**Sanga Kalagam (Sangam) &**

**BSChatZ Android App**

**A report submitted for the course named Internship (CS-410)**

*Submitted by*

Name: **Amit Kumar**

Semester: 8th Semester

**Supervised by**

**Dr. Balram sir**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY MANIPUR**

**MANTRIPUKHRI, IMPHAL-795002, INDIA**

**May, 2023**

**Declaration**

I hereby declare that this submission is our own work and that to the

Best of our knowledge and beliefs. It contains no material previously

Published or written by neither any person nor material which to a

Substantial extent has been accepted for the award of any other degree

Or diploma of the university or other institute of higher learning, except

Where due acknowledgement has been made in the text.

Signature of Student Signature of Supervisor

Name: Amit Kumar

Roll No: 19010111

****

**Certification**

The work embodied in the present report entitled “**Sanga kalagam android app**” has been carried out in the Department of Computer Science & Engineering The work reported herein is original and does not form part of any other report or dissertation on the basis of which a degree or award was conferred on an earlier occasion or to any other student.

I understand the Institute’s policy on plagiarism and declare that the report and publications are my own work, except where specifically acknowledged and has not been copied from other sources or been previously submitted for award or assessment.

Signature of Supervisor

**Abstract**

In this report, I am working on Android app.

This app was developed by noticing the social issue with neighbors

in our society where most of the people do not know about there

neighbours and area. By taking this into notice I have developed this

app to solve this issue.

It's Android app using java and PHP ,MySQL

Use PHP for serverside and Mysql for database

Four types of login: SUPER ADMIN, ADMIN, MANAGER and USER

App name: Sanga kalagam

User role: signup/login, reference, membership validity and member fee pay (fee per year), can view sangam details, register for joining meeting.

Login using username and password

Signup using email, name, address, phone num, Ref erred by

Dashboard : membership validity, sangam details, registers for joining meeting

Admin: login with username password, controls manager(access/denied) ,Admin sets membership fee for app(yearly/monthly) share details of sangam and current/upcoming events, Controls users access(block/release) , create registration form for users fo meeting (meeting can be paid/free (admin sets fee amount) and admin can see all details of user from email to their address.

Super admin same as admin with extra feature such that super admin can restrict admin too from login.

Manager role : can restrict users.

**Acknowledgement**

Firstly, I would like to thank Dr. Kishorjit sir for guiding us through

Each and every step of the process with knowledge and support. His

Thoughts have been a constant source of inspiration for us. I would also

Like to acknowledge the contribution of all faculty members of the

Department for their kind assistance, suggestions and cooperation

Throughout the development of the project.

Finally, we would like to thank our teammates for the encouragement

And help during the project.

Signature of Student

Name: Amit Kumar

Roll No: 19010111

Contents

[Chapter 1 8](#_Toc133945348)

[Introduction 8](#_Toc133945349)

[1.1 Problem statement and Objective 8](#_Toc133945350)

[1.2 Project Features 8](#_Toc133945351)

[1.3 Project Scope 9](#_Toc133945352)

[1.4 Summary 9](#_Toc133945353)

[Chapter 2: 10](#_Toc133945354)

[Existing System Study 10](#_Toc133945355)

[2.1 Introduction 10](#_Toc133945356)

[2.2 Software and Applications 10](#_Toc133945357)

[2.3 Project SetUp 11](#_Toc133945358)

[Chapter 3: 13](#_Toc133945359)

[System Design 13](#_Toc133945360)

[3.1 Introduction 13](#_Toc133945361)

[3.2 Life Cycle 13](#_Toc133945362)

[3.2.1 Prototype Model 13](#_Toc133945363)

[3.3 Justification of Chosen Lifecycle 15](#_Toc133945364)

[3.4 Use Case Diagram of Four types Login Access 16](#_Toc133945365)

[3.5 Summary 16](#_Toc133945367)

[Chapter 4: 17](#_Toc133945368)

[Implementation & Working of App 17](#_Toc133945369)

[4.1 Introduction 17](#_Toc133945370)

[4.2 Working of Sangam App 17](#_Toc133945371)

[4.3 Working of BSChatZ Android app 29](#_Toc133945372)

[Chapter 5: 40](#_Toc133945373)

[Conclusion 40](#_Toc133945374)

[5.1 Future Work of BSChatZ 40](#_Toc133945375)

[5.2 Bibliography 40](#_Toc133945376)

# Chapter 1

# Introduction

## 1.1 Problem statement and Objective

The Android app is something almost everyone using either on a daily basis or at least once in a while. I am working on Android app.

This app was developed by noticing the social issue with neighbours

in our society where most of the people do not know about there

neighbours and area. By taking this into notice I have developed this

app to solve this issue.

It's Android app using java and PHP ,MySQL.

## 1.2 Project Features

List the core features of the project which I am planning to build

Use PHP for serverside and Mysql for database

Four types of login: SUPER ADMIN, ADMIN, MANAGER and USER

User role: signup/login, reference, membership validity and member fee pay (fee per year), can view sangam details, register for joining meeting.

Login using username and password

Signup using email, name, address, phone num, Ref erred by

Dashboard : membership validity, sangam details, registers for joining meeting

Admin: login with username password, controls manager(access/denied) ,Admin sets membership fee for app(yearly/monthly) share details of sangam and current/upcoming events, Controls users access(block/release) , create registration form for users fo meeting (meeting can be paid/free (admin sets fee amount) and admin can see all details of user from email to their address.

Super admin same as admin with extra feature such that super admin can restrict admin too from login.

Manager role : can restrict users.

## 1.3 Project Scope

The app can be updated or modified in future with respect to user

requirements. The app can be updated by making changes in user

interface, user interaction and customer support.

## 1.4 Summary

This app was developed by noticing the social issue with neighbours

in our society where most of the people do not know about there

neighbours and area. By taking this into notice I have developed this

app to solve this issue.

