIAGRAM](#_bookmark10) 13
   3. [DATA DICTIONARY 14](#_Data_Dictionary)
5. [IMPLEMENTATION DETAILS 1](#_bookmark14)6
   1. [ACCOUNT MODULE](#_bookmark15) 16
   2. [CONTACT MODULE](#_Loan_Module) 16
   3. [GROUP MODULE 1](#_bookmark17)6
   4. [FUNCTION PROTOTYPES](#_bookmark18) 17
6. [TESTING](#_bookmark19) 19
7. [SCREENSHOTS](#_bookmark20) 21
8. [CONCLUSION](#_bookmark21) 34
9. [LIMITATIONS AND FUTURE EXTENSIONS 3](#_bookmark22)6
10. [REFERENCE / BIBLIOGRAPHY 37](#10._Reference_/_Bibliography)

# Abstract

* The Contact Management System is a web application that provides users the platform to manage their contacts individually as well as group wise.
* Users can add new contact and can manage contacts.
* Users can also create group and can manage group by adding their existing contacts to the particular group or by removing contacts from the particular group.
* Thus “CONTACT MANAGEMENT SYSTEM” provides a number of benefits like: no need to remember all the contact number like we do in traditional approach and can also group contacts based on certain requirement like family / friends.

# Introduction

### Brief Introduction

* In this fast-moving world it is necessary to develop contact management system as it allows a user to manage their contacts via internet and doesn’t need to remember them at all.
* The system is designed in such a way that it only works with end users.
* User can register themselves into the system and then after can login into the system.
* User can manage their contacts by adding, updating, and deleting their contacts from contact list and can also view contacts.
* User can manage their group by adding, updating, and deleting contacts from their respective groups and can also view groups.

### Tools/Technologies Used

### Technologies:

* + - WCF framework
    - ASP .Net framework
    - .Net core Web API framework
    - JavaScript
    - Bootstrap
    - CSS
    - SQL Server Database

### Tools:

* Visual Studio
* Git

### Platform:

* Local development server

# Software Requirement Specifications

### Product Scope

This system is designed to provide the user online contact management services.

### System Functional Requirements

.

1. **Account Module:**

**R.1.1 Register:**

* Description: User can register into the system.
* Input: User details.
* Output: User is redirected to login page with success message.

**R.1.2 Login:**

* Description: User can login to the system.
* Input: User details.
* Output: User is redirected to dashboard.

**R.1.3 Logout:**

* Description: User can logout from the system.
* Input: User selection.
* Output: User is redirected to the login page with success message.

1. **Contact Module:**

**R.2.1 Create contact:**

* Description: User can add contact to their contact list.
* Input: Contact details.
* Output: Success message.

**R.2.2 View contact:**

* Description: User can view individual contact from the contacts added.
* Input: User selection.
* Output: Contact details.

**R.2.3 Update contact:**

* Description: User can update individual contact from the contacts added.
* Input: Contact details.
* Output: Success message.

**R.2.4 Delete contact:**

* Description: User can delete individual contact from the contacts added.
* Input: User selection.
* Output: Success message.

**R.2.5 View recently added contacts:**

* Description: User can view last 3 recently added contacts from the contact list.
* Input: User selection.
* Output: Contacts list.

1. **Group Module:**

**R.3.1 Create group:**

* Description: User can add group to their group list.
* Input: Group details.
* Output: Success message.

**R.3.2 View group:**

* Description: User can view individual group from the groups added.
* Input: User selection.
* Output: Group details.

**R.3.3 Update group:**

* Description: User can update individual group from the groups added.
* Input: Group details.
* Output: Success message.

**R.3.4 Delete group:**

* Description: User can delete individual group from the groups added.
* Input: User selection.
* Output: Success message.

**R.3.5 View recently added groups:**

* Description: User can view last 3 recently added groups from the group list.
* Input: User selection.
* Output: Groups list.

**R.3.6 Add contacts to the group:**

* Description: User can add multiple contacts from the contact list to the selected group.
* Input: User selection of contacts.
* Output: Success message.

**R.3.6 Remove contacts from the group:**

* Description: User can remove multiple contacts from the selected group.
* Input: User selection of contacts.
* Output: Success message.

### 3.3 Other Non-functional Requirements

### Performance

The system must be interactive and must not involve long delays. Though in case of opening the website components or loading the page the system shows the delays less than 2 seconds.

### Safety

The user’s data is highly personal. The system has authorization to avoid any un- authorized access to user’s private data.

### Reliability

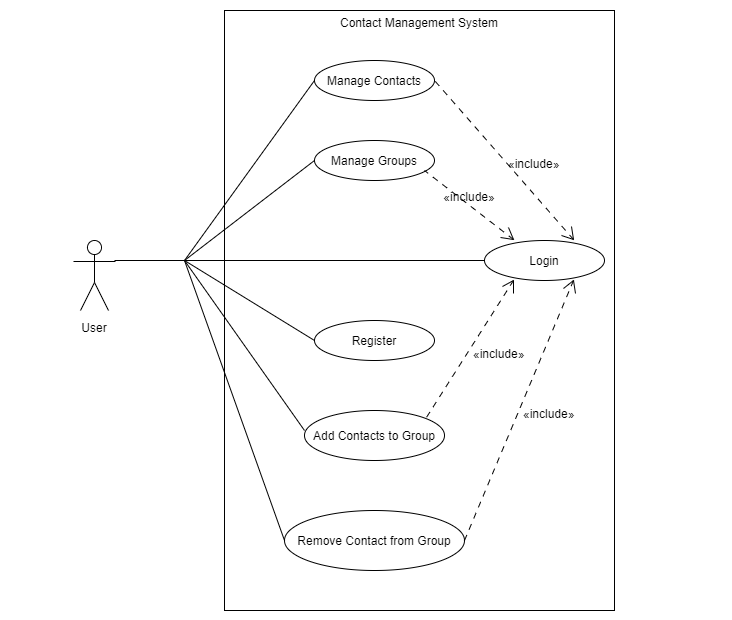
As the system has personal data, its reliability is the major factor for consideration.

### Database

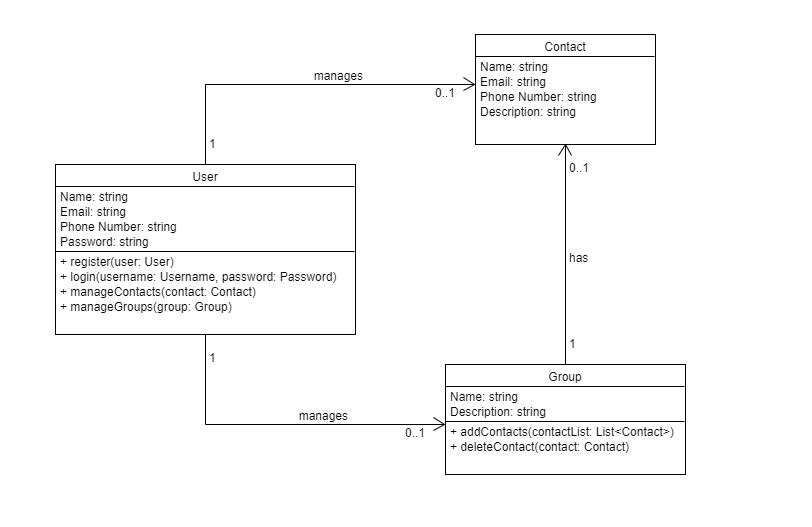
System requires to access user profile data fast to maintain the performance.

