PROJECT PROFILE	
➢ Project title	ShareCal
➢ Description	ShareCal is a web application that is helpful in storing and sharing events with friends and scheduling mails.
➢ Front end tool	PHP, HTML, CSS, Javascript, Jquery
➢ Back end tool	Mysql
➢ Platform utilized	Windows 10
➢ Project team size	1
➢ Prepared by	Dharmil K. Shah (1718034)
➢ Guide	Bhavesh Hariyani
➢ Submitted to	Shree Swaminarayan College of Computer Science, Gurukul Campus, Sardarnagar, Bhavnagar 364001

INTRODUCTION

1. INTRODUCTION

- The website 'ShareCal' is used to save and share your events and schedule mails.
- User will create an account through OTP and calendar will be provided.
- User can find friend and can have chat with them in personal or in group.
- User can also share events with friends or group of friends.

1.1 Background

Now day's people are having tight/busy schedule/days. It's common to forget to save or remember some important events or occasions. Easy way is to store/save your upcoming event details and information. This site will provide service to store and also share events with friends and family and we will remember and notify you upon your event occurrence. Apart from that user can also schedule their mails. This site will send that stored mail to the specified mail recipients on predefined time.

1.2 Objective

- Main objective is to provide better way to store and share events and emails to make scheduling easy and save time and work.
- User will have no difficulty to share their important events/occasions among friends and family with chatting facility.
- To reduce the work related to mails like send mail on particular time.
- Provide the way to share events and mails with chatting facility together.

1.3 Purpose and scope

- Existing system/apps will let you save events but you cannot share it with your contacts.
- User can see, update or delete events or scheduled mail anytime.
- In addition, you can also have conversation with your contacts and also can schedule mails to send them on particular time.
- User can also find their friends/contacts that are registered on ShareCal by searching through email or username.
- User can chat with friends and other users of the site.
- User can create group and add other friends and users of the site. Only group admin (creator) have the permission to add or remove user.
- In group, all members can chat and save group events and reminders of the group events will be sent to all members of group.
- If user don't want to receive reminder mails of group events then he/she can also mute the group reminders.
- User can leave group anytime he/she want.

1.4 Applicability

- User must have a valid e-mail address to be a part of ShareCal.
- No age restrictions are there on ShareCal.
- Anyone can use this system who is having difficulties in storing or sharing events and scheduling mails.

ShareCal

A Calendar for everyone

REQUREMENT AND ANALYSIS

2. REQUIREMENT AND ANALYSIS

2.1 Problem Definition

- Currently users having difficulties to save and share their important events or occasions.
- User cannot directly share saved events to another user or group of users.
- Also having difficulty to communicate via existing system/app.
- Having trouble to schedule mails.

2.2 Requirement Specification

- In ShareCal user can sign up and can get calendar to save events and will be notified on event occurrence.
- In ShareCal user can also find other users and have conversations with one or more users in private or group chat.
- Using ShareCal user can also get common calendar between/among one or more users. All authorised users can see and create events and information in common calendar.
- User can schedule mails by set particular date/time and recipient's mail address and ShareCal will send mail on pre-defined date/time to pre-defined users.(Not sure about this feature)

2.3 Planning and Scheduling

Planning	Duration
Analysis	15 days
Designing	20 days
Coding	20 days
Testing	10 days
Implementation	05 days
Reporting	10 days

2.4 Hardware and Software Requirement

- **Developer side**

Component	Description
Processor	64 bit Intel Celeron, 1.10GHz
Ram	4 GB
Hard Disk	1 TB
Operating System	Windows 10
Server	Xampp server(3.2.2)
Front-end	PHP, HTML, CSS, Javascript, Jquery
Back-end	MySQL
Editor	Brackets
Browser	Google Chrome

- **Client side**

Component	Description
Processor	32 bit Intel
Ram	1 GB
Hard disk	80 GB
Operating system	Windows 7 or higher
Browser	Google chrome or other

2.5 Tools and Technology

PHP :-

- ✓ PHP is a server side scripting language that is used for building dynamic, interactive web sites.

- ✓ Today, PHP has been a real success in many respects. PHP script run's on a web server. PHP stands for Hypertext pre-processor. PHP program's runs on web server, and serve Web pages to visitor on request.

- ✓ One of the key features of PHP that you can embed PHP code within HTML web pages, makings it's very easy for you to create dynamic web content quickly.

- ✓ PHP was created by Rasmus Lerdorf in 1994. PHP started out as a set of simple tools coded in the C language to replace the Perl script that Rasmus was using on his personal home page. He released PHP to the general public in 1995, and called it PHP version 2.

- ✓ In 1997, Zeev Suraski and Andi Gutmans along with Rasmus rewrite the PHP and released PHP version 3.0 in June 1998. After the released oh PHP version 3.0 PHP becomes so much popular.

- ✓ The PHP version 4.0 was launched in May 2000. This version includes session handling, output buffering, a richer core language, and support for a wider verity of web server platforms.

- ✓ The PHP version 5.0 was released in 2004 with object oriented programming (OOP) concept such as Private and Protected class member, final, private, protected, and static methods, abstract class, interface and standardized constructor/ destructor syntax.

HTML

- ✓ Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

- ✓ Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

- ✓ HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page.

Other tags such as <p> surround and provide information about document text and may include other tags as sub elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

- ✓ HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

JAVASCRIPT :-

- ✓ JavaScript is the scripting language of the Web. All modern HTML pages are using JavaScript. A scripting language is a lightweight programming language. JavaScript code can be inserted into any HTML page, and it can be executed by all types of web browsers. JavaScript is easy to learn.

Why to use JAVASCRIPT:

JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to specify the behavior of web pages

MYSQL :-

- ✓ MySQL is a database system used on the web

- ✓ MySQL is a database system that runs on a server
- ✓ MySQL is ideal for both small and large applications
- ✓ MySQL is very fast, reliable, and easy to use
- ✓ MySQL uses standard SQL
- ✓ MySQL compiles on a number of platforms
- ✓ MySQL is free to download and use
- ✓ MySQL is developed, distributed, and supported by Oracle Corporation
- ✓ MySQL is named after co-founder Monty Wideness's daughter:

The data in a MySQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Databases are useful for storing information categorically. A company. Many have a database with the following tables:

- ✓ Employees
- ✓ Products
- ✓ Customers
- ✓ Orders

Features of MYSQL?

1) Open Source Software:

- ✓ The MySQL package comes with the complete source code. This means that you may study the source code and modify it to suit your particular needs. You can find a high degree of support from third parties.
- ✓ So that is performed in open source software.

2) SQL Support

- ✓ As the "SQL" in MySQL suggests, MySQL supports SQL (Structured Query Language), a standard high –level language used to make queries for data required from a database. Although the SQL syntax is slightly different from one RDBMS to another. The underlying concepts are much the same. MySQL users its own set of SQL commands but the basic concept still survivors.

3) Superb Performance and Reliability:

- ✓ MySQL is remarkably fast and reliable even in a most demanding environment.so comparatively it is fast.

4) Free Support:

- ✓ You can get a certain degree of professional support from users who use MySQL by various means; MySQL newsgroups; mailing lists; independent websites that freely share their knowledge; among others; if you need a higher level of technical support, you can always ask for it from the developers at minimal cost.

5) Cross platform:

It is runs on a number of different platforms

JQUERY :-

- ✓ jQuery is a lightweight, "write less, do more", JavaScript library.
- ✓ The purpose of jQuery is to make it much easier to use JavaScript on your website.
- ✓ jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- ✓ jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- ✓ The jQuery library contains the following features:
  - ➢ HTML/DOM manipulation
  - ➢ CSS manipulation
  - ➢ HTML event methods
  - ➢ Effects and animations
  - ➢ AJAX
  - ➢ Utilities

2.6 Feasibility study

Operation of the proposed system depend on its various users. This various type of users mentioned below.

1) Admin
2) User

User can register or login and store events, find friends, create groups, schedule mails etc. Admin can check if mail sending script is running in server or not, admin can also execute that script.

- Technical feasibility:

  ShareCal is a web application which avoid time consuming and forgettable methods to remember and share useful events or dates. User can register or login and store events, find friends, create groups and group events, chat with other users and chat in group, schedule mails etc.

- Social feasibility:

  Proposed of system would be acceptable by the other people because this system avoid below issues.

  1. External system requires more efforts.
  2. Not accurate.
  3. Reminders on time.

- Economical feasibility:

  In offline system first you have to save events in diary/note or in your phones. This is okay, ShareCal provides you to share that events with you contacts or with group of people. Now a days people have their personal email addresses so no need for any extra effort.

2.7 Existing System Scenario

- Time consuming process to note down in diary or notes.
- Sometimes we forget some events after all we are humans and not robots.
- Difficult to share stored events with friends or with group of people.

2.8 Proposed of system

In online booking system of home serviceman there are three roles:

- User
- Admin

2.8.1 User

Store Events

User can store events with details.

Friends

User can find their friends registered on site.

Chatting

User can chat with friends.

Groups

User can create groups and add members and add group events and chat in group.

Mails

User can schedule mails.

2.8.2 Admin

Admin can do all things that user can do on site. Additionally admin can check status of mail sending script and handle that.

2.9 Methodology:

To develop this type of event sharing system there are following possible ways.

1) Desktop application.
2) Mobile application
3) Web base application.

- Develop as web application.

We have provides web-based application that can access from anywhere throughout world and access from any devices like computer, laptop or mobile.

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Data Dictionary

- A Data Dictionary is catalog a repository of the elements in a system. The major elements are data flows, data stores, and processes.
- The Data Dictionary stores details descriptions of these elements.
- If analysts want to know many characters are in a data item, by what other names it is referenced in the system, or where it is used in the system, they should be able to find answers in a properly developed Data Dictionary.
- However, its contents are used during systems design as well.

Why Data Dictionary is important?

