Chapter 1

INTRODUCTION

1.1Introduction

Pranathi Admin Dashboard is a one of kind web application dedicated to effective management of all the details of a college fest. Once the core team member has registered and entered all the details, this application gives all the statistics of the fest in one place.

The application has different pages like organisers, participants and users. The user can create or delete new events on the application. The users can also create new organisers to events.

The main feature of the application is to register participants to events of the fest. A smooth flow of user registration is implemented to make sure hassle-free registration of users to events.

All in all, our application enables easier management of the entire data of the college fest and can be potentially deployed to use in events with large footfall.

1.2 Problem Statement

To develop a web application using HTML, CSS, Javascript, PHP and MySQL that enables overall management of a college fest where users can sign up or login to view information that is relevant to them.

Chapter 2

LITERATURE SURVEY

2.1 Event Brite [1]

Eventbrite is an American event management and ticketing website. The service allows users to browse, create, and promote local events. The service charges a fee to event organizers in exchange for online ticketing services unless the event is free.[2]

Launched in 2006 and headquartered in San Francisco, Eventbrite opened their first international office in the United Kingdom in 2012. The company now[has local offices in Nashville, London, Cork, Amsterdam, Dublin, Berlin, Melbourne, Mendoza, Madrid, and São Paulo. The company went public on the New York Stock Exchange on September 20, 2018 under the ticker symbol EB.

2.2 Yapsody [2]

Yapsody India Pvt Ltd. was founded in January 2012 and since has been serving 3200+ presenters and 14,000+ events from around 55 countries. Yapsody is one of the fastest growing online ticketing software with a rapidly growing user base. It is a do it yourself online ticketing platform for event presenters of any size, from conferences, marathons, standing-room-only clubs to festival grounds and arenas with reserved seating. Yapsody is guaranteed to be the most cost-effective, value-packed ticketing and box office management system available for any type of venue from casinos to arenas – loaded with unique, innovative, never-before-seen features specially designed to support soft-ticket venues such as casinos, fairs and festivals.

2.3 Ticketbud [3]

Ticketbud is an event management, ticketing and registration SaaS platform for event organizers.[2] The software allows event organizers to sell tickets to events and also to promote and market their events on Facebook, Twitter, and Instagram. Founded in 2009 in Austin, Texas, Ticketbud's revenue is derived from a fee placed upon tickets sold through its software at $0.99 + 2% of the ticket's value per ticket. It places no fee upon events which cost nothing to enter, and due to its involvement with non-profits it offers a reduced rate for organizations under 501(c)(3) designation.

Chapter 3

REQUIREMENT ANALYSIS

3.1 Listing of requirements:

- Login Page
- Sign Up Page
- Home Page/ Dashboard
- Organisers Page
- Registrations Page
- Users Page
- Events Page
- Register a New Participant Page

3.2 Listing of screen design:

- The login and sign-up pages are must for a user to be able to use the application
- After successful registration or login, the user is taken to the dashboard where the overall details of the fest are shown
- The dashboard page consists of all the statistics of the fest including registrations, amount collected and number of participants.
- Individual screens are designed for different entities such as participants, users and organisers.
- An event page is created to create and delete events for the fest

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

4.1 Functional Requirements

- The user is required to either login or register before starting their activity on the website.
- The fest information is displayed to the user.
- Users on the platform should be able to register new participants to events
- Detailed information about organisers, users, events and participants should be shown on the dashboard

4.2 Non-Functional Requirements

- The user is required to either login or register before starting their activity on the website.
- The website displays the doctors list and an option to contact the clinic.
- **Scalability** -Is the ability of a system to grow in its capacity to meet the rising demand for its services offered.
- **Capacity** - How much it is needed to produce and determine if it is capable of meeting those production goals.
- **Availability** - Application of monitoring and management of software.
- **Reliability** - It is the probability of failure-free software operation for a specified period of time in a specified environment.
- **Recoverability** - How much time a system takes in order to recover.
- **Interoperability** - It is defined as "the ability to share information and services".
- **Serviceability** - It refers to the conditions where there is a limit for some factors such as

durability, overall stability, fire resistance, deflection, cracking and excessive vibration

- **Performance**– For example Response Time, Throughput, Utilization, Static Volumetric.
- **Security** - Services such as authentication, authorization, backup, server-clustering, etc.
- **Environmental**-Any legal requirement related to environment.
- **Data Integrity** – It is a typical requirement of documents that include timelines, goals, expectations, rules, and roles – as well as company-specific requirements

Chapter 5

HARDWARE AND SOFTWARE REQUIREMENTS

5.1 Software Requirements

- Front end: HTML, CSS, Javascript
- Back end: PHP
- Database: MySQL
- Operating System: Windows 10
- Architectural Platform: MVC

5.1 Hardware Requirements

- Processor : Intel i5 8th gen
- RAM: 8GB
- Hard Disk: 1TB

Chapter 6

WORK BREAKDOWN STRUCTURE

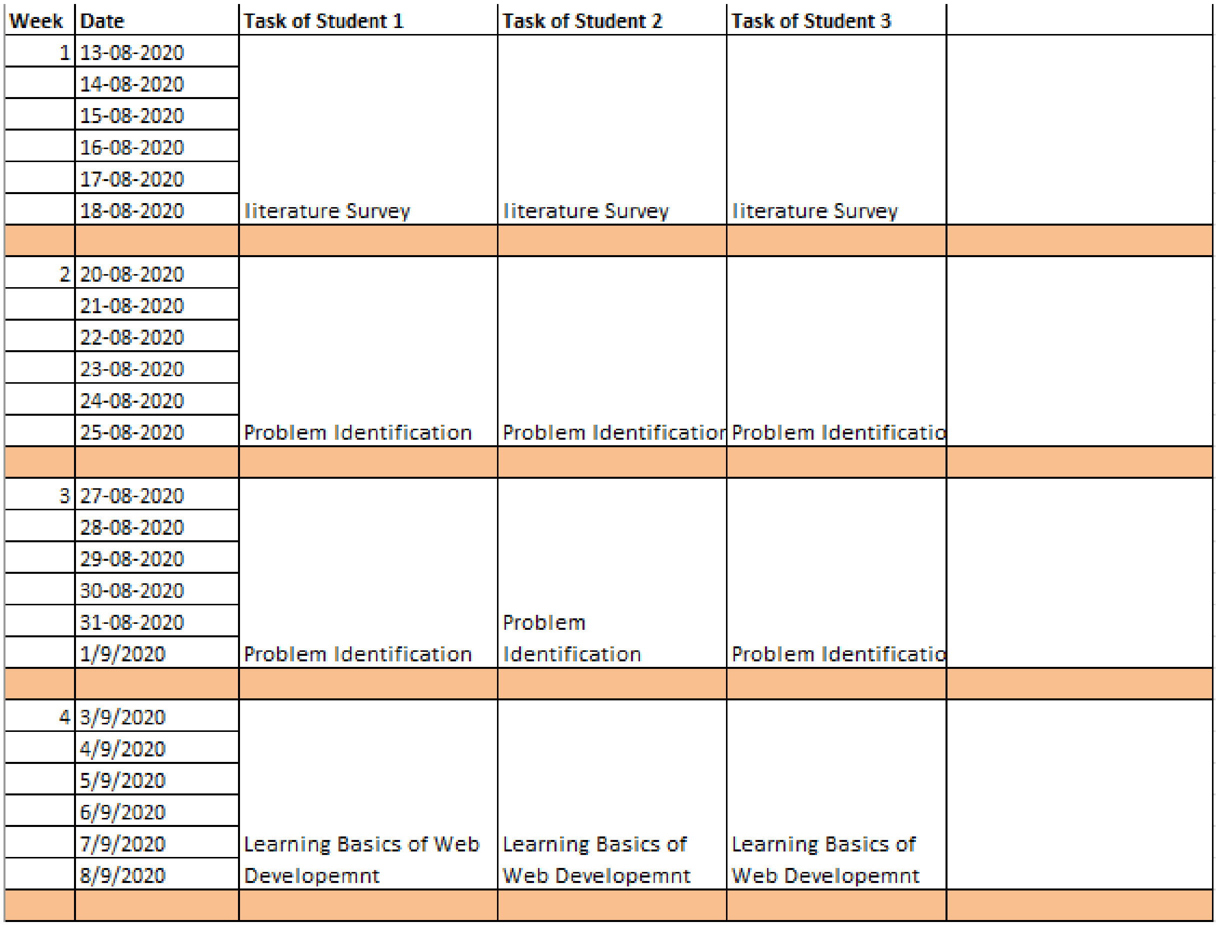
6.1 Planning Process:

The main goal of this project is to make a website for users to be able to find all donation organizations in a single website. The project is started by analyzing the similar websites for getting an overall idea on how to implement the website. To achieve the goal, we must consider a set of steps starting from literature survey to implementation and documentation of project.

6.2 List of tasks:

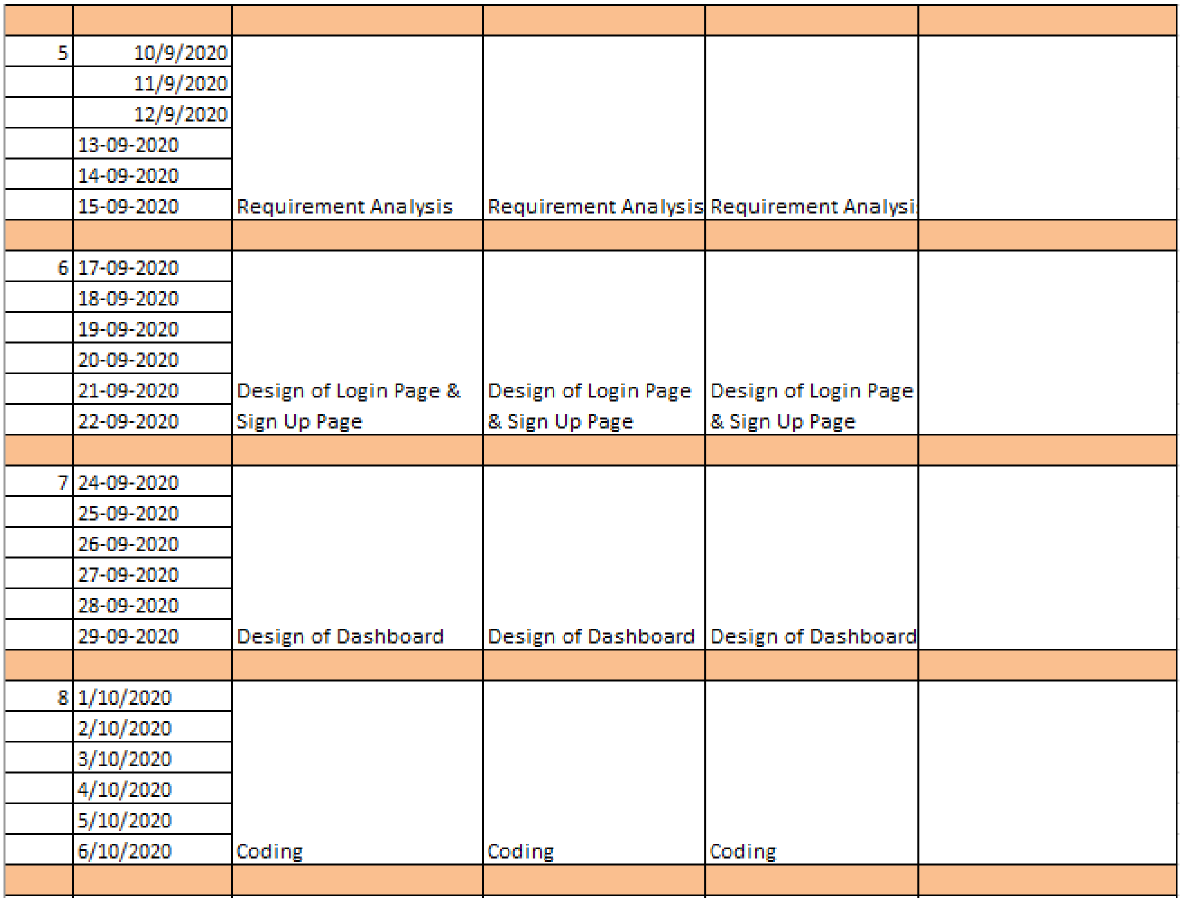
- Literature survey
- Problem Identification
- Learning activity
- Requirement analysis
- Design
- Coding
- Testing
- Report writing