# Design

## Use Case Diagram

z 

## Class Diagram

****

## Data Dictionary

### Users Model

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Users | | | | | | | | | |
| Sr no. | Field Name | Data Type | Width | Unique | Null | Default | Primary Key / Foreign  Key | Referred Table | Description |
| 1 | UserId | int | 4 | Yes | No | None | Primary Key | - | AUTO\_INCREMENT |
| 2 | Email | nvarchar | 50 | Yes | No | None | - | - | - |
| 3 | Name | nvarchar | 50 | No | No | None | - | - | - |
| 4 | Password | nvarchar | 50 | No | No | None | - | - | - |
| 5 | PhoneNumber | nvarchar | 15 | No | No | None | - | - | - |

### Contacts Model

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Contacts | | | | | | | | | |
| Sr no. | Field Name | Data Type | Width | Unique | Null | Default | Primary Key / Foreign  Key | Referred Table | Description |
| 1 | ContactId | int | 4 | Yes | No | None | Primary Key | - | AUTO\_INCREMENT |
| 2 | UserId | int | 4 | No | No | None | Foreign Key | Users | - |
| 3 | Name | nvarchar | 50 | No | No | None | - | - | - |
| 4 | Email | nvarchar | 50 | No | Yes | None | - | - | - |
| 5 | PhoneNumber | nvarchar | 15 | Yes | No | None | - | - | - |
| 6 | Description | nvarchar | 50 | No | Yes | None | - | - | - |
| 7 | PhotoPath | nvarchar | MAX | No | Yes | None | - | - | - |

### Groups Model

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Groups | | | | | | | | | |
| Sr no. | Field Name | Data Type | Width | Unique | Null | Default | Primary Key / Foreign  Key | Referred Table | Description |
| 1 | GroupId | int | 4 | Yes | No | None | Primary Key | - | AUTO\_INCREMENT |
| 2 | UserId | int | 4 | Yes | No | None | Foreign Keys | Users | - |
| 3 | Name | nvarchar | 50 | Yes | No | None | - | - | - |
| 4 | Description | nvarchar | MAX | No | Yes | None | - | - | - |

### Group\_Contacts Model

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Group\_Contacts | | | | | | | | | |
| Sr no. | Field Name | Data Type | Width | Unique | Null | Default | Primary Key / Foreign  Key | Referred Table | Description |
| 2 | Id | int | 4 | Yes | No | None | Primary Key | - | AUTO\_INCREMENT |
| 1 | GroupId | int | 4 | Yes | No | None | Foreign Keys | Groups | - |
| 3 | ContactId | Int | 4 | Yes | No | None | Foreign Keys | Contacts | - |

# Implementation Details

The system consists of 6 basic modules namely

1. Account Module
2. Contact Module
3. Group Module

Each module consists of several methods to implement the required functionality. Implementation is done using WCF framework, ASP .Net framework, .Net core Web API framework. Database used in these modules is SQL Server Database.

## Account Module

This module handles the functions related to user’s accounts like: register, login, and logout. New users can create their account by providing the required details. Existing users can login to the system by providing their email-id and password.

## Contact Module

This module handle the functions related to contacts like: create new contact, view contact, update contact, delete contact, and view recently added contacts. User can also view the list of all his contacts added.

## Group Module

This module handles the functions related to groups like: create group, view group, update group, delete group, view recently added groups, add contacts to the group, and remove contacts from the group. User can also view the list of all his groups added.

z

## Function Prototypes

### Function prototypes of Account Module:

* User Register(User user)
* User Login(User user)
* void Logout(User user)

### Function prototypes of Contact Module:

* List<Contact> GetContacts(int userId)
* Contact AddContact(Contact contact)
* Contact GetContact(Contact contact)
* Contact UpdateContact(Contact contact)
* Contact DeleteContact(Contact contact)

### Function prototypes of Group Module:

* List<Group> GetGroups(int userId)
* Group AddGroup(Group group)
* Group GetGroup(Group group)
* Group UpdateGroup(Group group)
* Group DeleteGroup(Group group)
* GroupContact AddGroupContact(GroupContact groupContract)
* List<GroupContact> GetGroupContacts(int GroupId)
* GroupContact DeleteGroupContact(GroupContact groupContact)

z

* GroupContact DeleteGroupContactByContactId(GroupContact groupContact)
* GroupContact DeleteGroupContactByGroupId(GroupContact groupContact)

z

# 6. Testing

Manual unit testing of each module was done after successfully completing the module.

Each module was tested individually before integrating them with the whole system.

After integrating each module with the system, manual integrating testing was done in order to check if all modules are working properly together.

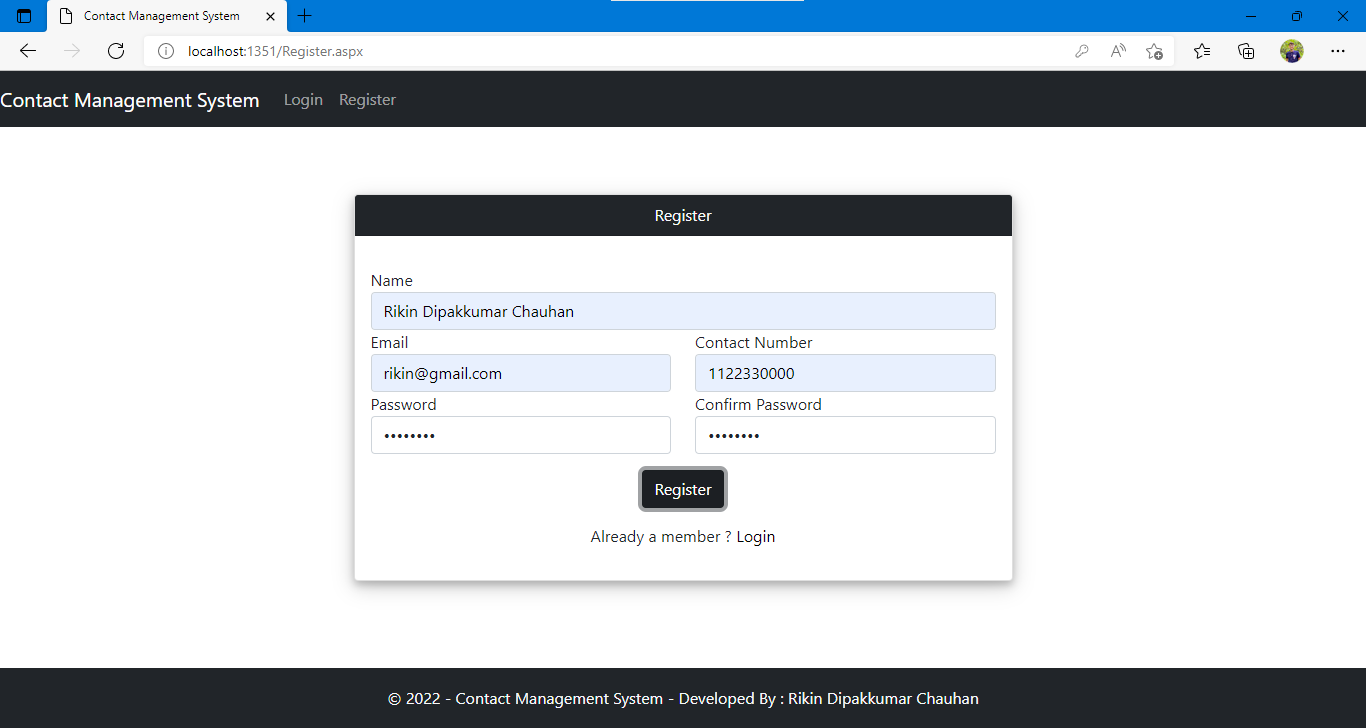
After completing all integrations, black-box testing of the whole system was carried out to ensure that the system works in a correct manner.

