VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A PROJECT REPORT

on

# "SAMPARK SETU"

*Submitted by*

| | |
|---|---|
| ABHIJITH MALLYA | 4SF20CI002 |
| HITHESH SHETTY | 4SF20CI024 |

*In partial fulfillment of the requirements for VI Sem. B.E CSE(AI & ML)*

MOBILE APPLICATION DEVELOPMENT LABORATORY

WITH MINI PROJECT - 18AIMP68

*Under the Guidance of*

Mr. MANJUNATH E C

Assistant Professor, Department of CSE(AI&ML)

at

SAHYADRI

College of Engineering & Management

An Autonomous Institution

Adyar, Mangaluru - 575 007

2022 - 23

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
MANGALURU

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

# CERTIFICATE

This is to certify that the Mobile Application Development Mini Project - 18AIMP68 work entitled **"SAMPARK SETU"** has been carried out by **Abhijith Mallya (4SF20CI002), and Hithesh Shetty (4SF20CI024)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering (Artificial Intelligence & Machine Learning) of Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the departmental library. This project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

| | |
|---|---|
| **Guide** | **HOD** |
| Mr. Manjunath E C | Dr. Pushpalatha K |

**External Viva:**

| Examiner's Name | Signature with Date |
|---|---|
| 1. .................... | .................... |
| 2. .................... | .................... |

SAHYADRI

COLLEGE OF ENGINEERING & MANAGEMENT

An Autonomous Institution

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

# DECLARATION

We hereby declare that the entire work embodied in this Mobile Application Development Mini Project - 18AIMP68 Report titled **"SAMPARK SETU"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Mr. Manjunath E C** for the award of **Bachelor of Engineering** in **Computer Science & Engineering(Artificial Intelligence & Machine Learning)**. This report has not been submitted to this or any other University for the award of any other degree.

**ABHIJTH MALLYA (4SF20CI002)**

**HITHESH SHETTY (4SF20CI024)**

Dept. of CSE (AI & ML), SCEM, Mangaluru

# Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Project Report on **"Sampark Setu"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Computer Science & Engineering (Artificial Intelligence & Machine Learning).

We are profoundly indebted to our guide, **Mr. Manjunath E C**, Assistant Professor, Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning) for their constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Pushpalatha K**, Head & Professor, Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning) for her invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering & Management, who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

**ABHIJTH MALLYA (4SF20CI002)**

**HITHESH SHETTY (4SF20CI024)**

# Abstract

”Sampark Setu” is a mobile chat application developed using Android Studio and Firebase Database. The project aims to provide users with a seamless and secure platform for communication through instant messaging. The application is built on the Android platform, allowing users to connect with their contacts in real-time.The project utilizes various features and functionalities offered by Android Studio to create an intuitive user interface and a smooth user experience. Users can register and create their profiles within the application, enabling them to manage their contacts and start conversations easily. The application also supports features like group chats, file sharing, and multimedia messaging to enhance communication possibilities.

The development process involves designing and implementing the user interface, integrating Firebase services for authentication and database functionalities, and optimizing the application for performance and reliability. The project also focuses on ensuring data security and privacy by implementing appropriate encryption and access control measures.”Sampark Seva” serves as a practical demonstration of building a mobile chat application using Android Studio and Firebase Database. The project report details the development process, challenges encountered, and the implementation of various features. The application aims to provide a reliable and user-friendly platform for communication, addressing the growing need for efficient and secure messaging applications in today’s digital world.

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

The project "Sampark Setu" aims to contribute to this domain by developing a mobile chat application using Android Studio and Firebase Database. The goal of the project is to provide users with a seamless and secure platform for communication, enabling them to connect with their contacts effortlessly. With the increasing reliance on mobile devices, the demand for feature-rich and user-friendly chat applications has risen. "Sampark Setu" aims to address this need by leveraging the capabilities of Android Studio and Firebase Database.