6.3 Assignment of tasks:



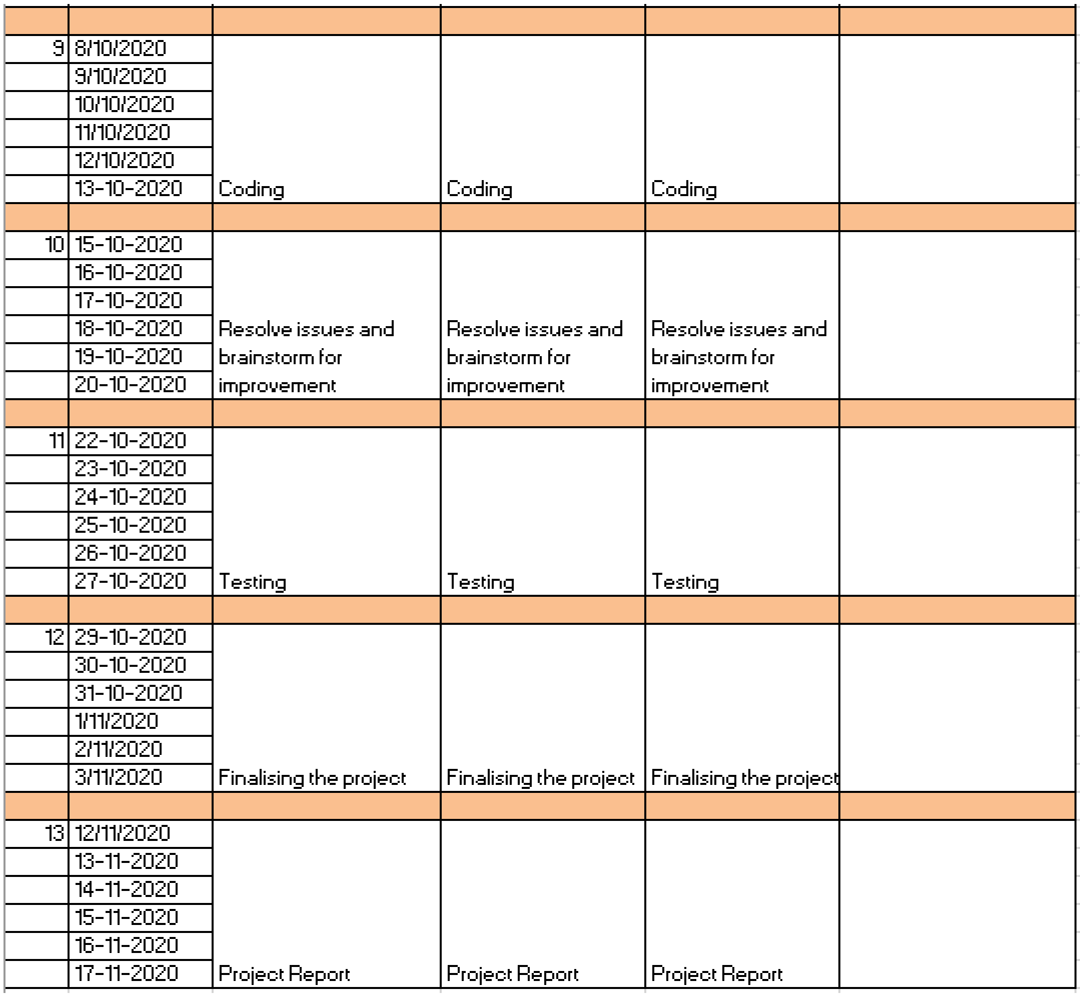
Week	Date	Task of Student 1	Task of Student 2	Task of Student 3	
1	13-08-2020	literature Survey	literature Survey	literature Survey	
	14-08-2020				
	15-08-2020				
	16-08-2020				
	17-08-2020				
	18-08-2020				
2	20-08-2020	Problem Identification	Problem Identificatior	Problem Identificatio	
	21-08-2020				
	22-08-2020				
	23-08-2020				
	24-08-2020				
	25-08-2020				
3	27-08-2020	Problem Identification	Problem Identification	Problem Identificatio	
	28-08-2020				
	29-08-2020				
	30-08-2020				
	31-08-2020				
	1/9/2020				
4	3/9/2020	Learning Basics of Web Developemnt	Learning Basics of Web Developemnt	Learning Basics of Web Developemnt	
	4/9/2020				
	5/9/2020				
	6/9/2020				
	7/9/2020				
	8/9/2020				

Fig 6.1: Assignment of tasks



5	10/9/2020	Requirement Analysis	Requirement Analysis	Requirement Analysi	
	11/9/2020				
	12/9/2020				
	13-09-2020				
	14-09-2020				
	15-09-2020				
6	17-09-2020	Design of Login Page & Sign Up Page	Design of Login Page & Sign Up Page	Design of Login Page & Sign Up Page	
	18-09-2020				
	19-09-2020				
	20-09-2020				
	21-09-2020				
	22-09-2020				
7	24-09-2020	Design of Dashboard	Design of Dashboard	Design of Dashboard	
	25-09-2020				
	26-09-2020				
	27-09-2020				
	28-09-2020				
	29-09-2020				
8	1/10/2020	Coding	Coding	Coding	
	2/10/2020				
	3/10/2020				
	4/10/2020				
	5/10/2020				
	6/10/2020				

Fig 6.3: Assignment of tasks



9	8/10/2020	Coding	Coding	Coding	
	9/10/2020				
	10/10/2020				
	11/10/2020				
	12/10/2020				
	13-10-2020				
10	15-10-2020	Resolve issues and brainstorm for improvement	Resolve issues and brainstorm for improvement	Resolve issues and brainstorm for improvement	
	16-10-2020				
	17-10-2020				
	18-10-2020				
	19-10-2020				
	20-10-2020				
11	22-10-2020	Testing	Testing	Testing	
	23-10-2020				
	24-10-2020				
	25-10-2020				
	26-10-2020				
	27-10-2020				
12	29-10-2020	Finalising the project	Finalising the project	Finalising the project	
	30-10-2020				
	31-10-2020				
	1/11/2020				
	2/11/2020				
	3/11/2020				
13	12/11/2020	Project Report	Project Report	Project Report	
	13-11-2020				
	14-11-2020				
	15-11-2020				
	16-11-2020				
	17-11-2020				

Fig 6.3: Assignment of tasks

Chapter 7

DESIGN AND ANALYSIS

7.1 Database Design

A database is simply an organized collection of related data, typically stored on disk, and accessible by possibly many concurrent users. Databases are generally separated into application areas. For example, one database may contain Human Resource (employee and payroll) data; another may contain sales data; another may contain accounting data; and so on. Databases are managed by a DBMS.

The choice of a database product is often influenced by factors such as:

- Computing platform (i.e., hardware, operating system)
- The volume of data to be managed.
- The number of transactions required per second.
- Existing applications or interfaces that an organization may have.
- Support for heterogeneous and/or distributed computing.
- Cost.
- Vendor support.

7.1.1 Introduction to MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language. A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language, programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).[8] In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.[9]

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

7.1.2 Structure of the Database

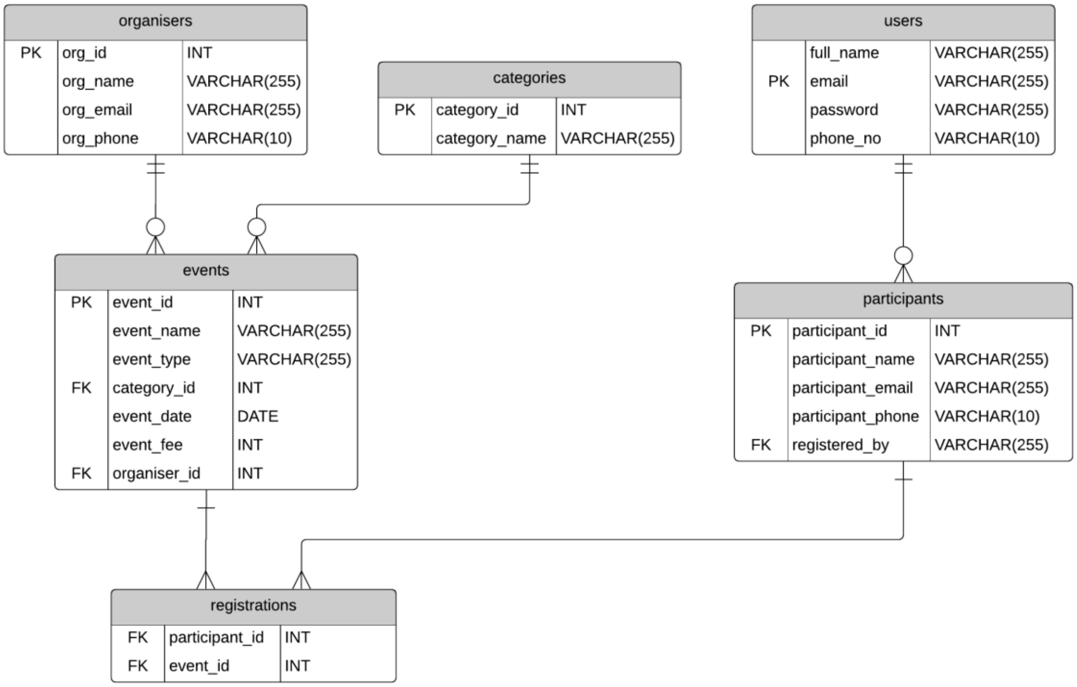
The list of tables that are present in the database are:

- User
- Participant
- Event
- Category
- Organiser

7.2 Schema Diagram:

In database terms, a schema is the organisation and structure of a database. A schema contains schema objects, which could be tables, columns, data types, views, stored procedure, relationships, primary keys, foreign_keys, etc. A database schema can be represented in a visual diagram, which shows the database objects and their relationship with each other.