- To communicate a common meaning for all system elements.
- To document the features of the system.
- To facilitate analysis of the details on order to evaluate characteristics and determine where system changes should be made.
- There is a style of data dictionaries known as a middleware data dictionary.
- Middleware is computer software that connects software components or applications.
- The software consists of a set of services that allows multiple processes running on one or more machines to interact.
- Traditional data dictionaries provide structure and basic function to the database. Middleware data dictionaries are located within the DBMS itself and operate on a higher level. Middleware data dictionaries can provide alternate entity relationship structures that can be tailored to fit different users that interact with the same database.
- Middleware data dictionaries can also assist in query optimization as well as distributed database.

- Middleware also helps database designer by reducing the amount of time it takes to create form, queries, reports, menu and many other database components. They do this by automatically generated SQL code for common item such as forms and views.
- Some middle wear database dictionary can also help with data security as well as database optimization.

TABLE 1 : This table contains basic details about all users.

main_table					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	id	Integer	4	Primary key	This field contain Id of users which assign by auto increment.
2	u_name	Varchar	34	--	User name of user which is displayed to other users.
3	password	Varchar	34	--	Password of user.
4	mail	Varchar	48	--	Email-id of all users by which user can login or create their account. One account per mail.

TABLE 2 : This table is used to store the personal chat between 2 particular users(personal chat table).

chat					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	mid	Integer	6	Primary key	Each message will be assigned a unique id by auto-increment.
2	id	Integer	4	Foreign key	Id of user who has sent the message. References main_table (id) field.
3	rid	Integer	4	Foreign key	Id of user who has received the message. References main_table (id) field.
4	msg	Text	--	--	It stores the text of the message.
5	mtime	Datetime	--	--	It stores the message sending time. Default value is CURRENT_TIMESTAMP

TABLE 3 : This table contains details about all events that are stored by the users.

events					
SR. No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	id	Integer	4	Foreign key	This field contains id of user which added the event. References main_table (id).
2	event_id	Integer	5	Primary key	This field contains id of event which is assigned by auto increment.
3	event_subject	Varchar	25	--	Subject of an event. Default value is 'No Subject'
4	event_description	Varchar	256	--	Stores the details/text of an event that is stored by user.
5	event_date_time	Datetime	--	--	Stores the date-time at which the event will occur and reminder mail will be sent.
6	event_add_date	Datetime	--	--	This will stores the date-time when the event is stored. Default value is CURRENT_TIMESTAMP

TABLE 4 : This table is used to store group details like members , permissions , mute event reminders etc.

group_details					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	group_name	Varchar	36	--	This field will store the Group name.
2	Id	Integer	4	Foreign key	This will store group member(user) id. References main_table(id).
3	permission	Integer	1	--	This is used to know group member privileges. Like add, delete member etc.
4	notify	Varchar	3	--	Used to know that user is interested in group event reminders or not(yes/no).
5	time	Datetime	--	--	It will store the time when the user is added.

TABLE 5 : This table is used to store group chat of group members.

group_chat					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	Id	Integer	4	Foreign key	Id of user who has sent the message. References main_table (id).
2	mid	Integer	5	Primary key	Each message will be assigned a unique id by auto-increment.
3	group_name	Varchar	36	--	This field contains group name in which the message is sent.
4	msg	Text	--	--	This field contains message that is sent by a user into the group.
5	mtime	Datetime	--	--	It stores the message sending time. Default value is CURRENT_TIMESTAMP.

TABLE 6 : This table contains details about all events that are stored by the users.

group_events					
SR. No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**

1	group_name	Varchar	36	--	This field contains group name in which the event is stored.
2	id	Integer	4	Foreign key	This field contains id of user which added the event. References main_table (id).
3	event_id	Integer	5	Primary key	This field contains id of event which is assigned by auto increment.
4	event_subject	Varchar	25	--	Subject of an event. Default value is 'No Subject'
5	event_description	Varchar	256	--	Stores the details/text of an event that is stored by user.
6	event_date_time	Datetime	--	--	Stores the date-time at which the event will occur and reminder mail will be sent.
7	event_add_date	Datetime	--	--	This will stores the date-time when the event is stored. Default value is CURRENT_TIMESTAMP

TABLE 7 : This table contains details of mails stored by users to send on particular time.

mail_queue					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	mail_id	Integer	4	Primary key	It is used to store the unique mail id by auto increment for mail stored in queue.
2	Id	Integer	4	Foreign key	It stores the id of the user. References main_table(id).
3	mail_subject	Varchar	100	--	It stores the subject of the mail.
4	mail_text	Text	--	--	It stores the content of the mail.
5	mail_send_datetime	Datetime	--	--	It stores date and time for sending mail.
6	mail_stored_datetime	Datetime	--	--	It stores date and time of storing mail. Default value is current timestamp.
7	mail_recipient	Varchar	48	--	It stores the mail address of recipient.
8	status	Integer	1	--	It shows that if mail is sent or not. Default value is 0.

TABLE 8 : This table contains only one row and one column. It will store latest time at which the mail sending script was executed through which mails and event reminders will be sent automatically through active server. It is useful for admin to know that script is running in background or not, if not then run script.

mail_send_time					
SR No	**Field Name**	**Data type**	**Size**	**Constraint**	**Description**
1	time	Datetime	--	--	It will store the time at which mail sending script was executed.

3.2 Data Flow Diagram

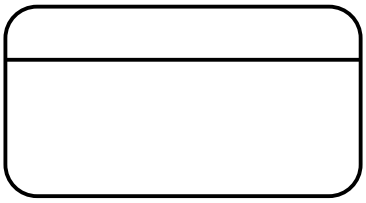
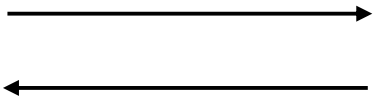
- A picture is worth a thousand words. A Data Flow Diagram (DFD) is traditional visual representation of the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or combination of both.
- It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.
- It is usually beginning with a context diagram as the level 0 of DFD diagram, a simple representation of the whole system. To elaborate further from that, we drill down to a level 1 diagram with lower level functions decomposed from the major functions of the system. This could continue to evolve to become a level 2 diagram when further analysis is required. Progression to level 3, 4 and so on is possible but anything beyond level 3 is not very common. Please bear in mind that the level of details for decomposing particular functions really depending on the complexity that functions.

Rules

- Each process must have a minimum of one data flow going into it and one data
- Each data store must have at least one data flow going into it and one data flow leaving it.
- A data flow out of a process should have some relevance to one or more of the data flows into a process.
- Data stored in a system must go through a process.
- Filing systems within an organization cannot logically communicate with one another unless there is a process involved.
- All processes in DFD must be linked to either another process or a data store.

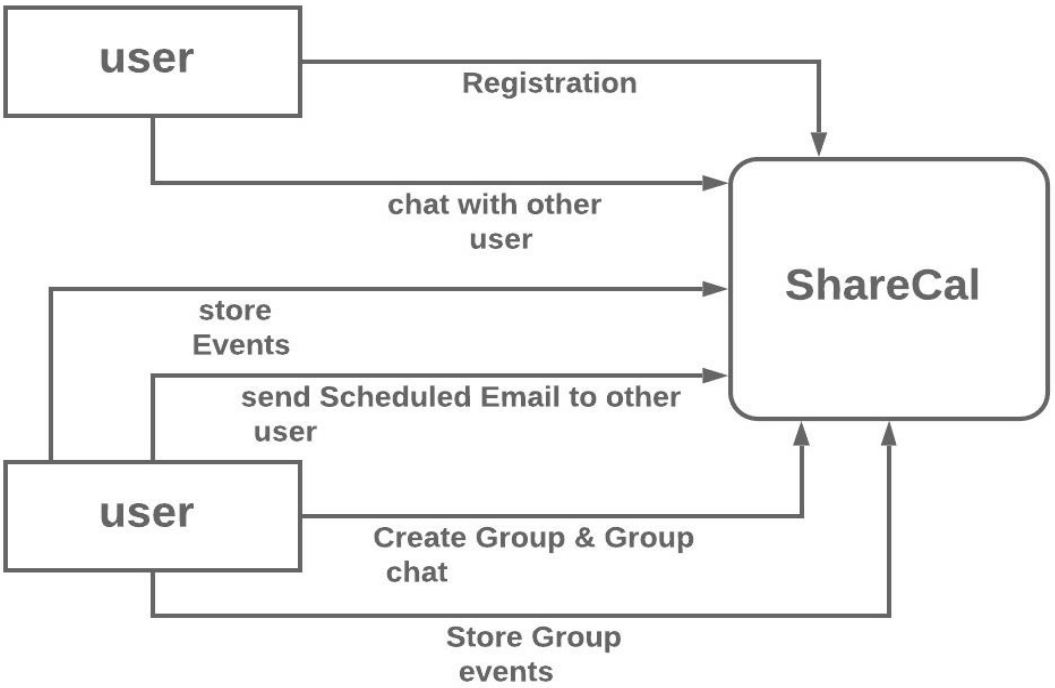
We usually begin withdrawing a context diagram, a simple representation of the whole system.