Android Studio serves as the primary development environment, offering a robust set of tools and resources for creating Android applications. It provides a user-friendly interface design and allows developers to utilize various features to enhance the application's functionality and user experience. By leveraging Android Studio's capabilities, "Sampark Setu" ensures an intuitive user interface and smooth navigation for its users.Firebase Database acts as the backend infrastructure for the application, enabling real-time data synchronization and storage. With Firebase, users can experience instant message delivery and synchronization across multiple devices, ensuring a seamless communication experience. Additionally, Firebase's authentication services provide secure user registration and login functionalities, ensuring data privacy and protection.

This project report provides an in-depth overview of the development process, methodologies, and challenges encountered while building "Sampark Setu." It explores the technical aspects of using Android Studio for application development and integrating Firebase Database for efficient data storage and synchronization. The report also highlights the considerations given to user experience, security, and performance optimization.Overall, the "Sampark Setu" project serves as a practical demonstration of developing a mobile chat application using Android Studio and Firebase Database. By providing users with an intuitive and secure platform for communication, the application aims to meet the growing demand for efficient and reliable messaging applications in today's digital era.

## 1.1 About Android

Android [1]is a versatile and widely used operating system that powers a multitude of mobile devices, including smartphones and tablets. Developed by Google, Android offers a user-friendly interface and a vast ecosystem of applications, making it a favorite among users and developers alike. With its open-source nature, Android allows for customization and innovation, enabling manufacturers to create diverse and unique devices to cater to different user preferences. Its robust features, seamless integration with Google services, and constant updates contribute to its widespread popularity, making Android a dominant force in the mobile industry.

## 1.2 Languages Used

### 1.2.1 Java

Java [2]is a fundamental programming language widely used in Android development, providing a solid foundation for building robust and versatile mobile applications. As the primary language for Android development, Java offers a rich set of libraries and frameworks that enable developers to create engaging user interfaces, handle complex data processing, and implement efficient algorithms. Its object-oriented nature facilitates code reusability and modularity, enhancing productivity and maintainability. With Java's extensive community support and continuous updates, it remains a key language choice for crafting innovative and feature-rich Android applications.

### 1.2.2 XML (Extensible Markup Language)

XML (Extensible Markup Language) plays a crucial role in Android development, serving as a powerful tool for defining and organizing the user interface (UI) components of an application. With its hierarchical structure and human-readable syntax, XML enables developers to describe the visual elements and their properties, such as layouts, widgets, and styles, in a clear and concise manner. By separating the presentation layer from the application logic, XML empowers developers to create flexible and customizable interfaces, supporting efficient design iteration and localization efforts. Additionally, XML facilitates the binding of UI elements to code through data binding techniques, promoting a more modular and maintainable approach to Android app development.

### 1.2.3 Google Firebase

Google Firebase [3, 4] is a powerful and versatile platform that plays a vital role in Android development. It offers a wide range of tools and services designed to simplify the development process and enhance app functionality. Firebase provides features such as real-time database, cloud storage, user authentication, and push notifications, enabling developers to effortlessly create robust and scalable Android applications. With its extensive set of APIs and intuitive interface, Firebase empowers developers to focus on building innovative and engaging user experiences while seamlessly integrating backend functionality into their Android apps.

## 1.3 Android Studio

Android Studio [5]is an integrated development environment (IDE) designed specifically for creating Android applications. It provides a comprehensive set of tools and features that enable developers to efficiently build, test, and debug Android apps. With a user-friendly interface and a wide range of built-in tools, Android Studio streamlines the development process by offering code editing, project management, and emulator support all in one place. Its robust features, such as real-time code analysis, intelligent code completion, and powerful debugging capabilities, make it an indispensable tool for developers looking to create high-quality Android applications.