In any data model, it is important to distinguish between the description of the database and the database itself. The description of a database is called the database schema, which is specified during database design and is not expected to change frequently. Most data models have certain conventions for displaying schemas as diagrams. A displayed schema is called a **schema diagram.**



organisers

PK	org_id	INT
	org_name	VARCHAR(255)
	org_email	VARCHAR(255)
	org_phone	VARCHAR(10)

categories

PK	category_id	INT
	category_name	VARCHAR(255)

users

	full_name	VARCHAR(255)
PK	email	VARCHAR(255)
	password	VARCHAR(255)
	phone_no	VARCHAR(10)

events

PK	event_id	INT
	event_name	VARCHAR(255)
	event_type	VARCHAR(255)
FK	category_id	INT
	event_date	DATE
	event_fee	INT
FK	organiser_id	INT

participants

PK	participant_id	INT
	participant_name	VARCHAR(255)
	participant_email	VARCHAR(255)
	participant_phone	VARCHAR(10)
FK	registered_by	VARCHAR(255)

registrations

FK	participant_id	INT
FK	event_id	INT

Fig 7.1: Schema Diagram

7.3 ER Diagram:

An entity relation diagram(ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

An ER diagram is a means of visualizing how the information a system produces is related. ERD diagrams are commonly used in conjunction with a data_flow_diagram to display the contents of a data store

There are five main components of an ERD:

- **Entities**, which are represented by rectangles.
- **Actions**, which are represented by diamond shapes

- **Attributes**, which are represented by ovals.
- **Connecting lines**, solid lines connect attributes to show the relationships of entities in the diagram.
- **Cardinality** specifies how many instances of an entity relate to another entity instance.
- **Key Attribute -** The attribute which **uniquely identifies each entity** in the entity set is called key attribute.
- **Composite Attribute –** An attribute **composed of many other attribute** is called as composite attribute.
- **Multivalued Attribute –** An attribute consisting **more than one value** for a given entity.
- **Derived Attribute –** An attribute which can be **derived from other attributes** of the entity type is known as derived attribute.

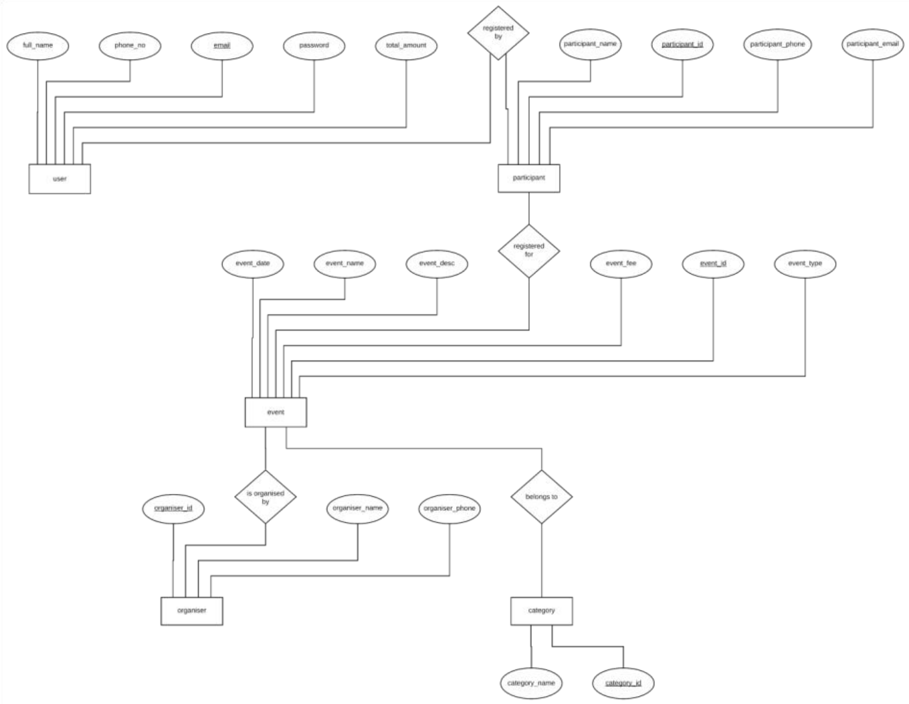


Fig 7.2: ER Diagram

7.4 SQL File

```
CREATE TABLE `categories` (
  `category_id` int(11) NOT NULL,
  `category_name` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;


CREATE TABLE `events` (
  `event_id` int(11) NOT NULL,
  `event_name` varchar(255) NOT NULL,
  `event_type` varchar(255) NOT NULL,
  `category_id` int(11) NOT NULL,
  `event_date` date NOT NULL,
  `event_fee` int(11) NOT NULL,
  `event_desc` text NOT NULL,
  `organiser_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;


CREATE TABLE `logs` (
  `log_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `log_user` varchar(255) NOT NULL,
  `log_message` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

```
CREATE TABLE `participants` (
  `participant_id` int(11) NOT NULL,
  `participant_name` varchar(255) NOT NULL,
  `participant_email` varchar(255) NOT NULL,
  `participant_phone` varchar(10) NOT NULL,
  `registered_by` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

CREATE TABLE `registrations` (
  `participant_id` int(11) NOT NULL,
  `event_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;

CREATE TABLE `users` (
  `full_name` varchar(255) NOT NULL,
  `email` varchar(255) NOT NULL,
  `pass` varchar(255) NOT NULL,
  `phone` varchar(10) NOT NULL,
  `contribution` int(11) DEFAULT '0'
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_general_ci;
```

Insertions

```
INSERT INTO `categories` (`category_id`, `category_name`) VALUES

(98707, 'Technical'),

(98708, 'Brainstorming'),

(98709, 'Cultural'),

(98710, 'Sports'),

(98711, 'Gaming'),

(98712, 'Fun');
```

```
INSERT INTO `events` (`event_id`, `event_name`, `event_type`, `category_id`, `event_date`, `event_fee`, `event_desc`, `organiser_id`) VALUES

(313813, 'Coding & Debugging', 'Individual', 98707, '2019-12-18', 100, 'Can you beat the time? Most importantly, the errors?', 67547);
```

```
INSERT INTO `organisers` (`organiser_id`, `organiser_name`, `organiser_phone`) VALUES

(67547, 'Revanth', '7895374954');
```

```
INSERT INTO `participants` (`participant_id`, `participant_name`, `participant_email`, `participant_phone`, `registered_by`) VALUES

(213648, 'Samarth A', 'samarth@gmail.com', '8874298334', 'gurunn@gmail.com');
```

```
INSERT INTO `registrations` (`participant_id`, `event_id`) VALUES (213648, 313813);
INSERT INTO `users` (`full_name`, `email`, `pass`, `phone`, `contribution`) VALUES ('Siddharth Subramanian', 'sidsbrmnn@gmail.com', '332b3091416bc4687821c4653f1c6eb1', '9535572838', 0);
```

Triggers

```
DELIMITER $$

CREATE TRIGGER `log_user` AFTER INSERT ON `users` FOR EACH ROW INSERT INTO logs (log_user, log_message)

VALUES (NEW.email, 'Signed up as a user')

$$

DELIMITER
```

Stored Procedure

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `calc_contribution`()
BEGIN
    DECLARE
        finished INTEGER DEFAULT 0 ; DECLARE emailAddress VARCHAR(255) DEFAULT "" ;
DECLARE curEmail CURSOR FOR
    SELECT
        email
    FROM
        users ; DECLARE CONTINUE
    HANDLER FOR NOT FOUND
SET
    finished = 1 ; OPEN curEmail ; getEmail: LOOP FETCH curEmail
INTO emailAddress ; IF finished = 1 THEN LEAVE getEmail ;
END IF ;
UPDATE
    users
SET
    contribution =(
    SELECT
        SUM(event_fee) AS contribution
    FROM
        participants
    NATURAL JOIN registrations NATURAL JOIN EVENTS WHERE registered_by = emailAddress
)
WHERE
    email = emailAddress ;
END LOOP getEmail ; CLOSE curEmail ;
END$$
DELIMITER ;
```

7.5 Design Process

7.5.1 Setting Goals and Objectives

We conducted contextual inquiries with our existing core team and registration team members to understand the current state of the process. The team used Google Sheets to store the data of the participants and events. Many of our existing team members did not like the current process because there was no connectivity between the registration team members during registration apart from the common Google Sheet that was created, every registration was manually entered and nothing was standardized.

7.5.2 Goals of the dashboard

The gal of the dashboard is to create a seamless connection between all the registration team users to register/view/modify participants and events without multiple steps and track total entry fee in real time.

7.5.3 Target Audience

Primary audience: Core team and Registration team

Secondary audience: Volunteers

7.5.4 User Flow

A site map describes the different content pieces on the site and the relationship between them. It is an important step of the user centered process as it ensures content is in places users would expect to find it.