Following are the basic notations used to create data flow diagram:

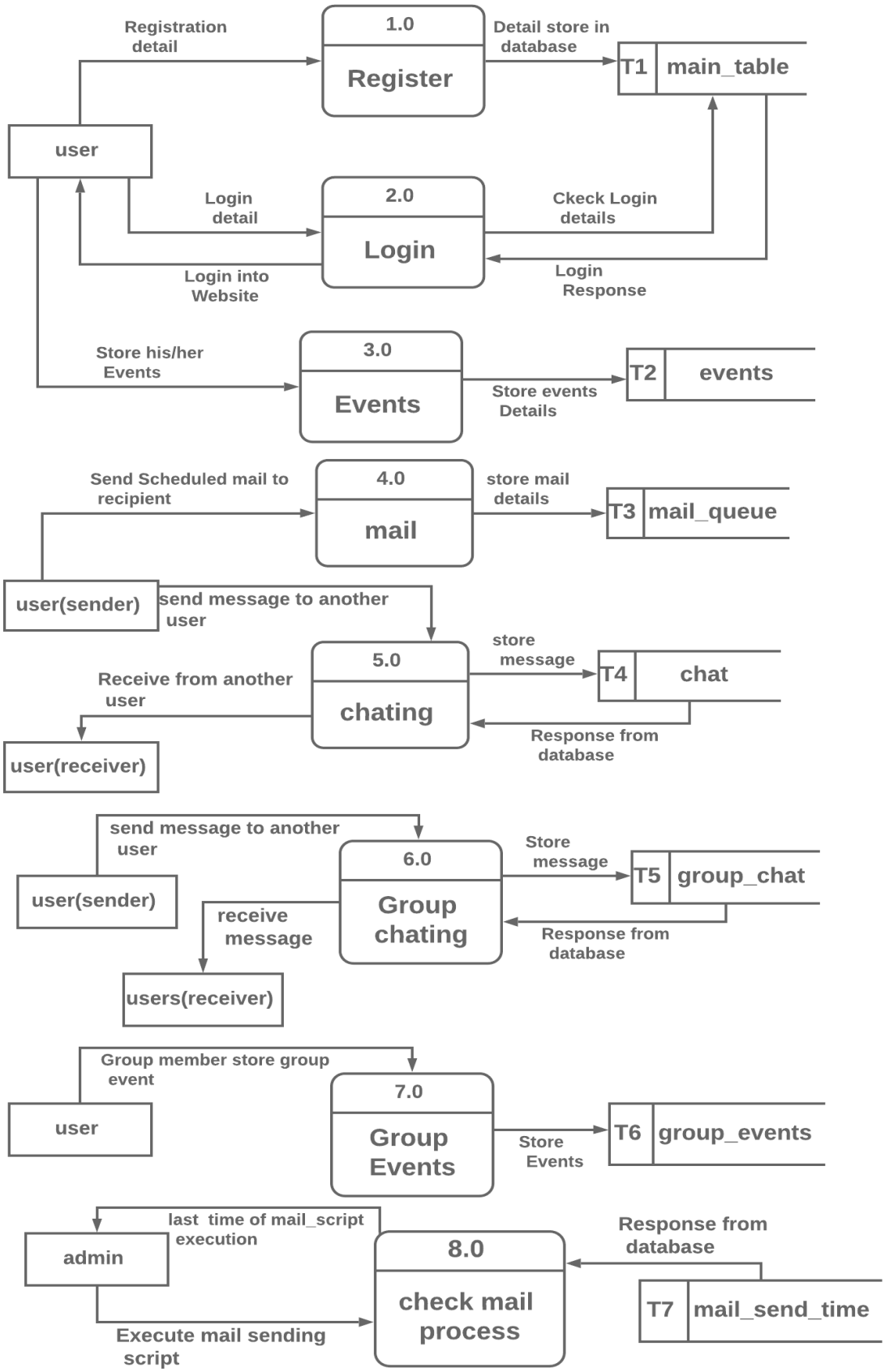
Symbol	Name
	External Entity
	Process
	Data Flow
	Data store

- An external entity, which are also known as terminators, sources, sinks, or actors, are an outside system or process that sends or receives data to and from the diagrammed system. They're either the sources or destinations of information, so they're usually placed on the diagram's edges.
- Process is a procedure that manipulates the data and its flow by taking incoming data, changing it, and producing an output with it. A process can do this by performing computations and using logic to sort the data or change its flow of direction. Processes usually start from the top left of the DFD and finish on the bottom right of the diagram.
- Data stores hold information for later use, like a file of documents that's waiting to be processed. Data inputs flow through a process and then through a data store while data outputs flow out of a data store and then through a process.
- Data flow is the path the system's information takes from external entities through processes and data stores. With arrows and succinct labels, the DFD can show you the direction of the data flow.
- But before you start mapping out data flow diagrams, you need to follow four rules of thumb to create a valid DFD.

Context Level DFD of ShareCal



1st Level DFD of ShareCal



3.3 Entity Relationship Diagram (ERD) ?

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is an object, a component of data. An entity set is a collection of similar entities. These entities can have attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, an ER diagram illustrates the logical structure of databases. ER diagrams are used to sketch out the design of a database.

Documenting an Existing Database Using Data

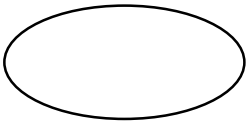
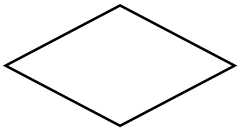
There are two reasons to create a database diagram. You're either designing a new schema or you need to document your existing structure. If you have an existing database you need to document, you can create a database diagram using data directly from your database. You can export your database structure as a CSV file (there are some scripts on how to this here), then have a program generate the ERD automatically. This will be the most accurate portrait of your database and will require no drawing on your part.

The History of Entity Relationship Diagrams

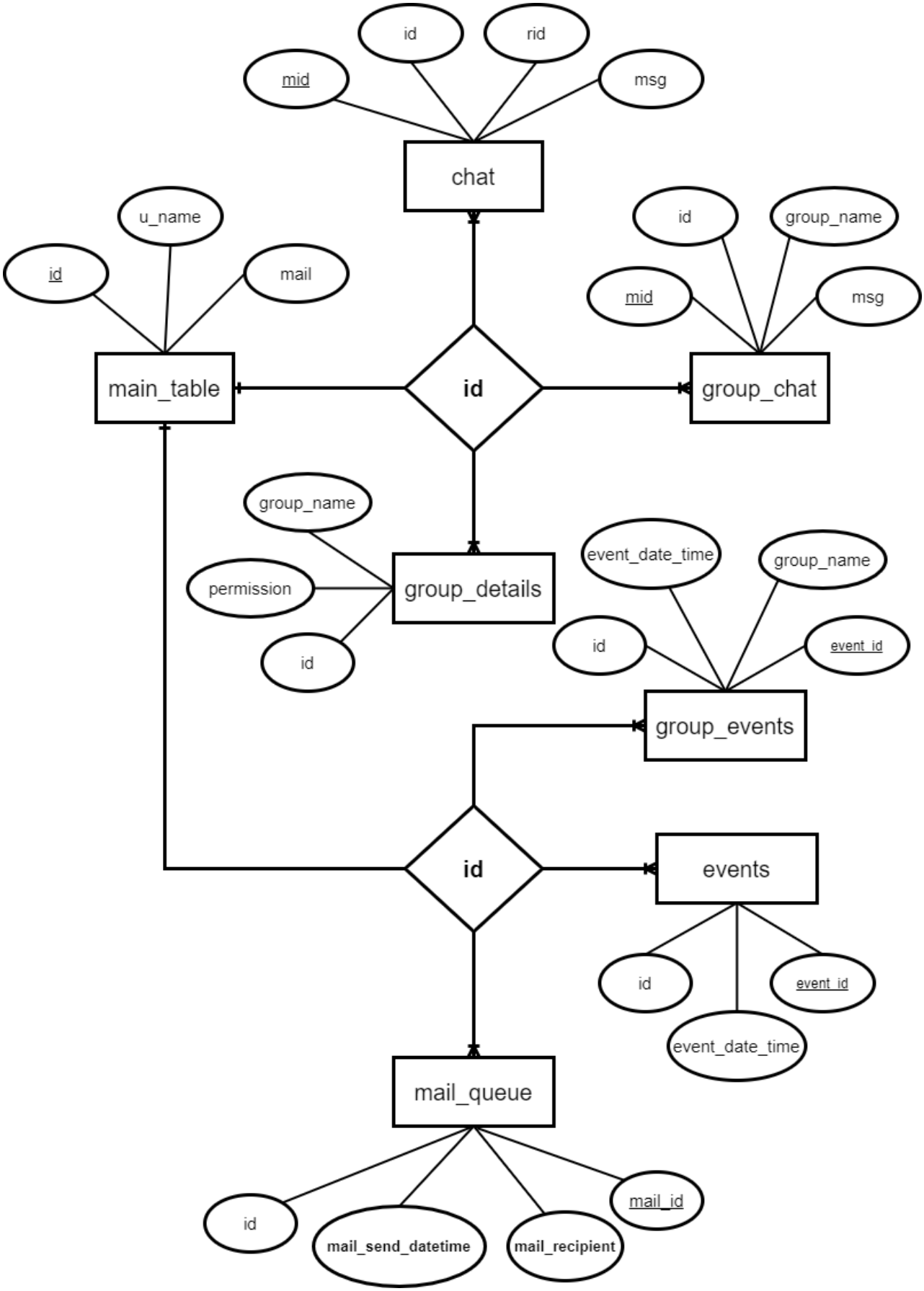
Peter Chen developed ERDs in 1976. Since then James Martin have added some slight refinements to the basic ERD principles.