## 1.4 Firebase Database

Firebase Database [3] is a flexible and scalable cloud-based NoSQL database that enables developers to store and synchronize data for their web and mobile applications. It offers real-time data synchronization and offline capabilities, allowing seamless data updates across different devices. With Firebase Database, developers can easily manage and manipulate data using a simple JSON-based structure, making it straightforward to retrieve, update, and delete data. Its robust security features and built-in authentication options ensure data privacy and integrity.

# Chapter 2

# Requirements Specification

## 2.1 Project Requirements

The package is designed such that users with an Android phone having minimum configuration can also use it. It does not require complex computing. The App requires a simple daily use android phone in which this app can be installed and then user can run it. For a developer it is required to install android studio or IntelliJ Application software which is used to design the app using Xml and Java. Firebase Realtime database is used as backend to store the data.

## 2.2 Hardware Specification

- Processor : Intel(R) Core(TM) i5-1005G1 CPU @ 1.20GHz
- RAM : 8GB
- Hard Disk : 512GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

## 2.3 Software Specification

- Programming Language :Java and XML
- IDE :Android Studio
- Database: Firebase

# Chapter 3

# System Design Methodology

## 3.1 Basic Layout

1. A Splash page for application.
2. A Login page for users.
3. A Main interface page for users
4. A Main text page.
5. A Chat interface page

## 3.2 Architecture Diagram

The architecture diagram of the application is as shown in the below figure:

The Model represents the data and business logic, including data models and repositories. The Firebase database serves as the data source, storing chat messages and user profiles.The View handles UI components, such as activities and fragments, while the ViewModel acts as a mediator between the Model and View, managing the application state and performing business logic. It interacts with the Firebase database to fetch and update data.Firebase Realtime Database is used to store and synchronize chat messages in real-time. It provides event listeners to observe changes in the data and automatically update the UI.Firebase Authentication is integrated for user registration, login, and managing user sessions. It offers various authentication methods like email/password,Other components include the ChatAdapter to manage the chat message list and UI, and Push Notification Service using Firebase Cloud Messaging to notify users about new chat messages when the app is in the background.By leveraging the features of Firebase and integrating it with

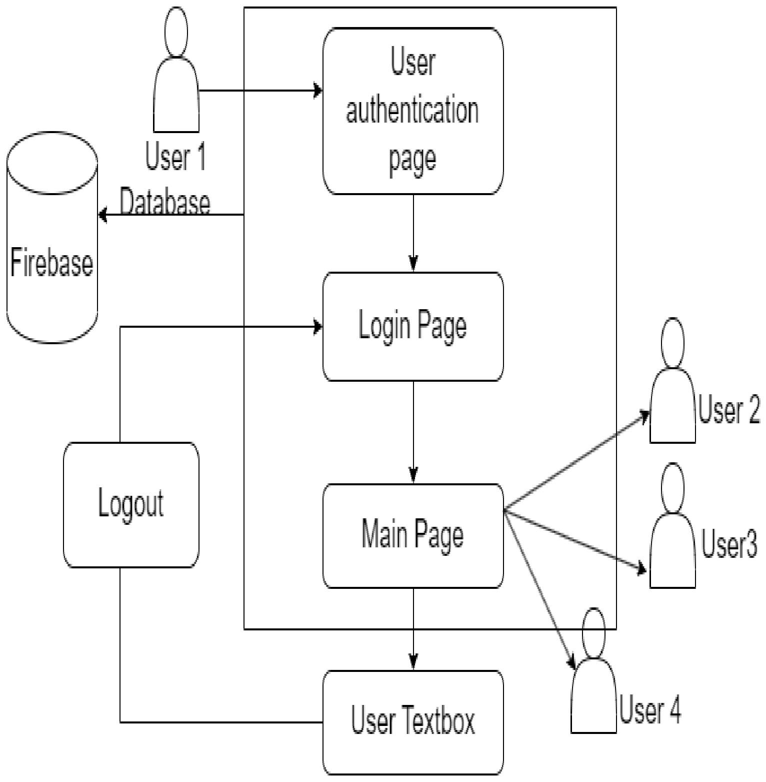


Figure 3.1: Architect ure Diagram of Ration Distribution System

Android Studio, the architecture provides a scalable and reliable solution for building a chat application with real-time updates, user authentication, and seamless data synchronization.