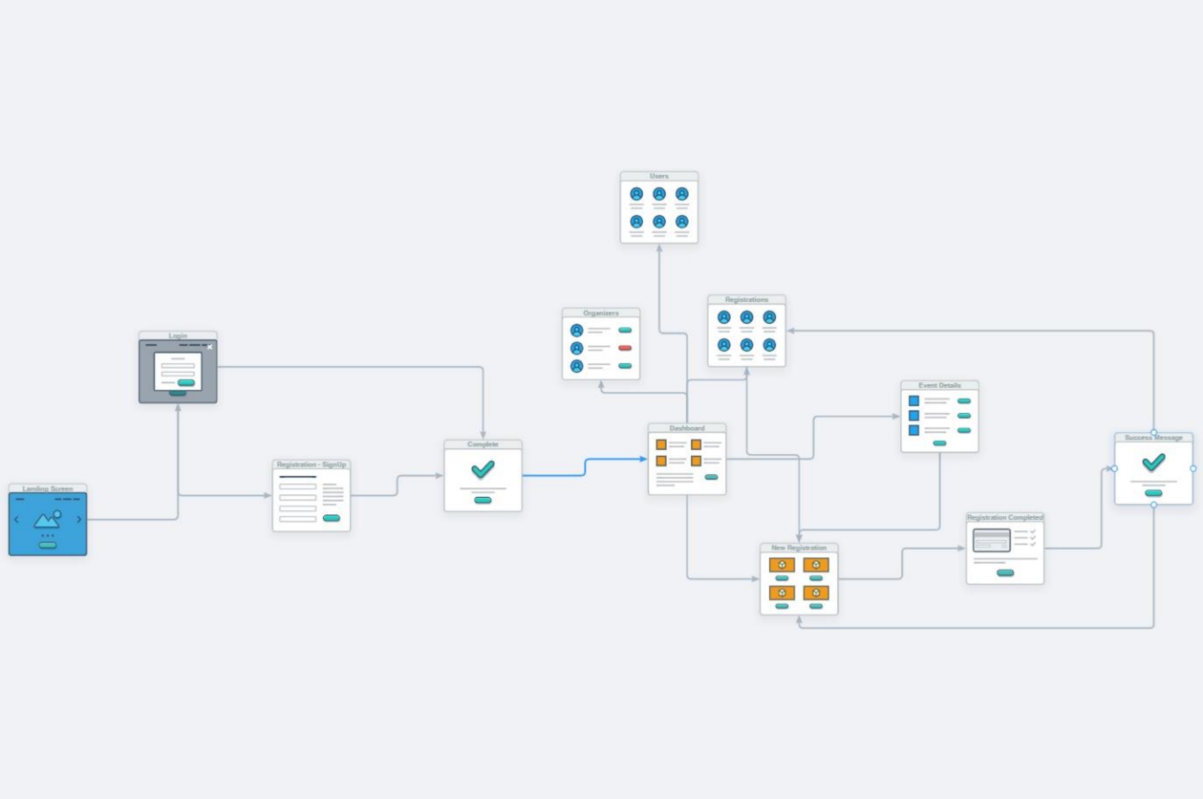


Fig 7.3: User Flow Diagram

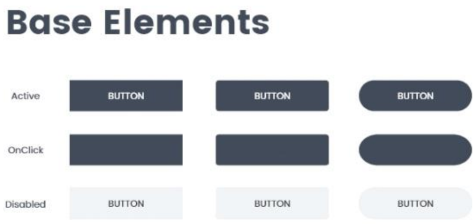


Fig 7.4: Base Elements

7.5.5 Design Elements

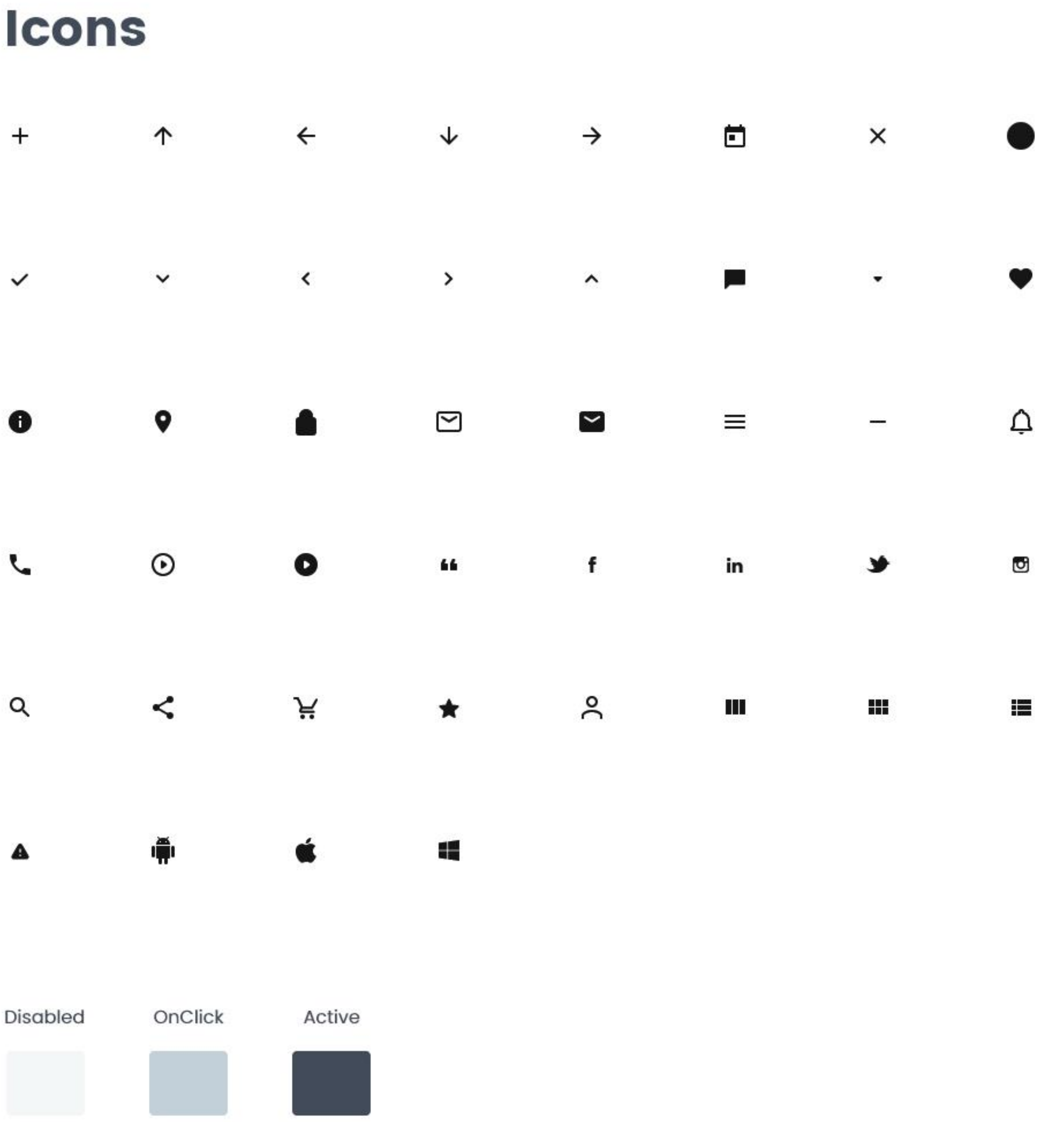


Fig 7.5: Icons

7.5.6 Font Systems

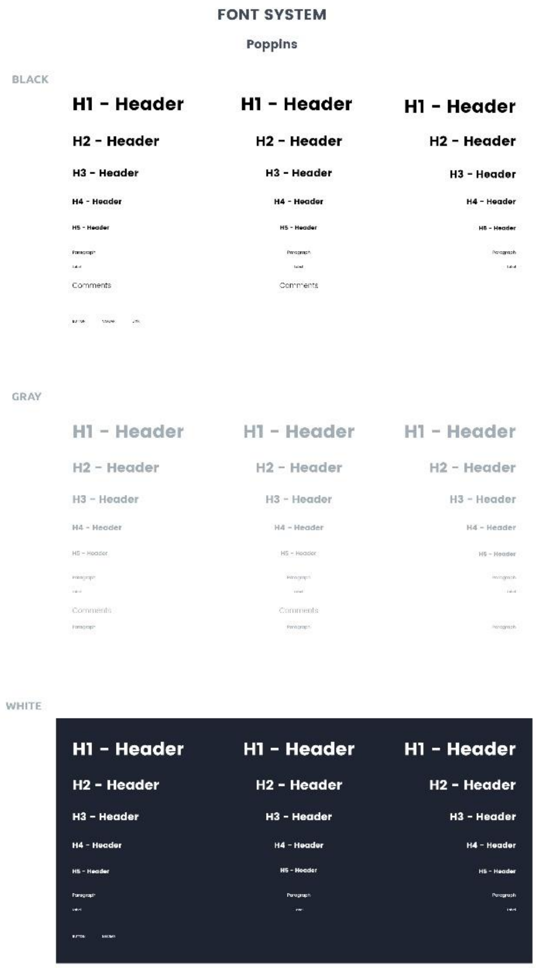


Fig 7.6: Font Systems

7.5.7 Wireframes

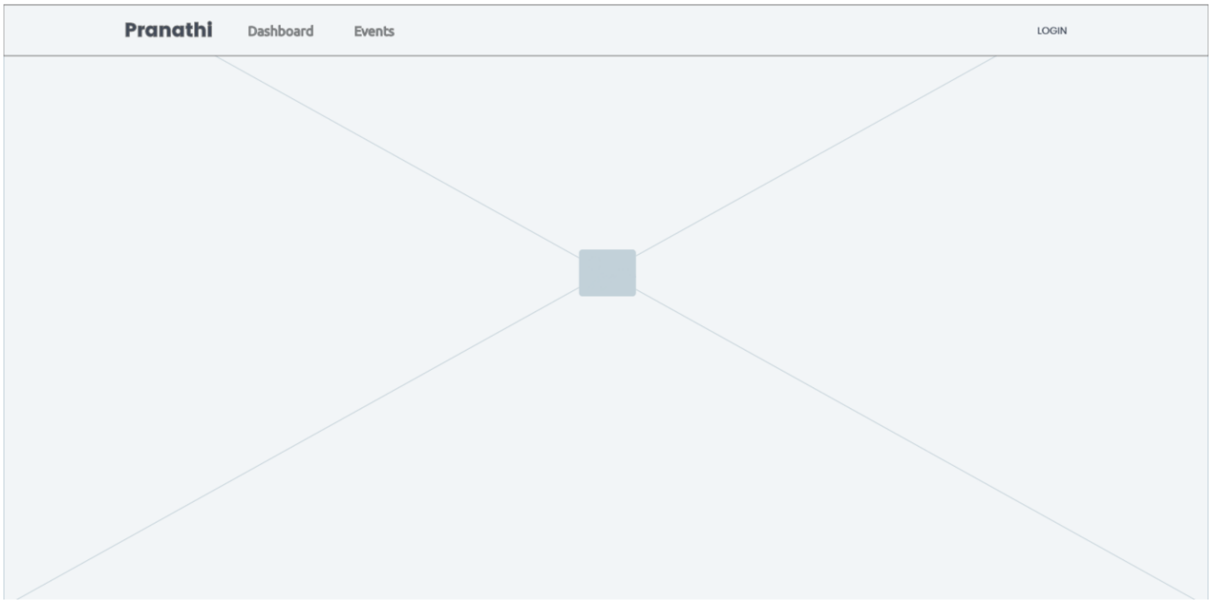


Fig 7.7: Landing Page

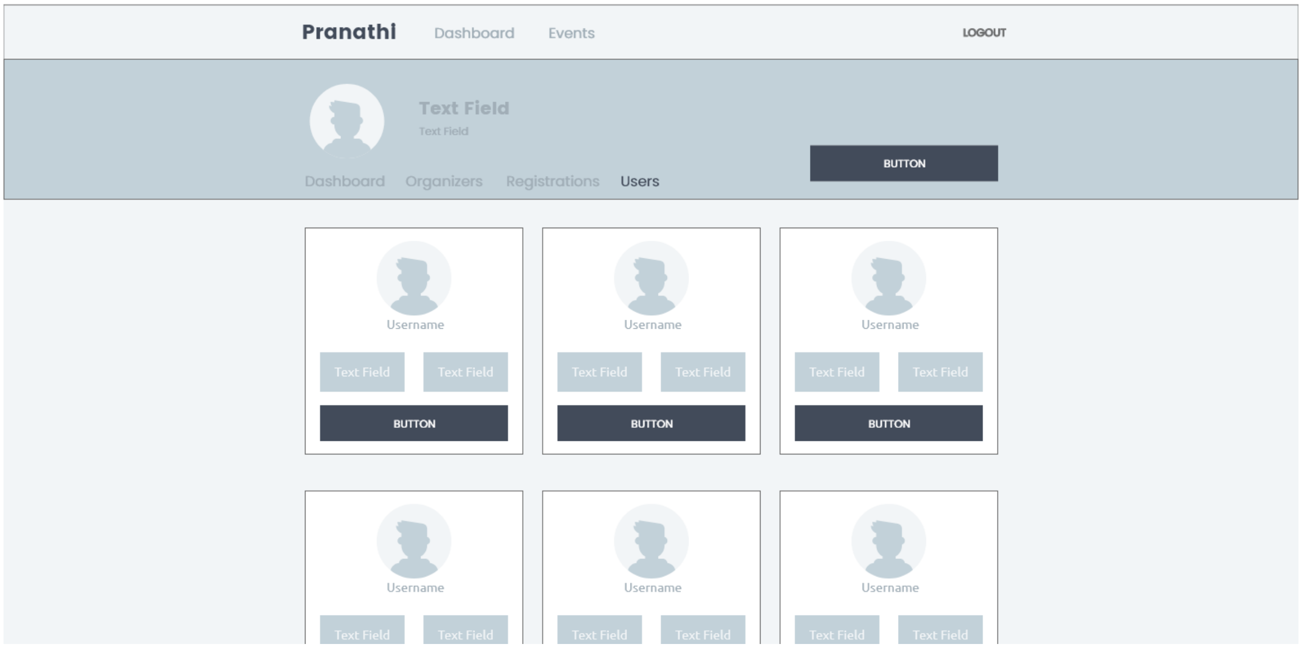


Fig 7.8 Users

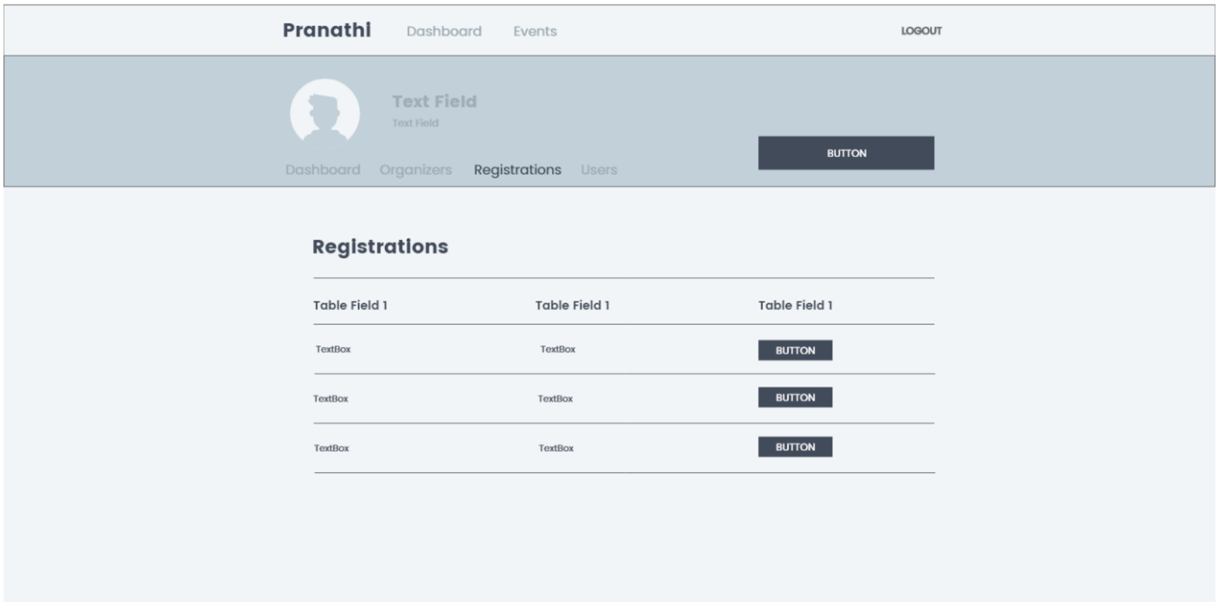


Table Field 1	Table Field 1	Table Field 1
TextBox	TextBox	BUTTON
TextBox	TextBox	BUTTON
TextBox	TextBox	BUTTON