**Testing Method**: Manual Testing

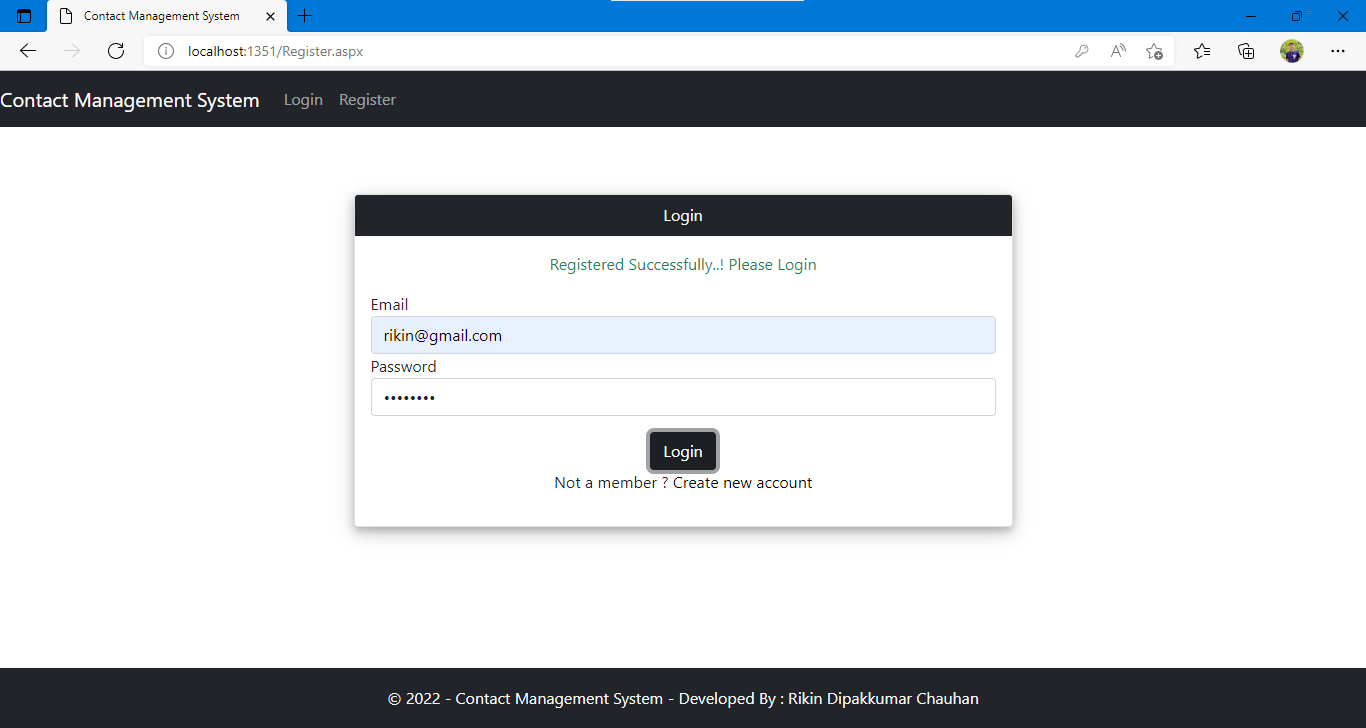
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Test Scenario** | **Expected Result** | **Actual Result** | **Status** |
| 1 | Login with incorrect credentials | User should not be able to log-in and shown an error message | The system shows the message ‘Invalid Credentials’ and redirected to login page | Success |
| 2 | Login with correct credentials | User should be able to log-in and redirected to dashboard page | User is logged in and redirected to dashboard page | Success |
| 3 | Register with Email which is already used | User should not be able register and shown an error message | The system shows the message ‘Email is already registered’ and user is redirected to register page | Success |
| 4 | Edit contact number to an existing number in the contact list | User should not be able to edit the contact and shown an error message | The system shows the message ‘Contact is already added’ and doesn’t update the contact | Success |
| 5 | Edit contact by entering all correct details | User should be able to edit the contact and redirected to view contact page | The system successfully edits the contact and redirected to view contact page | Success |
| 6 | User tries to access any unauthorized page without login | User should not be able to access it and redirected to login page with an error message | The system doesn’t allow to access any page without authorization and user is redirected to login page with the message ‘Access Denied! Please Login’ | Success |
| 7 | User tries to access any page that doesn’t exists | User should be redirected to error 404 page | The system redirects the user to error 404 page | Success |