Tips for Effective ER Diagrams

- Make sure that each entity only appears once per diagram.
- Entity, relationship, and attribute on your diagram.
- Examine relationships between entities closely. Are they necessary? Are there any relationships missing? Eliminate any redundant relationships. Don't connect relationships to each other.
- Use colours to highlight important portions of your diagram

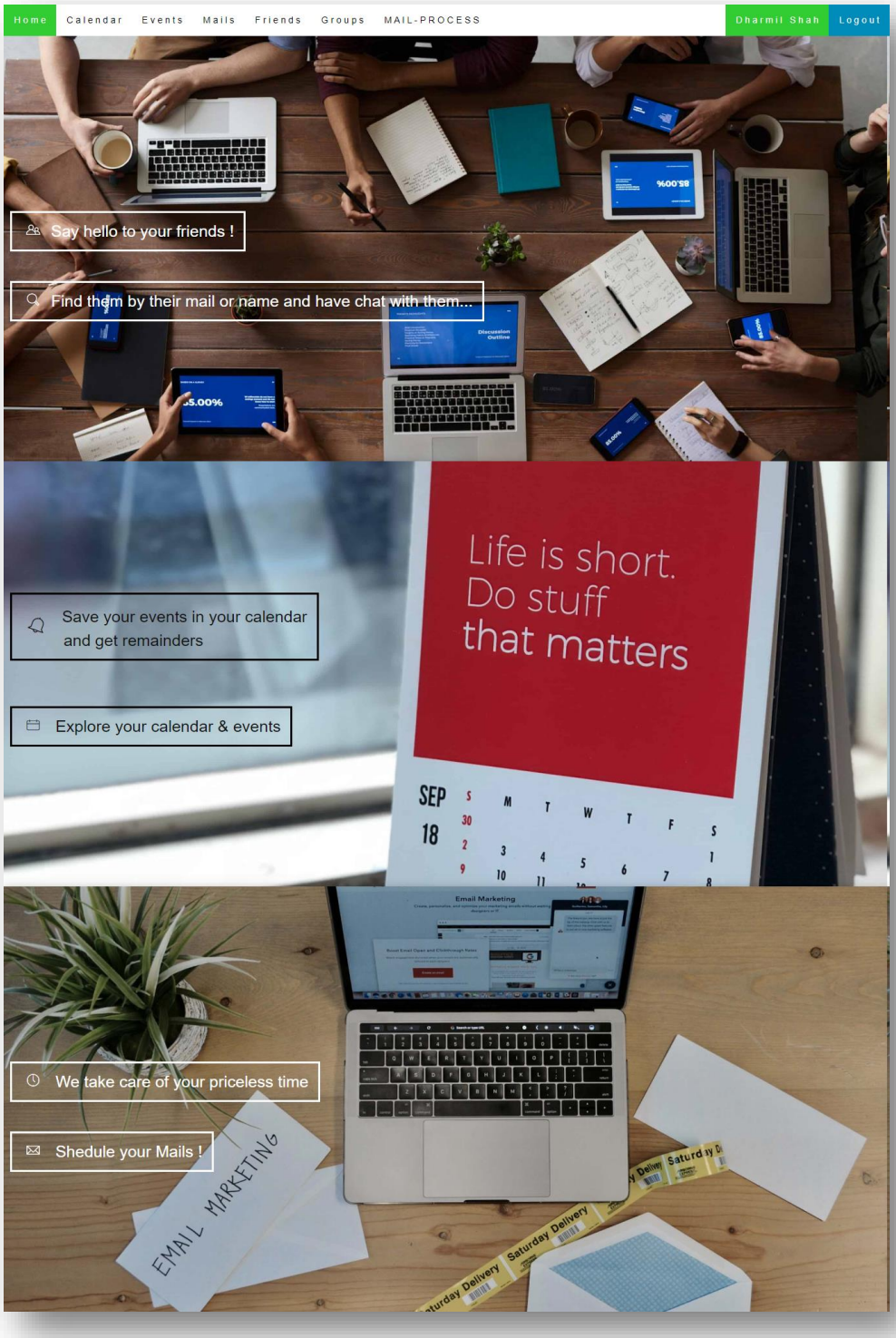
Name	Symbol	Meaning
Rectangle		Representing Entity Set
Oval		Representing Attributes
Diamond		Representing relationships among entity sets
Line		Links attributes to entity sets And entity sets to relationship

ER Diagram


id
rid
mid
msg
chat
u_name
id
mail
main_table
id
id
group_name
mid
msg
group_chat
group_name
permission
id
group_details
event_date_time
group_name
id
event_id
group_events
id
events
id
event_id
event_date_time
mail_queue
id
mail_send_datetime
mail_recipient
mail_id

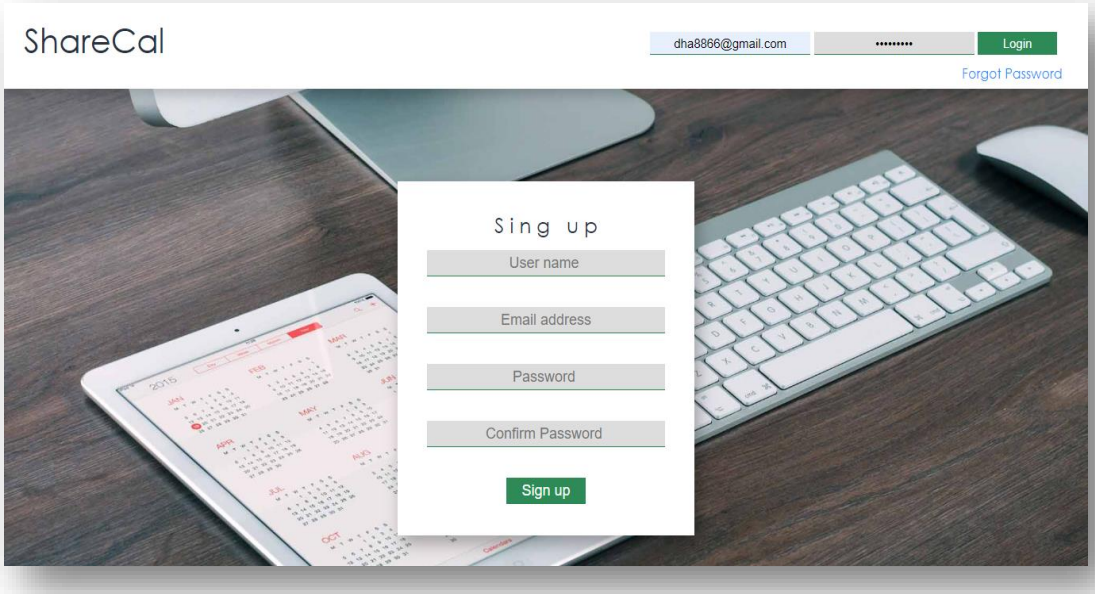
3.4 Screen shots

Home page



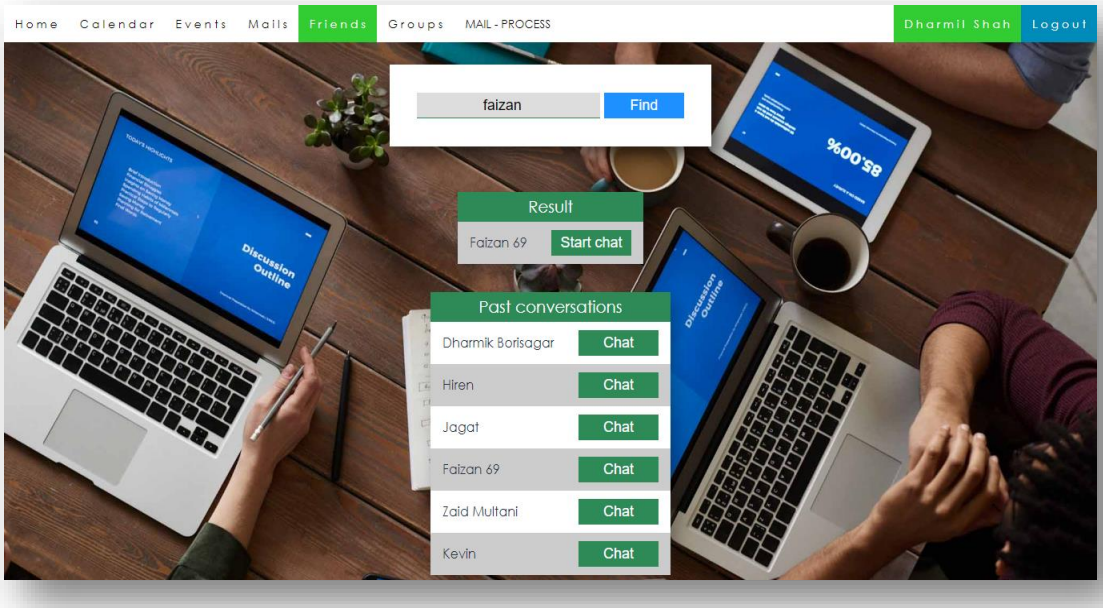
- Home page simply contains brief overview of some main features of the site.
- Like finding friends and chat with them, event storing and sharing and mail scheduling etc.

Login / Signup page



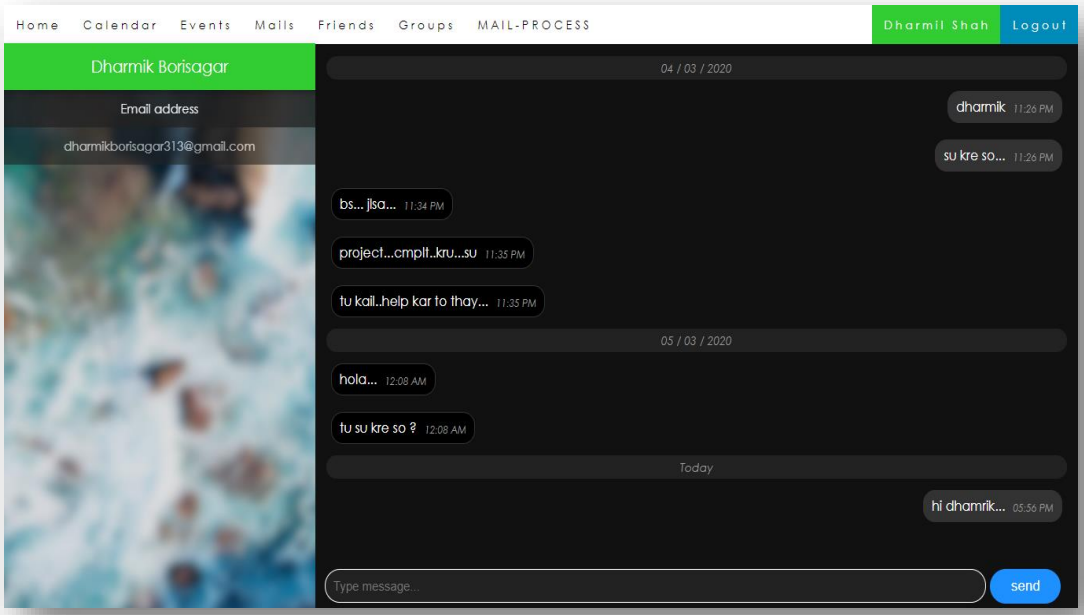
- User can login through this page if he/she is registered.
- If not registered then user can easily signup in the site by filling up signup form and OTP confirmation through mail he/she entered.
- User can also change password if password is forgotten. By clicking on Forgot Password link under login button.
- If user enters invalid details in login or signup form then appropriate message will be displayed.