Fig 7.9: Registrations

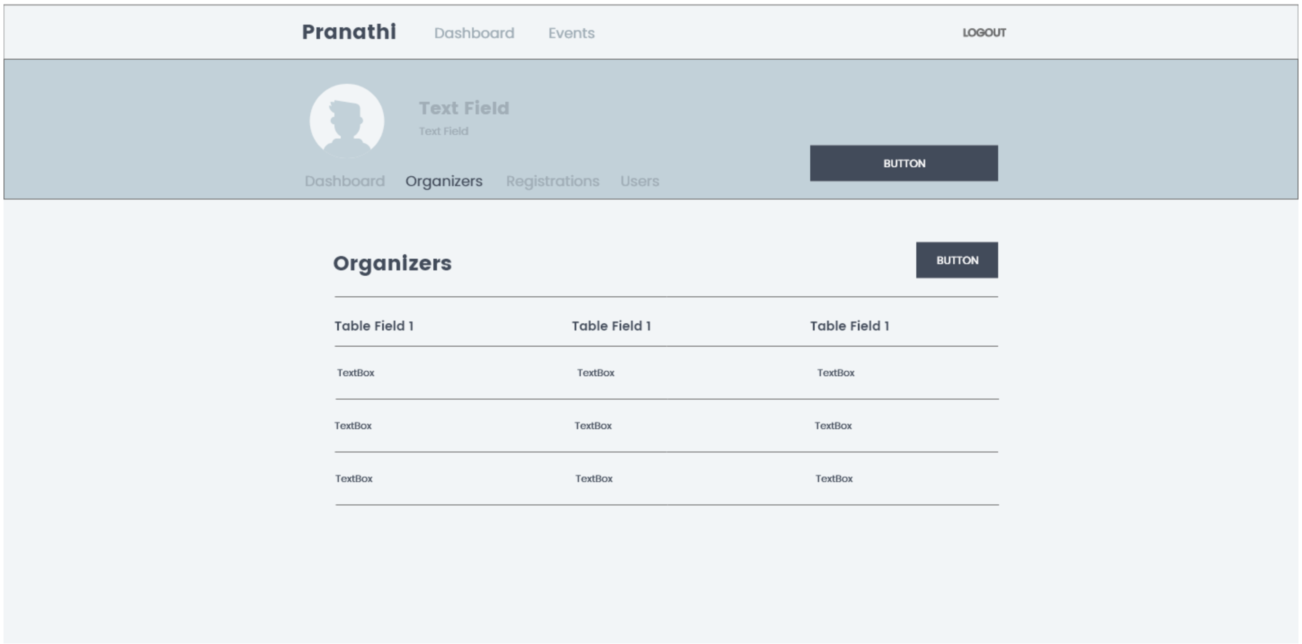


Table Field 1	Table Field 1	Table Field 1
TextBox	TextBox	TextBox
TextBox	TextBox	TextBox
TextBox	TextBox	TextBox

Fig 7.10: Organisers

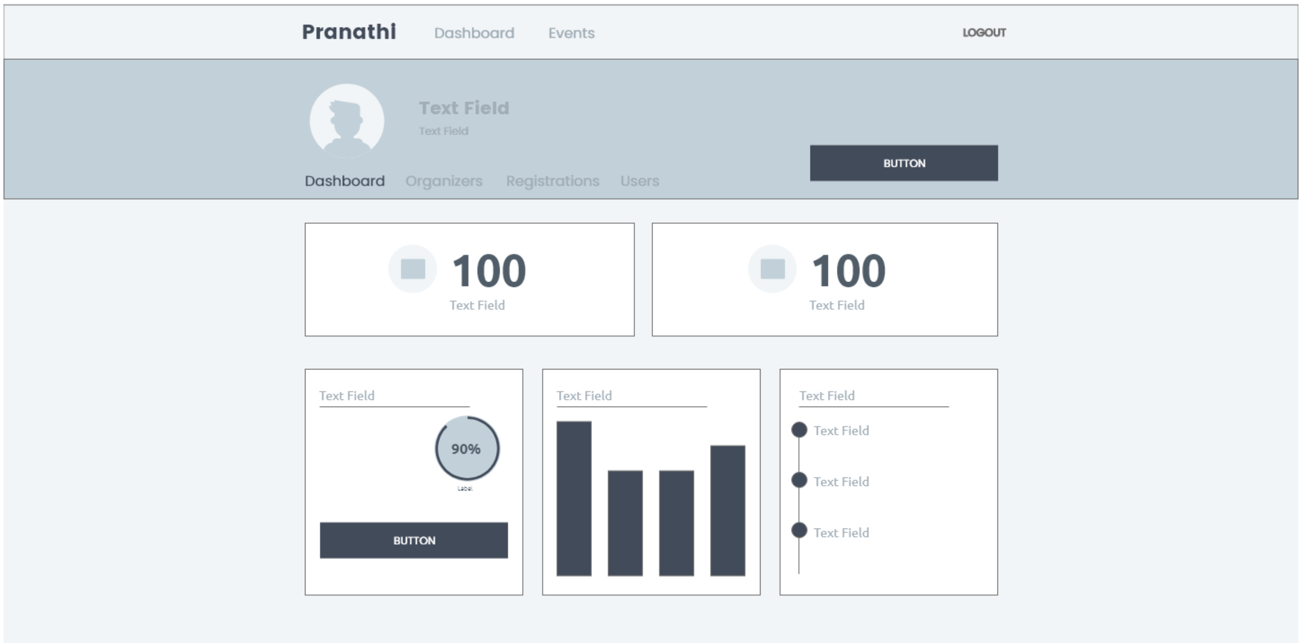


Fig 7.11: Dashboard

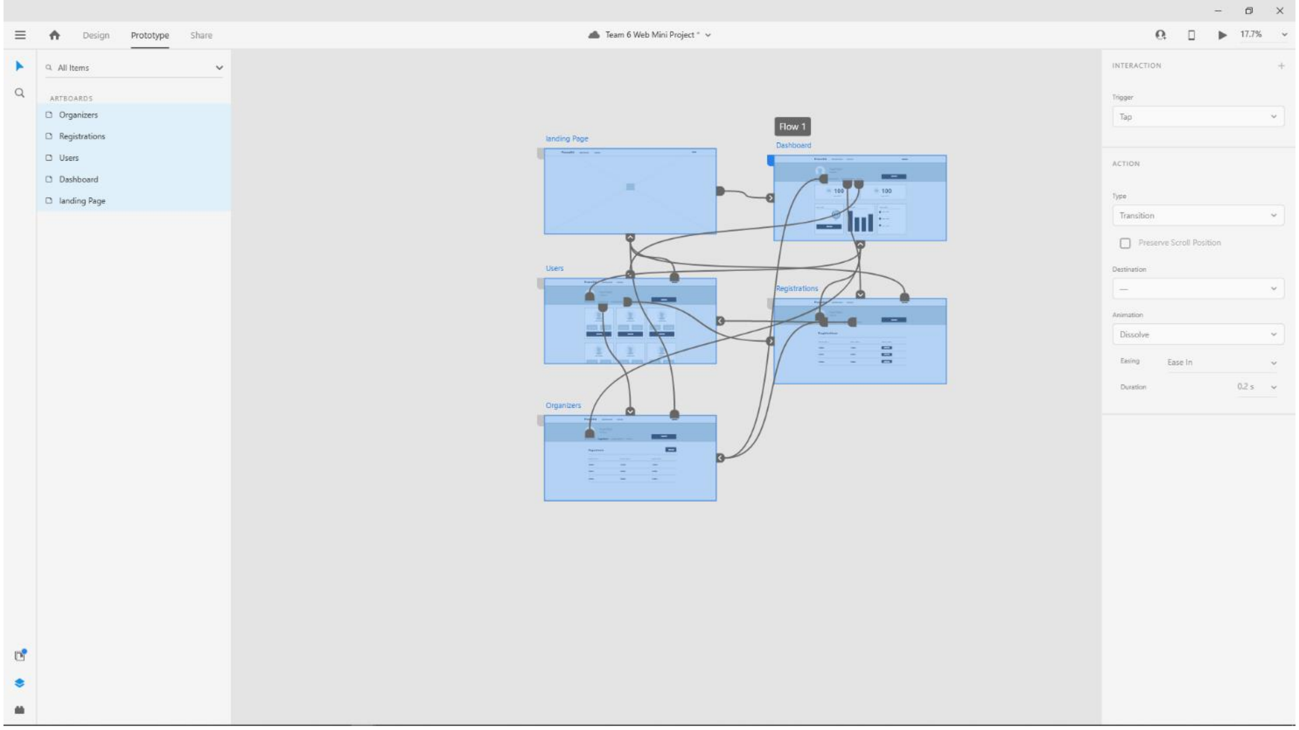


Fig 7.12: Flow Diagram

Chapter 8

IMPLEMENTATION

8.1 List of Program Files

Sl No	File Name	Description
1	Login.php	Login Form
2	Signup.php	Sign Up Form
3	Index.php	Landing Page
4	Dashboard.php	Dashboard Overview Page
5	Events.php	Events Page
6	Edit_Event.php	Edit Events
7	Add_Event.php	Add New Events
8	Delete_Event.php	Delete Events
9	Organisers.php	Organisers Page
10	Users.php	Users Page
11	Registrations.php	Participants Page
12	Select_Events.php	Select Events to register
13	Checkout.php	Checkout Form for Participant Registration
14	Update_Cart.php	Update Event Selection