# Chapter 2:

# Existing System Study

## 2.1 Introduction

Sanga kalagam Android app using java and PHP ,MySQL. So in this chapter we talk about software and application,project setup.

## 2.2 Software and Applications

1. **Android Studio** : Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.
2. **Java**: Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java succeeds at this by being platform-independent at both the source and binary levels.
3. **PHP:**  PHP is a server-side scripting language used to develop websites and web applications. It is a go-between for the server and the front end of a website or web application that manages dynamic content, databases and such. PHP's purpose is **to create websites and web applications**.
4. **MySQL :** MySQL is ideal for storing application data, specifically web application data. Additionally you should use MySQL if you need a relational database which stores data across multiple tables. As MySQL is a relational database, it's a good fit for applications that rely heavily on multi-row transactions.
5. **XML :** XML stands for eXtensible Markup Language, which is a way of describing data using a text-based document. Because XML is extensible and very flexible, it's used for many different things, including defining the UI layout of Android apps.

## 2.3 Project SetUp

**Android Studio**: firstly we create new folder name Sanga then I will install dependecies for my project .Then I will create folder and file for coding.

plugins **{** id 'com.android.application'  
 id 'com.google.gms.google-services'  
  
**}**android **{** compileSdk 32  
  
 defaultConfig **{** applicationId "com.visanka.sangam"  
 minSdk 23  
 targetSdk 32  
 versionCode 1  
 versionName "1.0"  
  
 testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"  
 **}** buildTypes **{** release **{** minifyEnabled false  
 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'  
 **}  
 }** compileOptions **{** sourceCompatibility JavaVersion.*VERSION\_1\_8* targetCompatibility JavaVersion.*VERSION\_1\_8* **}  
}**dependencies **{** implementation 'androidx.appcompat:appcompat:1.5.0'  
 implementation 'com.google.android.material:material:1.6.1'  
 implementation 'androidx.constraintlayout:constraintlayout:2.1.4'  
 implementation 'androidx.constraintlayout:constraintlayout-core:1.0.4'  
 implementation 'com.google.firebase:firebase-messaging:23.1.2'  
  
 testImplementation 'junit:junit:4.13.2'  
 androidTestImplementation 'androidx.test.ext:junit:1.1.3'  
 androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'  
 implementation 'com.android.volley:volley:1.2.1'  
 implementation 'com.razorpay:checkout:1.6.26'  
  
**}**

package com.visanka.sangam;  
  
public class params {  
  
 public static final String *FILE\_URL* ="http://192.168.239.234/php/Register.php";  
 public static final String *MANAGER\_URL* ="http://192.168.239.234/php/manager.php";  
 public static final String *LOGIN\_URL* ="http://192.168.239.234/php/login.php";  
 public static final String *SUPERADMIN\_URL* ="http://192.168.239.234/php/superadmin.php";  
  
 public static final String *ADMIN\_URL* ="http://192.168.239.234/php/admin.php";  
  
}

# Chapter 3:

# System Design

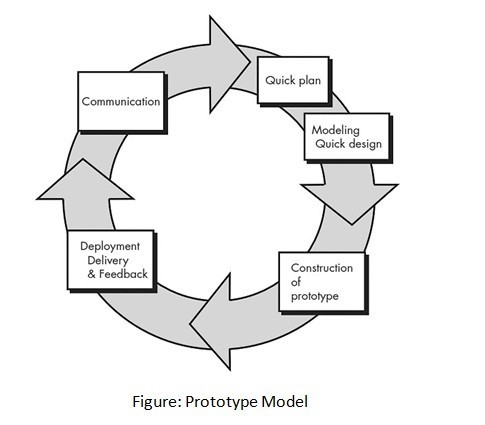
## 3.1 Introduction

This chapter focuses on the systematic approaches to be adopted to guide help in the starting to the completion of the project, it will mainly focus on a development life cycle model and its advantages and disadvantages. The chapter aims to conclude with why the methodology was chosen for the duration of the project and the reasons behind choosing it. If the correct methodology is chosen and followed correctly, it should help to ensure the project stays on schedule.

## 3.2 Life Cycle

### 3.2.1 Prototype Model

In Prototype Model, a throw-away prototype is built with potentially few features included to closely understand the requirements. The prototype is not the complete system because many of the features are not built in the prototype, it is simply a prototype of what the final system will look like so that the client/user can get close feel of the system before the final system is even built.



**Figure 1 prototype model**

**Advantages of Prototype Model to the project:**

* Errors can be detected much earlier in the lifecycle since therefore requirements can be changed with the feedback of the client, who are the students and myself in this case, if the error is time consuming to fix, as the project will have to be delivered on time.
* Feedback can be gathered early on in the development lifecycle through prototypes, this means that the feedback received from supervisor and participants students can be applied earlier than later in the lifecycle, saving time and helping to complete the project on schedule. This also means that the final system will satisfy the problem identified, as feedbacks from both supervisors and students will be regularly inputted in very prototype developed.

**Disadvantages of Prototype Model to the project:**

* Prototype have to be regularly developed which means, this could lead to continuous implantation and repairing, which could be time consuming.
* This methodology could increase problems, such as getting attached to prototype built and using that prototype design to the final system rather than changing it according to the feedback received. This means the final system cannot be according to the requirements and the requirements are not met.

## 3.3 Justification of Chosen Lifecycle

The chosen lifecycle methodology to be followed by the system was Prototype Model. This methodology was chosen because it allows to work on different aspects of the system requirements separately, this allows to get feedbacks on different prototypes very quickly and allows changes to be made, so that the final system matches with the requirements specified. Furthermore, it also helps to see that the project schedule, Gantt chart, is accurate and achievable. The chosen lifecycle has different variations which can be used, chosen is that Evolutionary Prototyping variation is very effective for myself. It is first time making a system which requires different components to be researched and implemented because usually this was done in a group which meant shared workload and made it easier as each member were allocated certain aspects of the system. However, as this time it is an individual work, Evolutionary Prototyping lifecycle helps to create a prototype and keep adding new prototypes on top of the initial prototype and tested along with the components of the first prototype, i.e., Database, will have to bear in mind that the first prototype must be robust. This is better than the Waterfall .