Find friends



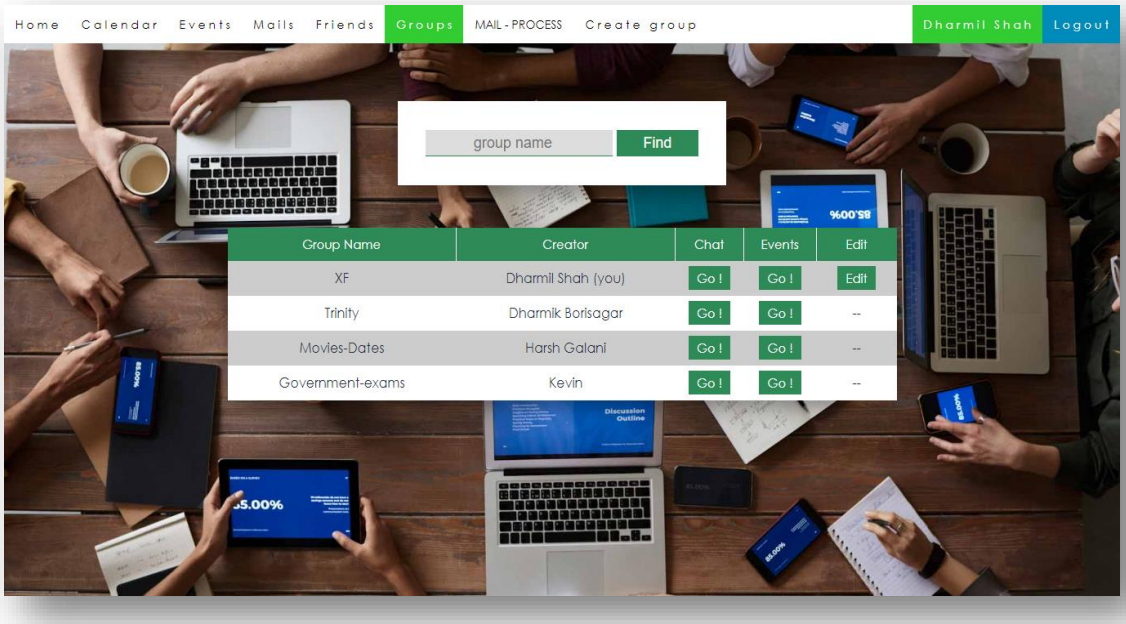
- Here user can find other users registered on site.
- User can enter username or proper email address of other users for searching and the result will be displayed.
- User will also get list of users with whom he/she had conversations in past. So no need to find friend every time.
- There is a chat button besides every user. By clicking there, chat module/page will be opened and user can chat with selected user.

Chat



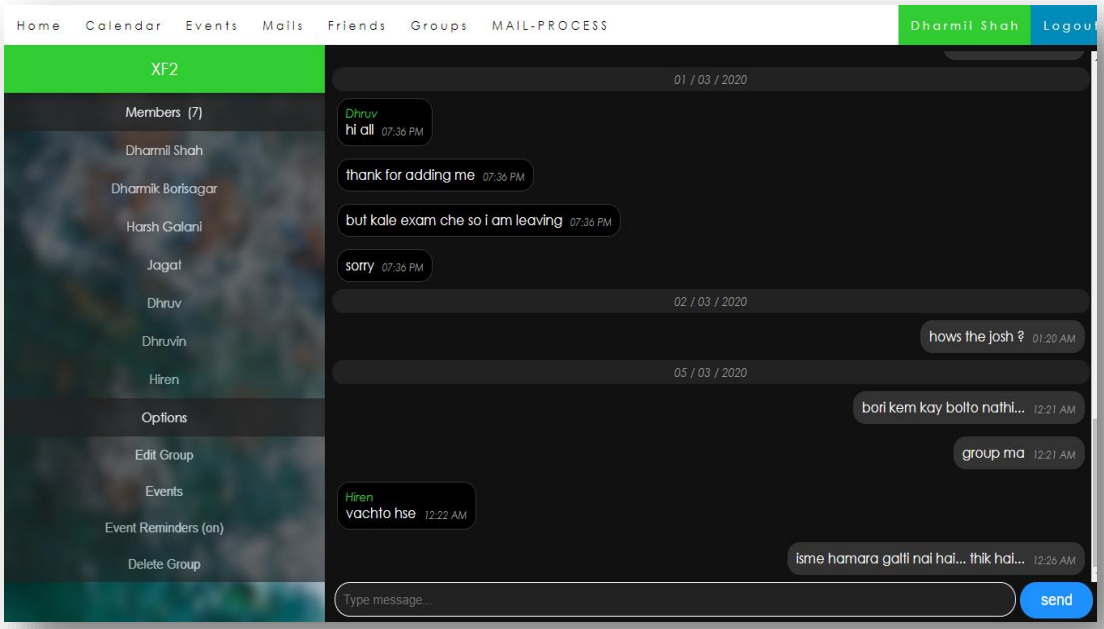
- Username of other user will be displayed in left panel with his/her email-address.
- Chat is displayed with message and message time in top to bottom method and it will be scrolled automatically to the bottom till last message.
- Chatting division will be refreshed automatically in 5 second's time intervals. So no need to refresh every time for new messages.

Groups



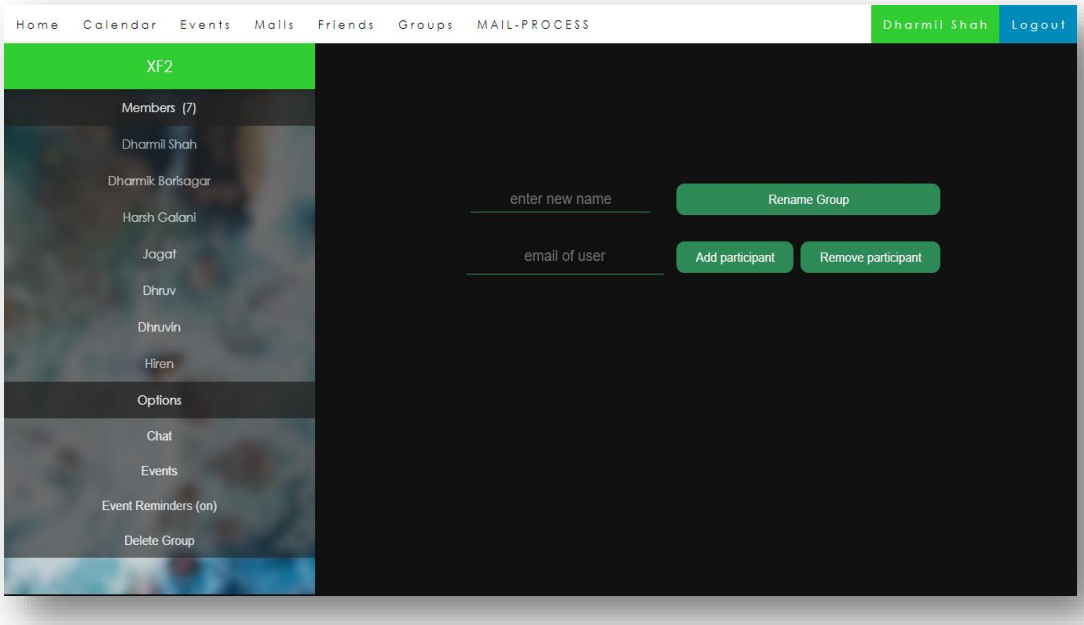
- In this page user will get the list of all groups in which he/she is a participant or an admin.
- Table includes Group name, Name of group admin (creator) and 3 options/buttons.
- Clicking on chat button will redirect user to the group chat.
- Events button will redirect user to the group event's page.
- Edit button will redirect user to group edit page.
- If user is admin of that group then 'Edit' button will be displayed to him/her.