8.2 Results and Snapshots

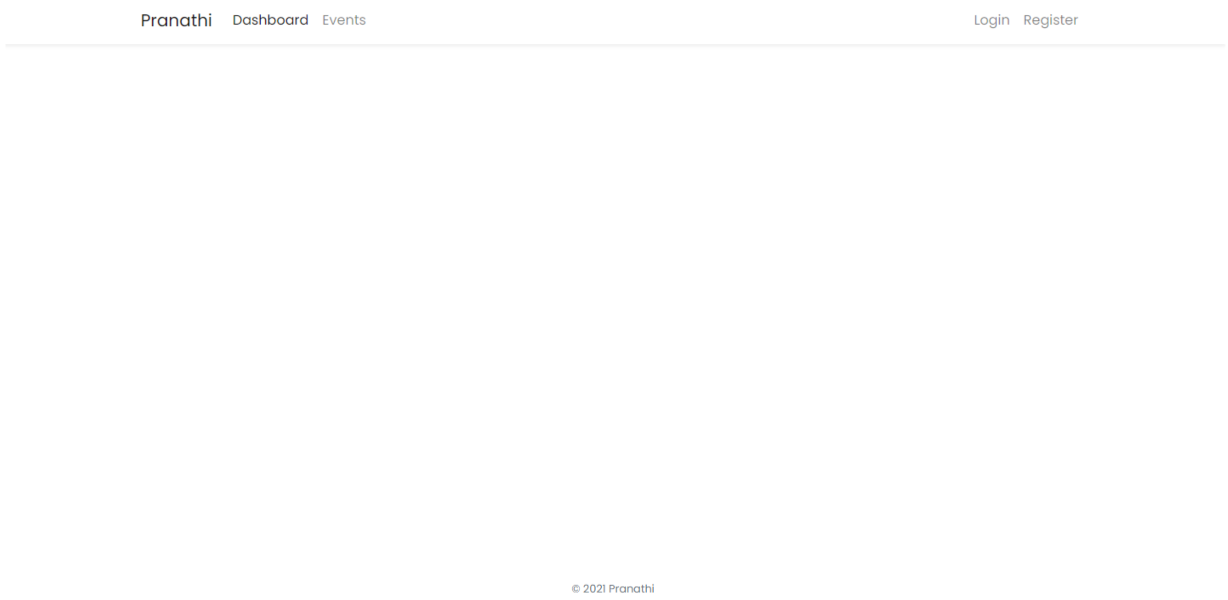
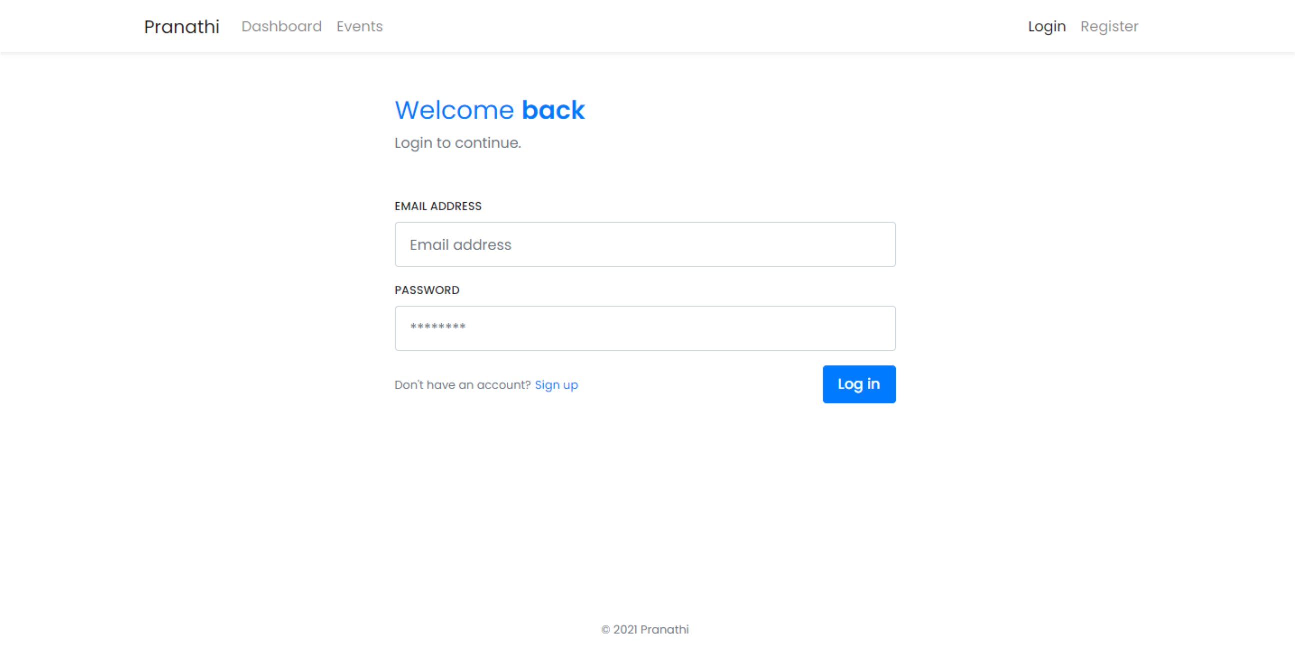


Fig 8.1: Landing Page



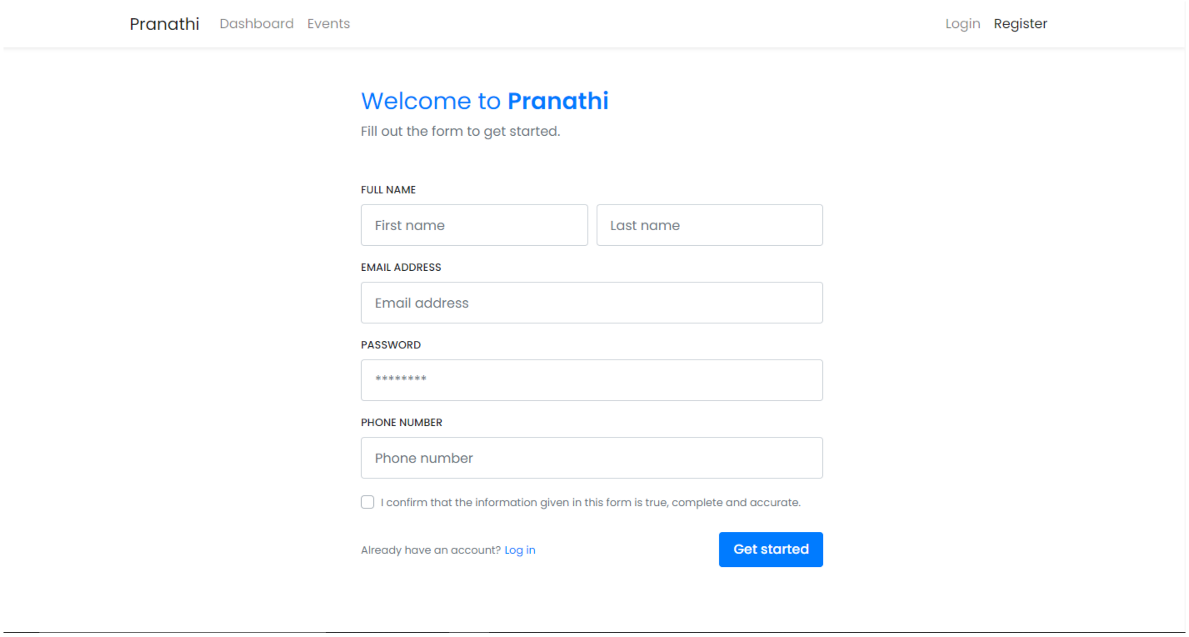
Welcome **back**

Login to continue.

Don't have an account? Sign up

© 2021 Pranathi

Fig 8.2: Login Page



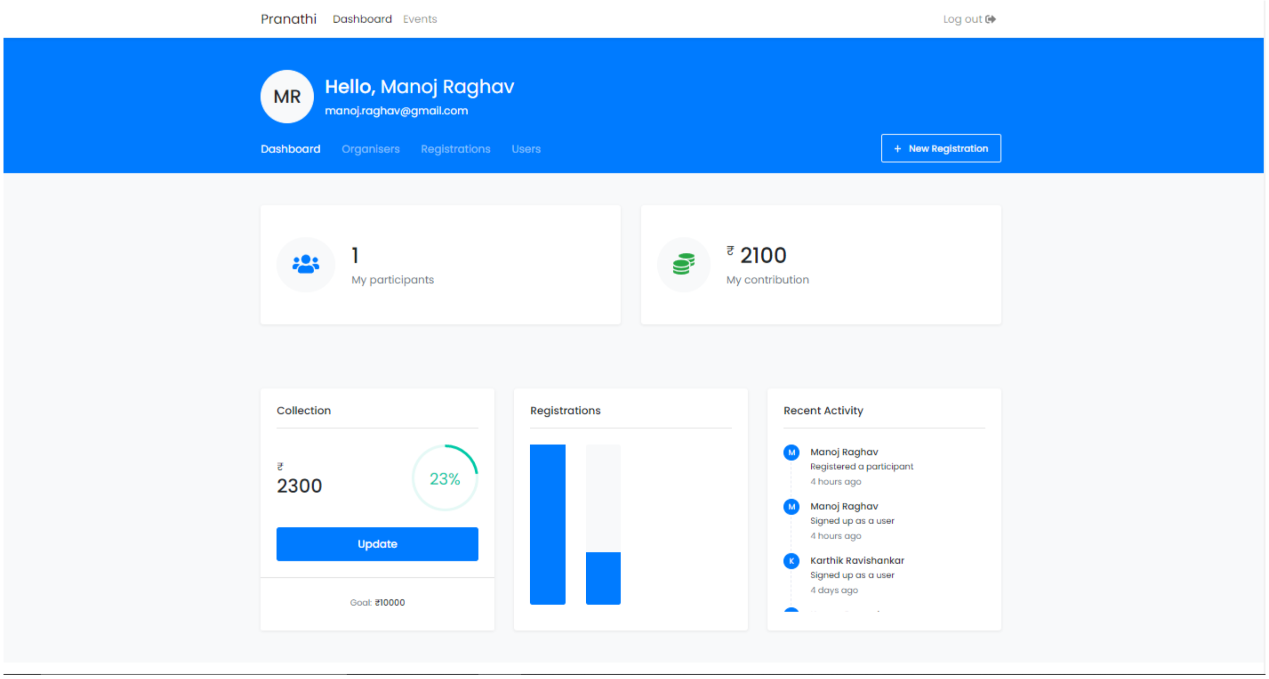
Welcome to **Pranathi**

Fill out the form to get started.

□ I confirm that the information given in this form is true, complete and accurate.