# 7. Screenshots

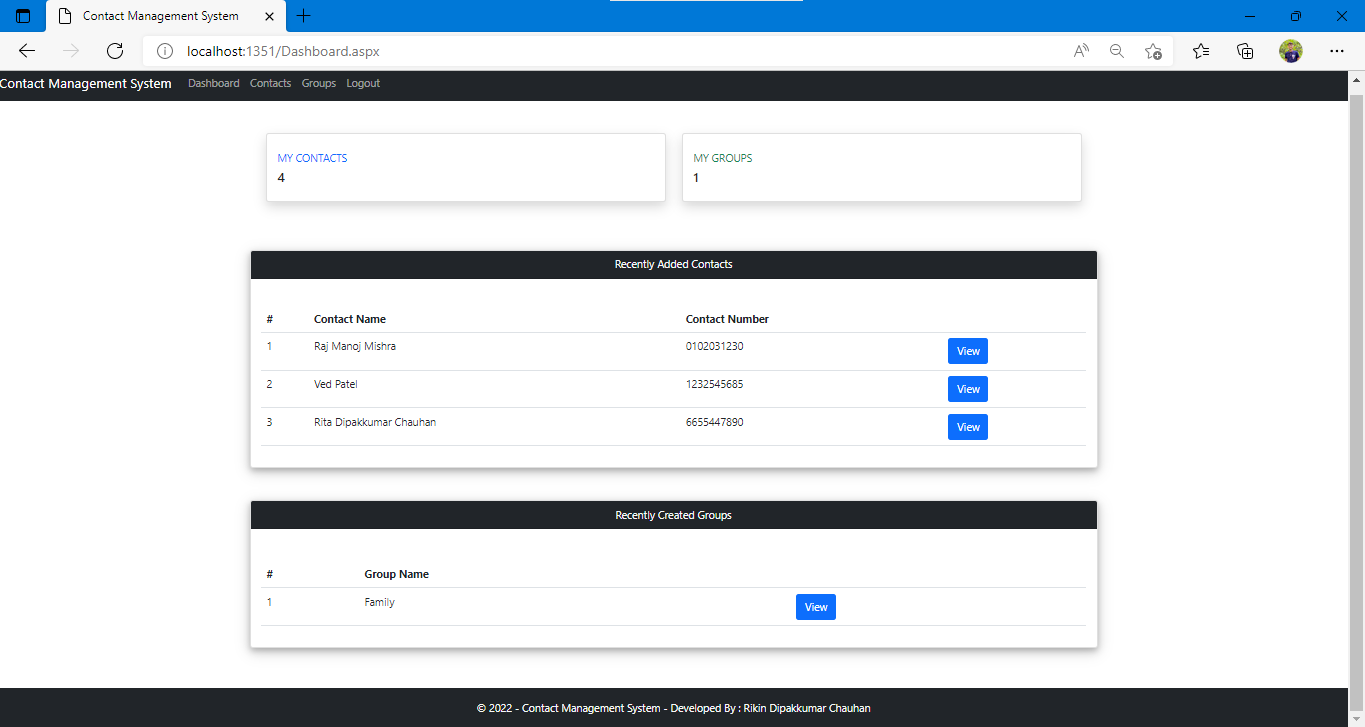
### Register



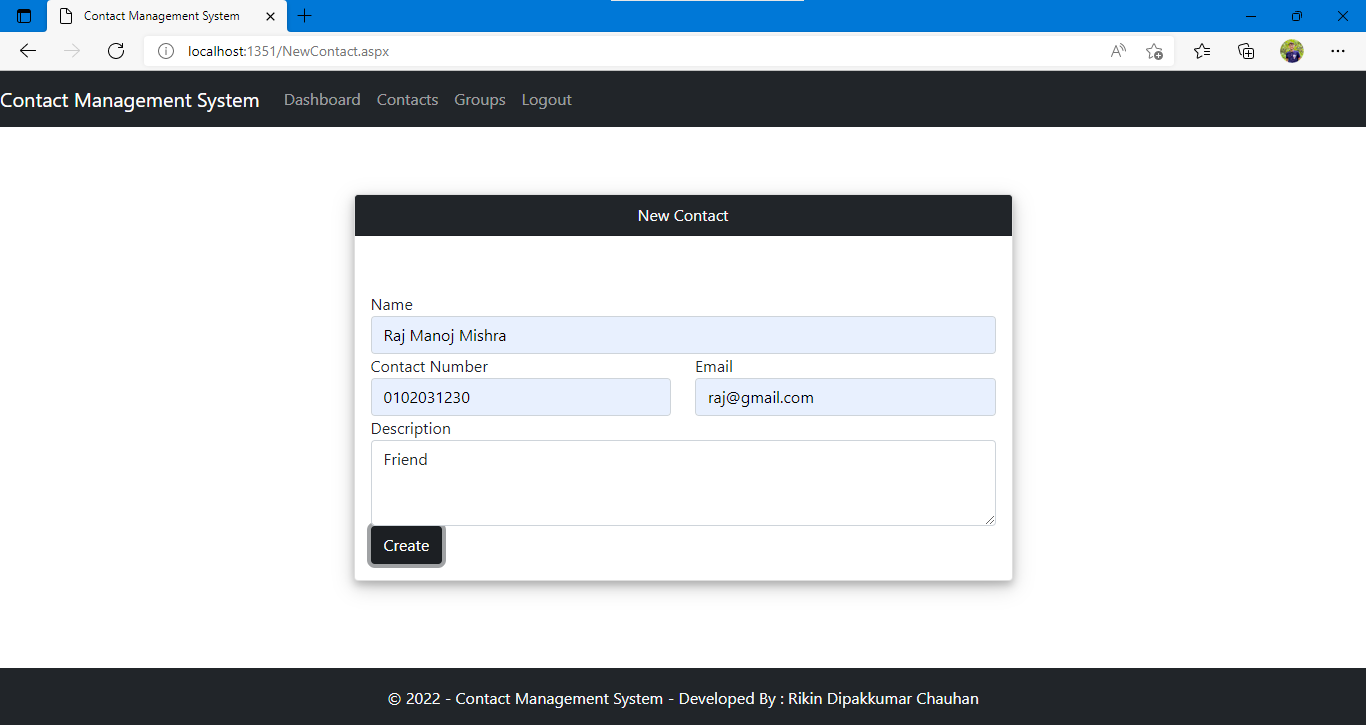
### Login



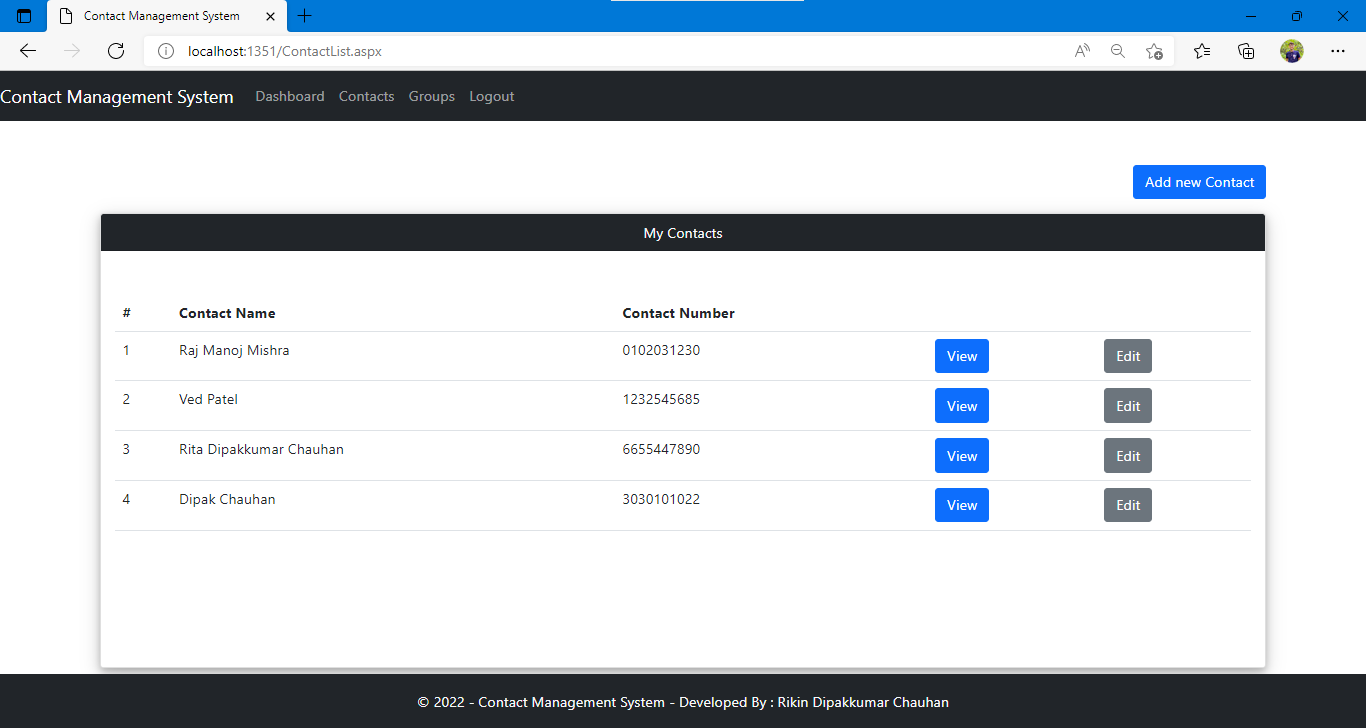
### Dashboard



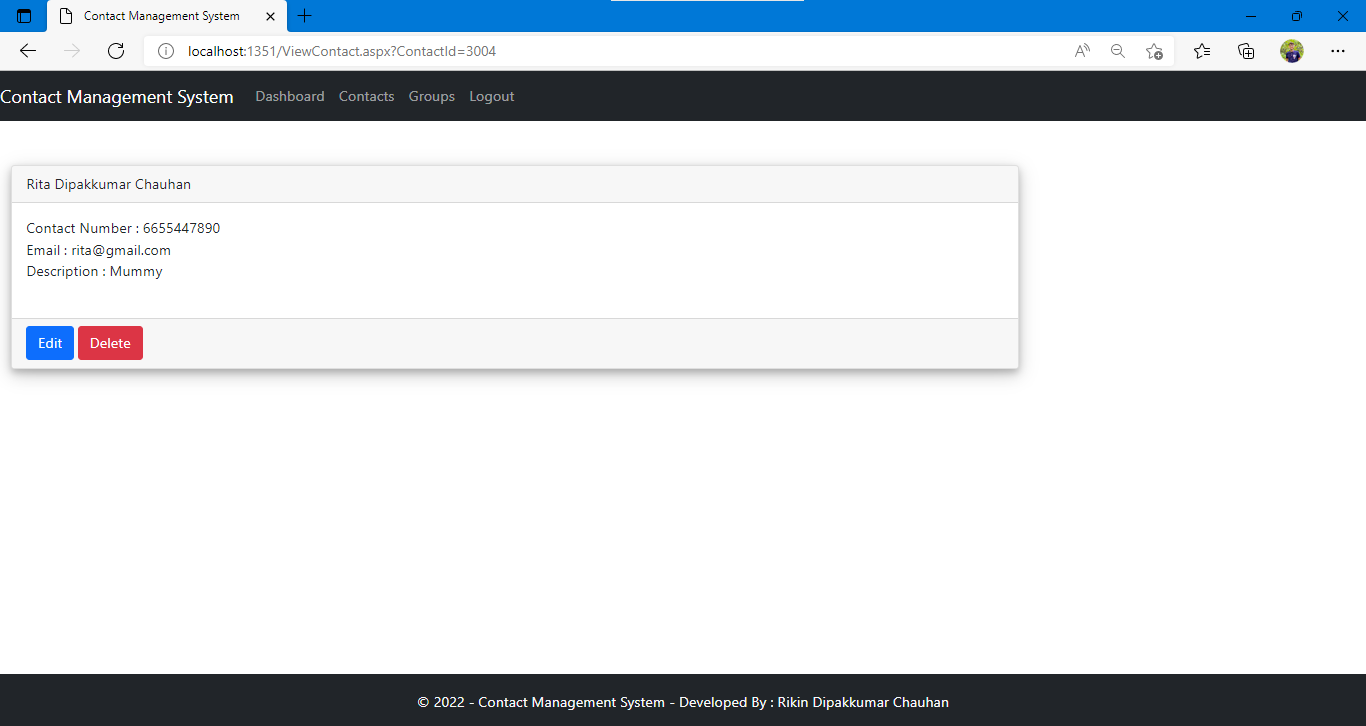
### Create New Contact



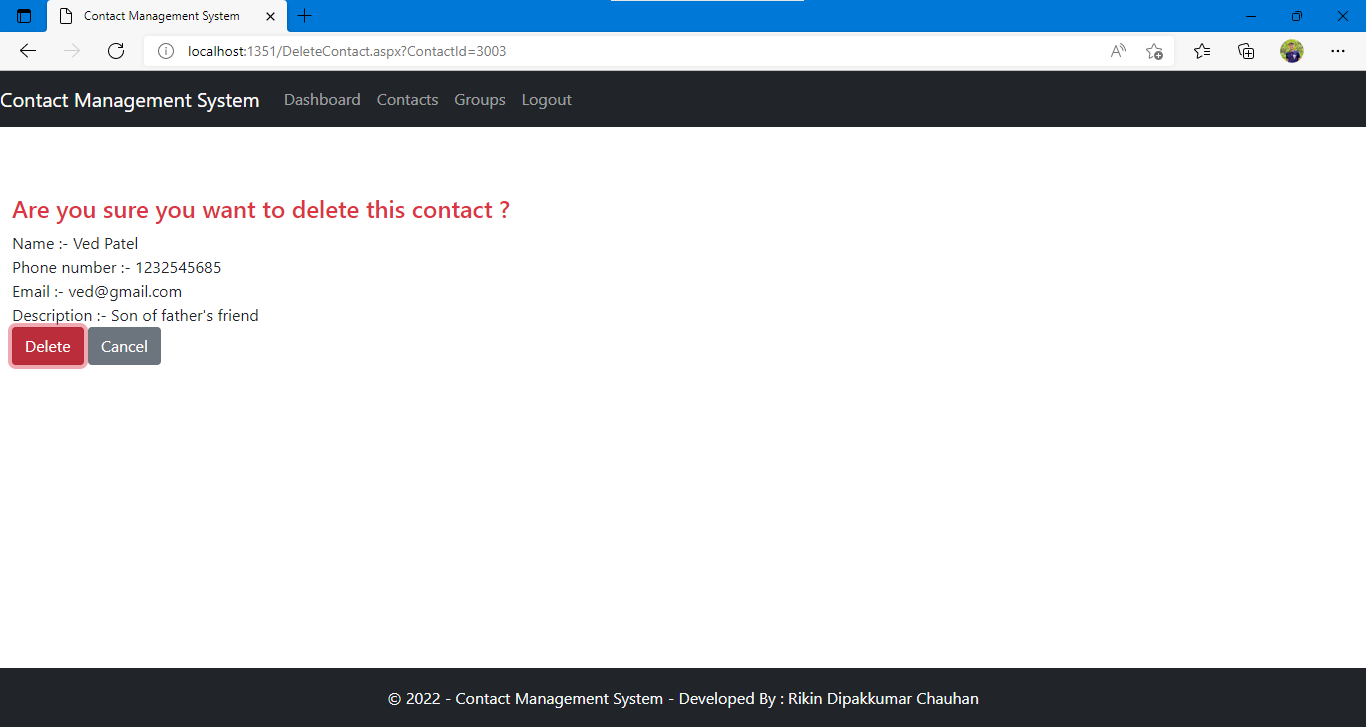
### Contact List

****

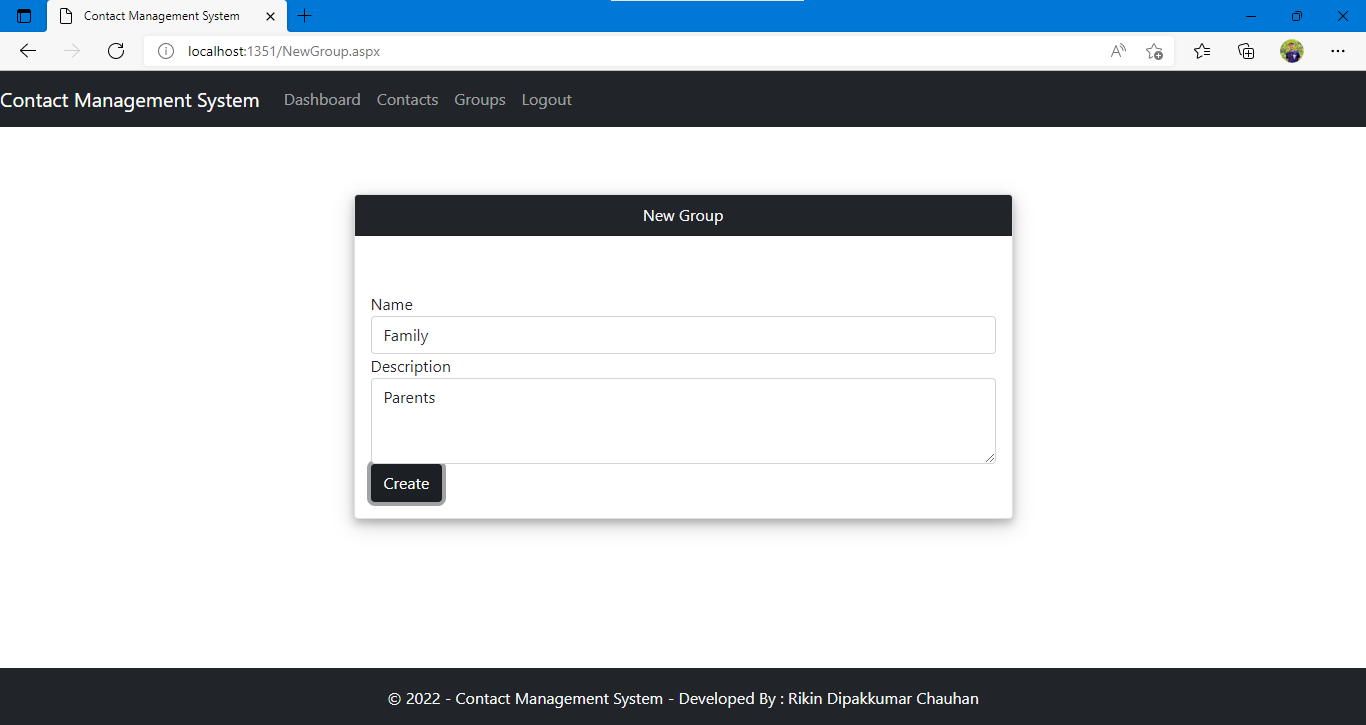
### View Contact

****

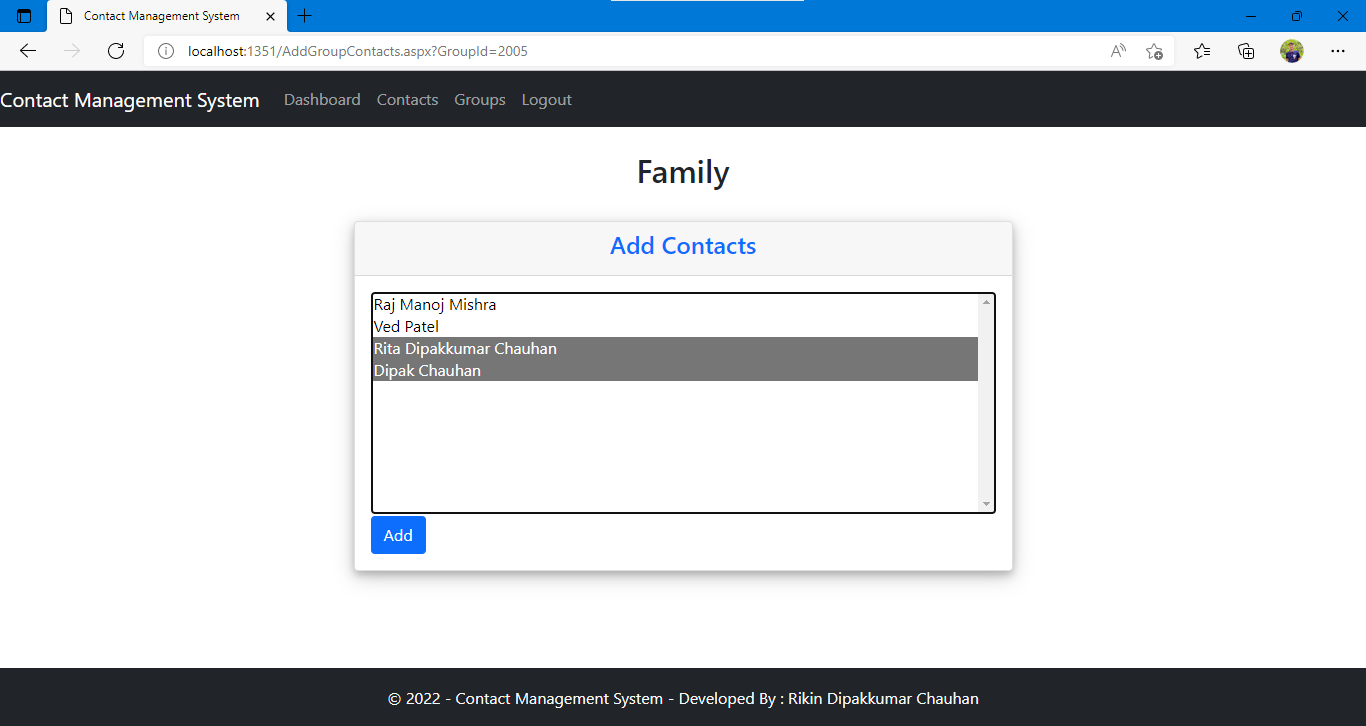
### Delete Contact

****

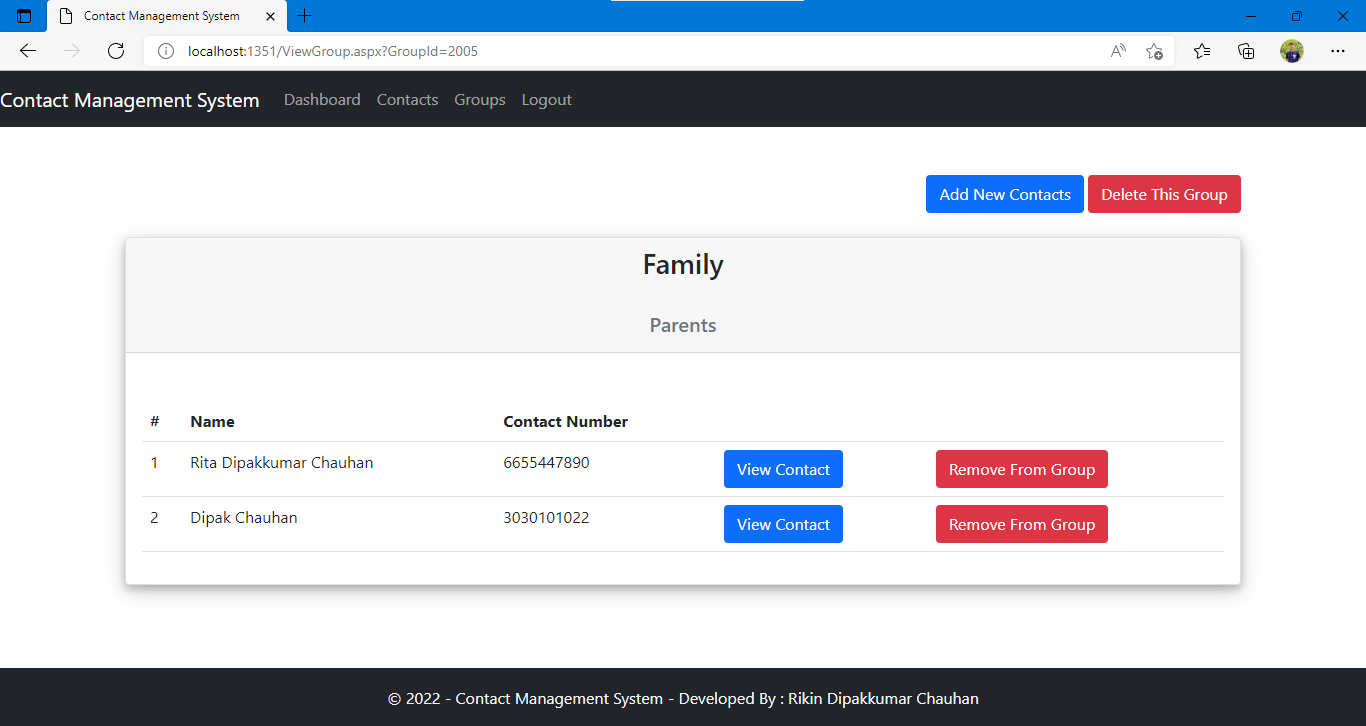
### Create New Group

****

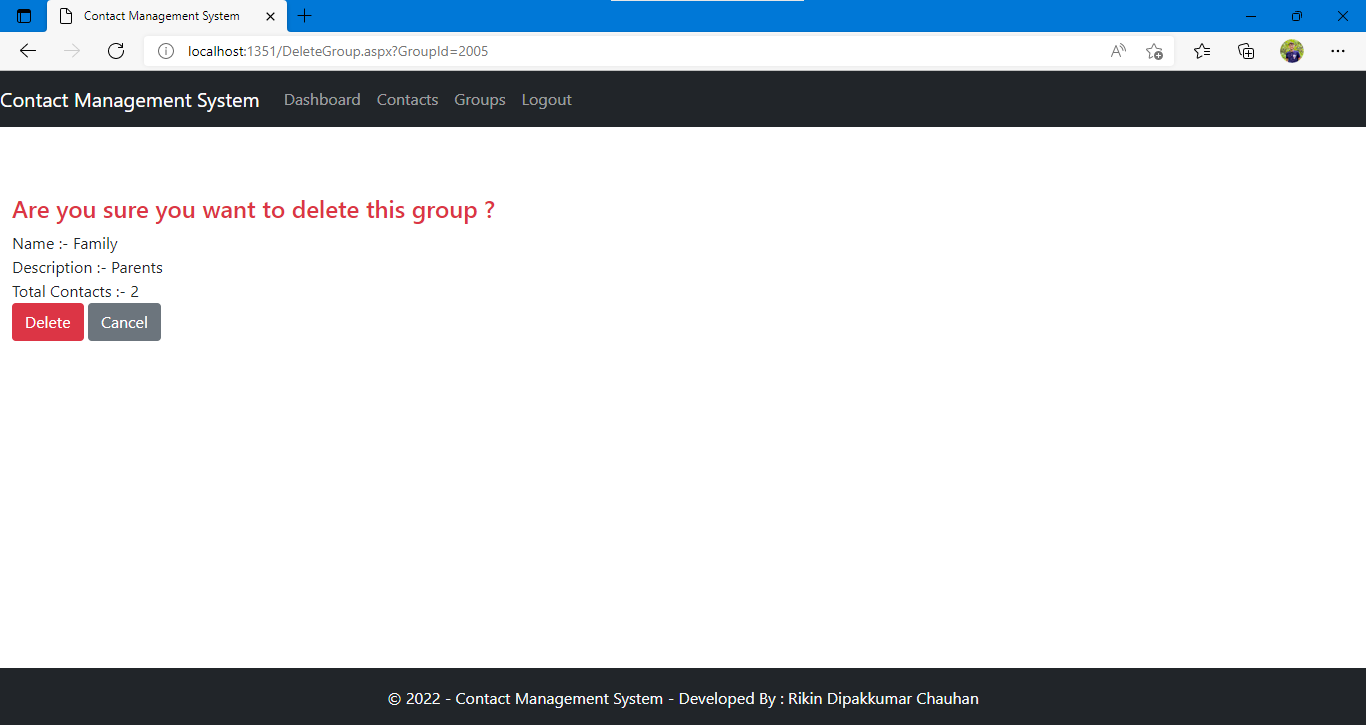