Group Chat



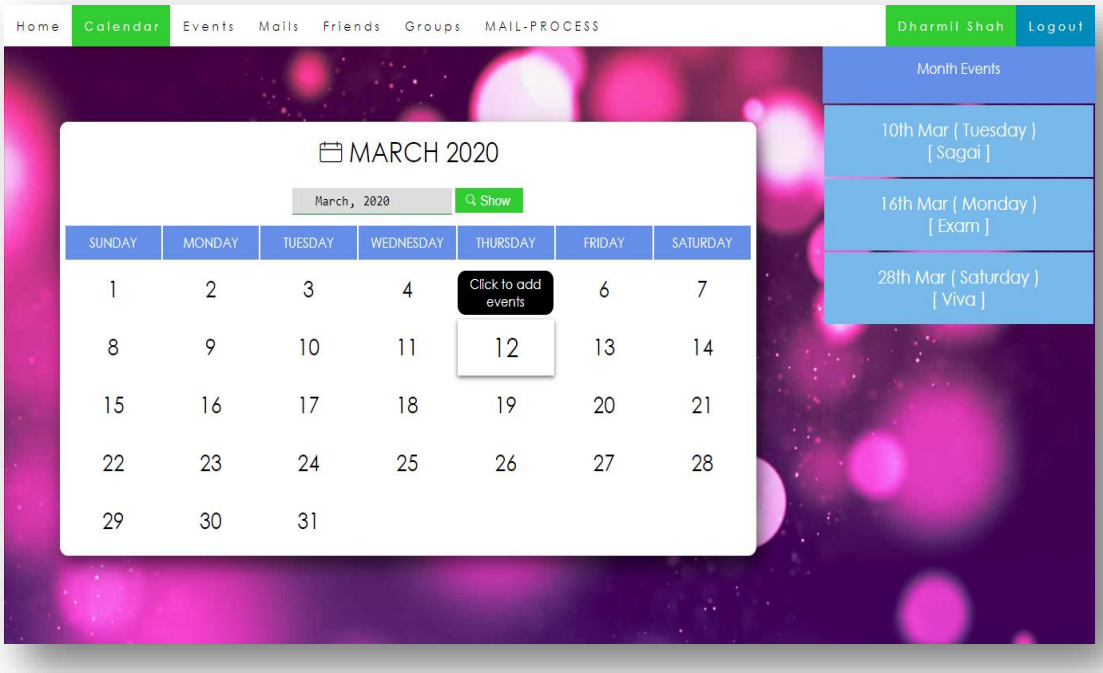
- Like chat module it is same with messages, additionally the username of the message sender will be displayed on the top of the message.
- The left panel includes user name of all group participants/members.
- You can see their mail address by hover on their username.
- Below that, there are some settings/options related to group. Like edit group will redirect user to group edit page, events will redirect user to group event's page, event reminder (on/off) will turn-on or turn-off group event's reminders.
- Delete group will be displayed to the admin and by clicking that whole data of group will be removed from DB.
- Normal member will see 'Leave Group' option instead of delete group.

Edit group



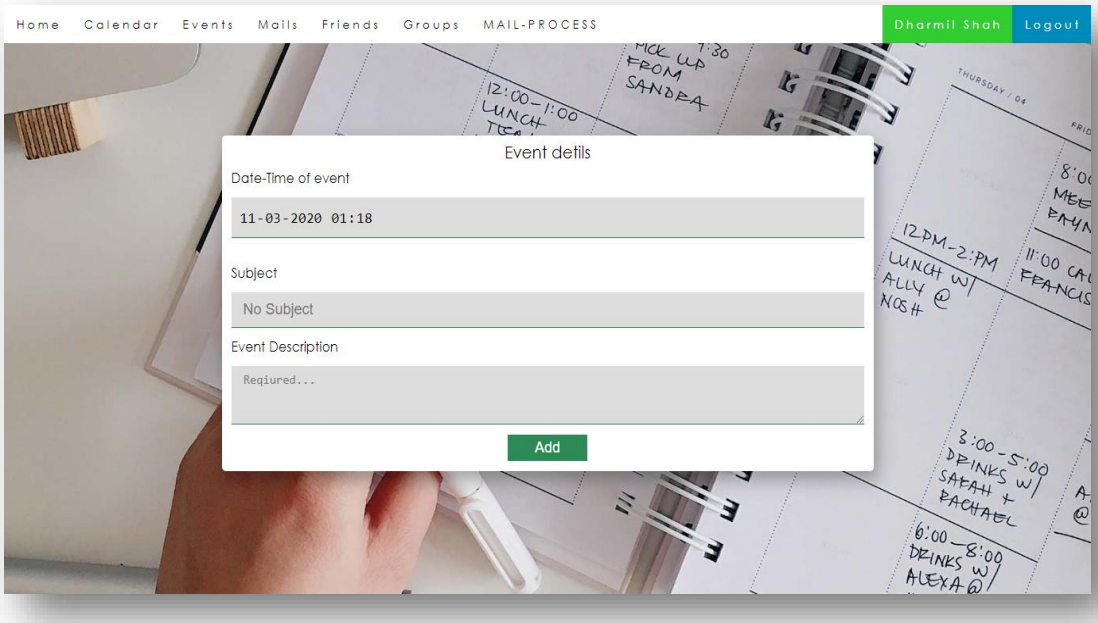
- This page is available just for group admin/creator.
- In this page, admin can perform group operations like add participant, remove participant and rename group.
- Possible validations are included.
- Left panel is as same as group chat.

Calendar



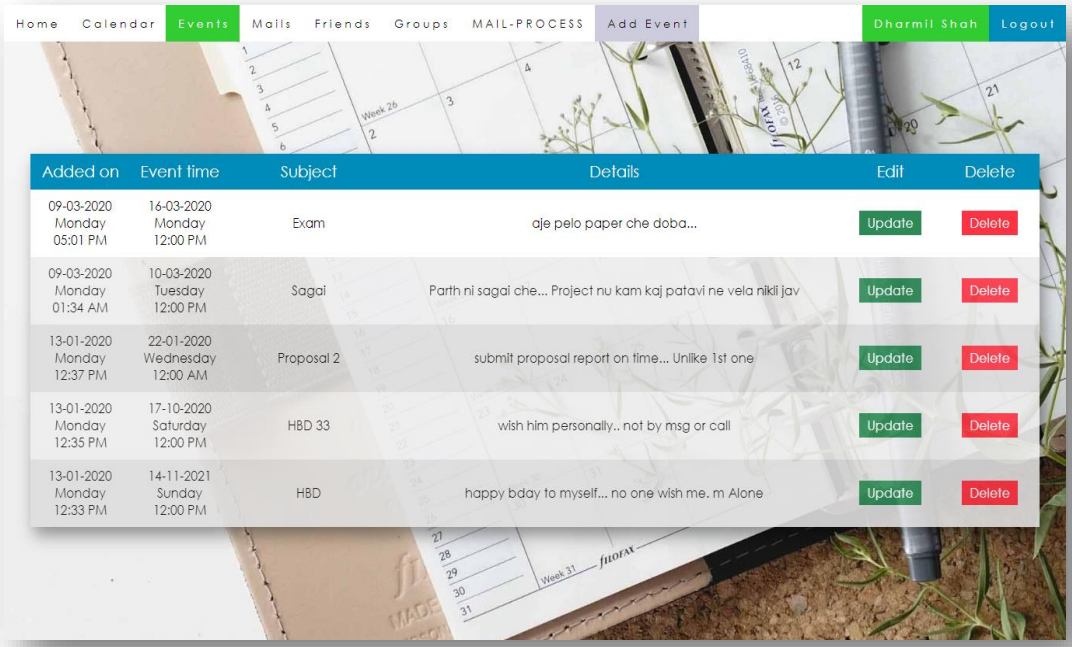
- This page is available for all registered users of site.
- Calendar provide easy way to see the month calendar.
- A different page with same UI and structure is there for group_events named as group_calendar.php
- User can select any month and year and can see calendar of that month.
- By clicking on any of the date, user will be redirected to add event in that particular date as shown in tooltip.
- In right side panel, user will see all the stored events in that particular month. Event date and event subject are shown.

Add Event



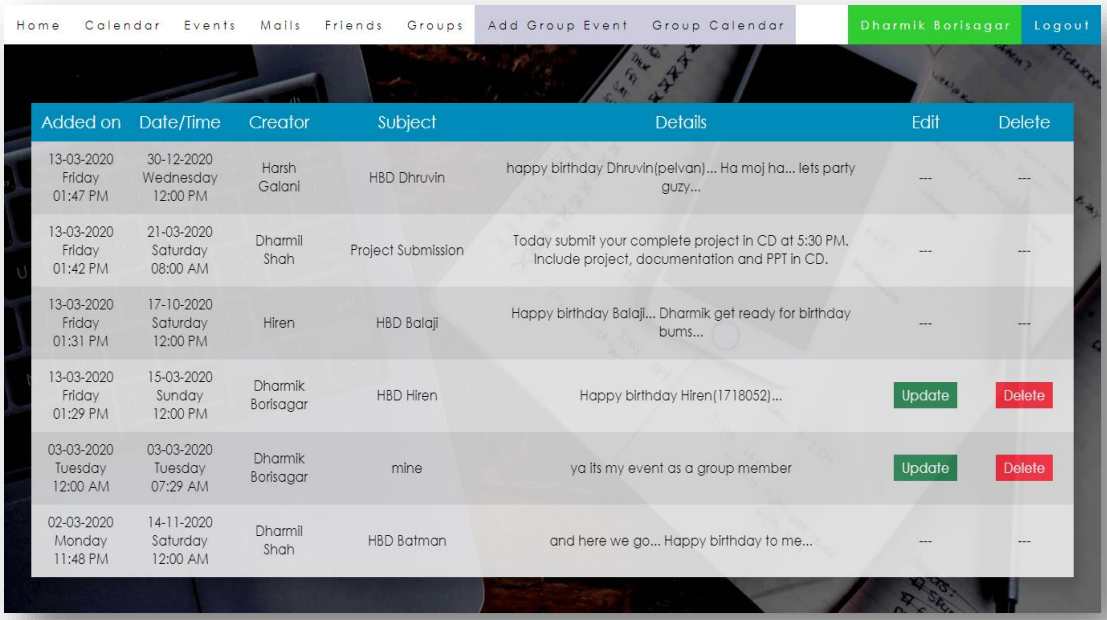
- In this page user can add his/her personal events.
- There is a different page with same form, UI and Structure for adding group events called add_group_events.php
- Form requires some event details which includes event date-time, event subject and event text/description.
- All fields are required.