Already have an account? Log in

Fig 8.3: Sign Up Page



Hello, Manoj Raghav
manoj.raghav@gmail.com

1
My participants

₹ 2100
My contribution

₹
2300
23%

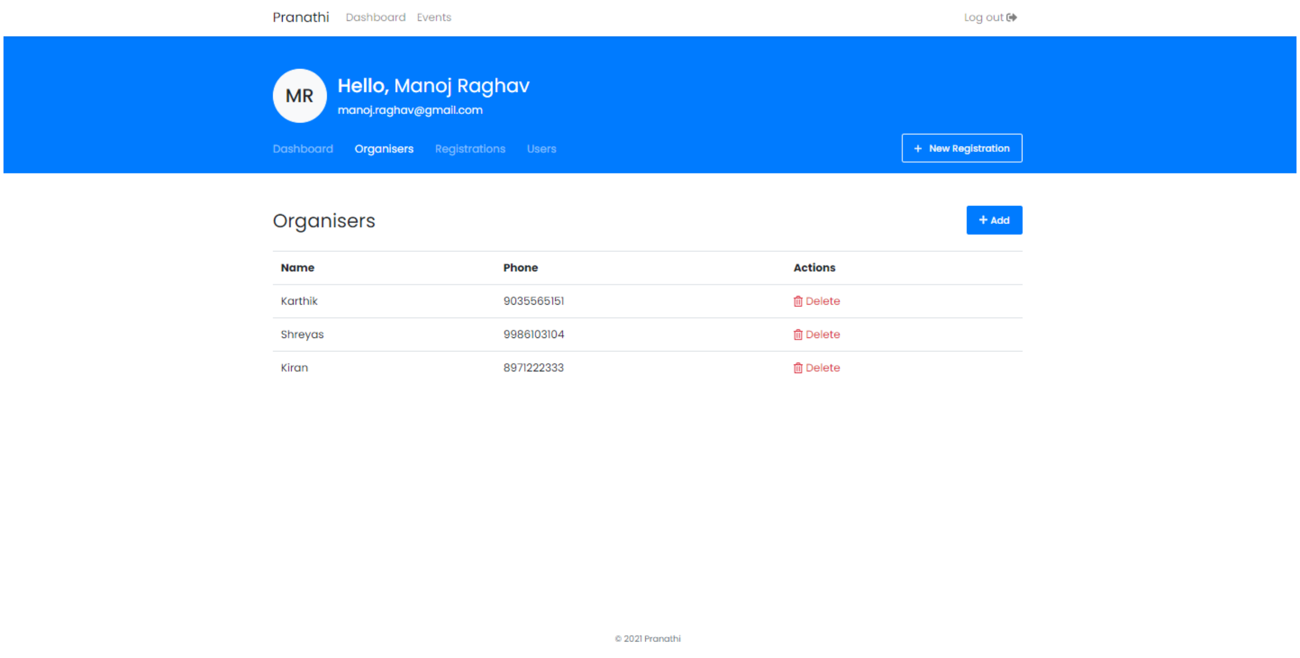
Goal: ₹10000

Manoj Raghav
Registered a participant
4 hours ago

Manoj Raghav
Signed up as a user
4 hours ago

Karthik Ravishankar
Signed up as a user
4 days ago

Fig 8.4: Dashboard Page



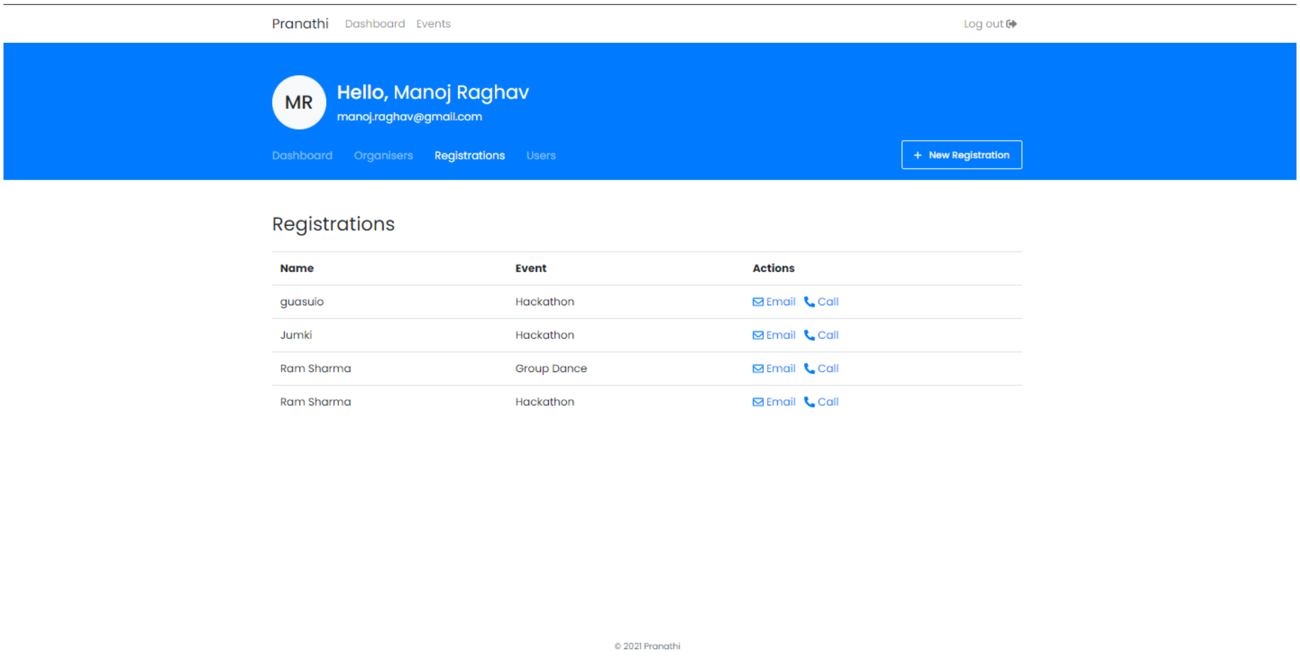
Hello, Manoj Raghav
manoj.raghav@gmail.com

Organisers

Name	Phone	Actions
Karthik	9035565151	Delete
Shreyas	9986103104	Delete
Kiran	8971222333	Delete

© 2021 Pranathi

Fig 8.5: Organisers Page



Hello, Manoj Raghav

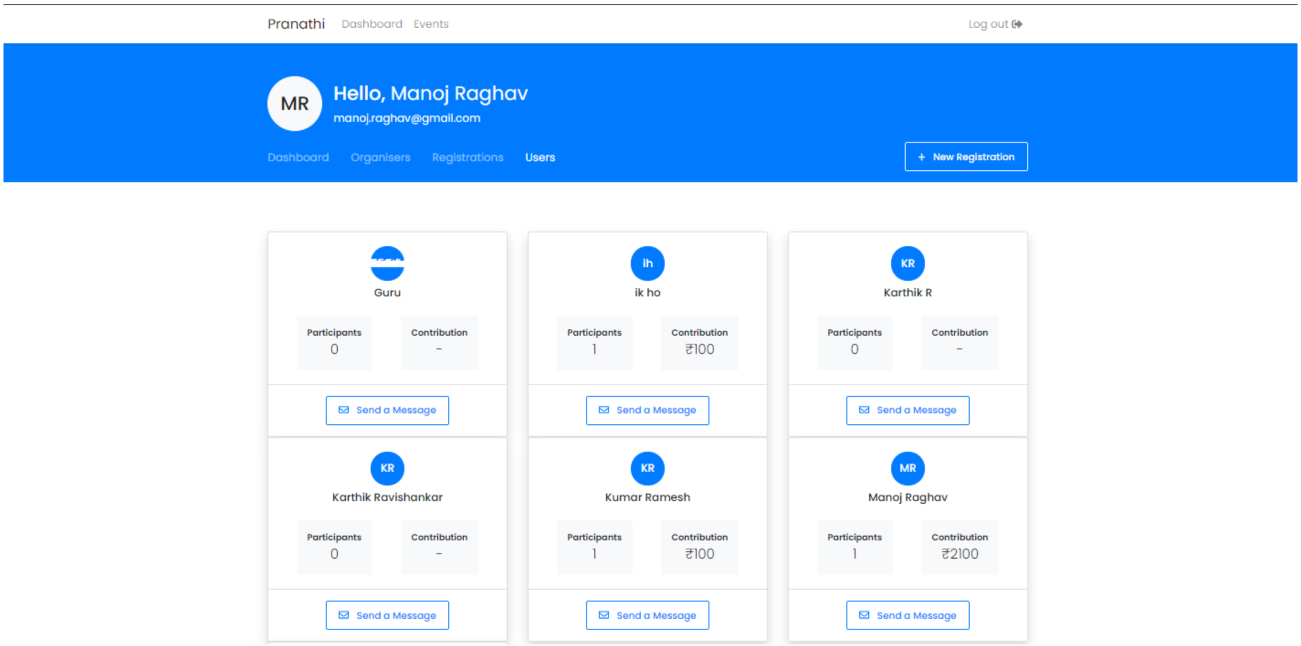
manoj.raghav@gmail.com

Registrations

Name	Event	Actions
guasuio	Hackathon	Email Call
Jumki	Hackathon	Email Call
Ram Sharma	Group Dance	Email Call
Ram Sharma	Hackathon	Email Call

© 2021 Pranathi

Fig 8.6: Registrations page

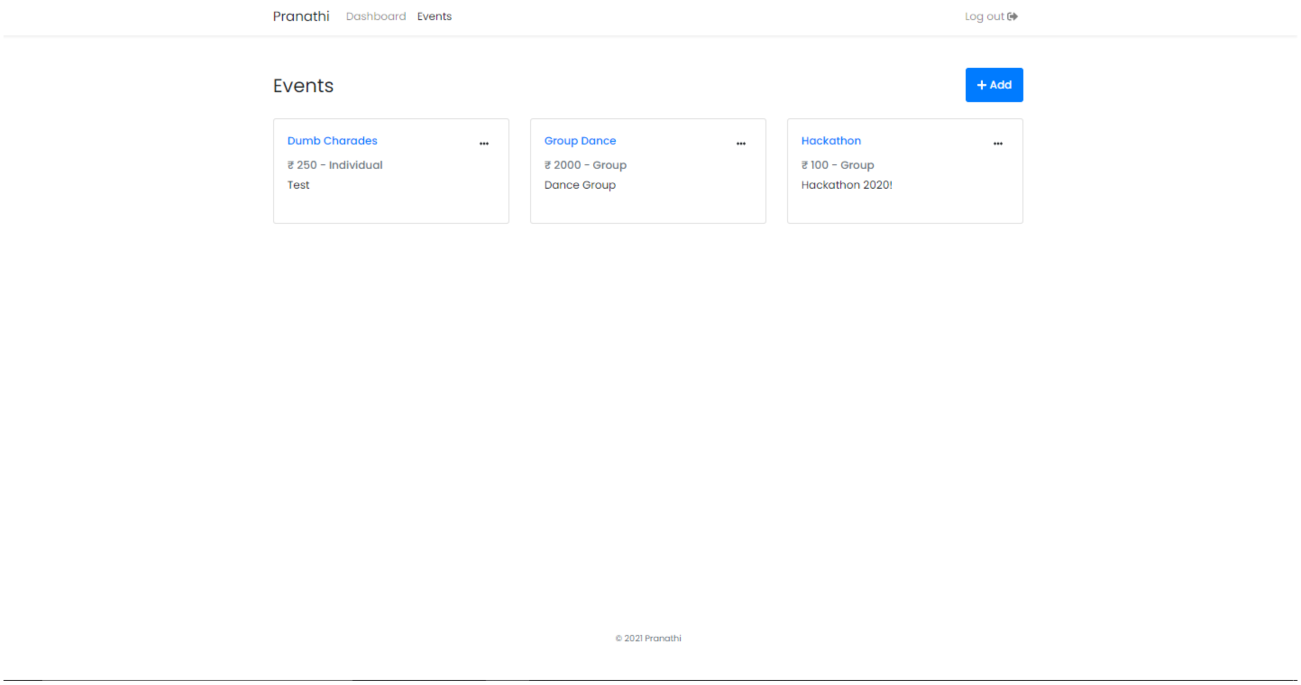


Hello, Manoj Raghav

manoj.raghav@gmail.com

Name	Participants	Contribution
Guru	0	–
ik ho	1	₹100
Karthik R	0	–
Karthik Ravishankar	0	–
Kumar Ramesh	1	₹100
Manoj Raghav	1	₹2100