# Chapter 4

# Implementation

## 4.1 Login page

The below contains Java code for login the user into the application. If the data is not entered properly then an error message will be displayed and if the user is a new user then it will redirect into sign up page.

```
package com.av.SamparkSetu;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class login extends AppCompatActivity {
    TextView logsignup;
    Button button;
    EditText email, password;
    FirebaseAuth auth;
    String emailPattern = "[a-zA-Z0-9._-]+@[a-z]+\\.+[a-z]+";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

//-----------------------------    google login start--------------------------

        googleBtn = findViewById(R.id.google_btn);
```

```
gso = new
gsc = GoogleSignIn.getClient(this, gso);



googleBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        signIn();
    }
});



button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String Email = email.getText().toString();
        String pass = password.getText().toString();

        if ((TextUtils.isEmpty(Email))) {
            progressDialog.dismiss();
            Toast.makeText(login.this, "Enter The Email",
                Toast.LENGTH_SHORT).show();
        } else if (TextUtils.isEmpty(pass)) {
            progressDialog.dismiss();
            Toast.makeText(login.this, "Enter The Password",
                Toast.LENGTH_SHORT).show();
        }else {
            auth.signInWithEmailAndPassword(Email,
                pass).addOnCompleteListener(new
                OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        progressDialog.show();
                        try {
                            Intent intent = new Intent(login.this,
                                MainActivity.class);
                            startActivity(intent);
                            finish();
                        } catch (Exception e) {
                            Toast.makeText(login.this, e.getMessage(),
                                Toast.LENGTH_SHORT).show();
                        }
                    } else {
                        Toast.makeText(login.this,
                            task.getException().getMessage(),
                            Toast.LENGTH_SHORT).show();
                    }
                }
            });
        }
    }
```

## 4.2 Registration page

The below section contains Java code to add new users to the database. In this the user can have options to add his details to the database by giving required fields in this page like email id and password.

```
package com.av.avmessenger;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

public class registration extends AppCompatActivity {
    TextView loginbut;
    EditText rg_username, rg_email , rg_password, rg_repassword;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registration);
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Establishing The Account");
        loginbut.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(registration.this,login.class);
                startActivity(intent);
                finish();
            }
        });

        rg_signup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String namee = rg_username.getText().toString();
if (TextUtils.isEmpty(namee) || TextUtils.isEmpty(emaill) ||
        TextUtils.isEmpty(Password) ||
            TextUtils.isEmpty(cPassword)){
    progressDialog.dismiss();
    Toast.makeText(registration.this, "Please Enter Valid
        Information", Toast.LENGTH_SHORT).show();
}else if (!Password.equals(cPassword)){
    progressDialog.dismiss();
    rg_password.setError("The Password Doesn't Match");
}else {
        public void onComplete(@NonNull Task<AuthResult> task) {
             if (task.isSuccessful()){
                 String id = task.getResult().getUser().getUid();
                storage.getReference().child("Upload").child(id);
            else {
                Toast.makeText(registration.this, "Error in
                creating the user", Toast.LENGTH_SHORT).show();
                              }
                          }
                      });
                  }
             }else {
                 Toast.makeText(registration.this,
                     task.getException().getMessage(),
                     Toast.LENGTH_SHORT).show();
             }
        }
    });
}
```