Show Events



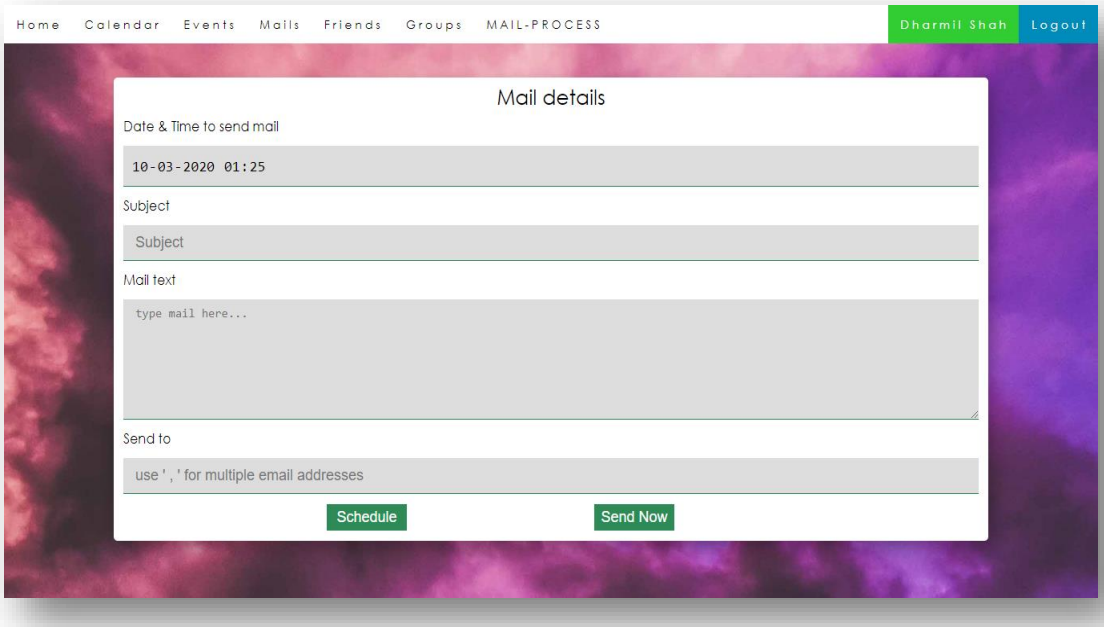
- In this page user will see all the events that are stored by him/her.
- Table includes Added on (date-time at which the event was added).
- Event time (date-time of event occurrence).
- Event subject and Details (event description).
- And operations like delete or update event.

Show Group Events



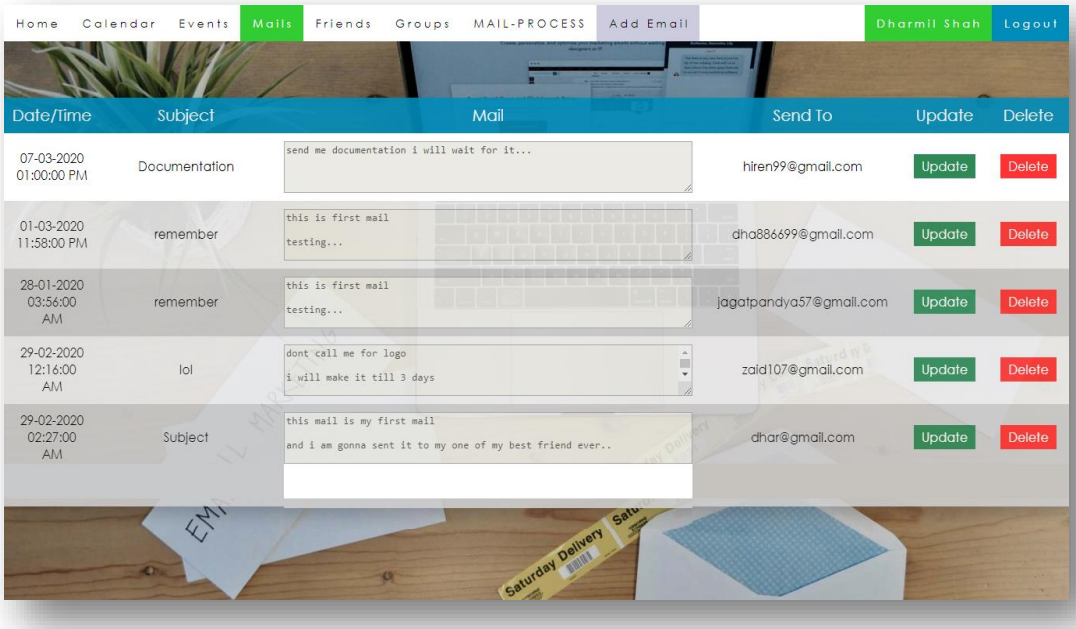
- In this page user will see all group events of particular group.
- Table includes Added on (date-time at which the event was added).
- Date/time (date-time of event occurrence).
- Creator (username of group member who added this event).
- Event subject and Details (event description).
- And operations like delete or update event. These operations are just available for event creator and group admin.

Add Mail



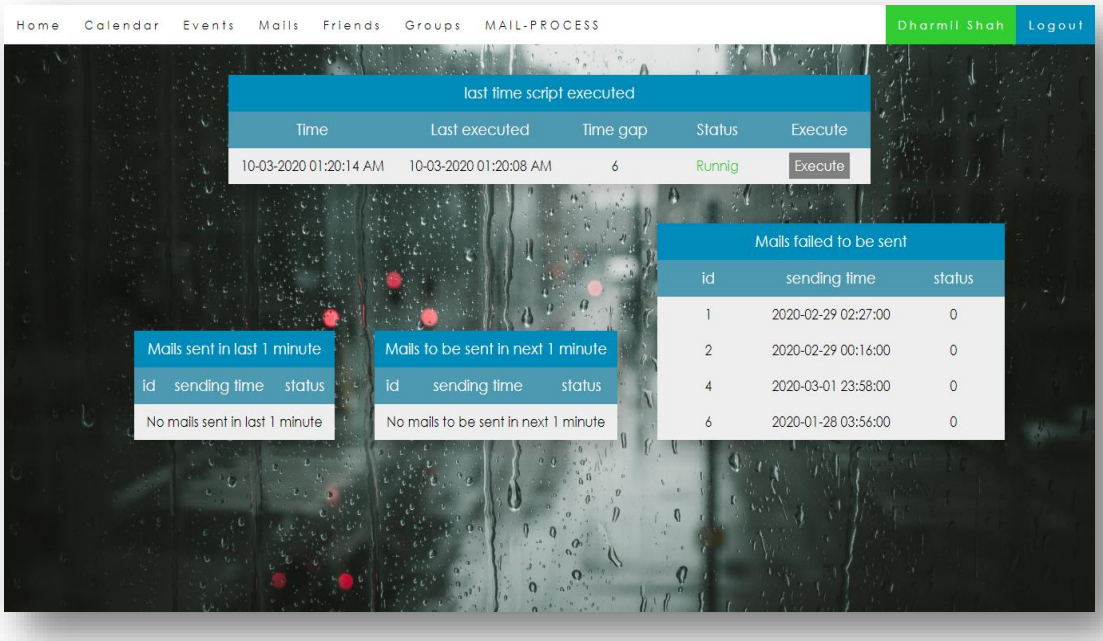
- User can send scheduled mail or normal mail through this page.
- All fields are required.
- Details include date-time to send mail (if it is scheduled mail).
- Subject of mail and then content of the mail.
- At last there is e-mail address of the mail recipient(s). If there are more than one recipients than user has to separate them by putting coma (,) in between them.
- There are 2 options are given which are Schedule and Send now.

Show Scheduled Mails



- In this page user will see the list of mails scheduled by him/her.
- It includes date time of sending mail, mail subject, mail content, mail recipient and operations like update or delete mail.

Mail Process



- This page is available only for site admins. It is important to keep an eye on mail sending process since this is one of the mail features of the site.
- This page contains details about scheduled mails and event mails that are to be sent in next few minutes.
- First table contains information about mail script. If mail sending script is running in background then Status will be ‘Running’ else ‘Not running’.
- If it is not running then admin can click on execute button. It also shows that when the script was run last time and time gap between current time and script run time.
- Other tables show upcoming mails, sent mails and mails that failed to be sent.

ShareCal

A Calendar for everyone

TESTING AND IMPLEMENTATION

4. TESTING AND IMPLEMENTATION

4.1 Testing Approaches :

Software testing is the process used to assess the quality of computer software. Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs. Quality is not an absolute; it is value to some person. With that in mind, testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification. An important point is that software testing should be distinguished from the separate discipline of Software Quality Assurance (S.Q.A.), which encompasses all business process areas, not just testing.

- **Testing Plan**

  - The main Objective of doing testing is to identify all defects existing in software. Basically, the testing of software consists of providing the program with a set of test inputs (test case) and observing that whether the software behaves as expected.
  - Testing is the process of executing a program with the explicit intention of finding errors, which makes the program fail. The tester is actually trying to make the program fail. A successful test is the one that finds errors.
  - Regardless of which strategies the analysts follow, there are preferred practices to ensure that the testing is useful. The levels of tests and practices to ensure that testing is useful. The levels of tests and types of test data, combined with testing libraries are important aspects of test process. Among the various testing practices or strategies that are followed by analysts, the two important ones are unit testing and system testing.
  - Software testing is crucial step in determining whether a software application is viable, ready for market and free of bugs. No software will be completely free of glitches but through software testing can and will make sure that it is as error free as humanly possible.

- **Testing Strategy**

1. Unit Testing

In this testing individual components and modules are tested to ensure that they operate correctly. We had tested each and every module such as valid admin login, user login, photographer registration, etc.

2. Integration Testing

This testing is a systematic technique for constructing the program structure while at same time conducting tests to uncover errors associated with interfacing. All the modules testing in the unit testing are integrated and are tested for their inter-dependency. This system mainly works on the integration of all the departments. Human Factor testing.

In this type of testing the user can't use this management system without its accessibility.

3. Acceptance Testing

This type of testing is done when the system is being deployed. The testing data are supplied by the system procurer. If they think it capable and only after a series of thorough testing the system will be ready to use by the company.

4. Control Testing

Control is a management tool to that processing is performed in accordance to what management desire or intents of management. The objective of doing so are accurate and complete data, Authorized transactions of doing so are accurate and complete data, and process meeting needs of the user.