### Add Contacts to Group

****

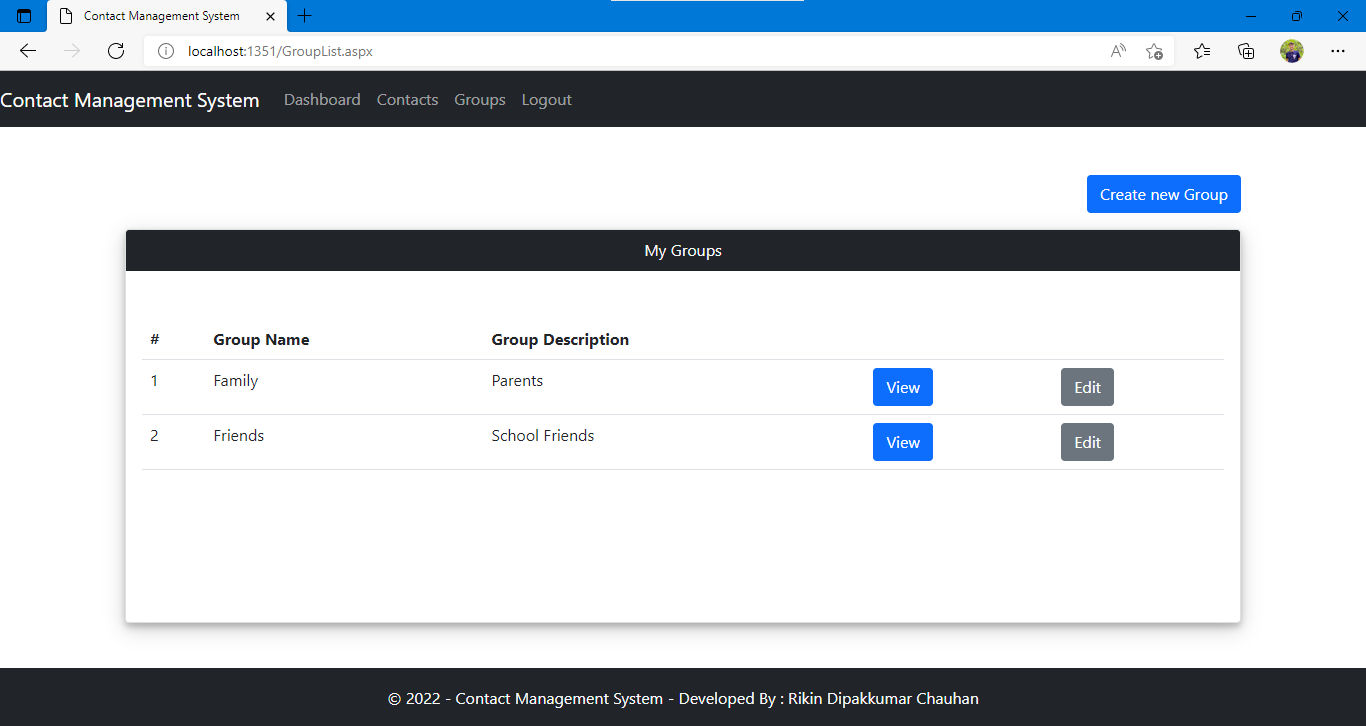
### View Group

****

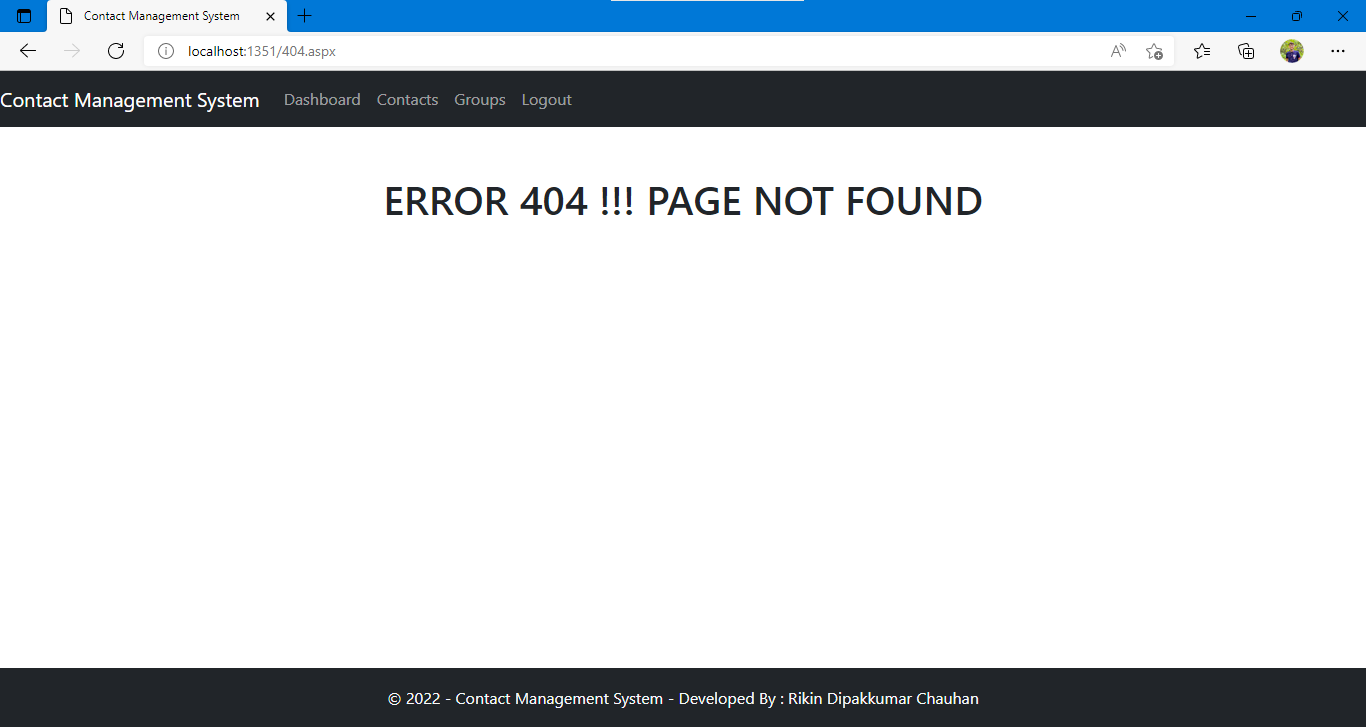
### Delete Group

****

### Group List

****

### Error ‘404’

****

# 8. Conclusion

The functionalities are implemented in system after understanding all the system modules according to the requirements. Functionalities that are successfully implemented in the system are:

* Register
* Login
* Logout
* Create Contact
* View Contact
* Update Contact
* Delete Contact
* View Recently Added Contacts
* Create Group
* View Group
* Update Group