## 3.4 Use Case Diagram of Four types Login Access

## 

## 3.5 Summary

The chapter aims to conclude with why the methodology was chosen for the duration of the project and the reasons behind choosing it. If the correct methodology is chosen and followed correctly, it should help to ensure the project stays on schedule.

# Chapter 4:

# Implementation & Working of App

## 4.1 Introduction

So in this chapter we talk about coding and testing part.

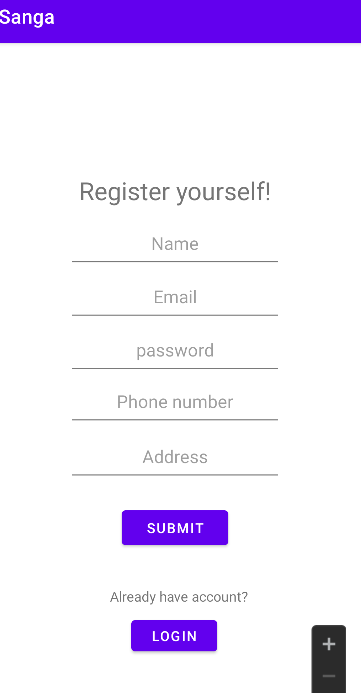
## 4.2 Working of Sangam App

 Created 1 st Registration page with Name, Email, Password,

Phone Number, Address, as Input Fields. Provided two button,

one is to submit the registration and another button is to visit

login page if already registered.

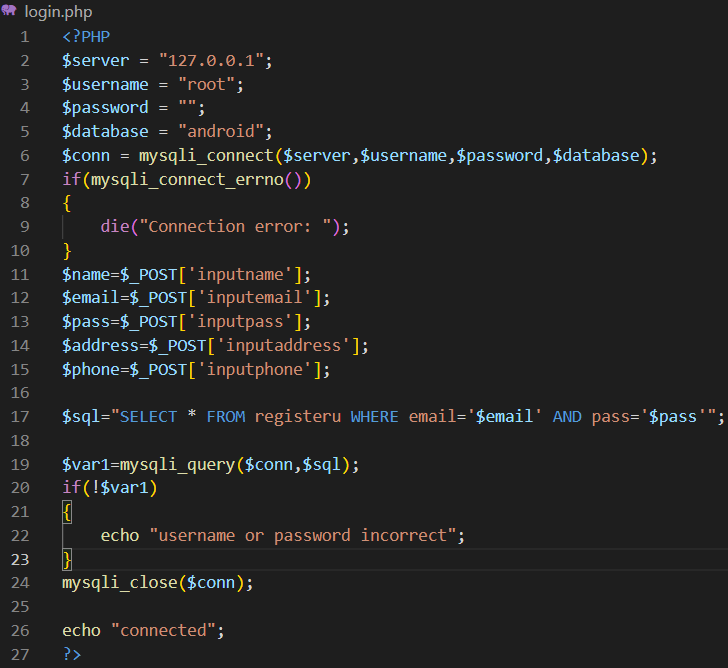


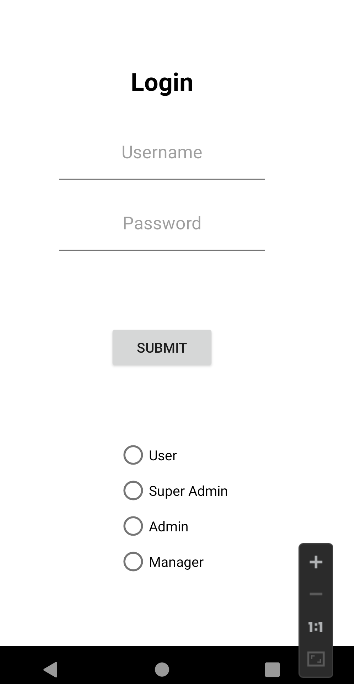


 Created 2 nd Login Page with username and password input

fields with multiple user type logins. User, Super Admin, Admin,

Manager are 4 types of login types.





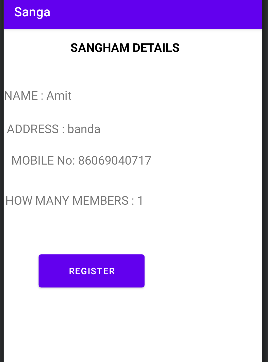
User: User performs login and lands on dashboard page which

is 3 rd page with respect to its login types. User finds different

types of sangams in their dashboard.



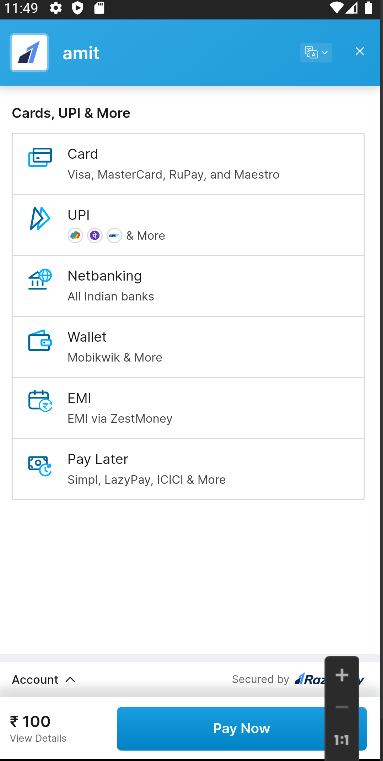
Selecting on of the sangam will send user on ‘sangam details’ page which is 4 th page of application.