5. Software Testing

Involves executing an implementation of the software with test data and examining the outputs of the software and its operational behavior to check that it is performing as required.

6. Statistical Testing

Used to test the program's performance and reliability and to check how to works under operational conditions. Tests are designed to reflect the actual user inputs and their frequency. The stages involved in the static analysis for this system are follows.

7. Data use analysis

- Variable used before initialization
- Variable declared but never used
- Variable assigned twice but never used between assignments
- Possible array bound violations
- Declare variables Interface analysis
- Parameter type mismatches
- Non-usage of results of functions
- Uncalled function and procedures.

8. Defect Testing

Intended to find inconsistencies between a program and its specification. These inconsistencies are usually due to program faults or defects.

9. Structural Testing

We have done path testing to exercise every independent execution path through a component or program. If every independent path is executed, then all statements in the components must have been executed at least once.

We checked graphics module and database access module, which have independent execution path. They are not related to each other. The structure of our program is also checked.

10. Performance Testing

Performance testing is designed to test the runtime performance of the system within the context of the system. These tests were performed as module level as well as system level. Individual modules were tested for required performance

In Performance testing we counted the processing time and response from the server with respect to request.

We also checked out the total-execution time. On the click ofthe mouse we get the result.

11. Condition Testing

Coding testing is a test case design method that exercises the logical conditions in a program module. If the condition is incorrect then at least one component of the condition is incorrect.

12. Stress Testing

Stress testing is to test the system for emergent properties such as performance and reliability. Performance tests have to design to ensure that the system can process its intended load.

13. Object Testing

Object testing is to test objects as individual components, which are often larger than functions. Here following activities taken place.

- Testing the individual operations associated with objects.
- Testing individual object classes.
- Testing clusters of objects.
- Testing the object-oriented system.

- **Types of Testing :**

1. Black box Testing:

- Black box testing treats the software as a black-box without any understanding of internal behavior. It aims to test the functionality according to the requirements. Thus, the tester inputs data and only sees the output from the test object. This level of testing usually requires thorough test cases to be provided to the tester who then can simply verify that for a given input, the output value (or behavior), is the same as the expected value specified in the test case. Black box testing methods include: equivalence

partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix etc.

2. White box Testing:

- White box testing, however, is when the tester has access to the internal data structures, code, and algorithms. White box testing methods include creating tests to satisfy some code coverage criteria. For example, the test designer can create tests to cause all statements in the program to be executed at least once. Other examples of white box testing are mutation testing and fault injection methods. White box testing includes all static testing.

- White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. Two common forms of code coverage are function coverage, which reports on functions executed and statement coverage, which reports on the number of lines executed to complete the test. They both return coverage metric, measured as a percentage.

2. Gray box Testing:

- In recent years the term grey box testing has come into common usage. This involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as grey-box because the input and output are clearly outside of the black-box we are calling the software under test. This is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed for test. Grey box testing may also include reverse engineering to determine, for instance, boundary values.

Special methods exist to test non-functional aspects of software. Performance testing checks to see if the software can handle large quantities of data or users. Usability testing is needed to check if the user interface is easy to use and understand. Security testing is essential for software which processes confidential data and to prevent system intrusion by hackers. To test internationalization

and localization aspects of software a pseudo localization method can be used.

A common practice of software testing is performed by an independent group of testers after the functionality is developed before it is shipped to the customer. This practice often results in the testing phase being used as project buffer to compensate for project delays, thereby compromising the time devoted to testing. Another practice is to start software testing at the same moment the project starts and it is a continuous process until the project finishes.

In counterpoint, some emerging software disciplines such as extreme programming and the agile software development movement, adhere to a "test-driven software development" model. In this process unit tests are written first, by the software engineers (often with pair programming in the extreme programming methodology). Of course these tests fail initially; as they are expected to. Then as code is written it passes incrementally larger portions of the test suites. The test suites are continuously updated as new failure conditions and corner cases are discovered, and they are integrated with any regression tests that are developed. Unit tests are maintained along with the rest of the software source code and generally integrated into the build process (with inherently interactive tests being relegated to a partially manual build acceptance process).

Levels of Testing :

- **Unit testing** tests the minimal software component, or module. Each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. Corresponding to elements of the architectural design are integrated and tested until the software works as a system.

- **System testing** tests a completely integrated system to verify that it meets its requirements.

- **System integration testing** verifies that a system is integrated to any external or third party systems defined in the system requirements.

Before shipping the final version of software, alpha and beta testing are often done additionally:

- **Alpha testing** is simulated or actual operational testing by potential users/customers or an independent test team at the developers' site. Alpha testing is often employed for off-the shelf software as a form of internal acceptance testing, before the software goes to beta testing.

- **Beta testing** comes after alpha testing. Versions of the software, known as **beta versions,** are released to a limited audience outside of the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or **bugs.** Sometimes, beta versions are made available to the open public to increase the feedback field to a maximal number of future users.

4.2 Test Cases

- A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Test Case 1

Test Case ID	TCID-1
Name	User Login Test
Test Scenario	It will check if the user is registered or not.
Expected Result	If user is not registered then Appropriate message will be displayed, otherwise logged in.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 2

Test Case ID	TCID-2
Name	User Signup Test
Test Scenario	New User can Register easily.
Expected Result	User can register with a unique email-address. If that email is already registered in database then give appropriate message. Otherwise store registration details in database.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 3

Test Case ID	TCID-3
Name	Add Event Test
Test Scenario	User can Add his/her events.
Expected Result	User can add event details like subject, description, date-time and it will be stored in database and user can see, update and delete all events stored by his/her.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 4

Test Case ID	TCID-4
Name	Personal Chat Test
Test Scenario	User can Chat with Other Users.
Expected Result	User can find other users registered on site through their email or username and then they can chat with each other. Chat will be stored in database and will be retrieved successfully every time. If user not found then give appropriate message.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 5

Test Case ID	TCID-5
Name	Create Group Test
Test Scenario	User can Create group(s).
Expected Result	User can create group(s) with other registered users and they all can chat in group. All group details including members, chats and group events will be stored in database.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 6

Test Case ID	TCID-6
Name	Group Events Test
Test Scenario	User can Add Group Events.
Expected Result	Group members can see, add, update and delete group events. Event's data will be stored and retrieved from database.
Actual Results	As Expected
Pass/Fail	Pass

Test Case 7

Test Case ID:	TCID-7
Name:	Group Edit Test
Test Scenario:	Group admin can rename group if group with same name & admin doesn't exist. Admin can add and remove any group member through email address.
Expected Result:	If that user is already a member or doesn't registered on site then display appropriate message. Or update database as per the operation.
Actual Results:	As Expected
Pass/Fail:	Pass

Test Case 8

Test Case ID:	TCID-8
Name:	Forget Password Test
Test Scenario:	In login page, if user want to reset password then he can click on Forgot Password link.
Expected Result:	If given email is registered then verify OTP and then enter new password and update data in database. Else display appropriate message.
Actual Results:	As Expected
Pass/Fail:	Pass

Test Case 9

Test Case ID:	TCID-9
Name:	Schedule Mail Test
Test Scenario:	User can Schedule mail. User enters all details like date-time, subject, content and e-mail address of recipient.
Expected Result:	Scheduled mails will be stored in Database with all details including multiple recipient's email addresses.
Actual Results:	As Expected
Pass/Fail:	Pass

Test Case 10

Test Case ID:	TCID-10
Name:	Update Events and Mails
Test Scenario:	User can update details of events and scheduled mails stored by him/her.
Expected Result:	Update data into database according to event_id or mail_id.
Actual Results:	As Expected
Pass/Fail:	Pass

Test Case 11

Test Case ID:	TCID-11
Name:	Mail sending process
Test Scenario:	Admin users are allowed to se status of mail sending script and list of mails/events that will be sent in next few minutes.
Expected Result:	Admin user can execute/run mail sending script if script is not running.
Actual Results:	As Expected
Pass/Fail:	Pass

4.3 Bug Report

Developer faces many bugs in development phase. I have also faced many and some of them are listed below.

Bug	Status
Designing issues in some pages like calendar and chat.	Solved
Fetch date in proper format as required in forms databases and to display.	Solved
Resubmission of form each time when the page is reloaded in chat modules.	Solved
Unauthorised surfing among web pages through writing URL.	Solved
Difficulties in storing & updating events and mail data from database.	Solved

CONCLUSION

5. CONCLUSION

5.1 Future Enhancement

As opposed to our current development, a few more functionality can be added to further expand the project.

- ✓ Providing the more facility and operations to admin users. Like deleting or blocking users and groups from user's reports.
- ✓ Adding more facilities for users like notifications of chat or any other activities.
- ✓ We can add functionality to make group public so any user can search and join it. Like public channels on telegrams.
- ✓ More functionality for users like sending attachments with mail, user's profile picture, sending pics and emoji in chatting etc.

5.2 Conclusion

In the beginning the project seemed to be just one another subject that we needed to pass in order to get a through our semester. But with passing time and with each passing day (even nights), we found the development of this project became a great learning process.

Not only the project taught us how to analyse and code any software/web-site but it also managed to show me important resource management skills. Also teach us time management in within deadline is reach so it is a part of management skill and important things for future.

We had to complete the project in time and budget. It was very important that we got proper information from the end-users in order to provide them with a fully functional and simple enough to understand system.

All in all this entire project has prepared us in many ways for what we are going to face in the IT field in future.

5.3 Bibliography (References) :

Bibliography means that we have refer while doing project like, which sites, we gone through and which books we have referred or which magazines. There are books and websites that can help me proper guideline to implements my application in the right direction and without any errors.

Web site references :

https://www.w3schools.com/

https://stackoverflow.com/

https://www.geeksforgeeks.org/

https://www.youtube.com/

https://www.php.net/