* Delete Group
* View Recently Added Group
* Add Contacts to the Group
* Remove Contacts from the Group

After the implementation and coding of system, comprehensive testing was performed on the system to determine the errors and possible flaws in the system.

# 9. Limitations and Future Extensions

## 9.1 Limitations

* We are able to implement most of the functionality of all modules.
* We aim to complete all the functionality of all modules and make this product ready to be used practically in all scenarios.
* Currently, the project runs completely fine if all the inputs / selections are given within proper criteria but it doesn’t cover all the corner cases.
* User can’t login in case if the password is not known.

## 9.2 Future Extensions

* User interface will be improved to provide better interaction of user with the system.
* Forget Password functionality will be implemented.
* User Photo Upload functionality will be implemented.
* Update User Profile functionality will be implemented.
* Import User Contacts using .VCF File functionality will be implemented.
* Export User Contacts using .VCF File functionality will be implemented.
* The project can be extended to run robust.

# 10. Reference / Bibliography

Following links and websites were referred during the development of this project:

* [Windows Communication Foundation - WCF | Microsoft Docs](https://docs.microsoft.com/en-us/dotnet/framework/wcf/)
* [Create web APIs with ASP.NET Core | Microsoft Docs](https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-5.0)
* [Bootstrap · The most popular HTML, CSS, and JS library in the world. (getbootstrap.com)](https://getbootstrap.com/)
* [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)