## 4.3 Chat interface page

A page comprises of a code for the working interface where two users can interact with one another in the form of chat using realtime firebase.

```
package com.av.avmessenger;


import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;


public class MainActivity extends AppCompatActivity{
    ImageView imglogout;
    ImageView cumbut,setbut;


    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        getSupportActionBar().hide();
        imglogout = findViewById(R.id.logoutimg);


        imglogout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Dialog dialog = new Dialog(MainActivity.this,R.style.dialoge);
                dialog.setContentView(R.layout.dialog_layout);
                Button no,yes;
                yes = dialog.findViewById(R.id.yesbnt);
                no = dialog.findViewById(R.id.nobnt);
                yes.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        FirebaseAuth.getInstance().signOut();
                        Intent intent = new Intent(MainActivity.this,login.class);
                        startActivity(intent);
                    finish();
                }
            });
            no.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    dialog.dismiss();
                }
            });
            dialog.show();
        }
    });

    setbut.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(MainActivity.this, setting.class);
            startActivity(intent);
        }
    });
}
```

## 4.4 User Model

It gives the structure of user model. It has getter and setter functions for the attributes to be accessed by other class.

```java
package com.av.avmessenger;


public class Users {
    String profilepic,mail,userName,password,userId,lastMessage,status;


    public Users(){}


    public Users(String userId, String userName, String maill, String password,
        String profilepic, String status) {
        this.userId = userId;
        this.userName = userName;


    }


    public String getProfilepic() {
        return profilepic;
    }
    public void setProfilepic(String profilepic) {
        this.profilepic = profilepic;
    }
    public String getMail() {
        return mail;
    }
    public void setMail(String mail) {
        this.mail = mail;
    }
    public String getUserName() {
        return userName;
    }
}
```

## 4.5 User Adapter

It handles the user data from the firebase real-time database and maps the data into the recycler views.

```
package com.av.avmessenger;


import android.content.Context;
import android.content.Intent;
public class UserAdpter extends RecyclerView.Adapter<UserAdpter.viewholder> {
    Context mainActivity;
    ArrayList<Users> usersArrayList;
    public UserAdpter(MainActivity mainActivity, ArrayList<Users>
        usersArrayList) {
        this.mainActivity=mainActivity;
        this.usersArrayList=usersArrayList;
    }

    @NonNull
    @Override
    public void onBindViewHolder(@NonNull UserAdpter.viewholder holder, int
        position) {

        Users users = usersArrayList.get(position);
        holder.username.setText(users.userName);
        holder.userstatus.setText(users.status);
        Picasso.get().load(users.profilepic).into(holder.userimg);

        holder.itemView.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(mainActivity, chatwindo.class);
                intent.putExtra("nameeee",users.getUserName());
                intent.putExtra("reciverImg",users.getProfilepic());
                intent.putExtra("uid",users.getUserId());
                mainActivity.startActivity(intent);
            }
        });

    }

    @Override
    public int getItemCount() {
        return usersArrayList.size();
    }

    public class viewholder extends RecyclerView.ViewHolder {
        CircleImageView userimg;
        TextView username;
        TextView userstatus;
        public viewholder(@NonNull View itemView) {
            super(itemView);
            userimg = itemView.findViewById(R.id.userimg);
            username = itemView.findViewById(R.id.username);
            userstatus = itemView.findViewById(R.id.userstatus);
        }
    }
}
```