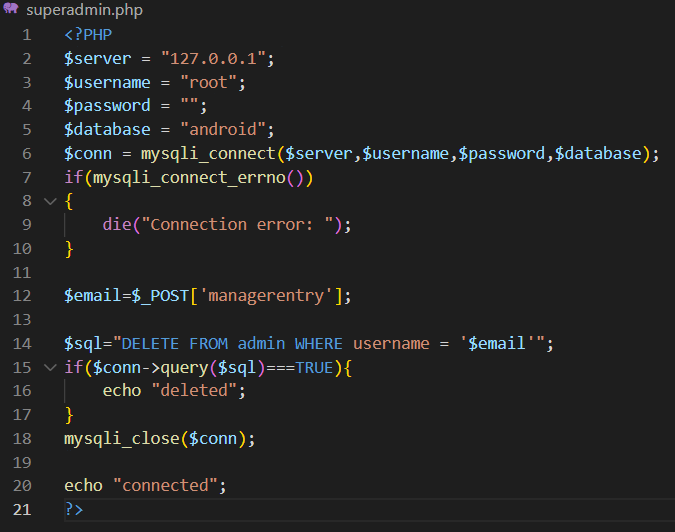
On Selecting Register button ,this will go to payment page.



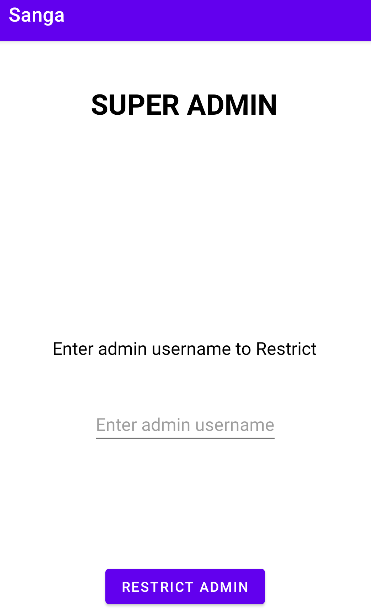
Now select ‘pay’ button.



Super Admin: Super Admin has power to restrict or delete admin from database by using PHP code.



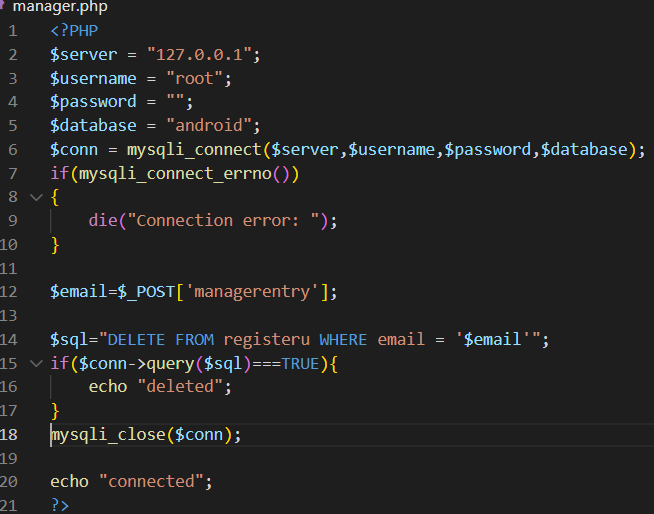
Super Admin has to provide username or admin ID in input field which is provided on 5 th Super Admin page to delete/restrict admin

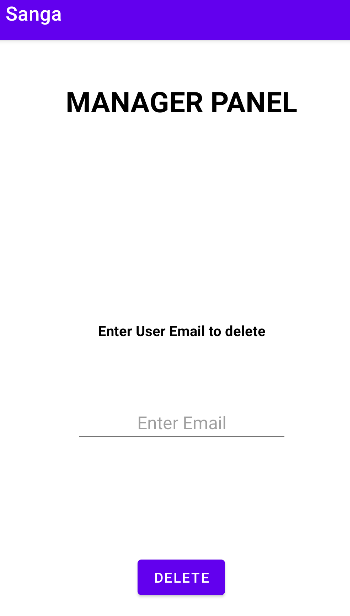


Manager: Manager has power to delete users from their

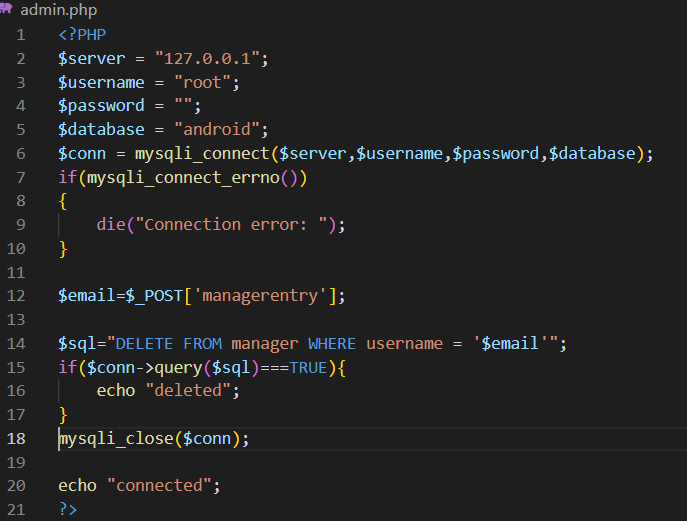
application on basis of their email. This page is also 6 th page

of application.

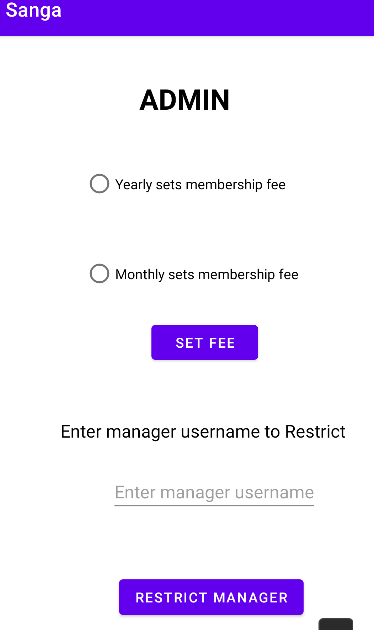




Admin: Admin has power to restrict or delete manager from database by using PHP code.