Fig 8.7: Users page



Events

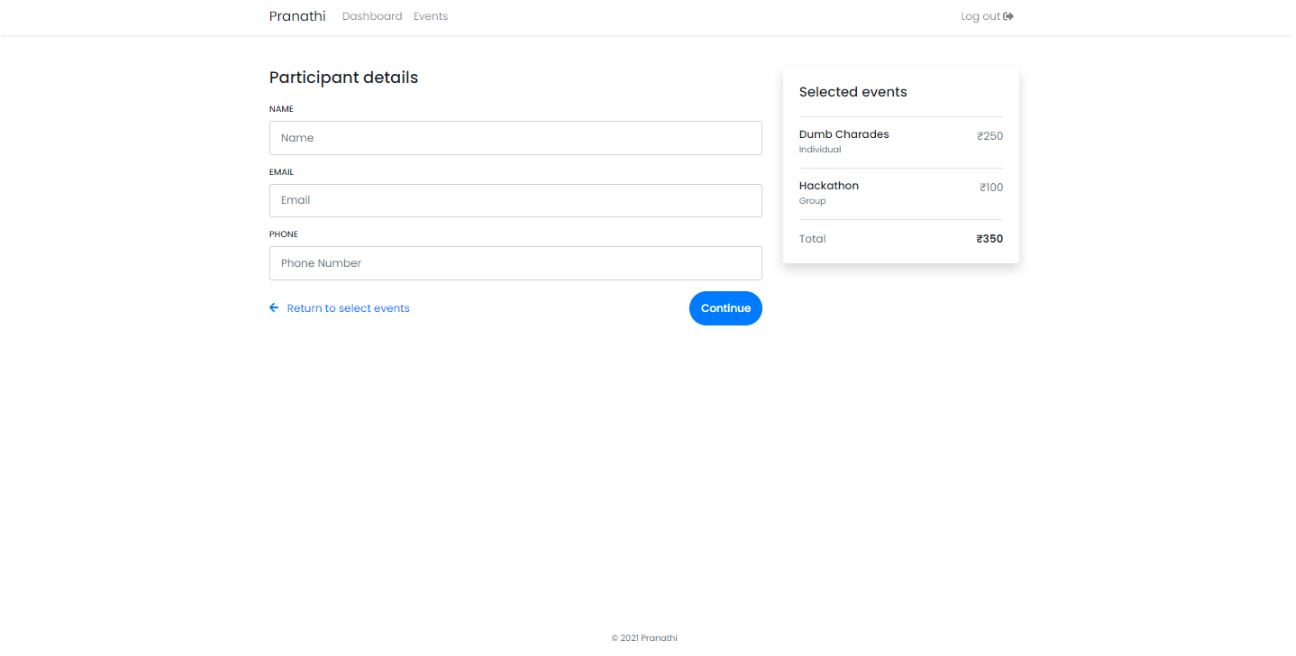
Dumb Charades
₹ 250 - Individual
Test

Group Dance
₹ 2000 - Group
Dance Group

Hackathon
₹ 100 - Group
Hackathon 2020!

© 2021 Pranathi

Fig 8.8: Events page



Participant details

NAME: Name

EMAIL: Email

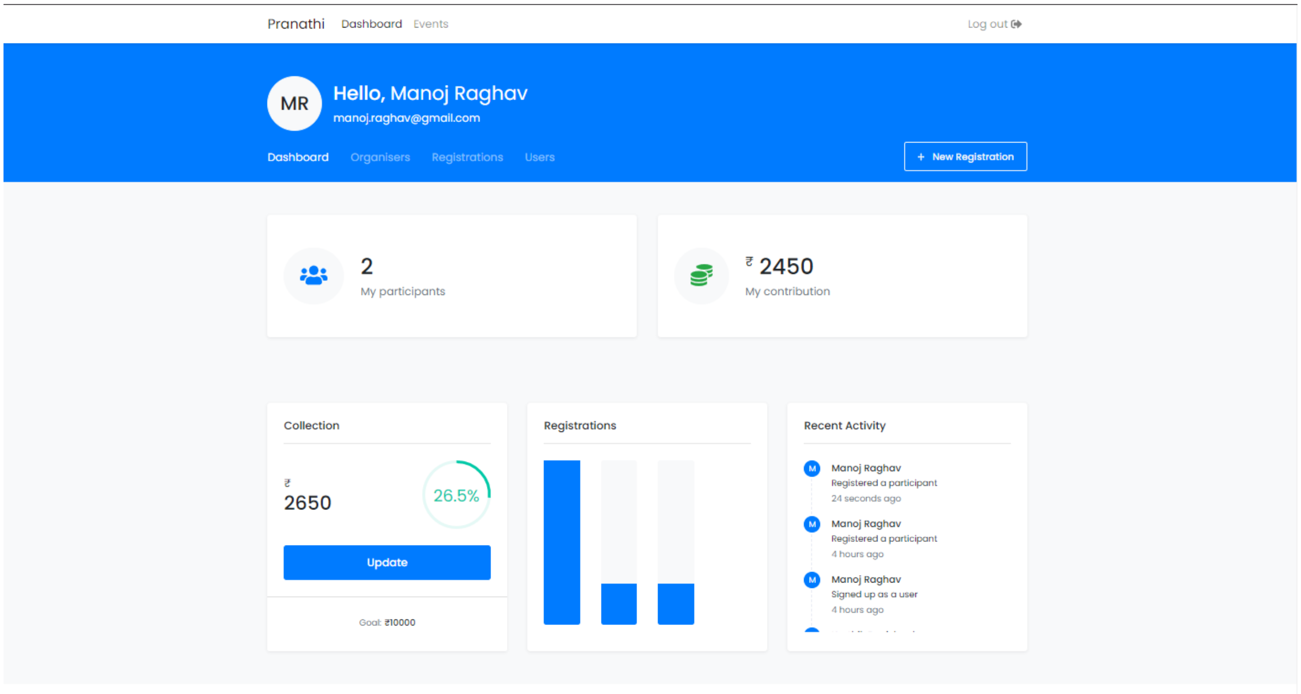
PHONE: Phone Number

Selected events

Dumb Charades Individual	₹250
Hackathon Group	₹100
Total	**₹350**

© 2021 Pranathi

Fig 8.9: Events Registration page



Hello, Manoj Raghav
manoj.raghav@gmail.com

2
My participants

₹ 2450
My contribution

Collection
₹
2650
26.5%
Goal: ₹10000

Registrations

Recent Activity

Manoj Raghav
Registered a participant
24 seconds ago

Manoj Raghav
Registered a participant
4 hours ago

Manoj Raghav
Signed up as a user
4 hours ago

Fig 8.10: Updating of Registrations

Chapter 9

TESTING

9.1 Sample tests

ID	Description	Input	Expected Output	Observed Output	Result
1	On entering unregistered email	br.karthik9@gmail.com	Exception- Username not found	Exception - Username not found	Pass
2	On entering email in invalid format	wu@gmail	Exception- Please enter a valid email address	Exception- Please enter a valid email address	Pass
3	On entering phone number less than specified size	9886611	Exception- Please specify a valid phone number	Exception- Please specify a valid phone number	Pass
4	On entering phone number more than specified size	99861041031	Exception- Please specify a valid phone number	Exception- Please specify a valid phone number	Pass
5	On entering the password wrong	11222	Exception-Enter correct password	Exception-Enter correct password	Pass

6	On trying to register participants without selecting at least one event	No event selected	Error- Select at least one event	Error- Select at least one event	Pass
7	On editing an existing event	Appropriate editing is done	Event updated successfully	Event updated successfully	Pass
8	On deleting an event	Desired event is deleted	Event deleted successfully	Event deleted successfully	Pass

Fig 9.1: Sample tests

Chapter 10

TOOLS

10.1 Editor

Visual Studio Code is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring. It is highly customizable, allowing users to change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. The source code is free and open source and released under the permissive MIT License. The compiled binaries are freeware and free for private or commercial use.

Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. Although it uses the Electron framework, the software does not use Atom and instead employs the same editor component used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services).

10.2 XAMPP

XAMPP is a free and open source cross-platform web server solution stac package developed by Apache friends consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server to a live server possible.

XAMPP's ease of deployment means a WAMP or LAMP stack can be installed quickly and simply on an operating system by a developer. With the advantage a number of common add-in applications such as WordPress and Joomla can also be installed with similar ease using Bitnami.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing content management system(CMS) like Joomla or WordPress. It is also possible to connect to local host via FTP with an HTML editor.

10.3 MySQL Query Browser

MySQL Query Browser is a cross-platform GUI client program that's intuitive and easy to use. It provides a graphical interface to the MySQL server for querying and analyzing data. It's similar in style of use to MySQL Administrator but is oriented toward accessing database contents rather than server administration.

The following list describes some of the ways that you can use MySQL Query Browser:

- Interactively enter, edit, and execute queries.
- Navigate result sets with scrolling. Multiple result sets are tabbed so that you can switch between them easily by selecting the appropriate tab.
- Browse the databases available on the server, the tables and stored routines in databases, and the columns in tables.
- Browse your query history to see what queries you've issued, or recall and re-execute previous queries.
- Bookmark queries for easy recall.
- Create or drop databases and tables, and modify the structure of existing tables.

- Create and edit SQL scripts, with debugging.
- Edit connection profiles that can be used to connect to servers more easily.
- Access information from the MySQL Reference Manual, such as statement syntax and function descriptions.

MySQL Query Browser supports multiple server connections and opens a separate window for each connection that you establish.

10.4 MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more. MySQL Workbench is available on Windows, Linux and Mac OS X

MySQL Workbench enables a DBA, developer, or data architect to visually design, model, generate, and manage databases. It includes everything a data modeler needs for creating complex ER models, forward and reverse engineering, and also delivers key features for performing difficult change management and documentation tasks that normally require much time and effort.

MySQL Workbench delivers visual tools for creating, executing, and optimizing SQL queries. The SQL Editor provides color syntax highlighting, auto-complete, reuse of SQL snippets, and execution history of SQL. The Database Connections Panel enables developers to easily manage standard database connections, including MySQL Fabric. The Object Browser provides instant access to database schema and objects.

MySQL Workbench provides a visual console to easily administer MySQL environments and gain better visibility into databases. Developers and DBAs can use the visual tools for configuring servers, administering users, performing backup and recovery, inspecting audit data, and viewing database health. MySQL Workbench provides a suite of tools to improve the performance of MySQL applications. DBAs can quickly view key performance indicators using the Performance Dashboard. Performance Reports provide easy identification and access to IO hotspots, high-cost SQL statements, and more. Plus, with 1 click, developers can see where to optimize their query with the improved and easy to use Visual Explain Plan.

MySQL Workbench now provides a complete, easy to use solution for migrating Microsoft SQL Server, Microsoft Access, Sybase ASE, PostreSQL, and other RDBMS tables, objects and data to MySQL. Developers and DBAs can quickly and easily convert existing applications to run on MySQL both on Windows and other platforms. Migration also supports migrating from earlier versions of MySQL to the latest releases.

Chapter 11

CONCLUSION

In this mini project, we have created an application which is easy to access and user friendly. Here in this application, we try to build a connectivity bridge between the patient and doctor.

The purpose of this application is to make it easier to manage the registrations and data of the participants in an organised manner.

This work is an exploration of data captured by the fest management systems can potentially be used by the members of the core teams of college fests who want to know the statistics of the different components of the fest. The main goal of this project is to make a website for users to be able to find all the relevant details of a fest in a single website for a better conduction of the fest. This mini project has the potential to become a valuable resource that can be used in event that has large footfall.

As a part of future work, we want to migrate the application into newer and more capable methods which will make the application scalable.

REFERENCES

[1] www.eventbrite.com

[2] www.yapsody.com

[3] www.ticketbud.com

[4] Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development”, 1stEdition, Pearson Education India.

[5] Robin Nixon, “Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5”, 4thEdition, O’Reilly Publications, 2015.

[6] www.wikepedia.com

[7] www.google.com