## 4.6 Settings

It gives the user a choice to update his/her profile picture and set a status.

```
package com.av.SamparkSetu;
import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
public class setting extends AppCompatActivity {
    ImageView setprofile;
    EditText setname, setstatus;
    ProgressDialog progressDialog;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_setting);
        getSupportActionBar().hide();
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                email = snapshot.child("mail").getValue().toString();
                Picasso.get().load(profile).into(setprofile);
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });
        setprofile.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent();
                intent.setType("image/*");
                intent.setAction(Intent.ACTION_GET_CONTENT);
                startActivityForResult(Intent.createChooser(intent, "Select
                    Picture"), 10);
            }
        });
```

```
}

        donebut.setOnClickListener(new View.OnClickListener() {
                                @Override
                                public void onSuccess(Uri uri) {
                                    String finalImageUri = uri.toString();
                                    Users users = new Users(auth.getUid(),
                                        name,email,password,finalImageUri,Status);
                                }});}
                    });
                }else {
                    storageReference.getDownloadUrl().addOnSuccessListener(new
                        OnSuccessListener<Uri>() {
                        @Override
                        public void onSuccess(Uri uri) {
                            String finalImageUri = uri.toString();
                            reference.setValue(users).addOnCompleteListener(new
                                OnCompleteListener<Void>() {
                                @Override
                                public void onComplete(@NonNull Task<Void> task) {
                                    if (task.isSuccessful()){
                                        progressDialog.dismiss();
                                        Toast.makeText(setting.this, "Data Is save
                                            ", Toast.LENGTH_SHORT).show();
                                        Intent intent = new
                                            Intent(setting.this,MainActivity.class);
                                        startActivity(intent);
                                        finish();
                                    }else {
                                        progressDialog.dismiss();
                                        Toast.makeText(setting.this, "Some thing
                                            went romg", Toast.LENGTH_SHORT).show();
                                    }
                            });
}
```

# Chapter 5

# Results and Discussion

## 5.1 Registration Page

Users can easily sign up and connect with individuals worldwide. Leveraging the power of Android Studio, we have created a user-friendly interface, ensuring smooth navigation and interaction.

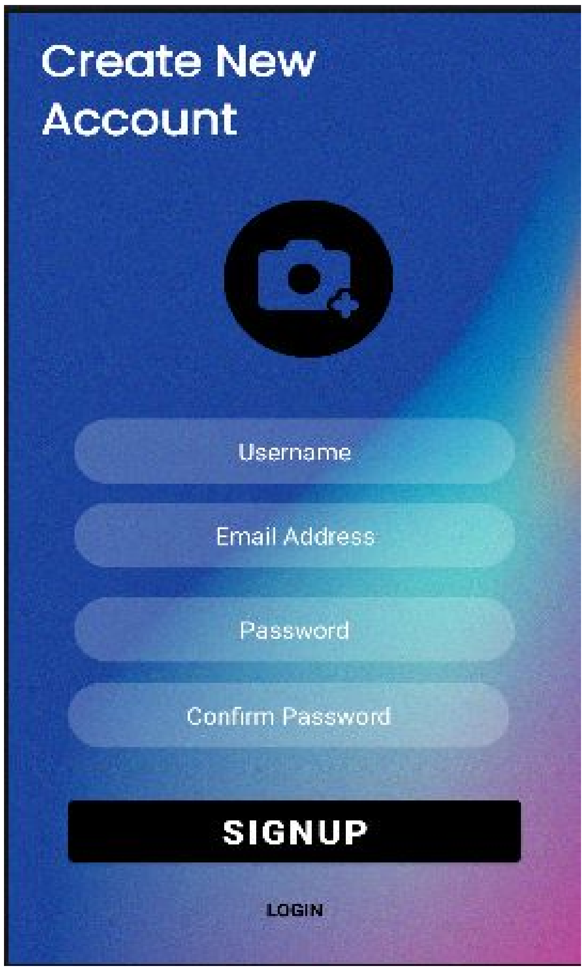


Figure 5.1: Customer Registration

## 5.2 Login Page

Sampark Seva login page, built with the expertise of Android Studio and the reliability of Firebase database integration. Allowing users to access their personalized accounts effortlessly. Our Android Studio development ensures a seamless interface, while Firebase database ensures data integrity and reliability.