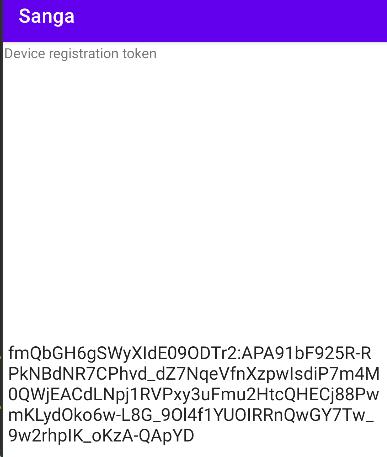


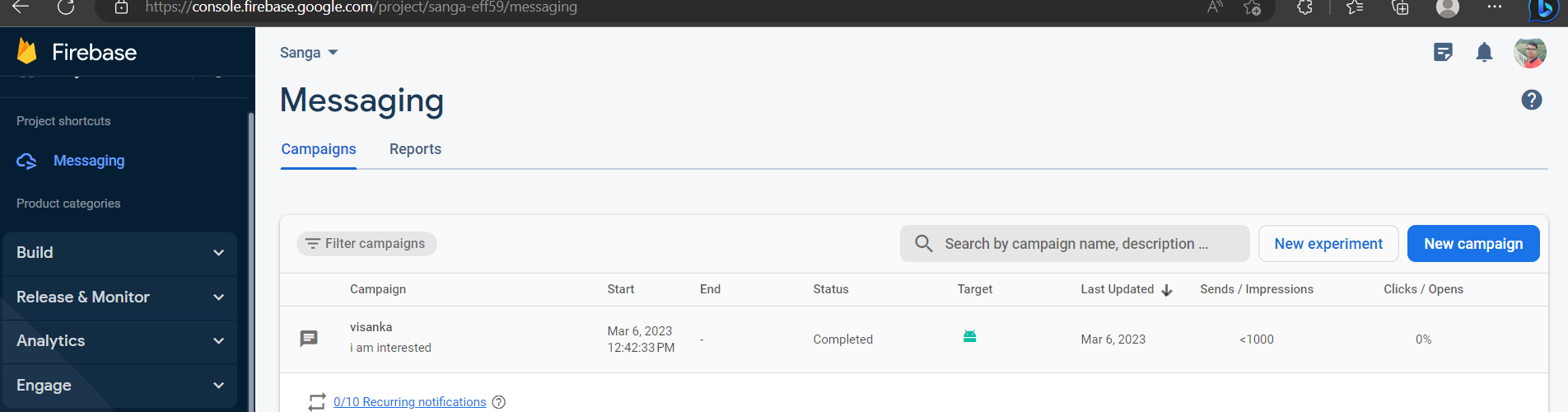
**Admin has to provide username of manager in input field which is provided on Admin page to delete/restrict manager. Admin can sets yearly or monthly membership fee for app.**



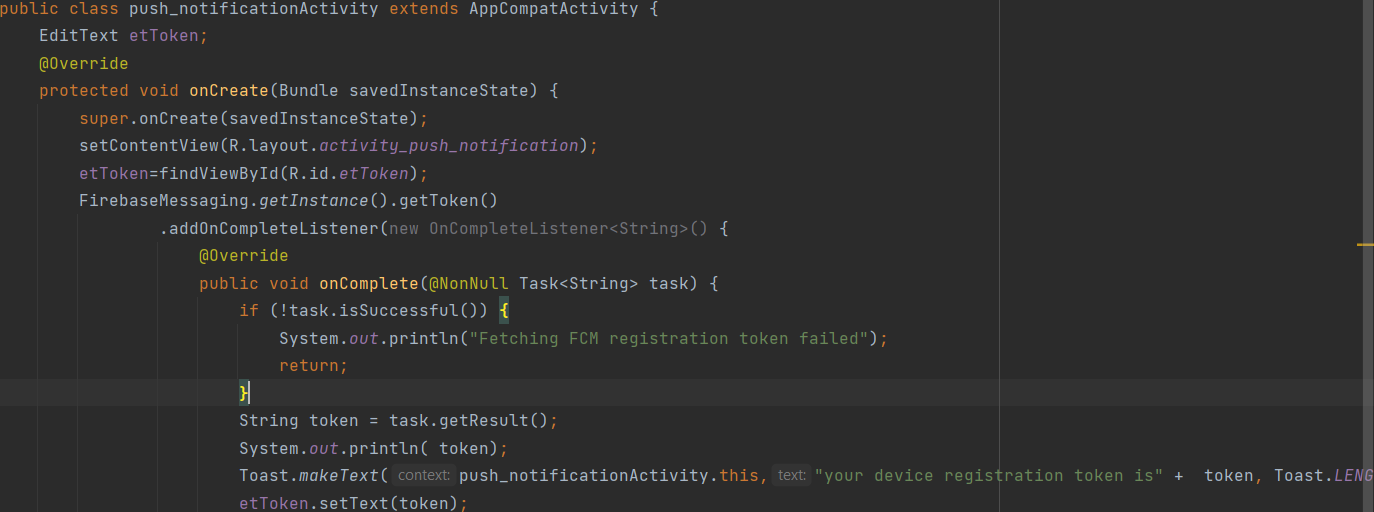


**Push Notification service by using Firebase.**



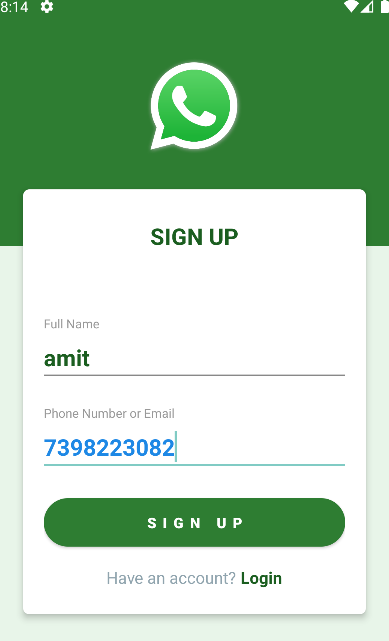


**Firebase Cloud Messaging (FCM) provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages and notifications on iOS, Android, and the web at no cost.**



## 4.3 Working of BSChatZ Android app

 Created 1 st Registration page with Name, Email or Phone Number. Provided two button,one is to submit the registration and another button is to visit login page if already registered.



<?php

    if(isset($\_POST['name'])){

        require\_once 'functions.php';

        $phone\_number = $\_POST['phone\_number'];

        $name = $\_POST['name'];

        $users = get\_user\_by\_phone($phone\_number);

        $resp['code'] = 0;

        $resp['message'] = "";

        $resp['data'] = "";

        if(!empty($users)){

            $resp['code'] = 0;

            $resp['message'] = "There is a user with same phone number.";

            $resp['data'] = "";

            echo  json\_encode($resp);

            die();

        }

        $reg\_date = time();

        $last\_seen = time();

        $profile\_photo = "";

        $sql = "INSERT INTO users (

                    name,

                    phone\_number,

                    reg\_date,

                    last\_seen,

                    profile\_photo

                ) VALUES (

                    '{$name}',

                    '{$phone\_number}',

                    {$reg\_date},

                    {$last\_seen},

                    '{$profile\_photo}'

            )";

        if($conn->query($sql)){

            $resp['code'] = 1;

            $resp['message'] = "Account was created successfully.";

            $resp['data'] = "";

            $users = get\_user\_by\_phone($phone\_number);

            if(!empty($users)){

                $resp['data'] = json\_encode($users[0]);

            }

            echo  json\_encode($resp);

            die();

        }else{

            $resp['code'] = 0;

            $resp['message'] = "Failed  because ".$conn->error;

            $resp['data'] = "";

            echo  json\_encode($resp);

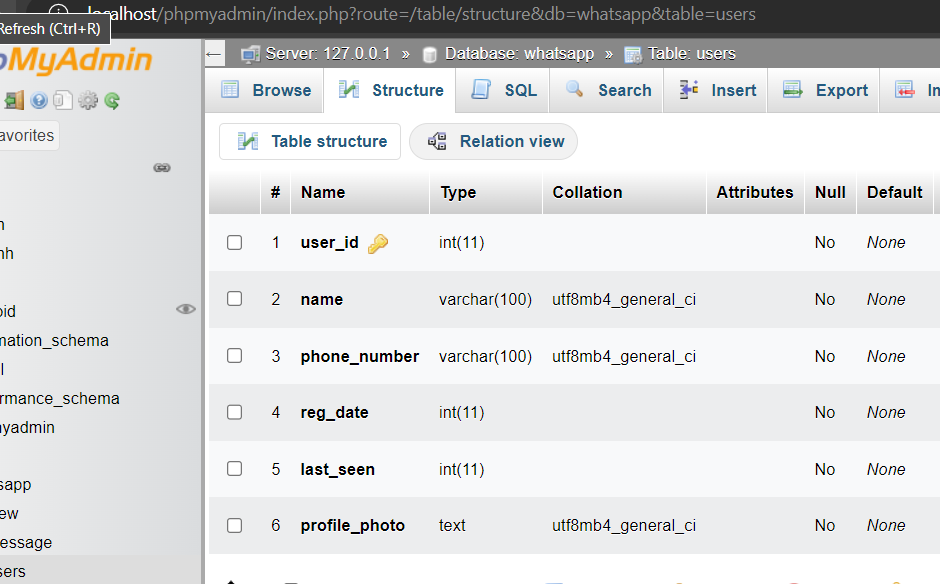
            die();

        }

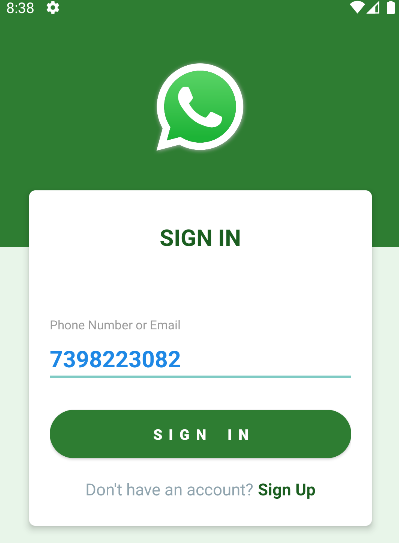
        die();

    }

 ?>



 Created 2 nd Login Page with Email or Phone Number.



function get\_user\_by\_phone($phone) {

    $sql = "SELECT \* FROM users WHERE phone\_number = '{$phone}'";

    global $conn;

    $res = $conn->query($sql);

    $users = array();

    while ($user = $res->fetch\_assoc()) {

        array\_push($users, $user);

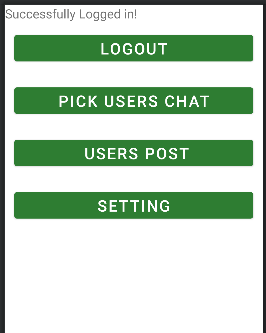
    }

    return $users;

}

User: User performs login and lands on dashboard page which

is 3 rd page. User finds different users for chat and logout.



function get\_user\_by\_phone($phone) {

    $sql = "SELECT \* FROM users WHERE phone\_number = '{$phone}'";

    global $conn;

    $res = $conn->query($sql);

    $users = array();

    while ($user = $res->fetch\_assoc()) {

        array\_push($users, $user);

    }

    return $users;

}

function get\_users() {

    $sql = "SELECT \* FROM users ORDER BY name ASC";

    global $conn;