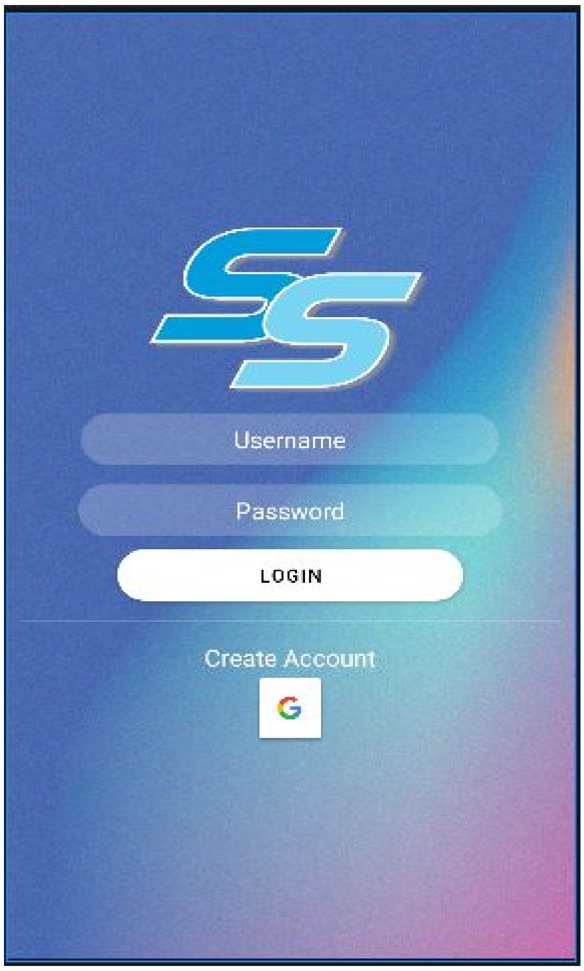
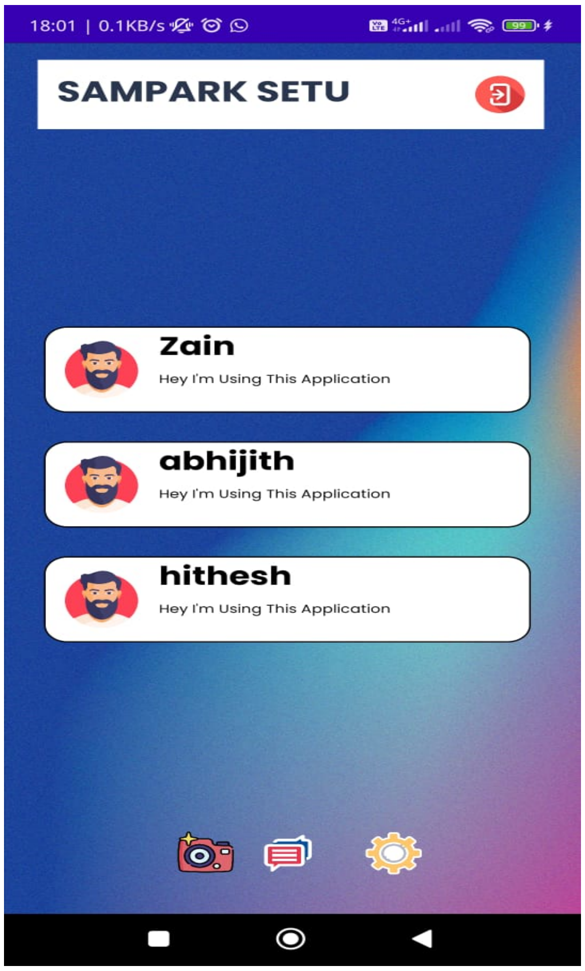


Figure 5.2: Customer Login

## 5.3 Home Page

This Home page provides users with a comprehensive list of all registered individuals, enabling easy navigation and communication. Android Studio ensures a visually appealing and user-friendly interface, while Firebase database integration guarantees real-time updates and secure data retrieval. Stay connected with a diverse community and foster meaningful connections through our intuitive contact interface.