    $res = $conn->query($sql);

    $users = array();

    while ($user = $res->fetch\_assoc()) {

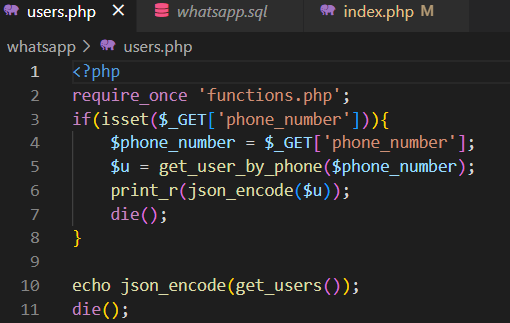
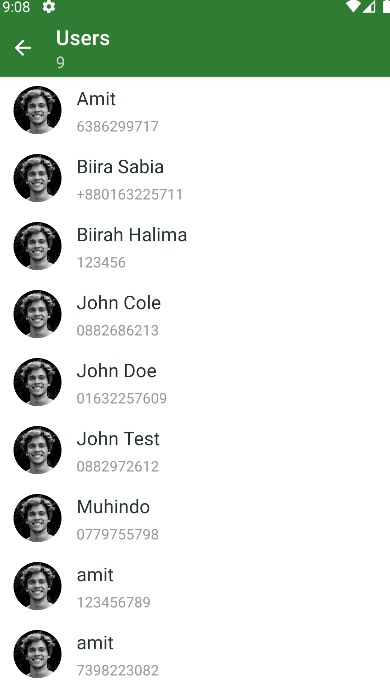
        array\_push($users, $user);

    }

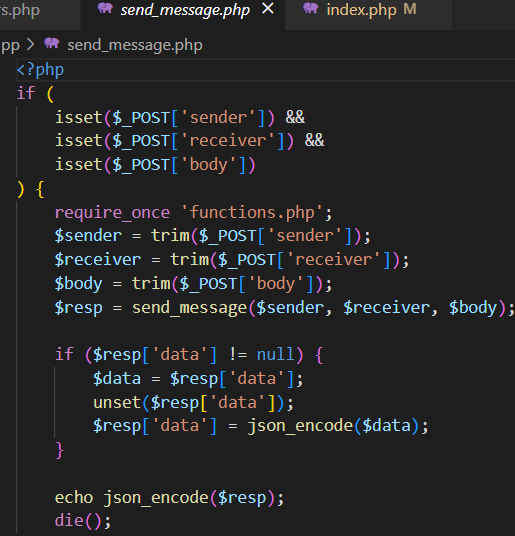
    return $users;

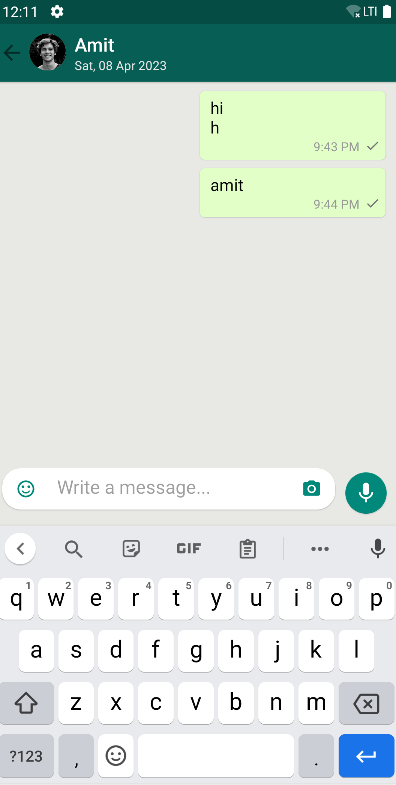
}

Selecting on of the Pick User will send user on different users for chats.



**Send messages work by this PHP code:**





function send\_message($sender, $receiver, $body) {

    global $conn;

    $chat\_thread = get\_chat\_thread($sender, $receiver);

    $body = mysqli\_real\_escape\_string($conn, $body);

    $message\_type = "text";

    $receive\_time = "sent";

    $sent\_time = time();

    $last\_update = time();

    $seen = 0;

    $sql = "

            INSERT INTO message (

                body,

                chat\_thread,

                message\_type,

                receiver,

                sender,

                receive\_time,

                sent\_time,

                last\_update,

                seen

            ) VALUES (

                '{$body}',

                '{$chat\_thread}',

                '{$message\_type}',

                '{$receiver}',

                '{$sender}',

                '{$receive\_time}',

                '{$sent\_time}',

                {$last\_update},

                {$seen}

            )

    ";

**Receive messages work by this PHP code:**

function get\_last\_message($chat\_thread) {

    global $conn;

    $sql = "SELECT \* FROM message,users WHERE chat\_thread = '{$chat\_thread}' AND user\_id = message.receiver  ORDER BY message\_id DESC LIMIT 1";

    $res = $conn->query($sql);

    if ($res->num\_rows > 0) {

        $msg = $res->fetch\_assoc();

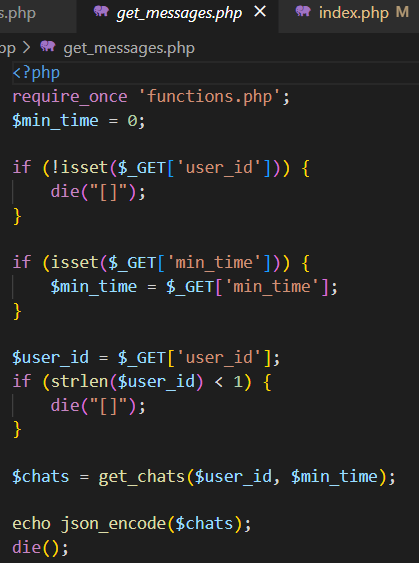
    } else {

        $msg = null;

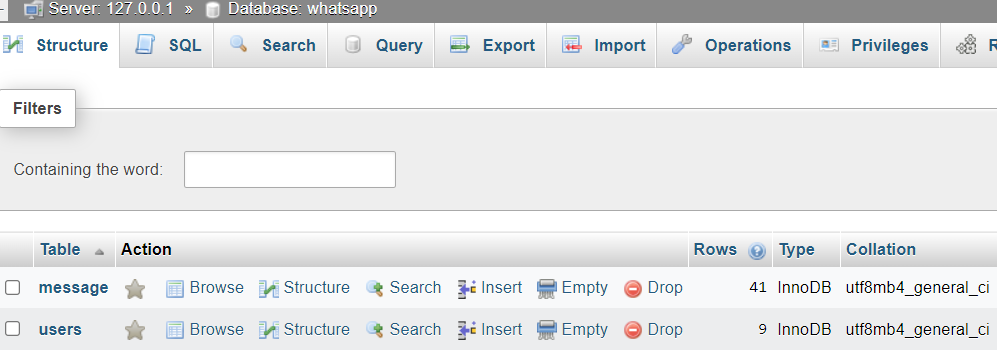
    }

    return $msg;

}



**MySQL use for Database:**



**Get chats work by this code:**

function get\_chats($user\_id = 0, $min\_time = 0) {

    $sql = "SELECT \* FROM  message,users

                WHERE

                (sender = {$user\_id} OR receiver = $user\_id)

                AND

                last\_update > {$min\_time}

                AND

                message.receiver = users.user\_id

            ";

    global $conn;

    $res = $conn->query($sql);

    $chats = array();

    while ($chat = $res->fetch\_assoc()) {

        $sender = $chat['sender'];

        if ($sender != $user\_id) {

            $message\_id = $chat['message\_id'];

            $receive\_time = "downloaded";

            $last\_update = time();

            $chat['receive\_time'] = $receive\_time;

            $chat['last\_update'] = $last\_update;

            $sql = "UPDATE message

                        receive\_time = '{$receive\_time}',

                        last\_update = '{$last\_update}'

                    WHERE

                        message\_id = {$message\_id}

            ";

        }

        $chats[] = $chat;

    }

    if ($chats == null) {

        return array();

    }

    return $chats;

}

function get\_chat\_thread($sender, $receiver) {

    $sql = "SELECT chat\_thread FROM  message

                WHERE

                (sender = {$sender} AND receiver = $receiver) OR

                (receiver = {$sender} AND sender = $receiver)

                ";

    global $conn;

    $res = $conn->query($sql);

    $chat\_thread = $res->fetch\_assoc();

    if ($chat\_thread != NULL) {

        return $chat\_thread['chat\_thread'];

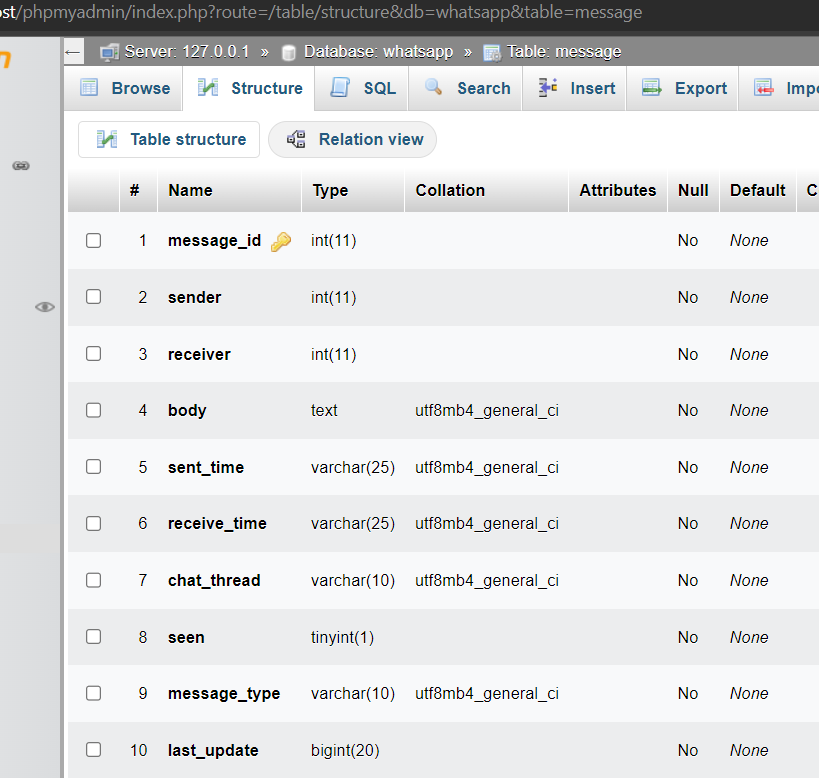
    }

    $new\_thread = $sender . "\_" . $receiver;

    return $new\_thread;

}

**Message DB structure:**



# Chapter 5:

# Conclusion

This app was developed by noticing the social issue with neighbours

in our society where most of the people do not know about there

neighbours and area. By taking this into notice I have developed this

app to solve this issue.

Sanga kalagam app allows individuals to keep in touch with friends and extended family. The app can be updated or modified in future with respect to user

requirment. The app can be updated by making changes in user interface, user interaction and customer support.

I am Amit Kumar as a developer with respect to application’s integrity

to other developers which fully supports the understandibility of

sophisticated developers.

## 5.2 Bibliography

<https://console.firebase.google.com/project/sangaeff59/inappmessaging/compoe>

<https://console.firebase.google.com/project/sanga-eff59/messaging>

[localhost / 127.0.0.1 / whatsapp | phpMyAdmin 5.2.0](http://localhost/phpmyadmin/index.php?route=/database/structure&db=whatsapp)

[localhost / 127.0.0.1 / whatsapp / users | phpMyAdmin 5.2.0](http://localhost/phpmyadmin/index.php?route=/sql&pos=0&db=whatsapp&table=users)