Zain
Hey I'm Using This Application

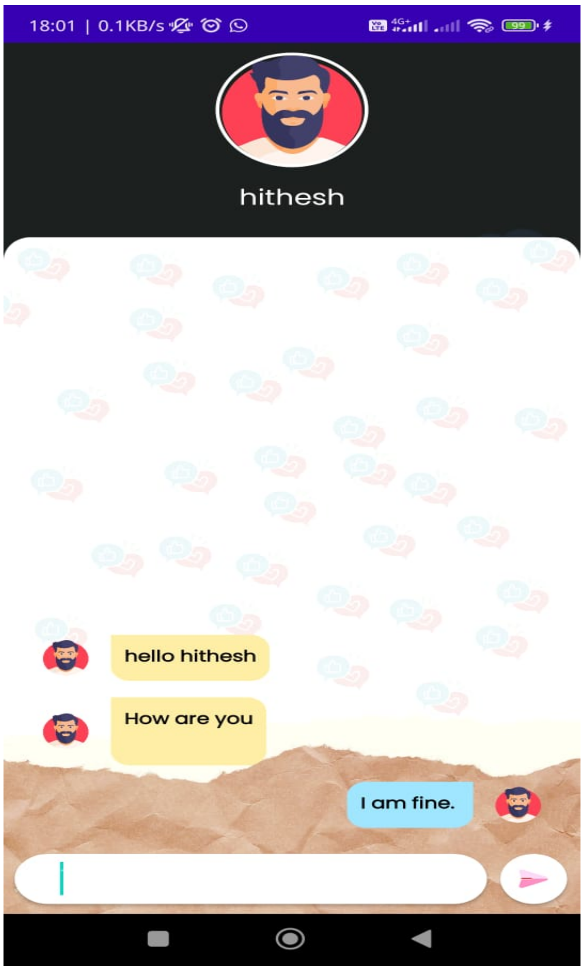
abhijith
Hey I'm Using This Application

hithesh
Hey I'm Using This Application

Figure 5.3: Home Page

## 5.4 User textpage Page

This intuitive page offers a convenient platform for users to engage in text-based conversations. With Android Studio.Leveraging Firebase database, we ensure real-time message delivery and storage for a reliable communication experience. Connect with other users effortlessly and exchange thoughts through our efficient text interface.



hello hithesh

How are you

I am fine.

Figure 5.4: Chatting window

# Chapter 6

# Conclusion and Future work

The integration of Firebase's robust database infrastructure ensures efficient data storage and retrieval. Extensive testing has validated the application's functionality and user-friendliness. Future work includes enhancing the user interface design, enabling group chats and multimedia messaging, implementing end-to-end encryption, and integrating push notifications for improved user engagement. Additional features such as message status indicators, user presence tracking, and analytics can further enhance the application's usability and performance.

**User Authentication**: Implement a secure user authentication system to ensure that only authorized users can access the chat application. Firebase provides authentication services that can be integrated into the app for this purpose.

**Multimedia Support**: Extend the chat functionality to support the exchange of multimedia content such as images, videos, and audio files. Users can send and receive various media types, enhancing the communication experience.

**Group Chats**: Introduce group chat functionality, allowing users to create or join groups to have conversations with multiple participants. This feature facilitates collaboration and communication among a larger set of users.

**Push Notifications**: Implement push notifications to alert users about new messages even when they are not actively using the app. Firebase Cloud Messaging (FCM) can be used to send push notifications to the users' devices.

**Message Encryption**: Enhance the security of the application by implementing end-to-end encryption for messages. This ensures that only the intended recipients can read the messages, providing privacy and data protection.

# References

[1] E. Hellman, *Android Programming-Pushing the Limits.* Wiley India Pvt Ltd, 1st ed., 2014.

[2] “Java.” https://www.java.com, 1995. Programming Language.

[3] “Firebase.” https://firebase.google.com. Accessed: 12-05-2023.

[4] YouTube, “TechCoder AV.” https://www.youtube.com/channel/UC$_oXh9DV6HLoD$.

[5] “Android Studio.” https://developer.android.com/studio, 2013. Integrated Development Environment (IDE) for Android app development.